

TON ROOSENDAL, STEFANO SELLERI et al

EL OFICIAL

BLENDER

2.3 GUIA

LA SUITE ABIERTA DE CREACION 3D



EN ESPAÑOL



INDICE

I - INTRODUCCIÓN

Introducción
Instalando desde Ejecutables
La Interfaz
Navegando en el Espacio 3D
Las Funciones Vitales
Su Primera Animación en 30 + 30 Minutos I
Su Primera Animación en 30 + 30 Minutos II

II - MODELADO

Modo Objeto
Booleanos
Objetos Básicos
Modo Edición
Modelado Básico de Malla
Subdivisión de Superficies
Herramientas Arista y Cara
Modelado Avanzado de Malla
Curvas
Superficies
Texto
Meta Objetos

III - MATERIALES

Difusión
Reflexión Especular
Los Materiales en la Práctica
Sombreadores de Rampa
Reflexiones y Transparencias
Materiales Múltiples
Materiales Especiales

IV - TEXTURAS

Canales de Textura
Map Input
Map To
Botones de Textura
Texturas Procedurales
Texturas Imagen
Mapeado UV
Mapas de Relieve y de Normales
Mapas de Ambiente
Mapas de Desplazamiento
Plugins de Textura

V - ILUMINACIÓN

Tipos de Luces
Sombras con el Trazador de Rayos
Sombras con el Buffer

VI - EL MUNDO Y EL UNIVERSO

El Fondo del Mundo
Oclusión Ambiental
Exposición y Rango
Niebla
Estrellas

VII - CONCEPTOS DE ANIMACIÓN

Bloque IPO
Fotogramas Clave
Curvas IPO y Claves IPO
Siguiendo un Camino

VIII - DEFORMACIONES BÁSICAS

Garfios
Claves de Forma
Claves de Vértices Absolutas
Claves de Vértices Relativas
Animación de Enrejados

IX - EL SISTEMA ARMAZÓN

El Objeto Armature
FK, IK y otras Restricciones
Skinning
El Modo Pose
La Ventana Acción
Animación No Lineal
Rigging examples

X - CUERPOS BLANDOS Y FLUIDOS

Cuerpos Blandos
Simulación de Fluidos

XI - RENDERIZADO

Antialiasing
Animaciones
Formatos de Salida
Opciones de R nder
El Renderizador Unificado
YafRay
Preparando su trabajo para V deo

XII - RADIOSIDAD

Renderizado de Radiosidad
Radiosidad como Herramienta de Modelado
Un Jugoso Ejemplo de Radiosidad

XIII - HERRAMIENTAS ESPECIALES DE MODELADO

Part culas
Interacci n de Part culas
Build y Ondas
DupliVerts
DupliFrames

XIV - SECUENCIADO

Editor de Secuencia de V deo
Editor de Secuencia de Audio

XV - EXTENDIENDO BLENDER

Scripts en Python
Sistema de plugins de Blender

ESTE MANUAL FUE COPIADO DE LA WEB:

<http://mediawiki.blender.org/index.php/Manual.es/Manual>

LA IMAGEN DEL MANUAL FUE COPIADA DE LA WEB:

http://www.blender.org/e-shop/product_info.php?products_id=79

La imagen fue modificada por LEONO con el fin de dotar al manual de su portada en espa ol, adem s agregando el logotipo de BLENDER en la parte inferior derecha, esta recopilaci n de material no es con motivo de lucro sino con el fin de facilitar el aprendizaje a los usuarios hispanohablantes que tienen las ganas de aprender el 3D con esta fabulosa herramienta de dise o 3D, que adem s es open source. No me hago responsable del mal uso de este material que es con fines educativos.

Un cordial saludo a todos los usuarios de blender.....aprender aprender aprender.... con blender

PARTE I - INTRODUCCIÓN

La primera parte de la documentación te guiará a través de la descarga, instalación y, si elige descargar también las fuentes, la compilación de Blender.

Blender posee una interfaz muy peculiar, altamente optimizada para la producción de gráficos en 3D. Al principio, puede parecer un poco complicada a un usuario sin experiencia, pero comprobará su poder a la larga. Es muy recomendable leer La Interfaz detenidamente para familiarizarse tanto con la interfaz como con las convenciones usadas en esta documentación.

El último capítulo de esta parte, Su Primera Animación en 30 + 30 Minutos, le permitirá entrever las capacidades de Blender. Por supuesto, Blender puede hacer mucho más que esto, ya que esto es sólo una introducción rápida al programa.

Capítulos

- Introducción
- Instalando desde Ejecutables
- Compilando las Fuentes
- La Interfaz
- Navegando en el Espacio 3D
- Las Funciones Vitales
- Su Primera Animación en 30 + 30 Minutos I
- Su Primera Animación en 30 + 30 Minutos II

Introducción

¿Qué es Blender?

Blender es un programa que integra una serie de herramientas para la creación de un amplio rango de contenidos 3D, con los beneficios añadidos de ser multiplataforma y tener un tamaño de unos 5MB.

Destinado a artistas y profesionales de multimedia, Blender puede ser usado para crear visualizaciones 3D, tanto imágenes estáticas como vídeos de alta calidad, mientras que la incorporación de un motor de 3D en tiempo real permite la creación de contenido interactivo que puede ser reproducido independientemente.

Originalmente desarrollado por la compañía 'Not a Number' (NaN), Blender es ahora desarrollado como 'Software Libre', con el código fuente disponible bajo la licencia GNU GPL.

Características principales:

- Paquete de creación totalmente integrado, ofreciendo un amplio rango de herramientas esenciales para la creación de contenido 3D, incluyendo modelado, mapeado uv, texturizado, rigging, weighting, animación, simulación de partículas y otros, scripting, renderizado, composición, post-producción y creación de juegos.
- Multiplataforma, con una interfaz unificada para todas las plataformas basada en OpenGL, listo para ser usado en todas las versiones de Windows (98, NT, 2000 y XP), Linux, OSX, FreeBSD, Irix y Sun, y otros sistemas operativos.
- Arquitectura 3D de alta calidad permitiendo un rápido y eficiente desarrollo.
- Canales de soporte gratuito vía <http://www.blender3d.org>.
- Una comunidad mundial de más de 250.000 usuarios.
- Tamaño pequeño de ejecutable para una fácil distribución.

Historia de Blender

En 1988, Ton Roosendaal co-fundó el estudio de animación Holandés *NeoGeo*. NeoGeo rápidamente se convirtió en el estudio más grande de animación 3D en Holanda y en una de las más destacadas casas de animación en Europa. NeoGeo creó producciones que fueron premiadas (European Corporate Video Awards de 1993 y 1995) para grandes clientes corporativos tales como la compañía multinacional de electrónica Philips. En NeoGeo, Ton fue el responsable tanto de la dirección artística como del desarrollo interno del software. Después de una cuidadosa deliberación, Ton decidió que la actual herramienta 3D utilizada en el estudio de NeoGeo era demasiado vieja y voluminosa de mantener y actualizar y necesitaba ser reescrita desde el principio. En 1995, esta reescritura comenzó y estaba destinado a convertirse en el software de creación 3D que ahora conocemos como Blender. Mientras

NeoGeo continuaba refinando y mejorando Blender, Ton se dio cuenta que Blender podría ser utilizado como una herramienta para otros artistas fuera del estudio NeoGeo.

En 1998, Ton decidió crear una nueva compañía llamada Not a Number (NaN) derivada de NeoGeo para fomentar el mercado y desarrollar Blender. En la base de NaN, estaba el deseo de crear y distribuir gratuitamente una suite de creación 3D compacta y multiplataforma. En ese momento, esto fue un concepto revolucionario ya que la mayoría de los programas comerciales de modelado costaban miles de dólares. NaN esperaba conseguir una herramienta de modelado y animación de un nivel profesional al alcance del público en general. El modelo de negocio de NaN consistía en proporcionar productos comerciales y servicios alrededor de Blender. En 1999, NaN asistió a su primera conferencia en el Siggraph en un esfuerzo aún mayor para promocionar Blender. La primera convención del Siggraph para Blender en 1999 fue un auténtico éxito y provocó un enorme interés tanto de la prensa como de los asistentes a la convención. ¡Blender fue un gran éxito y se confirmó su tremendo potencial!

En alas del gran éxito del Siggraph, a principios del año 2000, NaN consiguió una financiación de 4,5 millones de euros procedente de unos inversores. Este gran aporte de dinero permitió a NaN expandir rápidamente sus operaciones. Pronto NaN alardeó de tener más de 50 empleados trabajando alrededor del mundo intentando mejorar y promocionar Blender. En el verano del 2000, Blender 2.0 fue publicado. Esta versión de Blender integraba un motor de juegos a la suite 3D. Al final del 2000, el número de usuarios registrados en el sitio web de NaN sobrepasó los 250.000.

Desafortunadamente, las ambiciones y oportunidades de NaN no coincidieron con las capacidades de la compañía ni con la realidad del mercado de la época. Este sobredimensionamiento de la empresa condujo a una reestructuración creando una compañía (NaN) mas pequeña y con nuevos fondos procedentes de los inversores. Seis meses mas tarde, el primer producto comercial de NaN, *Blender Publisher* fue lanzado. Este producto fue dirigido al emergente mercado de medios interactivos en 3D basados en entornos web. Debido a las decepcionantes ventas y al continuo clima de dificultades económicas, los nuevos inversores decidieron dar por terminadas las actividades de NaN. Esto también incluía parar el desarrollo de Blender. Si bien existían claramente defectos en la actual versión de Blender, con una arquitectura interna del software compleja, características inacabadas y una IGU no muy común, la magnífica ayuda de la comunidad y los clientes que habían comprado Blender Publisher en el pasado provocó que Ton no pudiera permitir que Blender desapareciera en el olvido. Como relanzar una nueva compañía con un equipo suficientemente grande de desarrolladores no era factible, en marzo de 2002, Ton Roosendaal fundó la organización no lucrativa *Blender Foundation* (Fundación Blender).

El primer objetivo de la Fundación Blender fue encontrar una manera de continuar el desarrollo y la promoción de Blender como un proyecto de código abierto basado en la comunidad de usuarios. En julio de 2002, Ton logró obtener de los inversores de NaN un "sí" para que la Fundación Blender llevara a cabo su plan de que Blender fuera código abierto. La campaña de "Liberad a Blender" tenía que obtener 100.000 EUR para que la Fundación pudiese comprar los derechos del código fuente y los de propiedad intelectual de Blender a los inversores de NaN y, posteriormente, liberar Blender a la comunidad de código abierto. Con un entusiasta grupo de voluntarios, entre los que se encontraban varios ex-empleados de NaN, fue lanzada la campaña de "Liberad a Blender". Para el deleite y sorpresa de todo el mundo, la campaña alcanzó el objetivo de 100.000 EUR en tan sólo 7 semanas. El domingo 13 de octubre de 2002, Blender fue liberado al mundo bajo los términos de la Licencia Pública

General de GNU (GPL). El desarrollo de Blender continúa hasta nuestros días conducido por un equipo de valientes y dedicados voluntarios procedentes de diversas partes del mundo y liderados por el creador de Blender, Ton Roosendaal.

Historia de Blender y hoja de ruta

- 1.00 - enero de 1995 - Desarrollo de Blender en el estudio de animación NeoGeo.
- 1.23 - enero de 1998 - Versión para SGI (IrisGL) publicada en la web.
- 1.30 - abril de 1998 - Versión para Linux y FreeBSD, se porta a OpenGL y a X.
- 1.3x - junio de 1998 - Creación de NaN.
- 1.4x - septiembre de 1998 - Versión para Sun y Linux Alpha publicada.
- 1.50 - noviembre de 1998 - Primer manual publicado.
- 1.60 - abril de 1999 - C-key (nuevas características necesitan ser desbloqueadas, \$95), la versión de Windows se libera.
- 1.6x - junio de 1999 - Versión para BeOS y PPC publicada.
- 1.80 - junio de 2000 - Fin de la C-key, Blender es totalmente gratuito de nuevo.
- 2.00 - agosto de 2000 - Motor en tiempo-real y reproductor interactivo.
- 2.10 - diciembre de 2000 - Nuevo motor, física y Python.
- 2.20 - agosto de 2001 - Sistema de animación de personajes.
- 2.21 - octubre de 2001 - Blender Publisher lanzado.
- 2.2x - diciembre de 2001 - Versión para Mac OSX publicada.
- **13 de octubre de 2002** - Blender se convierte en código abierto, primera Conferencia de Blender.
- 2.25 - octubre de 2002 - Blender Publisher está de nuevo disponible gratuitamente.
- Tuhopuu1 - octubre de 2002 - Se crea la rama experimental de Blender, un lugar de pruebas para los programadores.
- 2.26 - febrero de 2003 - La primera versión de Blender siendo código abierto.
- 2.27 - mayo de 2003 - La segunda versión de Blender siendo código abierto.
- 2.28x - julio de 2003 - La primera de las series 2.28x.
- 2.30 - octubre de 2003 - En la segunda conferencia de Blender, la interfaz 2.3x es presentada.
- 2.31 - diciembre de 2003 - Actualización a la interfaz de la versión 2.3x, ya que es estable.
- 2.32 - enero de 2004 - Gran revisión de la capacidad del render interno.
- 2.33 - abril de 2004 - Oclusión Ambiental, nuevos procedimientos de Texturas, ¡el motor de juego ha vuelto!.
- 2.34 - agosto de 2004 - Grandes mejoras: Interacciones de Partículas, mapeado LSCM UV, integración funcional de YafRay, Pliegues compensados en Subdivisión de Superficies, Sombreado de Inclinación (Ramp), OSA completo y muchas muchas más.
- 2.35 - noviembre de 2004 - Otra versión llena de mejoras: Object hooks, curve deforms and curve tapers, particle duplicators y mucho más.
- 2.36 - febrero de 2005 - Versión más estable, más trabajo en la escena, pocas nuevas mejoras, mapeado Normal y de Desplazamiento, principalmente.
- 2.37 - junio de 2005 - Un gran avance: herramientas de Transformación y controles, Softbodies, Force fields, deflections, incremental Subdivision Surfaces, sombras transparentes, y renderizado multihilo.
- 2.40 - diciembre de 2005 - Un mayor avance: reescritura del sistema de esqueletos, teclas de formas, piel con partículas, fluidos y cuerpos rígidos.
- 2.41 - enero de 2006 - Principalmente añadidos en el Motor de juego y correcciones.

Sobre el Software Libre y la licencia GPL

Cuando uno escucha hablar sobre "software libre", la primera cosa que le viene a la mente es "gratis". Aunque esto es verdad en la mayoría de los casos, el término "software libre" tal como es usado por la Fundación para el Software Libre (creadores del proyecto GNU y de la Licencia Pública General GNU) significa "libre, como en libertad" más que "gratis, como en cerveza gratis" (*N. del T.* En inglés tienen el problema de que la palabra "free" significa gratis y libre, por eso al definir "free software" tienden a hacer distinción entre software gratuito y software libre). Software libre es, en este sentido, aquel software que puedes usar, copiar, modificar y distribuir sin ningún límite. Esto contrasta con la licencia de la mayoría de paquetes de software comerciales, en donde se te permite usar el software en una sola computadora pero no está permitido hacer copias y menos aún ver el código fuente. El software libre da una increíble libertad al usuario. Como añadidura a todo esto, debido a que el código fuente está disponible para todo el mundo, los fallos pueden ser detectados y corregidos de una manera más eficiente y rápida.

Cuando un programa es licenciado bajo la Licencia Pública General GNU (la GPL):

- Tienes el derecho de usar el programa para cualquier propósito.
- Tienes el derecho de modificar el programa, y tener acceso al código fuente.
- Tienes el derecho de copiar y distribuir el programa.
- Tienes el derecho de mejorar el programa, y publicar tus propias versiones.

A cambio de esos derechos, también tienes algunas responsabilidades, que están diseñadas para proteger tu libertad y la libertad de otros, por lo que si distribuyes un programa GPL:

- Debes proporcionar una copia de la GPL con el programa, así el usuario es consciente de los derechos que le otorga la licencia.
- Debes incluir el código fuente o hacer que esté disponible gratuitamente.
- Si modificas el código y distribuyes una versión modificada, dichas modificaciones deben estar licenciadas bajo la GPL y tienes que facilitar el código fuente modificado. (No puedes usar código GPL como parte de un programa propietario.)
- No puedes licenciar el programa fuera de los términos de la licencia GPL. (No puedes transformar un programa GPL en un producto propietario).

Para más información sobre la licencia GPL, visita el sitio web del Proyecto GNU. Para referencia, una copia de la licencia pública general de GNU se incluye en el Manual de Referencia.

Consiga ayuda - La comunidad de usuarios de Blender

Al estar Blender disponible de forma gratuita desde un principio, incluso cuando el código era cerrado, ayudó mucho en su adopción. Una gran comunidad estable y activa de usuarios ha estado creciendo alrededor de Blender desde el año 1998.

La comunidad mostró lo mejor de sí misma en el momento crucial de la liberación de Blender, convirtiéndolo en código abierto bajo la licencia GNU GPL hacia finales del verano del año 2002.

La comunidad está ahora dividida en dos sitios con muchas características compartidas:

- La comunidad de desarrollo, centrada alrededor de la web de la Fundación Blender <http://www.blender.org>. Aquí es donde están los proyectos de desarrollo, las listas de funcionalidades y documentación, el repositorio CVS con las fuentes de Blender, todas las fuentes de documentación y foros de discusión. Los desarrolladores de Blender, los que hacen scripts de python, los documentadores, y cualquiera que trabaje en el desarrollo de Blender, generalmente, puede ser encontrado aquí
- La comunidad de usuarios, centrada alrededor de un sitio independiente <http://www.blenderartists.com>. Aquí es donde los artistas, los creadores de juegos y los aficionados a Blender se reúnen para mostrar sus creaciones, obteniendo críticas para mejorarlas, y donde pueden pedir ayuda para obtener una mejor comprensión de las funcionalidades de Blender. Algunos tutoriales de Blender y una base con el conocimiento básico se pueden encontrar también aquí.

Estas dos páginas web no son los únicos recursos de Blender. La comunidad alrededor del mundo ha creado una gran cantidad de sitios independientes, con diferentes idiomas o especializadas en diferentes áreas. Una constante actualización de los recursos para Blender se encuentra en cualquiera de los sitios mencionados anteriormente.

Para una crítica instantanea, hay tres canales de chat disponibles en irc.freenode.net. Puedes unirte a la conversación con tu programa IRC favorito.

Los canales son #blenderchat para discusiones generales sobre Blender, #blenderqa para preguntar sobre el uso de Blender, y #gameblender para discusiones sobre temas relacionados con la creación de juegos con el motor de Blender. Para los desarrolladores, también existen #blendercoders para preguntar y discutir temas de desarrollo, al igual que para cada reunión de los Sábados a las 7; #blenderpython para discutir sobre el API de Python y el desarrollo de scripts; #blenderwiki para preguntas relacionadas con la edición del wiki.

Instalando desde Binarios

Blender está disponible en la web de la de la Fundación (<http://www.blender.org/>) como ejecutable y como código fuente. En la página principal busca la sección de Descargas ('Downloads').

De todas formas, para una correcta comprensión de este libro, es altamente recomendable utilizar la versión incluida en el CDROM. Por lo tanto, cuando en el párrafo anterior mencionábamos "sección de descargas", también asumiremos que puede conseguirse desde el CDROM.

El ejecutable está disponible para 6 sistemas operativos:

- Windows
- Linux
- MacOSX
- FreeBSD
- Irix
- Solaris

Linux actualmente dispone de 4 opciones para las arquitecturas de Intel y PowerPC, con las librerías enlazadas estática o dinamicamente.

La diferencia entre enlazado dinámico y estático es importante. El estático tiene la librería OpenGL compilada en su interior. Esto hace que Blender corra en su sistema, sin utilizar la aceleración por hardware de su tarjeta gráfica. Utilice la versión estática para comprobar si Blender funciona cuando la versión dinámica falle. OpenGL se usa en Blender para todo el dibujado, incluyendo menús y botones. Esta dependencia hace que la instalación, en tu sistema, de la librería OpenGL sea necesaria. No todos los fabricantes de tarjetas gráficas 3D dan soporte a OpenGL, especialmente para las tarjetas más baratas destinadas al mercado de videojuegos.

Desde que los 'renders' se hacen en Blender con el motor de renderizado utilizando la memoria principal y el procesador de tu computadora, una tarjeta gráfica con aceleración por hardware no supone ninguna diferencia en el tiempo de renderizado.

Windows

Instalación Rápida

Descargue el fichero blender-2.3#-windows.exe, donde 2.3# es el número de la versión, de la sección de descargas del sitio web de Blender. Comience la instalación haciendo 'doble click' sobre el fichero descargado. En este proceso se le harán varias preguntas, para las cuales las respuestas que hay por defecto deberían bastar. Después que la instalación haya finalizado, puede comenzar a usar Blender directamente, o bien, puede usar la entrada que se habrá creado en el menú Inicio.

Instrucciones Detalladas

Descargue el fichero blender-2.3#-windows.exe de la sección de descargas del sitio web de Blender. Elija descargarlo, seleccione un directorio y pulse "salvar" (Save). Entonces, con el explorador, dirijase al directorio donde haya guardado el fichero y haga "doble-click" para comenzar la instalación.

El primer diálogo le muestra la licencia de Blender. Si quiere que la instalación continúe tendrá que aceptarla. Después de aceptar la licencia, seleccione los componentes que desee instalar (solo hay uno, Blender) y las acciones adicionales que quiera llevar a cabo. Hay tres: Añadir un enlace directo al menú de Inicio, añadir un icono de Blender al escritorio y asociar los archivos .blend con el programa Blender. Por defecto, todas las acciones están seleccionadas. Si no quiere que se ejecute alguna de esas acciones simplemente deseleccionela. Cuando esté hecho, simplemente presione en siguiente (Next).

Seleccione un lugar para instalar los archivos (el lugar que viene por defecto debería valer), y presione en siguiente (Next) para instalar Blender. Presione cerrar (Close) cuando la instalación haya terminado.

Por último, se le preguntará si quiere que el programa de instalación ejecute a continuación Blender. Una vez que esté instalado Blender puede ejecutarse desde el menú de Inicio (una entrada se ha creado por el programa de instalación con el nombre de "Blender Foundation") o bien haciendo "doble-click" en un archivo de Blender (*.blend).

OSX

Instalación

Descargue el fichero blender-2.3#-darwin-6.6-powerpc.dmg de la sección de descargas del sitio web de Blender. Descomprímalo haciendo "doble-click" en el archivo. Esto abrirá un directorio con varios ficheros.

Desde que Blender usa OpenGL para dibujar su interfaz, y MacOS X también lo utiliza para dibujar su escritorio, necesita verificar primero que dispone de suficiente memoria VRAM en su sistema. Por debajo de 8 MB de memoria VRAM Blender no funcionará. Por encima de 16 MB de memoria VRAM necesitará configurar su sistema para que tenga "miles de colores" (1000s of colors) en (System Preferences -> Displays).

Ahora puede usar Blender simplemente con hacer "doble-click" en el icono de Blender. O arrastrar el icono de Blender al panel principal (Dock) para crear un acceso directo. Blender, por defecto, se iniciará en una pequeña ventana. Utilice el botón "+" en la cabecera de la ventana para maximizarlo. Más consejos y avisos sobre la versión para MacOS X se encuentran en el archivo OSX tips.rtf en el directorio donde esté Blender instalado.

Linux

Instalación Rápida

Descargue el fichero blender-2.3#-linux-glibc#.#-ARCH.tar.gz de la sección de descargas del sitio web de Blender. Aquí 2.3# es la versión de Blender, #.# es la versión de la librería glibc y ARCH es la arquitectura de la computadora, que puede ser i386 o powerpc. Debería escoger la correspondiente a su sistema, recuerde elegir entre enlazado dinámico o estático.

Desempaquete el fichero en la localización de su elección. Esto creará un directorio denominado blender-2.3#-linux-glibc#.#-ARCH, en el cual encontrará el binario de blender.

Para comenzar Blender únicamente abre una consola y ejecuta el comando ./blender, por supuesto cuando el servidor gráfico 'X' esté funcionando.

Instrucciones detalladas

Descargue el fichero blender-2.3#-linux-glibc#.#-ARCH.tar.gz de la sección de descargas del sitio web de Blender. Elija descargarlo, seleccione un directorio y pulse "salvar" (Save). Entonces navegue hasta la localización que haya elegido para la instalación de blender (p.ej. /usr/local/) y desempaquete el fichero (con el comando tar xzf /ruta/donde/está/blender-2.3#-linux-glibc#.#-ARCH.tar.gz). Si usted desea, puede renombrar el directorio resultante del desempaquetado blender-2.3#-linux-glibc#.#-ARCH a algo más corto, como por ejemplo blender.

Ahora Blender está instalado y se puede ejecutar desde la línea de comandos (consola) simplemente poniendo ./ruta/donde/está/blender y presionando la tecla "enter". Si usa KDE o GNOME, puede comenzar Blender usando el explorador de archivos que utilice habitualmente navegando hasta donde esté el ejecutable de Blender y haciendo "doble-click" en él.

Si usa Sawfish, puede que quiera añadir una línea como esta ("Blender" (system "blender &")) a su archivo .sawfish/rc de configuración.

Añadir un icono para blender en KDE

1. Seleccione el "Editor de Menú" en el submenú System del menú de KDE.
1. Seleccione el submenú etiquetado como "Gráficos" en la lista.
1. Pinche en el botón "Nuevo dispositivo". Una caja de diálogo aparecerá para que elija un nombre. Cree el nombre y pinche en "OK". "Blender" o "Blender 2.3#" serían unas elecciones bastante lógicas, pero elegir otro cualquiera no afectará al funcionamiento del programa.
1. Será devuelto al listado del menú, y el submenú "Gráficos" se expandirá, con su nueva entrada resaltada. En la sección de la derecha, asegúrese de que los siguientes campos son rellenados: "Nombre", "Comentario", "Comando", "Tipo" y "Ruta de trabajo".
 - El campo "Nombre" debería estar ya rellenado, pero puede modificarlo en cualquier momento.
 - Rellene el campo "Comentario". Aquí es donde define la etiqueta que aparecerá sobre el icono cuando pase con el puntero por encima.
 - Pinche en el icono de la carpeta al final del campo "Comando" para buscar el icono de blender. Seleccione el icono y pulse en "OK" para volver al "Editor de Menú".
 - El campo "Tipo" debería ser "Application".
 - El campo "Ruta de trabajo" debería ser el mismo que el del campo "Comando", con el nombre del programa suprimido. Por ejemplo, si en el campo "Comando" pone `/home/user/blender-publisher-#.##-linux-glibc#.##-ARCH/blender`, el campo "Ruta de trabajo" debería ser `/home/user/blender-publisher-#.##-linux-glibc#.##-ARCH/`.
1. Pinche en "Aplicar" y cierre el "Editor de Menú".

Para añadir un enlace a Blender en el panel de KDE, pulse el botón derecho del ratón sobre un espacio vacío del panel, y entonces pulse "Añadir", luego "Botón", después "Gráficos" y por último selecciona "Blender" (o el nombre que haya elegido en el paso 3). Alternativamente, puede navegar a través del submenú "Panel de configuración" del menú de KDE y hacer "Añadir", "Botón", "Gráficos" y "Blender".

Para añadir un icono para Blender en el escritorio, abra el navegador Konqueror (que se puede encontrar en el Panel, o en submenú "Sistema" del menú de KDE) y navegue hasta el icono de Blender que se encuentra donde descomprimió por primera vez el programa. Pinche y seleccione el icono y muevalo desde el navegador hasta un espacio vacío de su escritorio. Se le preguntará si quiere copiarlo, moverlo o enlazarlo en esa localización. Elija enlazarlo.

Añadir un icono para blender en GNOME

1. Selecciona "Editar menús" en el submenú Panel del menú de GNOME.
1. Selecciona el submenú "Gráficos", y pinche en el botón "Nuevo dispositivo".
1. En el panel derecho, rellene los campos "Nombre:", "Comentario:" y "Comando:". Rellene el campo "Nombre:" con el nombre del programa, por ejemplo "Blender". Puede darle el nombre que más le guste, solo es lo que aparecerá en el menu, pero no afectará a la funcionalidad del programa. Rellene el campo "Comentario:" con un comentario descriptivo. Esto es lo que se mostrará en las etiquetas emergentes. Rellene el campo "Comando:" con la ruta hasta el programa blender, por ejemplo, `/home/user/blender-publisher-###-linux-glibc###-ARCH/blender`
1. Pinche en el botón "No Icono" para elegir un icono. Puede haber o no un icono para Blender en la localización por defecto. Puede hacer uno, o buscar el que viene con KDE. El icono de KDE debería estar en `/opt/kde/share/icons/hicolor/48x48/apps/blender.png`. Si su directorio de instalación es diferente, puede buscarlo usando este comando en un terminal o una consola: `find / -name "blender.png" -print`
1. Pinche en el botón "Salvar" y cierre el "Editor de Menú".

Para añadir un icono al panel, pincha con el botón derecho del ratón en un espacio vacío del Panel, entonces selecciona "Programas", luego "Gráficos", y después "Blender". Alternativamente, podría pinchar en el menú GNOME, seleccionar "Panel", entonces "Añadir al panel", luego "Lanzar desde el menu", después "Gráficos", y por último "Blender".

Para añadir un icono para Blender en el escritorio, abra el navegador Nautilus (doble-click en el icono de la casa en la esquina superior izquierda de su escritorio, o pinche en el menú GNOME, entonces "Programas", luego "Aplicaciones", y "Nautilus"). Navegue hasta el directorio que contiene el icono del programa Blender. Píche con el botón derecho del ratón, y muevaló hacia el escritorio. Un menú aparecerá preguntándole si quiere copiar, mover, enlazar o cancelar. Seleccione enlazar.

FreeBSD

Intalación

Descargue el archivo `blender-2.3#-freebsd-###-i386.tar.gz` de la sección de descargas del sitio web de Blender. Aquí 2.3# es la versión de Blender, ## es la versión de FreeBSD y i386 es la arquitectura de la computadora.

Para comenzar Blender únicamente abra una consola y ejecute el comando `./blender`, por supuesto cuando el servidor gráfico 'X' esté funcionando.

Irix

Instalación

Descargue el archivo blender-2.3#-irix-6.5-mips.tar.gz desde la sección de descargas de la web de Blender. Aquí 2.3# es la versión de Blender, 6.5 es la versión de Irix y mips es la arquitectura del ordenador.

Para comenzar Blender unicamente abra una consola y ejecute el comando ./blender, por supuesto cuando el servidor gráfico 'X' esté funcionando. Blender se desarrolló originalmente para la plataforma IRIX, pero actualmente esta plataforma no está siendo activamente mantenida en todas sus versiones. Para algunas estaciones de trabajo se ha informado de la existencia de algunos problemas que afectan al rendimiento.

Solaris

Instalación

Descargue el archivo blender-2.3#-solaris-2.8-sparc.tar.gz desde la sección de descargas de la web de Blender. Aquí 2.3# es la versión de Blender, 2.8 es la versión de Solaris y sparc es la arquitectura de la computadora.

Actualmente no hay instrucciones disponibles para Solaris. Por favor utilice los foros del sitio web de Blender para obtener ayuda.

La Interfaz

Si es nuevo en Blender, debería asimilar como trabajar con la interfaz de usuario antes de ponerse a modelar. Los conceptos detrás de la interfaz de Blender hacen que esta no sea muy estándar, sino que sea diferente de los otros paquetes 3D. Especialmente los usuarios de Windows necesitarán coger el truco a la manera en la que Blender maneja los controles, tales como botones o movimientos de ratón. Pero esta diferencia es de hecho la gran fuerza de blender: una vez comprenda la manera en la que funciona Blender, encontrará que puede hacer su trabajo extraordinariamente rápido y de manera muy productiva.

Por otro lado, la interfaz de Blender ha cambiado sustanciosamente desde la versión 2.28 a la versión 2.3, así que incluso los usuarios más experimentados podrán sacar partido a este capítulo.

Conceptos de la Interfaz de Blender

La interfaz de usuario es el mecanismo de interacción mutua entre el usuario y el programa. El usuario se comunica con el programa mediante el teclado y el ratón, el programa responde por medio de lo que muestra en pantalla.

El teclado y el ratón

La interfaz de Blender saca provecho de los ratones de tres botones y una amplia gama de atajos de teclado (en el Volumen II se encuentra una descripción detallada). Si el ratón sólo tiene dos botones, es posible emular el botón central (*interface_functions_settings* describe como). Es posible usar un ratón con rueda, pero no es obligatorio, ya que también existen atajos de teclado que cumplen la misma función. Este libro asume las siguientes convenciones para describir la entrada de usuario:

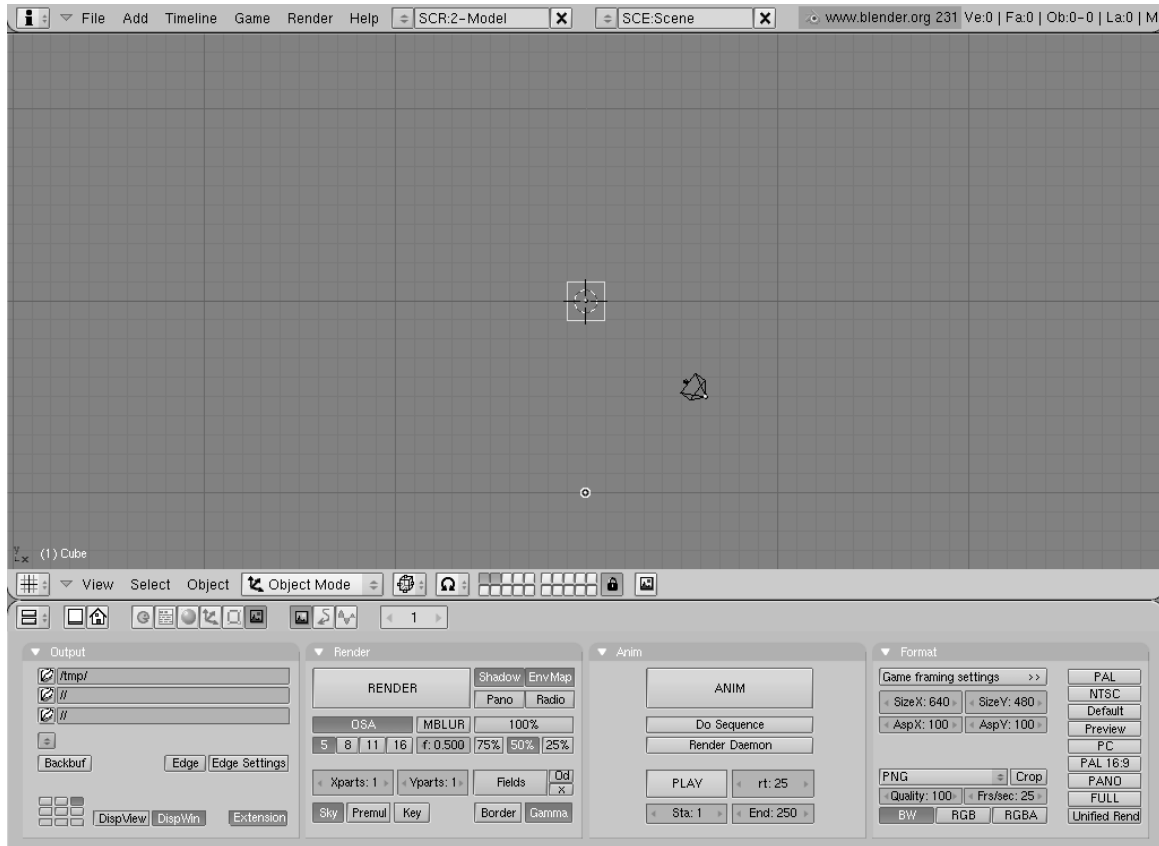
- Los botones del ratón se abrevian como **BIR** (botón izquierdo del ratón), **BMR** (botón medio del ratón) y **BDR** (botón derecho del ratón).
- Si el ratón tiene una rueda, **BMR** se refiere a hacer click con la rueda como si ésta fuera un botón, mientras que **RR** significa girar la rueda.
- Las letras de los atajos se nombran añadiendo **TECLA** a la letra, es decir **TECLAG** se refiere a la letra g en el teclado. Las teclas pueden ser combinadas con los modificadores **SHIFT**, **CTRL** y/o **ALT**. Generalmente para las teclas modificadas el sufijo **TECLA** se descarta, por ejemplo **CTRL-W** o **SHIFT-ALT-A**.
- **NUM0** a **NUM9**, **NUM+** y así sucesivamente, se refiere a las teclas ubicadas en el teclado numérico. Generalmente NumLock debería ser activada.
- En el caso de otras teclas, se les refiere usando sus nombres, tal como **ESC**, **TAB**, **F1** a **F12**.
- Otras teclas especiales para tener en cuenta son las teclas de dirección, **ARRIBA**, **ABAJO**, **IZQUIERDA** y **DERECHA**.

Dado que Blender hace un uso tan extensivo del ratón y el teclado, entre los usuarios se ha hecho popular una "regla de oro": mantener una mano en el ratón y la otra en el teclado! Si normalmente usa un teclado que es considerablemente diferente de la distribución de teclas inglesa, podría ir pensando en cambiar a uno con ésta o la distribución americana para el

trabajo con Blender. Las teclas de mayor uso están agrupadas de tal manera que pueden ser alcanzadas con la mano izquierda en posición estándar (dedo índice sobre **TECLAF**) en la distribución de teclado inglesa. Se asume que se usa el ratón con la mano derecha.

El sistema de ventanas

Ahora es tiempo de iniciar Blender y empezar a explorar.

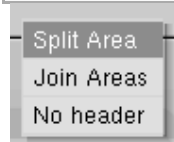


La escena por defecto de Blender.

La escena por defecto de Blender. muestra la vista de pantalla que debe obtenerse al haber iniciado Blender(excepto por las flechas y el texto añadido). Por defecto está separada en tres ventanas: el menu principal en la parte alta, la vista 3D grande y la Ventana de Botones abajo. La mayoría de las ventanas tiene un encabezado (la franja con un fondo gris claro que contiene botones con iconos - por esto se le llamará, también, *Barra de Herramientas*); si está presente, el encabezado puede ubicarse en la parte alta (como en la Ventana de Botones) o en la parte baja (como en la vista 3D) del área de una ventana. Si se mueve el ratón sobre una ventana, es posible notar que su encabezado cambia a un gris más claro. Esto significa que ésta está "enfocada"; todos los atajos de teclado que se presionen afectarán ahora el contenido de esta ventana. Es posible personalizar el sistema de ventanas de Blender para satisfacer las necesidades y gustos del usuario. Una ventana nueva puede ser creada fraccionando una existente. Esto se hace "enfocando" la ventana que se quiere fraccionar (poner el puntero del ratón dentro de esta), hacer click en el borde con **BCR** o **BDR** y seleccionar *Split Area (El menu de Split para crear ventanas nuevas.)*. Ahora es posible ubicar el nuevo borde haciendo click con **BIR**, o cancelar el fraccionamiento presionando la tecla **ESC**. La nueva ventana será

un clon de la ventana que fue fraccionada, pero ahora puede ser convertida a una ventana de otro tipo, o hacer que muestre la escena desde otro punto de vista.

Elementos de la Interface: Las etiquetas en los botones de la interface, los elementos de menú y, en general, cualquier texto mostrado en la pantalla se resalta en este libro de esta manera.

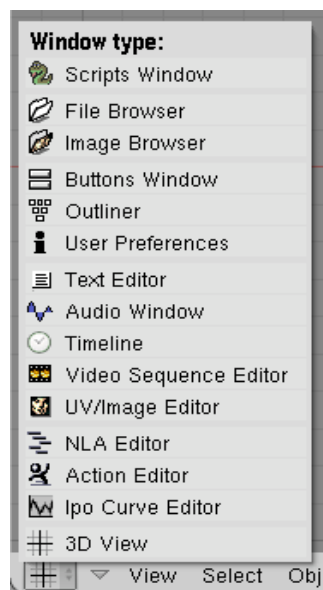


El menu de Split para crear ventanas nuevas.

Un nuevo borde vertical es creado al escoger *Split Area* en un borde horizontal y viceversa. Cada ventana puede ser redimensionada arrastrando un borde con el **BIR**. Para reducir el número de ventanas, se hace click en un borde entre dos ventanas con **BCR** o **BDR** y se escoge *Join Areas*. La ventana resultante recibe las propiedades de la ventana enfocada previamente. Para ajustar la posición de un encabezado se hace click con el **BDR** sobre el encabezado y se escoge *Top* o *Bottom*. También es posible ocultar el encabezado seleccionando *No Header*, pero esta opción sólo se recomienda si todos los atajos de teclado relativos se conocen. Un encabezado oculto se puede volver a mostrar haciendo click en el borde de la ventana con **BCR** o **BDR** y seleccionando *Add Header*.

Tipos de ventanas

El marco de cada ventana puede contener diferentes tipos y conjuntos de datos, dependiendo de lo que se está haciendo. Estos pueden incluir modelos 3D, animación, materiales de la superficie, scripts Python, y así sucesivamente. Se puede seleccionar el tipo de cada ventana haciendo click en el botón de más a la izquierda en su encabezado con el **BIR** (*El menú de selección del tipo de ventana.*).



El menú de selección del tipo de ventana.

Se explicarán las funciones y el uso de cada tipo de ventana a lo largo del libro. Por ahora sólo se prestará atención a los tres tipos de ventana que se encuentran en la escena por defecto de Blender:

- *Vista 3D*

Provee una vista gráfica de la escena en la cual se está trabajando. Es posible ver la escena desde cualquier ángulo con una variedad de opciones; para información detallada, vease *interface_3d*. Tener varias vistas en la misma pantalla es útil si se desea observar el efecto de los cambios desde distintas perspectivas al mismo tiempo.

- *Buttons Window(Ventana de Botones)*

Contiene la mayoría de las herramientas para editar objetos, superficies, texturas, luces y mucho más. Esta ventana se usa constantemente si los atajos de teclado no se saben de memoria. Por supuesto, es posible tener más de una ventana de estas, cada una con un conjunto de herramientas diferente.

- *Preferencias de Usuario (Menu Principal)*

Usualmente esta ventana está oculta, así que sólo la parte del menú está visible - para detalles, ver la *interface_functions_settings*. Raramente se usa, pues contiene ajustes de la configuración global.

En Blender 2.30 hay varias novedades. Primero que todo, los encabezados de ventana tienden a ser mucho más limpios, menos saturados por botones y los menús ahora están presentes en muchos encabezados. La mayoría de los encabezados, inmediatamente después del botón de menú "Tipo de Ventana", exhibe un grupo de menús; esta es una de las nuevas características principales de la interface 2.30. Los menús ahora permiten acceder directamente a muchas de las características y comandos que anteriormente sólo eran accesibles por medio de atajos de teclado o botones misteriosos. Los menús pueden ser ocultados y mostrados por medio del botón triangular junto a ellos. Los menús no son sólo dependientes de la ventana (cambian con el tipo de ventana) sino también dependientes del contexto (cambian de acuerdo con el objeto seleccionado) así que siempre son muy compactos, mostrando sólo las acciones que pueden ser realizadas. Todas las entradas de menú muestran el atajo de teclado relativo, si lo hay. El flujo de trabajo de Blender se optimiza al máximo cuando se usan atajos de teclado. Así que el resto de este libro ofrecerá mayormente atajos de teclado, en lugar de elementos de menú. De todas formas los menús son muy preciados, pues ofrecen una visión general, hasta donde es posible, de todas las herramientas y comandos que Blender ofrece. Una característica de las ventanas que a veces resulta práctica para hacer edición precisa es la de maximizar a pantalla completa. Si se usa el elemento de menú *View>Maximize Window* apropiado o el atajo de teclado **CTRL-ABAJO**, la ventana enfocada se extenderá para llenar la pantalla completa. Para regresar al modo de tamaño normal, se usa el elemento de menú *View>Tile Window* o **CTRL-ARRIBA**.

Contextos, Paneles y Botones

Los botones de Blender son mucho más emocionantes que los de la mayoría de otras interfaces de usuario y se volvieron aún mejores en la versión 2.30. Esto es mayormente debido al hecho de que son basados en vectores y dibujados en OpenGL, lo que los hace

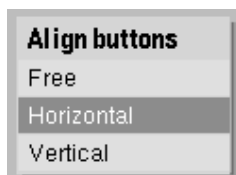
elegantes y permite hacer zoom en ellos. Los botones se hallan agrupados mayormente en la Ventana de Botones. Desde Blender 2.3 la Ventana de Botones muestra *seis* contextos principales, los cuales se pueden seleccionar por medio de alguno de los botones en el primer grupo en el encabezado (*Contextos y Sub-Contextos*), a su vez cada uno de estos puede ser subdividido en un número variable de sub-contextos, a los cuales se puede acceder a través de los botones del segundo grupo en el encabezado (*Contextos y Sub-Contextos*):



Contextos y Sub-Contextos

- *Lógica* - atajo F4
- *Scripts* - sin atajo
- *Shading* - atajo F5
- *Lámpara* - sin atajo
- *Material* - sin atajo
- *Textura* - atajo F6
- *Radiosidad* - sin atajo
- *Mundo* - atajo F8
- *Objeto* - atajo F7
- *Edición* - atajo F9
- *Escena* - atajo F10
- *Renderizado* - sin atajo
- *Animación/Reproducción* - sin atajo
- *Sonido* - sin atajo

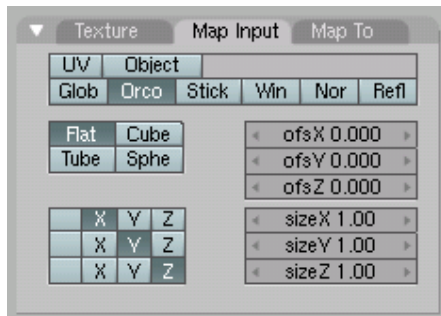
Usualmente, una vez el contexto ha sido seleccionado por el usuario, el subcontexto es determinado por Blender basándose en el objeto activo. Por ejemplo, en el contexto "Shading", si una lámpara está seleccionada, el subcontexto muestra los Botones de Lámpara, si una malla u otro objeto renderizable está seleccionado, entonces el subcontexto activo vendrá a ser el de Botones de Material y si una cámara está seleccionada el subcontexto activo es Mundo. La novedad más notable en la interface es probablemente la presencia de *Paneles* para agrupar lógicamente los botones. Cada panel es del mismo tamaño. Pueden ser movidos haciendo click con el **BIR** en su encabezado y arrastrando. Los paneles pueden ser alineados haciendo click con el **BDR** en la ventana de botones y escogiendo la distribución deseada en el menú que aparece (*Menu de la Ventana de Botones*).



Menu de la Ventana de Botones.

La **RR** desplaza los paneles en su dirección de alineamiento, **CTRL-RR** y **CTRL-BCR** hacen zoom de acercamiento y alejamiento. Paneles individuales pueden ser colapsados/expandidos haciendo click con el **BIR** en el triángulo ubicado a la izquierda de su encabezado. Los paneles particularmente complejos se organizan en *Fichas*. Haciendo click con el **BIR** en una ficha en el encabezado del panel cambian los botones mostrados (*Paneles con fichas*). Las

fichas se pueden "desligar" de un panel para formar paneles independientes haciendo click sostenido con el BIR en su encabezado y arrastrándolas fuera. De manera similar, paneles separados se pueden convertir en un solo panel con fichas soltando el encabezado de un panel sobre otro.



Panel con fichas.

Como un último elemento de interface, hay varios tipos de botones que se disponen en las fichas de los paneles:

- Botones de Operación.

Estos son botones que efectúan una operación cuando se les hace click (con el BIR, como todos los botones). Pueden ser identificados por el color pardusco que tienen en el esquema visual por defecto de Blender. (*Un botón de operación*).



Un botón de operación

- Botón Alternante.

Los botones alternantes vienen en varios tamaños y colores (*Botones alternantes*). Los colores verde, violeta y gris no modifican la función, sólo ayudan a facilitar la agrupación visual y reconocer el contenido de la interface más rápidamente. Hacer click en un botón de estos no genera acción alguna, sólo alterna un estado entre "on" y "off".

Algunos botones también tienen un tercer estado que se puede identificar por que el texto se vuelve amarillo (el botón Ref en *Botones alternantes*). Usualmente el tercer estado significa "negativo" y el estado normal "on" significa "positivo".



Botones alternantes

- Botones de Radio.

Los botones de radio son grupos particulares de botones alternantes mutuamente excluyentes. Tan sólo uno de estos botones puede estar "on" en un momento dado.

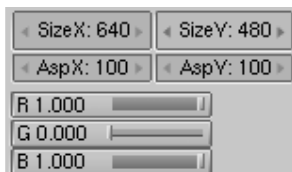
- Botones Numéricos.

Los botones numéricos (*Botones numéricos*) pueden ser identificados por sus títulos, que contienen ":" seguido por un número. Los botones numéricos son manipulados de varias formas:

Para incrementar el valor, se hace click con el **BIR** en la parte derecha del botón, donde se observa un triángulo pequeño; para decrementarlo, se hace click en la parte izquierda del botón, donde se encuentra otro triángulo.

Para cambiar el valor en un rango más amplio, se hace click sostenido con el **BIR** y se arrastra el ratón a la derecha o a la izquierda.

Si mantiene oprimida la tecla **CTRL** mientras lo hace, el valor cambia con saltos grandes; si sostiene oprimida la tecla **SHIFT**, tendrá un control más sutil sobre los valores. La tecla **ENTER** presta la misma utilidad que el **BIR**.



Botones numéricos

Es posible ingresar un valor usando el teclado si se sostiene la tecla **SHIFT** y se hace click con el **BIR**. Presionando la combinación **SHIFT-RETROCESO** se elimina el valor contenido en el botón; con **SHIFT-IZQUIERDA** se mueve el cursor al inicio; y **SHIFT-DERECHA** mueve el cursor al final. Presionando **ESC** se restaura el valor original. Algunos botones de número contienen un deslizador y no sólo un número con triángulos laterales. El mismo método de operación se aplica para estos, excepto que el click con el **BIR** se debe efectuar en la parte izquierda o derecha del deslizador, en tanto que hacer click en la etiqueta o el número automáticamente iniciará el modo de entrada por teclado.

- Botones de Menú.

Los botones de menú se usan para escoger de una lista de elementos creada dinámicamente. Los botones de menú son usados principalmente para enlazar "DataBlocks" entre sí. (DataBlocks son estructuras como Mallas, Objetos, Materiales, Texturas y otros; al enlazar un material a un objeto, éste queda asignado). Un ejemplo de este tipo de bloque de botones se muestra en *Botones de enlace de DataBlocks*. El primer botón (con los diminutos triángulos apuntando arriba y abajo) abre un menú que permite seleccionar el DataBlock al cual efectuar el enlace, esto se hace sosteniendo oprimido el **BIR** y liberándolo cuando el puntero se halle sobre el elemento deseado. El segundo botón muestra el tipo y nombre del DataBlock enlazado y permite editar su nombre después de hacer click con el **BIR**. El botón con la "X" elimina el enlace, el botón con el "carro" genera un nombre automáticamente para el DataBlock y el botón con la "F" especifica que el DataBlock deberá ser almacenado en el archivo aún si no está siendo usado (no tiene enlace alguno).

Objetos sin enlace alguno: Los datos sin enlace sólo se pierden hasta que se cierra Blender. Esta es una poderosa característica que permite deshacer errores. Si un objeto es eliminado el material que tiene asignado pasa a estar sin enlace, pero aún existe! Solo se debe re-enlazar a otro objeto o presionar el botón "F".



Botones de enlace de DataBlocks

Caja de Herramientas (ToolBox)

Al presionar la tecla **ESPACIO** en la Vista 3D, o sostener el **BIR** o el **BDR** con el ratón quieto durante más de medio segundo se abre la Caja de Herramientas. Esta contiene 6 contextos principales dispuestos en dos líneas, cada uno de los cuales abre menú y submenús. Tres de estos contextos abren los mismos tres menús presentes en el encabezado de la vista 3D; los otros tres, *Add* permite añadir objetos nuevos a la escena mientras *Edit* y *Transform* muestran todas las operaciones que se pueden efectuar sobre el(los) objetos seleccionado(s). (*La Caja de Herramientas*).

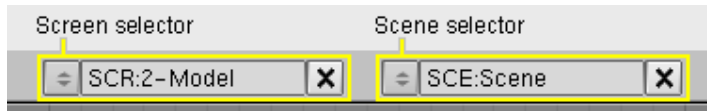


La Caja de Herramientas

Pantallas (Screens)

La flexibilidad de las ventanas de Blender permite crear entornos de trabajo personalizados para diferentes tareas, tales como modelado, animación y creación de scripts. A menudo resulta útil intercambiar rápidamente entre diferentes entornos dentro del mismo archivo. Esto es posible al crear varias Pantallas: Todos los cambios a las ventanas que se describen en *El sistema de ventanas* y *Tipos de ventanas* se almacenan en una pantalla, de modo que si las

ventanas de una pantalla son modificadas, las otras no resultarán afectadas en lo absoluto. Pero la escena en la cual se está trabajando es la misma en todas las pantallas. Blender incluye por defecto tres pantallas diferentes; están disponibles a través del Botón de Menú *SCR* en el encabezado de la Ventana de Preferencias de Usuario mostrado en *Selectores de Pantalla y Escena*. Para cambiar a la pantalla siguiente, alfabéticamente, se debe presionar la combinación CTRL-DERECHA; para cambiar a la pantalla previa, alfabéticamente, se presiona la combinación CTRL-IZQUIERDA.



Selectores de Pantalla y Escena

Escenas

También es posible tener varias escenas dentro del mismo archivo de Blender. Las escenas pueden usar objetos de otras o ser completamente independientes entre sí. Con el botón de menú *SCE*, ubicado en el encabezado de la Ventana de Preferencias de Usuario, se puede crear o seleccionar una escena (Selectores de Pantalla y Escena). *Cuando se crea una escena nueva, es posible escoger entre cuatro opciones para controlar su contenido:*

- *Empty* Crea una escena vacía.
- *Link Objects* crea la escena nueva con el mismo contenido de la escena seleccionada actualmente. Los cambios en una escena afectarán igualmente a la otra.
- *Link ObData* crea la nueva escena basada en la escena seleccionada actualmente, con enlaces a las mismas mallas, materiales, etc. Esto significa que la posición de los objetos puede ser cambiada, junto con otras propiedades, pero modificaciones de las mallas, materiales, etc. afectarán igualmente la otra escena, a menos que se indique manualmente la creación de copias con usuario único (single-user copies).
- *Full Copy* crea una escena completamente independiente con copias del contenido de la escena seleccionada actualmente.

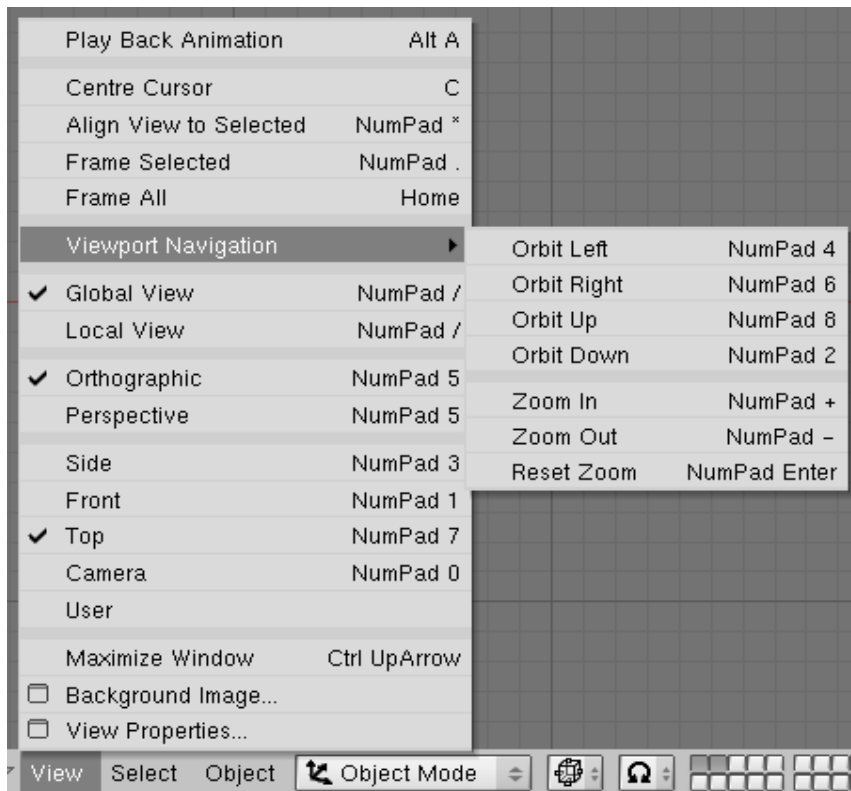
Navegando en el Entorno 3D

Blender te permite trabajar en un espacio tridimensional, pero las pantallas de nuestros monitores son sólo bi-dimensionales. Para poder trabajar en tres dimensiones, debes ser capaz de cambiar tanto tu punto de vista como la dirección en la que ves la escena. Esto es posible en todas las vistas 3D. Aunque vamos a describir la ventana de vista en 3D, la mayoría de las ventanas no-3D usan una serie de funciones equivalentes, por ejemplo es incluso posible arrastrar y escalar una Ventana de Botones y sus Paneles.

La dirección de la vista (rotando)

Blender ofrece tres direcciones de vista por defecto: Lateral, Frontal, and Superior. Como Blender usa un sistema de coordenadas "de mano derecha" con el eje Z apuntando hacia arriba, la vista "Lateral" corresponde a una mirada desde el eje X axis, en la dirección negativa; la vista "Frontal" es desde el eje Y; y la "Superior" desde el eje Z. Puedes seleccionar la dirección de la vista para una ventana 3D con las entradas de Menú *Vista (El menú de vistas de una ventana 3D.)* o presionando los atajos de teclado **NUM3** para "Lateral", **NUM1** para "Frontal", y **NUM7** para "Superior".

Atajos de teclado: Recuerda que la mayoría de los atajos de teclado afectan a la ventana activa, así que comprueba que el cursor del ratón está en el área donde quieres trabajar antes de usar los atajos de teclado!



El menú de vistas de una ventana 3D.

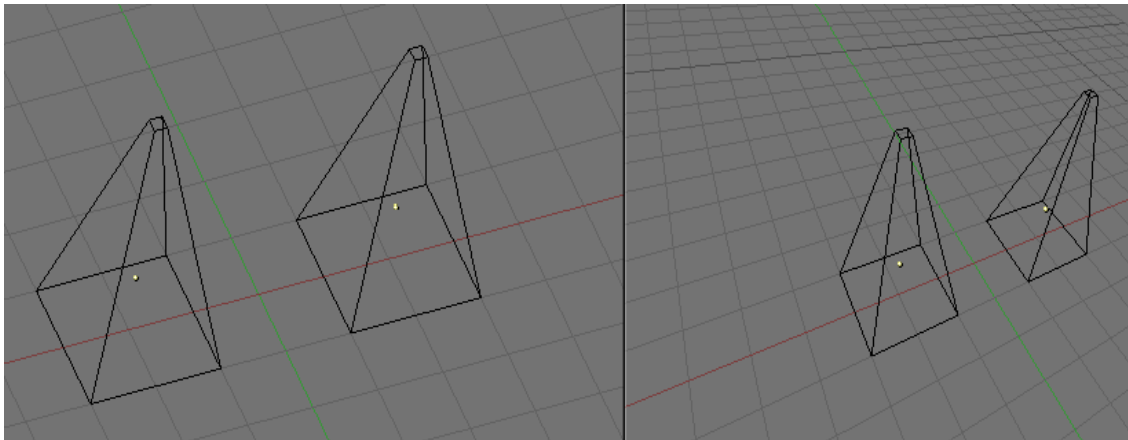
Aparte de estas tres direcciones por defecto, la vista puede ser girada con cualquier ángulo que necesites. Haz clic y arrastra **MMB** en el área de la ventana: si comienzas en el medio de la ventana y te mueves hacia arriba y hacia abajo o hacia la derecha y la izquierda, la vista gira alrededor del centro de la ventana. Si comienzas en el borde y no te mueves hacia el centro, puedes girar alrededor de tu eje de vista. Juega un poco con esta función hasta que controles su manejo. Para cambiar el ángulo de vista en pasos sucesivos, usa **NUM8** y **NUM2**, que corresponden al arrastre vertical. **MMB** o usa **NUM4** y **NUM6**, que corresponden al arrastre horizontal **MMB**.

Trasladando y Escalando la Vista

Para trasladar la vista, pulsa **SHIFT** y arrastra **MMB** en la Ventana 3D. Para hacerlo en pasos sucesivos, usa las combinaciones de teclas **CTRL-NUM8**, **CTRL-NUM2**, **CTRL-NUM4** y **CTRL-NUM6** como con las rotaciones. Puedes escalar la vista (acercarte y alejarte) pulsando **CTRL** y arrastrando **MMB**. Los atajos de teclado son **NUM+** y **NUM-**. Los sub-menús *View>>Viewport Navigation* contienen también estas funciones.

Proyección en Perspectiva y Ortográfica

Cada Ventana 3D soporta dos tipos diferentes de proyección. Los mostramos en *Ortográfica (izquierda)* y *perspectiva (derecha)*: ortográfica (izquierda) y perspectiva (derecha).



Ortográfica (izquierda) y perspectiva (derecha).

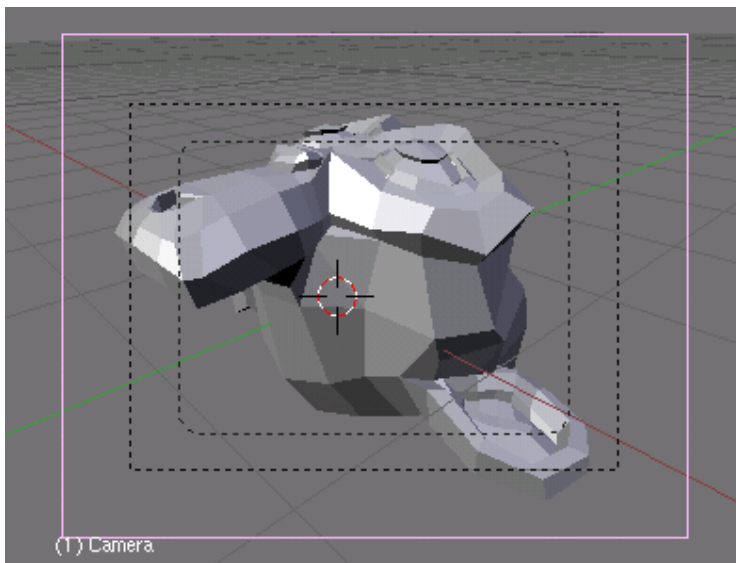
Nuestro ojo está acostumbrado a ver en perspectiva porque los objetos distantes parecen más pequeños. La proyección Ortográfica, a menudo, parece incorrecta al principio porque los objetos permanecen del mismo tamaño independientemente de su distancia: es como ver la escena desde un punto infinitamente distante. Al contrario, la vista ortográfica es muy útil (es el modo por defecto en Blender y en la mayoría de aplicaciones 3D), porque ofrece un aspecto más "técnico" de la escena, facilitando el dibujo y los cálculos de proporciones.

Perspectiva and Ortográfica: La vista en Perspectiva se construye geoméricamente de esta manera: tienes una escena en 3D y tu eres un observador situado en un punto O. La escena con perspectiva 2D se construye situando un plano, una hoja de papel donde la escena 2D se dibuja en frente del punto O, perpendicular a la dirección de la vista. Para cada punto P en la

escena 3D se dibuja una línea, pasando entre O y P. El punto de intersección S entre esta línea y el plano es la proyección en perspectiva de ese punto. Proyectando todos los puntos P de la escena obtendremos una vista en perspectiva. En una proyección ortográfica, también llamadas "ortonormales", por otra parte, tienes una dirección de vista pero no un punto de vista O. La línea que se dibuja atraviesa el punto P de forma paralela a la dirección de la vista. Las intersecciones S entre la línea y el plano conforman la proyección ortográfica. Y proyectando todos los puntos P de la escena obtenemos la vista ortográfica.

Para cambiar la proyección de una ventana 3D Viewport, elige *Vista>>Ortográfica* o *Vista>>Perspectiva* entradas de Menú (*El menú de vistas de una ventana 3D.*). El atajo de teclado **NUM5** cambia entre los dos modos.

La entrada de Menú *Vista>>Cámara* pone la vista 3D en modo Cámara (Atajo: **NUM0**). Entonces la escena se muestra como será renderizada luego (vea *Demostración de vista de Cámara.*): la imagen renderizada contendrá todo lo que abarque la línea de puntos exterior. Es posible acercarse y alejarse en esta vista, pero para cambiar el punto de vista tendremos que mover o girar la Cámara.



Demostración de vista de Cámara.

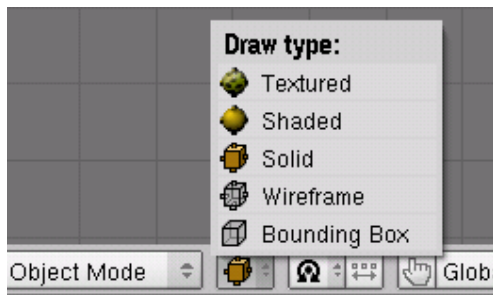
Modo de Dibujado

Dependiendo de la velocidad de tu ordenador, de la complejidad de tu escena y del tipo de trabajo que estés haciendo, puedes alternar entre diferentes modos de dibujado:

- *Texturizado* - Intenta dibujarlo todo tan completamente como sea posible, aunque no es aún una alternativa al renderizado. Nota que si no hay ninguna luz en tu escena todo saldrá negro.
- *Sombreado* - Dibuja superficies sólidas incluyendo el cálculo de iluminación. Como en el modo Texturado, no verás nada sin luces.

- *Sólido* - Las Superficies se dibujan como Sólidos, pero también funciona sin luces.
- *Enrejado* - Los objetos sólo consisten en líneas que hacen reconocibles las formas. Este es el modo de dibujado por defecto.
- *Caja de Límites* - Los objetos no son dibujados en absoluto; en vez de eso, este modo enseña sólo las cajas rectangulares que corresponden al tamaño y a la forma de cada objeto.

El modo de dibujado puede ser elegido con el botón de menú apropiado en la cabecera (*Un botón de modo de dibujado de una vista 3D.*) o con atajos de teclado: **Z**KEY cambia entre Enrejado y Sólido, **SHIFT-Z** cambia entre Enrejado y Sombreado.



Un botón de modo de dibujado de una vista 3D.

Vista Local

En vista local, sólo los objetos seleccionados son mostrados, lo que puede facilitar la edición en escenas complejas. Para entrar en vista local, primero selecciona los objetos que quieres (ver *mesh_modeling_objectmode_selecting*) y después usa la entrada de menú; *Vista>>Vista Local* usa la entrada de menú *Vista>>Vista Global* para volver a la vista global. (*El menú de vistas de una ventana 3D.*). El atajo es **NUM/**, que cambia entre vista Local y Global.

El sistema de Capas

Las escenas en 3D a menudo se hacen mucho más confusas a medida que aumenta su complejidad. Para mantenerlas bajo control, los objetos pueden ser agrupados en "capas", para que sólo las capas que elijas sean mostradas en cualquier momento. Las capas de 3D difieren de las capas que puedes conocer de las aplicaciones gráficas en 2D: no tienen influencia en el orden de dibujado y están ahí (a excepción de algunas funciones especiales) únicamente para ofrecer al modelador un mejor control de las vistas. Blender utiliza 20 capas; tú eliges las que se muestran con los pequeños botones sin título de la cabecera (*Los botones de Capas de una vista 3D.*). Para seleccionar sólo una capa, haz clic en el botón apropiado con **LMB**; para seleccionar más de una, presiona **MAYÚS** mientras haces clic.

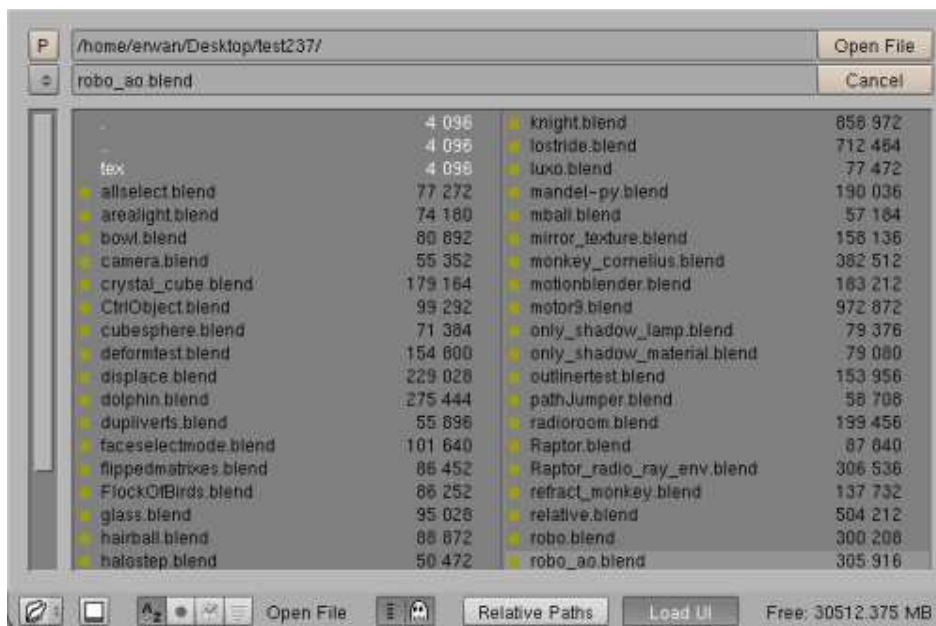


Los botones de Capas de una vista 3D.

Para seleccionar capas con el teclado, presiona desde la tecla **1KEY** hasta la tecla **0KEY** (en el área principal del teclado) para las capas de 1 a 10 (la fila superior de botones), y **ALT-1** hasta **ALT-0** para las capas 11 a 20 (la fila inferior). La tecla **MAYÚS** para selecciones múltiples también funciona con estos atajos. Por defecto, el botón de bloquear a la derecha del bloque de capas está activado; esto significa que los cambios a la visibilidad de las capas afecta a todas las ventanas 3D. Para seleccionar ciertas capas sólo en una ventana, desactiva primero el bloqueo. Para mover los objetos seleccionados a una capa diferente, presiona **MKEY**, selecciona la capa que quieres en el diálogo emergente, y finalmente haz clic en **OK**.

Cargando archivos

Blender usa el formato de archivo *.blend* para salvar casi todo: Objetos, escenas, texturas, e incluso todas tus opciones de la ventana de preferencias del usuario. Para cargar un archivo Blender del disco, presiona **F1**. La ventana activa se transformará temporalmente en la Ventana de selección de archivos como se muestra en la *Ventana de Selección de Archivos - cargando..* La barra de la izquierda puede ser arrastrada con **LMB** para ver el resto de la lista. Para cargar un archivo, selecciónalo con **LMB** y presiona **ENTER**, o simplemente haz clic en él con **MMB**.



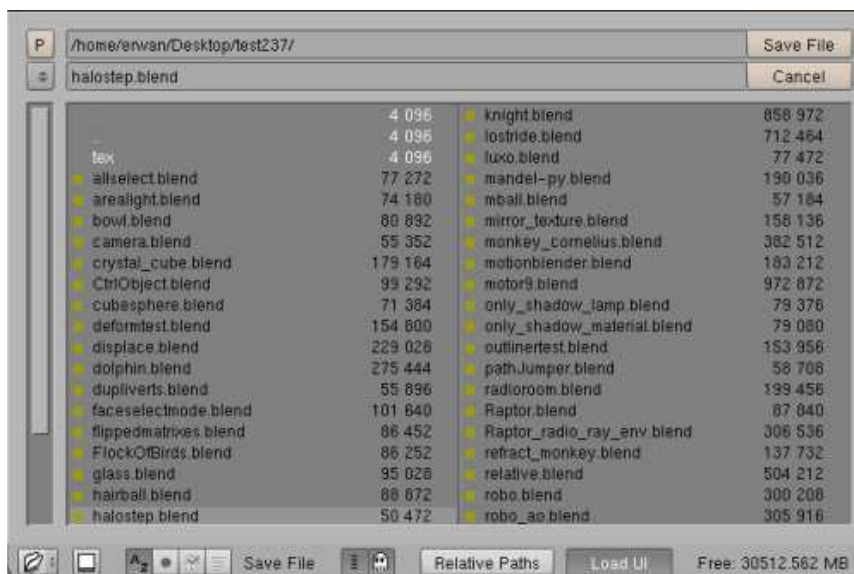
Ventana de Selección de Archivos - cargando.

La caja de texto superior muestra la ruta del directorio actual, y la inferior contiene el nombre de archivo seleccionado. El botón **P** (**PKEY**) te sitúa en el directorio anterior y el botón con la coma mantiene un listado de rutas usadas recientemente. En Sistemas operativos Windows, este último también contiene una lista de todos los discos (C:, D:, etc).

Nota: Blender espera que tú sepas lo que estás haciendo! Cuando cargas un archivo, no se te pregunta si quieres guardar los cambios de la escena que tenías abierta: completar el diálogo de carga de un archivo se considera suficiente confirmación de que no lo has hecho por accidente. Asegúrate de salvar tus archivos.

Guardando archivos


Guardar archivos es como cargarlos: Cuando presionas **F2**, la ventana activa cambia temporalmente a una ventana de selección de archivos, como se muestra en la *Ventana de Selección de Archivos - guardando..* Haz clic en la ventana de edición inferior para escribir un nombre de archivo. Si no lo acabas con ".blend," la extensión es añadida automáticamente. Después presiona **ENTER** para guardar el archivo. Si ya existe un archivo con el mismo nombre, tendrás que confirmar que quieres guardar el archivo en la pregunta de sobre-escribir.



Ventana de Selección de Archivos - guardando.

El diálogo de salvar tiene una pequeña cualidad añadida para ayudarte a crear múltiples versiones de tu trabajo: Presionando **NUM+** o **NUM-** incrementará o disminuirá un número contenido en el nombre del archivo. Para simplemente salvar sobre el archivo actualmente cargado evitando el diálogo de salvar, presiona **CTRL-W** en vez de **F2** y sólo tendrás que confirmar la pregunta.

Renderizando

Esta sección te ofrece sólo un repaso rápido de lo que necesitarás para renderizar tu escena. Encontrarás una descripción detallada de todas las opciones en *chapter_rendering*. Las Opciones de Render están en el contexto *Escena* y en el sub-contexto *Botones de Renderizado* (*Opciones de Renderizado en los Botones de Render.*) al que se accede haciendo clic en , o presionando **F10**.



Opciones de Renderizado en los Botones de Render.

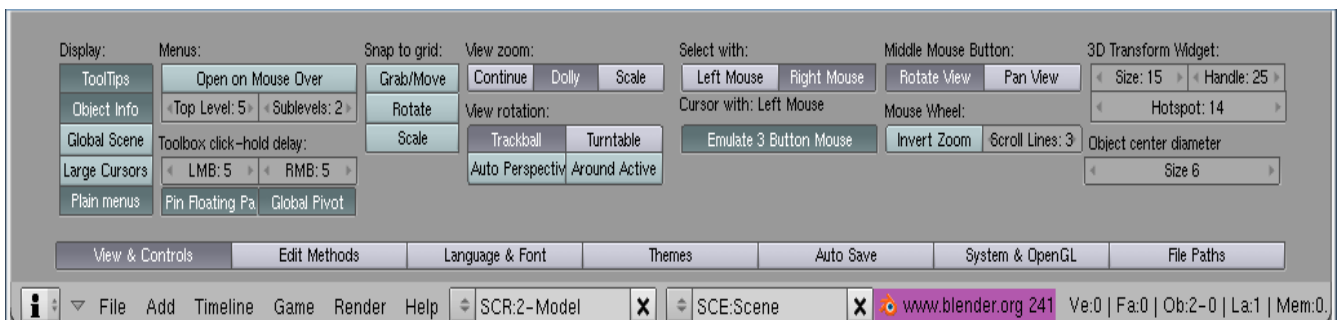
Por ahora sólo nos interesa el panel *Formato*. El tamaño (número de píxeles horizontales y verticales) y el formato de archivo de la imagen a crear se eligen aquí. Puedes fijar el tamaño usando los botones *SizeX* y *SizeY*. Haciendo clic en la caja de selección de debajo (en *Opciones de Renderizado en los Botones de Render.*, está seleccionado "Targa") abre un menú con todos los formatos de salida disponibles para imágenes y animaciones. Para imágenes fijas podemos elegir Jpeg, por ejemplo. Ahora que las opciones están completas, la escena puede ser renderizada presionando el botón *RENDER* en el panel *Render* o presionando **F12**. Dependiendo de la complejidad de la escena, esto puede tardar normalmente entre unos pocos segundos y varios minutos, y el progreso es mostrado en una ventana separada. Si la escena contiene una animación, sólo se renderiza el fotograma actual (Para renderizar la animación completa, ver *rendering_animation*). Si no ves nada en la vista renderizada, asegúrate de que tu escena está construida convenientemente. ¿Tiene luz? Está la cámara situada correctamente, y apunta en la buena dirección? ¿Están visibles todas las capas que quieres renderizar?

Nota: Una imagen renderizada no se salva automáticamente en el disco. Si estás satisfecho con el renderizado, debes salvarlo presionando F3 y usando el diálogo de salvar tal como describimos en Section 3.3.2. La imagen es salvada en el formato que hayas seleccionado previamente en los botones de Display.

Extensiones de archivo: Blender no añade la extensión del formato automáticamente a los archivos de imagen! Tienes que escribir la extensión explícitamente, si la necesitas.

Preferencias de usuario y Temas

Blender tiene unas pocas opciones que no son guardadas con cada archivo, sino que se aplican se aplican a todos los archivos de un usuario. Estas preferencias incluyen detalles de la interfaz del usuario, y propiedades del sistema como el ratón, las fuentes y los lenguajes. Como las preferencias del usuario son raramente necesarias, están escondidas bajo el menú principal. Para hacerlas visibles, baja el borde de la ventana del menú (normalmente el borde superior en la pantalla). Las opciones están agrupadas en siete categorías que pueden ser seleccionadas con los botones violeta mostrados en la *Ventana de Preferencias del Usuario*.



Ventana de Preferencias del Usuario.

Ya que la mayoría de los botones se explican por sí mismos o muestran una útil ayuda si mantienes el cursor parado sobre ellos, no vamos a describirlos en detalle aquí. En lugar de eso, te ofrecemos un repaso de las categorías de Preferencias:

- Vista & Controles

Las opciones concernientes a cómo debe reaccionar la interfaz a las acciones del usuario, como qué método de rotación debe usarse en las vistas 3D. Aquí puedes activar también la emulación del ratón de 3 botones si el tuyo tiene sólo dos. **MMB** puede ser emulado con **ALT-LMB**.

- Métodos de Edición

Te permite especificar los detalles del funcionamiento de ciertos comandos de edición como duplicar.

- Lenguaje & Fuentes

Selecciona una fuente TrueType alternativa para mostrar en la interfaz, y elige entre los lenguajes disponibles para la interfaz.

- Temas

Desde la versión 2.30 Blender permite la utilización de Temas para definir colores personalizados para la interfaz. Puedes crear y manejar temas desde aquí.

- Guardar Automáticamente

Puedes configurar Guardar Automáticamente para tener una copia de emergencia en caso de que algo vaya mal. Estos archivos se llamarán Nombre.blend1, Nombre.blend2, etc.

- Sistema & OpenGL

Debes consultar esta sección si experimentas problemas con los gráficos o con la salida de audio, o si no dispones de un teclado numérico y quieres imitarlo (para portátiles). Además, aquí puedes también configurar el esquema de luces para los modos de dibujo Sólido y Sombreado.

- Directorios de Archivos

Escoge el directorio por defecto para varios diálogos de carga de archivos.

Configurando la escena por defecto

No te gusta la configuración de ventanas por defecto de Blender, o quieres unas opciones específicas de render para cada nuevo proyecto que empieces, o quieres guardar tu Tema? No hay problema. Puedes usar cualquier archivo de escena como opciones por defecto cuando arranca Blender. Convierte la escena en la que estás trabajando en la opción por defecto presionando **CTRL-U**. La escena será copiada en un archivo denominado .B.blend en tu

directorio principal. Puedes borrar el proyecto actual y volver a la escena por defecto en cualquier momento presionando **CTRL-X**. Pero recuerda salvar primero tus cambios a la escena previa!

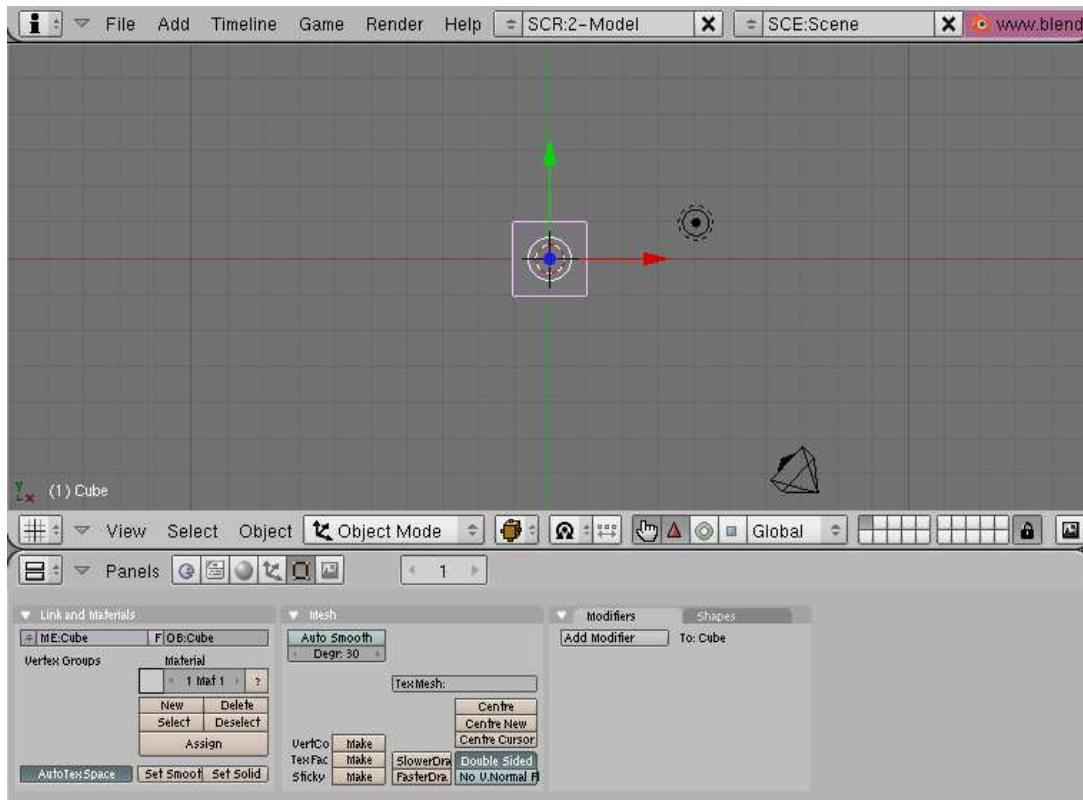
Su Primera Animación en 30 Minutos I

Este capítulo le guiará paso a paso a través de la animación de un pequeño personaje, "el hombre galleta". Describiremos completamente todas las acciones, pero asumiremos que se ha leído enteramente el capítulo La interfaz, y que entiende las convenciones que se van a usar en el libro. En los primeros 30 minutos de este tutorial construiremos un hombre de galleta *estático*. En los siguientes 30 minutos, le proporcionaremos un esqueleto y le animaremos para que camine (walk cycle).

Calentando

Comencemos.

1. Arranque Blender haciendo doble-click en su icono o ejecutelo desde la línea de comandos. Blender se abrirá mostrándole, desde la vista superior (top view), la configuración por defecto: una cámara y un plano. El plano está de color rosa, lo que significa que está seleccionado (*Blender recién iniciado*). Elimine el plano con la Tecla X y confirme su eliminación pinchando en *Erase Selected* en el dialogo que aparecerá.



Blender recién iniciado.

Ahora seleccione la cámara con el RMB y presione la Tecla M. Una pequeña caja de herramientas, como esta *Caja de herramientas para el control de capas.*, aparecerá bajo su puntero, con el primer botón ya seleccionado. En la fila de arriba seleccione el botón más a la derecha y luego pinche en el botón OK. Esto moverá su cámara a la capa (layer) 10.



Caja de herramientas para el control de capas.

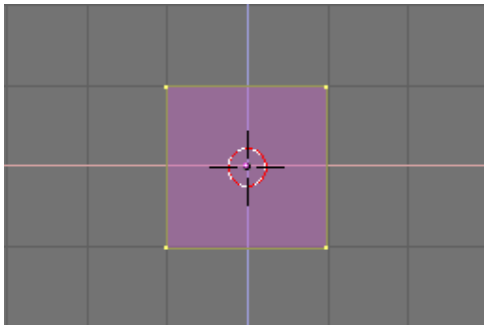
Blender le proporciona 20 capas para ayudarle a organizar su trabajo. Puede ver que capas son visibles actualmente a través del grupo de veinte botones en la barra de herramientas de la ventana 3D (3D windows toolbar) (*Controles de visibilidad de las capas.*). Puede elegir que capa quiere que sea visible con el LMB y seleccionar que varias capas sean visibles a la vez con SHIFT-LMB.



Controles de visibilidad de las capas.

Construyendo el cuerpo

Cambie a la vista frontal (front view) con NUM1 y añada un cubo presionando en la Barra Espaciadora (SPACE) y seleccione el menú *Add*, el submenú *Mesh*, y luego *Cube*. (En el libro usaremos SPACE >> Add >> Mesh >> Cube como abreviatura para esta clase de acciones). Un cubo aparecerá (*Nuestro cubo en Modo Edición, todos los vértices seleccionados.*). Este cubo está en *Modo Edición* (EditMode), un modo en el cual puedes mover los vértices que componen la malla (mesh). Por defecto, todos los vertices están seleccionados (amarillos). Los vértices sin seleccionar son rosas.



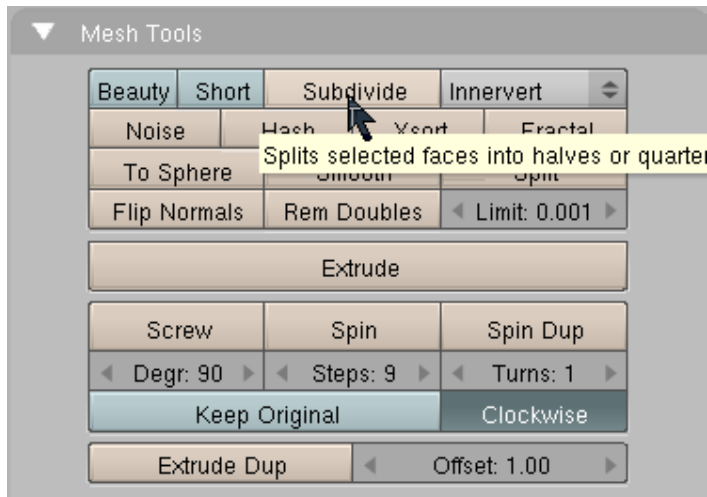
Nuestro cubo en Modo Edición, todos los vértices seleccionados.

Llamaremos a nuestro hombre galleta "Gus". Nuestro primer cometido es construir el cuerpo de Gus modificando nuestro cubo en Modo Edición. Para ver las herramientas que utilizaremos para este propósito, pulse el botón que muestra un cuadrado con vértices amarillos en la cabecera de la ventana (*El botón de la ventana de Edición.*), o presione F9.

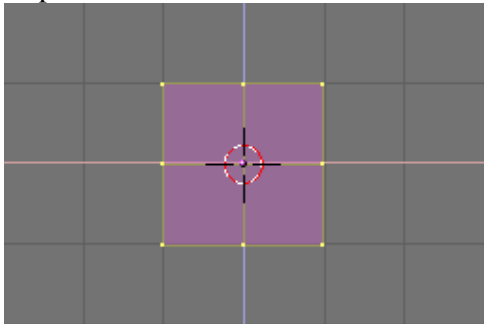


El botón de la ventana de Edición.

Ahora localice el botón *Subdivisión* (Subdivide) en el panel *Herramientas de Malla* (Mesh Tools) y presionele una vez (*El panel Mesh Tools dentro de la ventana de Edición.*). Esto partirá cada lado del cubo en dos, creando nuevos vértices y caras (*El cubo, subdividido una vez.*).



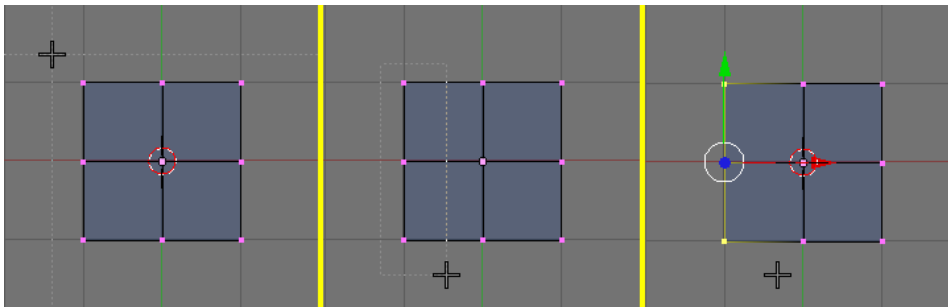
El panel Mesh Tools dentro de la ventana de Edición.



El cubo, subdividido una vez.

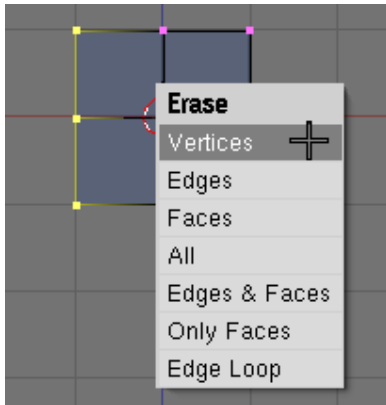
Con el cursor sobre la ventana 3D (3D Window), presione la **Tecla A** para deseleccionar todos los elementos. Los vértices se volverán rosas. Ahora presione la **Tecla B**; el cursor se transformará en un par de líneas grises ortogonales (Box select).

Mueva su ratón sobre la parte superior izquierda del cubo y a continuación presione y mantenga el **LMB**, entonces arrastre el ratón hacia abajo y hacia la derecha para seleccionar todos los vértices que se encuentren a la izquierda. Ahora deje de pulsar el **LMB**. Esta secuencia, la cual le permite seleccionar un grupo de vértices con una caja, está descrita en *La secuencia de seleccionar con una caja un grupo de vértices.*



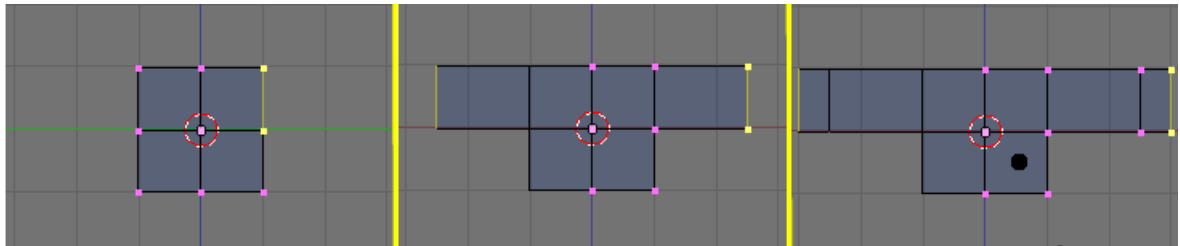
La secuencia de seleccionar con una caja un grupo de vértices.

Ahora presiona la **Tecla X** y, en el menú emergente, selecciona *Vertices* para eliminar los vértices seleccionados (*El menú emergente para borrar (Tecla X) en acción.*).



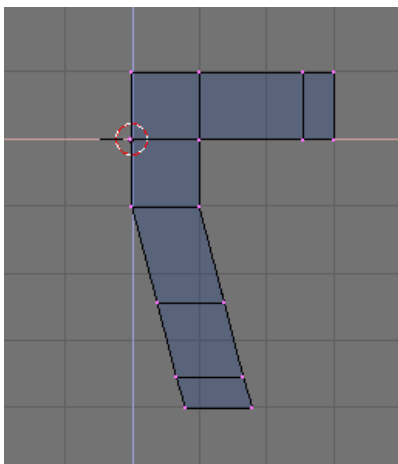
El menú emergente para borrar (**Tecla X**) en acción.

Ahora, usando la secuencia que acaba de aprender, seleccione con una caja los dos vértices de la esquina superior derecha (*Extruyendo los brazos en dos pasos.*, derecha). Presione la **Tecla E** y pinche en la entrada de menú *Extrude* para extruirllos. Esto creará nuevos vértices y nuevas caras que podrá mover y que seguirán al puntero del ratón. Muevalas a la derecha. Para restringir el movimiento horizontal o verticalmente, pulse **MMB** mientras se este moviendo. Puede cambiar a un movimiento sin restricciones pulsando **MMB** de nuevo. Alternativamente puede usar la **Tecla X** para restringir el movimiento al eje x, la **Tecla Y** para restringirlo al eje y, etc. Vamos a crear brazos y piernas a Gus. Mueva esos vértices nuevos un cuadrado y medio a la derecha, entonces pulse **LMB** para fijar su posición. Extruya de nuevo con la **Tecla E** entonces mueva los vértices nuevos otro medio cuadrado a la derecha. *Extruyendo los brazos en dos pasos.* muestra esta secuencia.



Extruyendo los brazos en dos pasos.

Gus debería tener actualmente un brazo izquierdo (ey! Gus nos está mirando). Contruiremos la pierna izquierda de la misma manera, extruyendo los vértices inferiores. Intente hacerlo tal y como se muestra en *Medio cuerpo*.. Usaremos la herramienta de extruir 3 veces para crear la pierna. ¡No se preocupe por los codos, pero necesitaremos una rodilla más tarde!

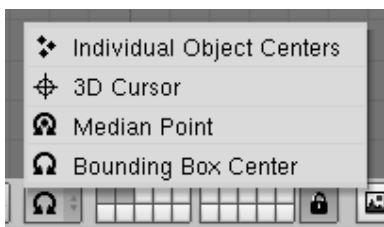


Medio cuerpo.

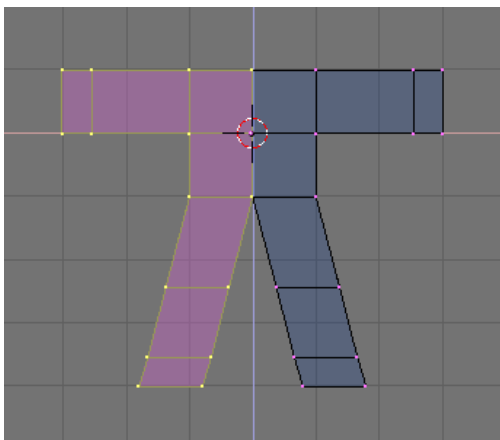
Nota: El CD contiene un fichero .blend con este ejemplo, guardado en varias fases del modelado. El primer fichero, Quickstart00.blend contiene lo que debería haber conseguido hasta ahora. Los pasos subsiguientes son numerados progresivamente, Quickstart01.blend, Quickstart02.blend y así, mientras Quickstart.blend contiene el resultado final. Este estándar se aplica a todos los otros ejemplos en este libro.

Ahora crearemos la otra mitad de Gus:

- Seleccione todos los vértices (Tecla **A**) y elige la entrada *3D Cursor* en el menú *Rotation/Scaling Pivot* (eje o punto de rotación/escalado) de la cabecera de la ventana 3D (*Configure el centro de referencia a el cursor.*).
- Presione **SHIFT-D** para duplicar todos los vértices, aristas y caras seleccionadas. Al duplicar nos encontraremos en modo de movimiento, presione **ESC** para salir de este modo sin mover los vértices.
- Presione la Tecla **M** para abrir el menú *Mirror Axis* (eje de reflexión). Elija *Global X* (eje X global). El resultado es el mostrado en *Voltee la copia de la mitad del cuerpo para obtener un cuerpo completo..*



Configure el centro de referencia a el cursor.

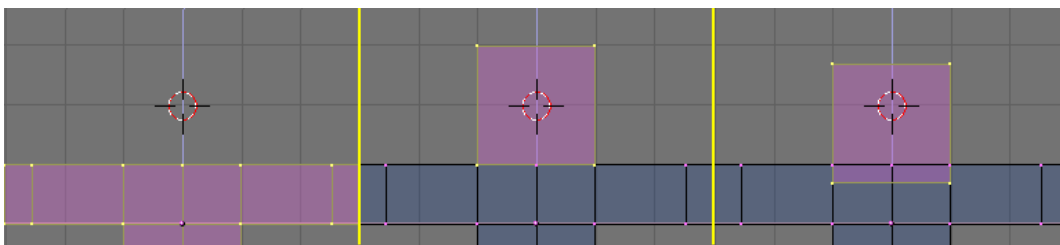


Voltee la copia de la mitad del cuerpo para obtener un cuerpo completo.

- Deseleccione todo y entonces vuelva seleccionarlo presionando la **Tecla A** dos veces, entonces elimine los vértices coincidentes presionando el botón *Remove doubles* (La ventana de botones de edición.). Una ventana aparecerá, notificándole que ocho vértices han sido eliminados.

Gus Necesita una cabeza:

- Mueva exactamente el cursor a un cuadrado de la rejilla por encima del cuerpo de Gus (La secuencia de añadir una cabeza., izquierda). Añada un nuevo cubo aquí (SPACE>>ADD>>Mesh>>Cube).
- Presione la **Tecla G** para cambiar al Modo Movimiento y mover los vértices creados hacia abajo restringiendo el movimiento con **MMB**, más o menos una tercera parte de una unidad de rejilla (La secuencia de añadir una cabeza., derecha).



La secuencia de añadir una cabeza.

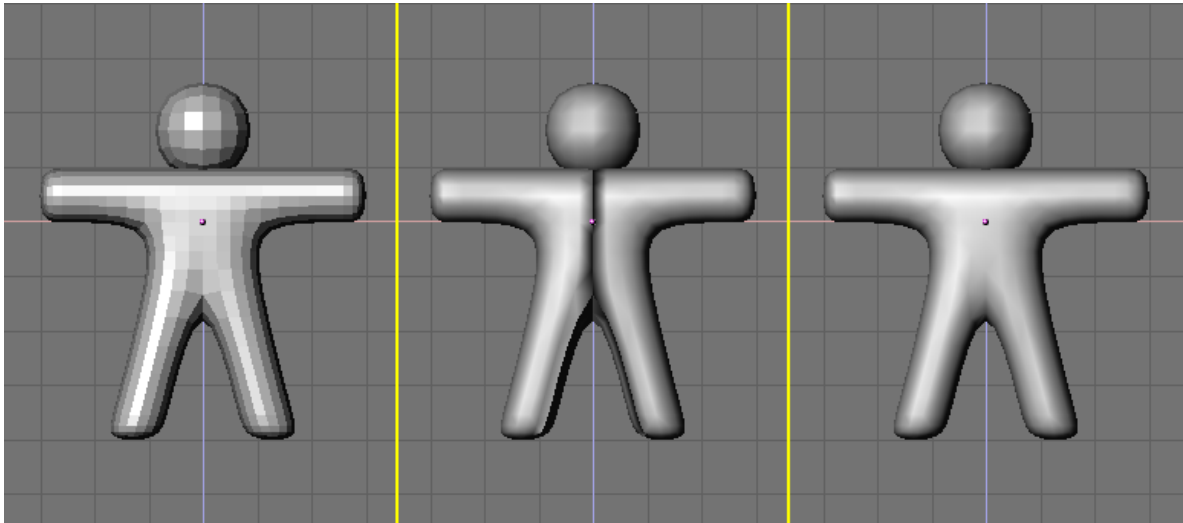
- Esto produce una figura muy tosca. Para hacerla mas suave, localice el botón *SubSurf* (Subdivisión de superficies) Figure 4-16 en el panel de *Mesh* y actívelo. Asegurese de poner a 2, ambos botones inferiores.

Nota: Subdivisión de superficies es una herramienta de modelado avanzado, que dinamicamente crea una malla mucho más densa a partir de los vértices de la malla inicial, haciendo de esta forma una malla mas suave y uniforme que sigue la forma original de la malla que es mucho más tosca. La forma del objeto está aún controlada por la localización de los vértices de la malla original, pero la forma que se renderiza es la malla subdividida, que es mucho más suave.

- Salga del *modo Edición* (TAB) y cambie la visión del objeto del *Modo Alambre* (Wireframe mode) al *Modo Sólido* (Solid Mode) con la **Tecla Z**. para echarle un vistazo a Gus (*Haciendo a Gus más suave.*, izquierda).

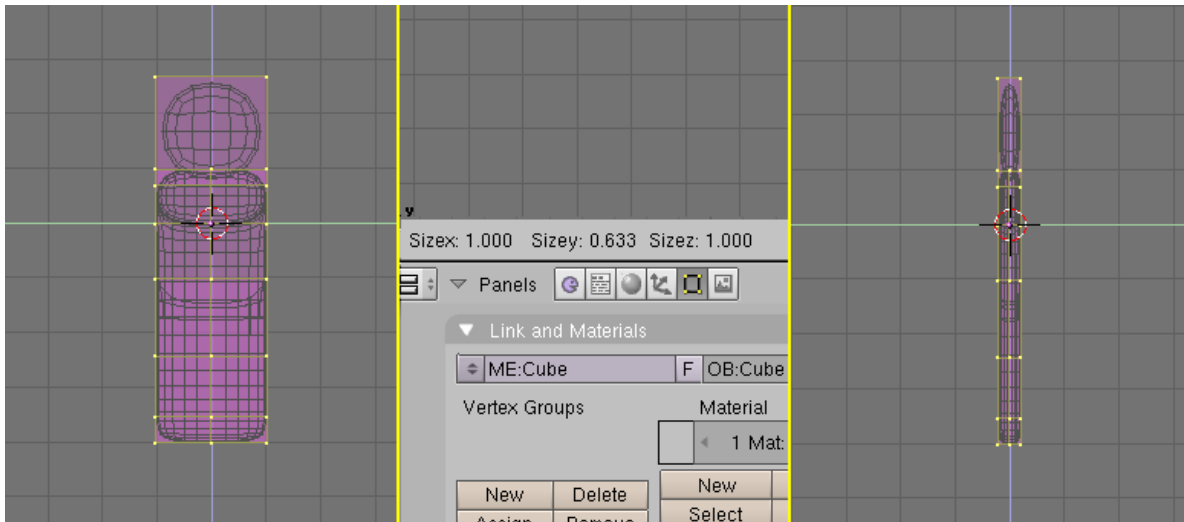


La ventana de botones de Edición.



Haciendo a Gus más suave.

- Para hacer que Gus parezca más liso, presione el botón *SetSmooth* (*La ventana de botones de Edición.*). Ahora Gus aparecerá liso pero con divertidas líneas negras en medio (*Haciendo a Gus más suave.*, medio). Esas líneas aparecen porque la malla subdividida está calculada usando las direcciones normales de la malla tosca, las cuales pueden no ser muy consistentes, esto es, que algunas normales de las caras pueden apuntar hacia dentro, algunas hacia fuera, si extrusiones o volteados han sido hechos. Para resetear las normales, regrese al *modo Edición* (TAB), seleccione todos los vértices (AKEY), y presione CTRL-N. Pulse con LMB en *Recalc normals outside* (Recalcular las normales hacia fuera) en la ventana que aparecerá. Ahora Gus debería aparecer correctamente, tal como se muestra *Haciendo a Gus más suave.*. Presione MMB y arrastre el ratón alrededor de Gus para verle desde todos los ángulos. Oh, ¡es demasiado grueso!. Para corregir esto, cambia a la vista lateral NUM3. Ahora, entra en el *Modo Edición* (si no te encuentras en él), y regresa al *modo Alambre* (Tecla Z), y selecciona todos los vértices con la **Tecla A** (*Adelgazando a Gus utilizando el escalado restringido.*).



Adelgazando a Gus utilizando el escalado restringido.

Vamos a hacer a Gus más delgado:

- Presione la **Tecla S** y comience a mover el ratón horizontalmente. Pulse **MMB** para restringir el escalado solo a un eje o presione la **Tecla Y** para obtener el mismo resultado. Si ahora movemos el ratón veremos como Gus comienza a hacerse más delgado pero manteniendo la misma altura.
- Los tres numeros en la barra de herramientas de la ventana 3D muestran el factor de escalado. Una vez que el escalado está restringido, solo uno de esos numeros variará. Presione y mantenga **CTRL**. El factor de escalado ahora variará de 0.1 en 0.1. Reduzca a Gus con un factor de 0.2, introduciendo su dimensión pulsando **LMB**.
- Regrese a la vista frontal y al Modo Sólido (**Tecla Z**), entonces rote su vista con **MMB**. ¡Gus está mucho mejor ahora!

Veamos como se ve Gus

Estamos listos para ver nuestro primer renderizado, pero antes de eso, tenemos algún trabajo que hacer.

- **Shift-LMB** en la esquina superior derecha para accionar el pequeño botón de los botones de visibilidad de las capas en la barra de herramientas de la ventana 3D (*Haciendo visibles las capas 1 y 10.*) para conseguir que tanto la capa 1 (la capa de Gus) como la capa 10 (la capa de la cámara) sean visibles.



Haciendo visibles las capas 1 y 10.

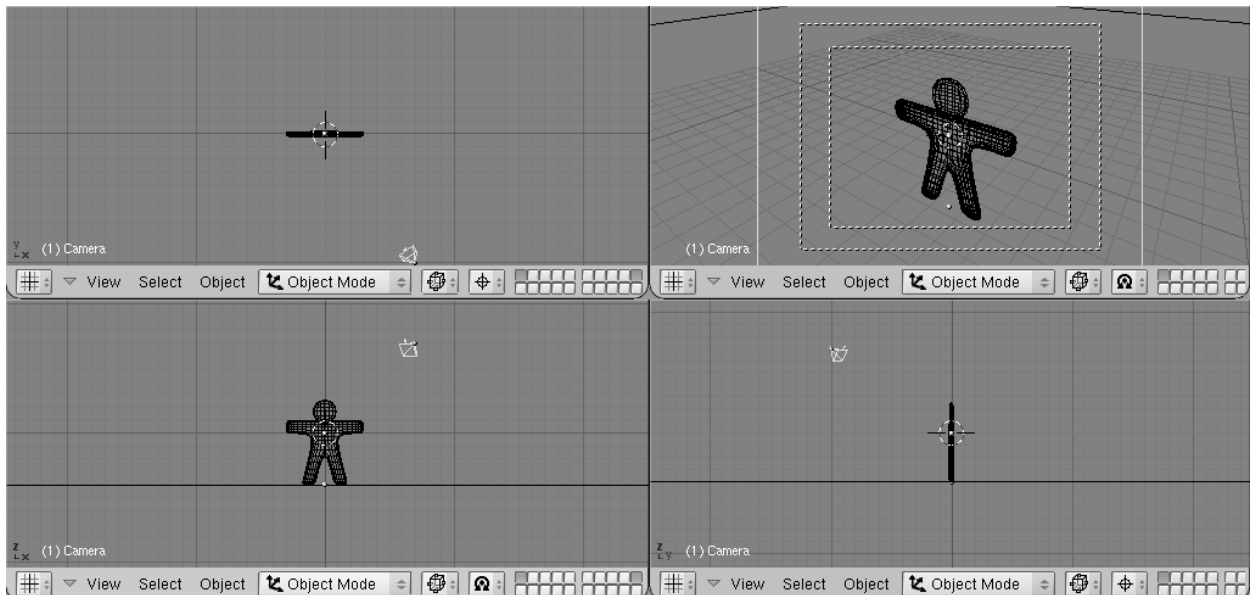
Nota: Recuerde que la última capa seleccionada es la capa activa, por lo que todas los añadidos subsecuentes se harán automáticamente en la capa 10.

- Seleccione la cámara (RMB) y muevala a la siguiente localización (x=7, y=-10, z=7). Puede hacer esto presionando la Tecla G y arrastrando la cámara mientras mantiene presionado CTRL para moverla en pasos de 1 unidad en 1 unidad de rejilla.

Para conseguir que la cámara apunte a Gus, mantenga su cámara seleccionada y entonces seleccione a Gus con SHIFT-RMB. La cámara debería estar de color rosa y Gus de color rosa claro. Ahora presione CTRL-T y seleccione la entrada *Old Track* en la ventana emergente. Esto forzará a la cámara a seguir a Gus y siempre le seguirá. Esto significa que puede mover la cámara donde quiera y puede estar seguro que Gus siempre estará en el centro de la vista de la cámara.

{{Tip|Seguimiento (Tracking)| Si el objeto a seguir ha sufrido ya una rotación, como ocurre muy a menudo, el resultado de pulsar CTRL-T puede no ser el esperado. Si le ocurre esto, seleccione el objeto que realiza el seguimiento (en nuestro caso la cámara), y presione ALT-R para borrar cualquier rotación del objeto. Una vez que haga esto la cámara realmente seguirá a Gus.

La posición de la cámara con respecto a Gus. muestra las vistas superior, frontal, lateral y de cámara de Gus. Para obtener una vista de cámara presione la Tecla NUM0.



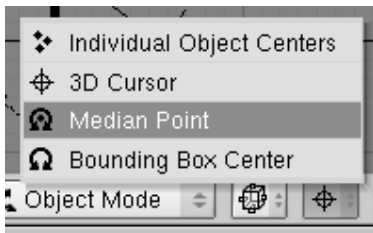
La posición de la cámara con respecto a Gus.

Ahora necesitamos crear el terreno para que Gus se mantenga de pie.

- En la vista superior (NUM7), y fuera del modo Edición, añada un plano (SPACE>>ADD>>Mesh>>Plane).

Nota: Es importante salir del modo Edición, porque si no el objeto añadido formará parte del objeto que actualmente se encuentra en modo Edición, como hicimos con la cabeza de Gus cuando la añadimos. Si el cursor está donde se muestra Figure 4-21, el plano se añadirá en mitad del cuerpo de Gus.

- Cambia a modo Objeto y a la vista frontal (NUM1) y mueva (Tecla G) el plano hasta los pies de Gus, usando CTRL para mantenerlo alineado con Gus.
- Cambiamos el centro de referencia de cursor (donde lo configuramos al principio) a objeto presionando el botón resaltado *Seleccione el centro de referencia al centro del objeto..*
- Ve a la vista de cámara (NUM0) y, con el plano aún seleccionado, presiona la Tecla S para comenzar a escalarlo.

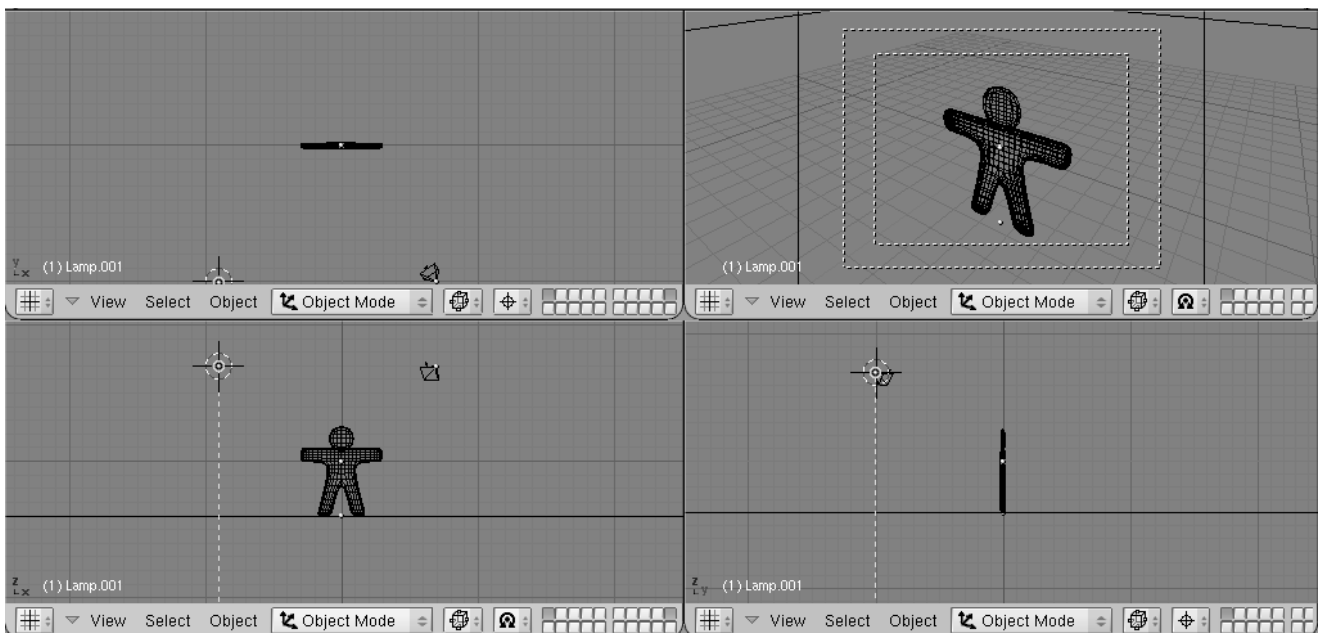


Seleccione el centro de referencia al centro del objeto.

- Alargue el plano hasta que sus lados se extiendan más allá del area de visión de la cámara , que está indicado por el rectángulo de rayas blancas en la vista de cámara.

Ahora, ¡algo de luz!

- En la vista superior (NUM7), añada una lámpara de luz (Lamp) (SPACE>>ADD>>Lamp) en frente de Gus, pero en el otro lado de la cámara; por ejemplo en (x=-9, y=-10, z=7) (*Insertando una lámpara.*).



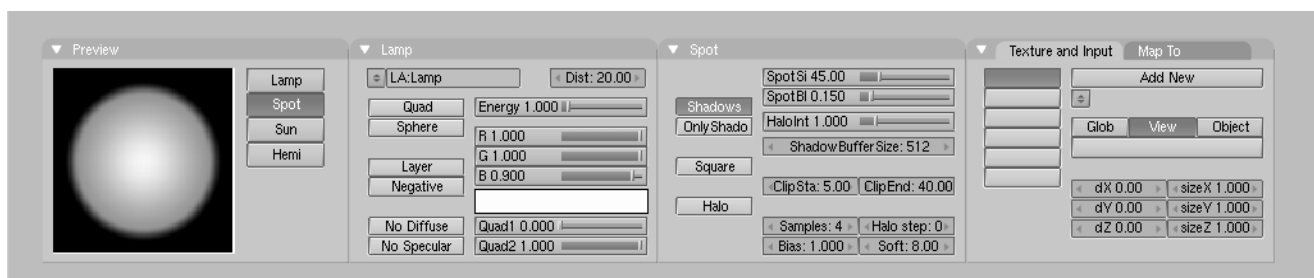
Insertando un lámpara.

- Cambie a los botones de lámpara (Lamp Buttons) en el contexto de materiales a través del botón con una lámpara en la barra de herramientas de la ventana de botones (*El botón para el panel de lámparas.*) o presione simplemente F5.



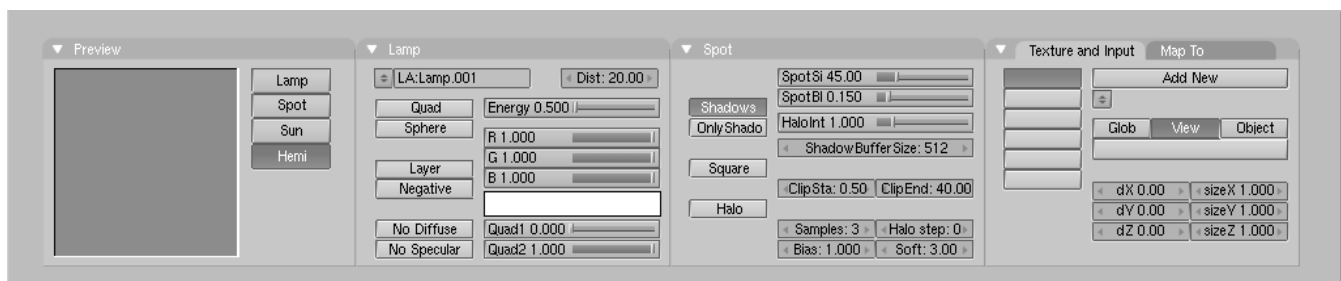
El botón para el panel de lámparas.

- En la ventana de botones, en el panel *Preview*, presione el botón *Spot* para convertir la lámpara en una Spotlight (luz tipo foco) (*Configuración de una luz Spot.*) de color amarillo claro (R=1, G=1, B=0.9). Ajuste *ClipSta:* a 5, *Samples:* a 4, y *Soft:* a 8.



Configuración de una luz Spot.

- Haga que la luz Spot siga a Gus, tal como hizo con la cámara, seleccionando la luz, luego presionando SHIFT, luego Gus, y por último Ctrl-T. Si ha añadido la luz Spot en la vista superior no debería necesitar borrar su rotación a través de Alt-R.
- Añada una segunda lámpara en la misma localización que la luz Spot, y de nuevo en la vista superior, con (SPACE>>ADD>>Lamp). Haga esta lámpara, una lámpara *Hemi* con una energía de valor 0.6 (*La configuración de la lámpara Hemi*).



La configuración de la lámpara Hemi

¿Dos lámparas?: Usar dos o más lámparas ayuda a producir una iluminación más suave y realística porque en realidad la luz natural nunca viene de un solo punto. Aprenderá más sobre esto en *chapter_lighting*.

Estamos casi listos para renderizar. Como primer paso, vaya al panel Scene (Escena) presionando el botón que muestra una imagen en la barra de herramientas de la ventana de botones (*El panel de botones de renderizado.*) o presione F10.



El panel de botones de renderizado.

En los botones de render, en el panel *Format*, ponga el tamaño de la imagen a 640x480. En el panel *Render* pulsa el botón *Shadows* (sombras) que esta arriba en el centro, y el botón *OSA* en el centro a la izquierda (*La ventana de botones de renderizado*). Esos últimos botones habilitarán las sombras y el oversampling (alisado) el cual prevendrá los bordes de sierra.



La ventana de botones de renderizado

Ahora presione el botón *RENDER* o pulse F12. El resultado, tal como se muestra en *Su primer renderizado*. ¡Felicidades!, es actualmente bastante lamentable. Aún necesitamos poner materiales, y montones de detalles, tales como ojos, y cosas por el estilo.



Su primer renderizado. ¡Felicidades!

Materiales y Texturas

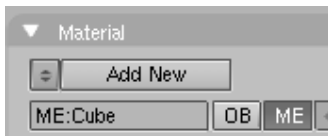
Es hora de dar a Gus algún buen material que simule una galleta:

- Seleccione a Gus. Entonces, en la cabecera de la ventana de botones, seleccione la ventana de materiales presionando el botón de la bola roja (*El botón de la ventana de botones de materiales.*) o usando la tecla F5.



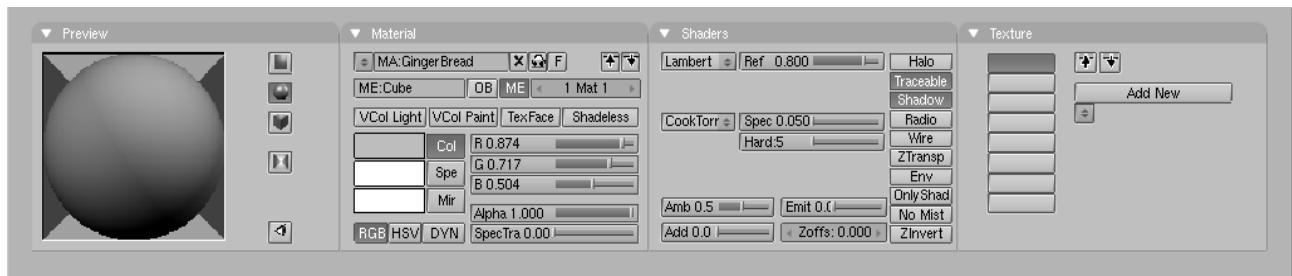
El botón de la ventana de botones de materiales.

- La ventana de botones estará casi vacía porque Gus no tiene aún materiales asignados. Para añadir un material, pincha en el Panel *Material* (donde están los dos triángulos, uno hacia arriba y otro hacia abajo) y selecciona *Add New* (Añadir nuevo) (*El botón en el menú de los materiales.*).



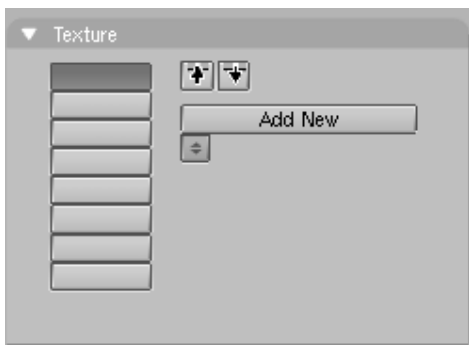
El botón en el menú de los materiales.

- La ventana de botones se llenará con Paneles y botones y un hueco con el nombre del material, que por defecto será "Material", que aparecerá próximo al botón cuadrado de color blanco. Cambie esto a algo más significativo, como GingerBread (galleta de jengibre).
- Modifique los valores como aparece en *La ventana de botones del material y un primer material de jengibre.* para obtener un primer material áspero.



La ventana de botones del material y un primer material de jengibre.

- Presione el botón de menú en el área del panel de texturas (*Panel texturas en la ventana de materiales*) y seleccione *Add new*. Con esto añadiremos una textura en el primer canal. Llamela "GingerTex."



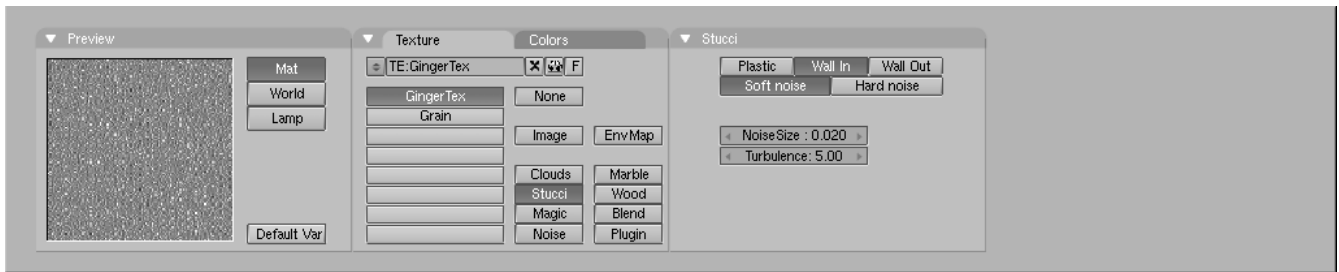
Panel texturas en la ventana de materiales.

- Seleccione la ventana de texturas pinchando en el botón *El botón de la ventana de texturas.* o presionando F6.



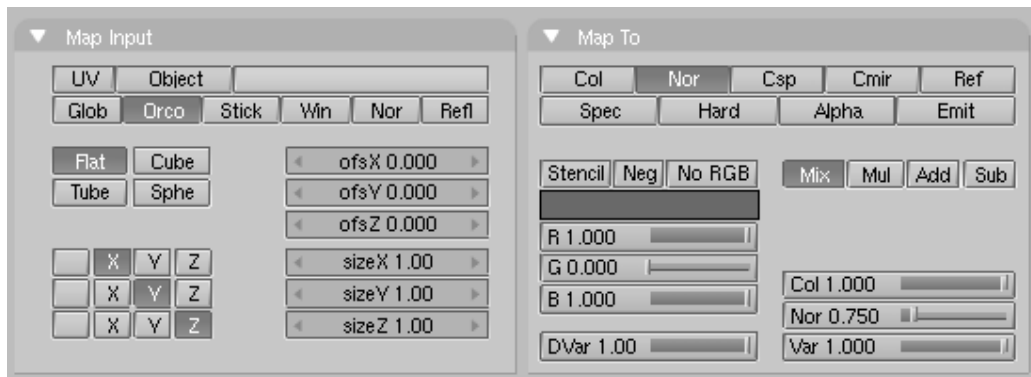
El botón de la ventana de texturas.

- De las columnas de botones que aparecen en el panel *Texture* (Textura) seleccione *Stucci* y configure todos los parámetros como en *La ventana de los botones de textura con una textura stucci*.



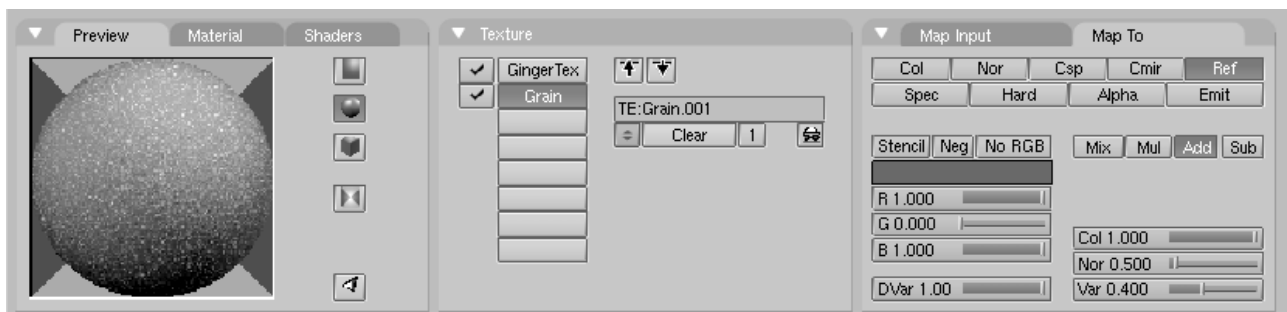
La ventana de los botones de textura con una textura stucci.

- Vuelva a la ventana de materiales (F5) y configure las pestañas *Map Input* y *Map To* del panel de *Texturas* como en *Configuración para la textura Stucci en la ventana de botones del material*. Deseleccione el botón *Col* y seleccione el botón *Nor*, entonces mueva el deslizador *Nor* a 0.75. Esos cambios harán que nuestra textura stucci actúe como un "bumpmap" (mapa de rugosidad) y haga que Gus parezca más de galleta.



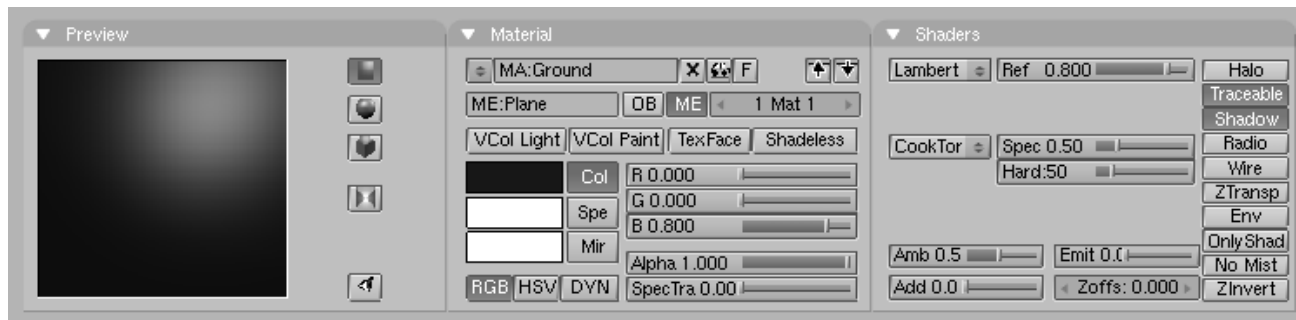
Configuración para la textura Stucci en la ventana de botones del material.

- Ahora añada una segunda textura, llámela "Grain", y seleccione solo la propiedad *Ref* y un valor de 0.4 en el deslizador *Var* (*Configuración para una textura adicional de ruido en el canal 2*). La textura es una textura de *Noise* (Ruido).



Configuración para una textura adicional de ruido en el canal 2.

Dele al suelo un material apropiado, como el azul oscuro que se muestra en *Un material muy simple para el suelo*.



Un material muy simple para el suelo.

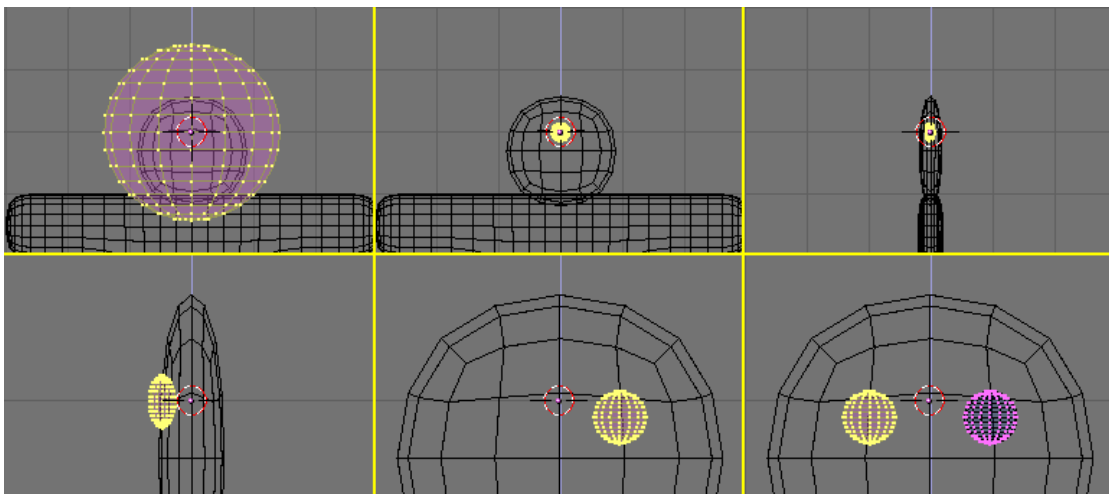
Para darle algunos toques finales, le añadiremos ojos y otros detalles.

- Primero haremos que la única capa visible sea la 1 pinchando con **LMB** en el botón de la capa 1 (*Los botones de visibilidad de las capas en la barra de herramientas.[]*). Esto ocultará las luces, la cámara y el suelo.



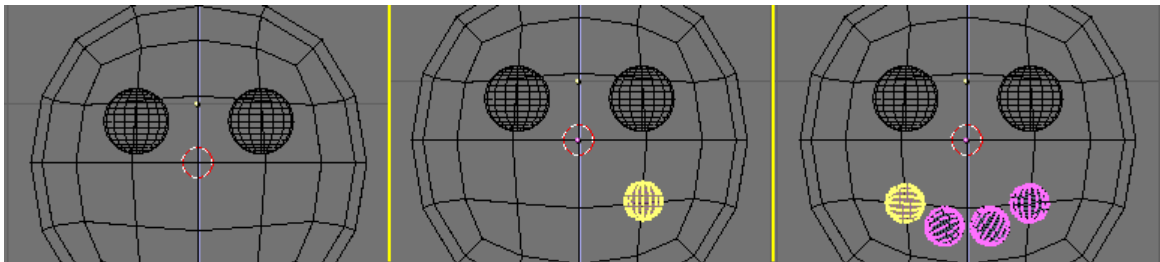
Los botones de visibilidad de las capas en la barra de herramientas.

- Situé el cursor en el centro de la cabeza de Gus. ¡Recuerde que está en 3D así que asegúrese de comprobar al menos dos vistas para comprobarlo!.
- Añada una esfera (**SPACE**>>**ADD**>>**Mesh**>>**UVsphere**). Se le preguntará por el número de *Segments* (Segmentos) (meridianos) y *Rings*: (Anillos) (paralelos) en los que se dividirá la esfera. El valor por defecto de 32 es más de lo que necesitamos aquí, así que usaremos un valor de 16 para ambos. La esfera está en la primera imagen arriba a la izquierda en la secuencia *Secuencia de creación de los ojos*.
- Disminuya la esfera escalándola (**SKEY**) con un factor de 0.1 en todas las dimensiones, entonces cambie a la vista lateral (**NUM3**) y escálela solo en la dirección horizontal (Tecla **Y**) un valor de 0.5 (vea las siguientes dos imágenes en *Secuencia de creación de los ojos*).



Secuencia de creación de los ojos.

- Aplique un poco el Zoom si es necesario con **NUM+**, **MW**, o **CTRL-MMB**, y mueva la esfera (**GKEY**) a la izquierda para que esté en la mitad de la cabeza (como se muestra en la primera imagen de la segunda fila de *Secuencia de creación de los ojos*.).
- Vuelva a la vista frontal (**NUM1**) y mueva la esfera a la derecha. Colóquela donde Gus debería tener un ojo.
- Voltee un duplicado alrededor del cursor siguiendo la secuencia que aprendió cuando volteó el cuerpo de Gus. (Seleccione el cursor 3D, en modo Edición seleccione todo **AKEY**, y haga **SHIFT-D**, **ESC**, **MKEY**, *Global X*). Ahora Gus tiene dos ojos.
- Salga del modo Edición (**TAB**), y sitúe el cursor tan cerca como pueda del centro de la cara de Gus. Añada una nueva esfera y escálela y muevala exactamente como antes, pero hágala más pequeña y Colóquela más abajo y a la derecha del cursor, centrada sobre el vértices de la malla subdividida *Creando una boca con herramientas de spinning (revolución)*.).



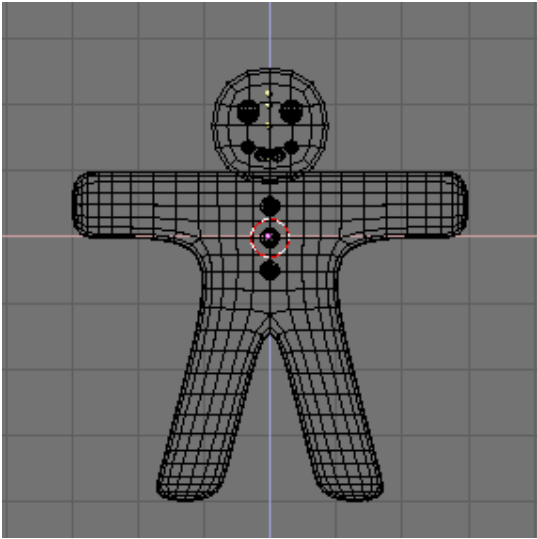
Creando una boca con herramientas de spinning (revolución).

- Ahora, en los botones de Edición (**F9**), localice el grupo de botones en la parte superior del Panel *Mesh Tools* (Herramientas de malla) (*Los botones de la herramienta Spin en la ventana de botones de Edición.*). Configure *Degr:* a 90, *Steps:* a 3, y verifique que esta pulsado el botón *Clockwise:*. Entonces, con todos los vértices aún seleccionados, presione *SpinDup*. Esto creará tres duplicados de los vértices seleccionados en un arco de 90 grados, centrados alrededor del cursor. El resultado debería ser la boca de Gus, tal como se muestra en la última imagen de secuencia mostrada en *Creando una boca con herramientas de spinning (revolución)*.



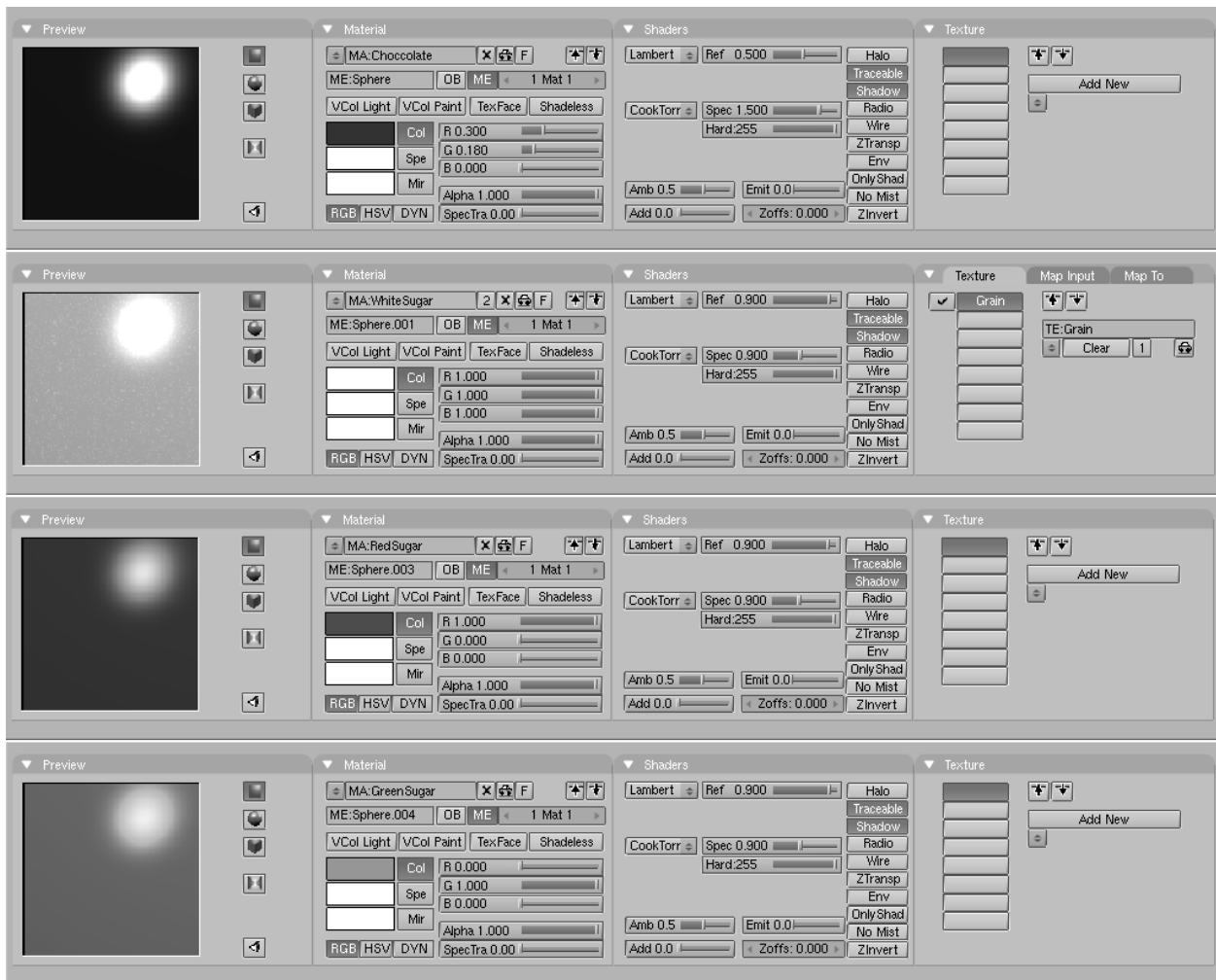
Los botones de la herramienta Spin en la ventana de botones de Edición.

Ahora que ha aprendido el truco, añada tres más de esas elipsoides para crear los botones de Gus. Una vez que haya hecho un botón, puede simplemente salir del modo Edición, presionar **SHIFT-D** para crear un duplicado, y mover el duplicado a su lugar, tal y como se muestra en ¡Gus completado!.



¡Gus completado!

Dé a los ojos un material parecido al chocolate, como el que se muestra en la parte superior *Algunos materiales de caramelo..* Dé a la boca un material blanco parecido al azúcar, como el segundo que se muestra en *Algunos materiales de caramelo.*, y dé a los botones un material rojo, blanco y verde parecido de nuevo al azúcar. Estos materiales son mostrados de arriba a abajo en *Algunos materiales de caramelo..*



Algunos materiales de caramelo.

Una vez que haya terminado de asignar los materiales, haga la capa 10 visible de nuevo (ya debe saber como), para que esas luces y la cámara también aparezcan, y haga un nuevo renderizado (F12). El resultado debería asemejarse más o menos a esto *Renderizado estático de Gus una vez terminado..*



Renderizado estático de Gus una vez terminado.

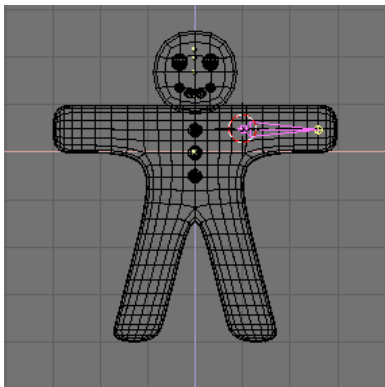
Salve su imagen, si así lo desea, presionando F3. Introduzca el nombre de su imagen en la ventana de archivos y guárdelo.

Su Primera Animación en 30 Minutos II

Construyendo un esqueleto (Rigging)

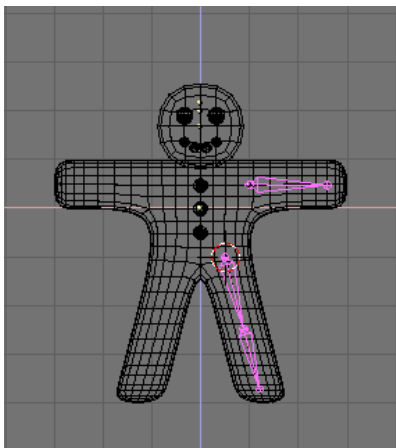
Si hubiésemos querido hacer una imagen estática, nuestro trabajo hasta este punto sería suficiente, pero ¡queremos que Gus se mueva!. El siguiente paso será darle a Gus un esqueleto, o armadura, con la cual se podrá mover. Esto se llama el arte del rigging. Gus tendrá un esqueleto muy simple: cuatro extremidades (dos brazos y dos piernas) y una pocas uniones (sin codos, solo con rodillas), pero sin pies o manos. Para añadir el esqueleto:

- Situé el cursor donde debería estar el hombro, presione **SPACE>>Add>>Armature**. Un objeto romboidal aparecerá, un hueso del sistema del esqueleto, que va desde el cursor 3D al puntero del ratón. Situé el fin del hueso en la mano de Gus (*Añadiendo el primer hueso del brazo.*) con **LMB**. Esto fijará el hueso y creará uno nuevo que partirá del punto final del anterior, produciendo una cadena de huesos. De momento no necesitamos más huesos, así que presione **ESC** para salir.



Añadiendo el primer hueso del brazo.

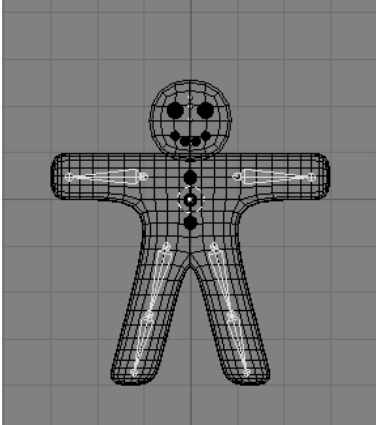
2. Estando en modo Edición, mueva el cursor hasta la unión de la pierna con el tronco y añada un nuevo hueso (**SPACE>>ADD>>Armature**) hasta la rodilla. Presione **LMB** y un nuevo hueso debería aparecer estomáticamente ahí. Lleve este hueso hasta el pie (*Añadiendo el segundo y el tercer hueso, una cadena de huesos en la pierna.*).



Añadiendo el segundo y el tercer hueso, una cadena de huesos en la pierna.

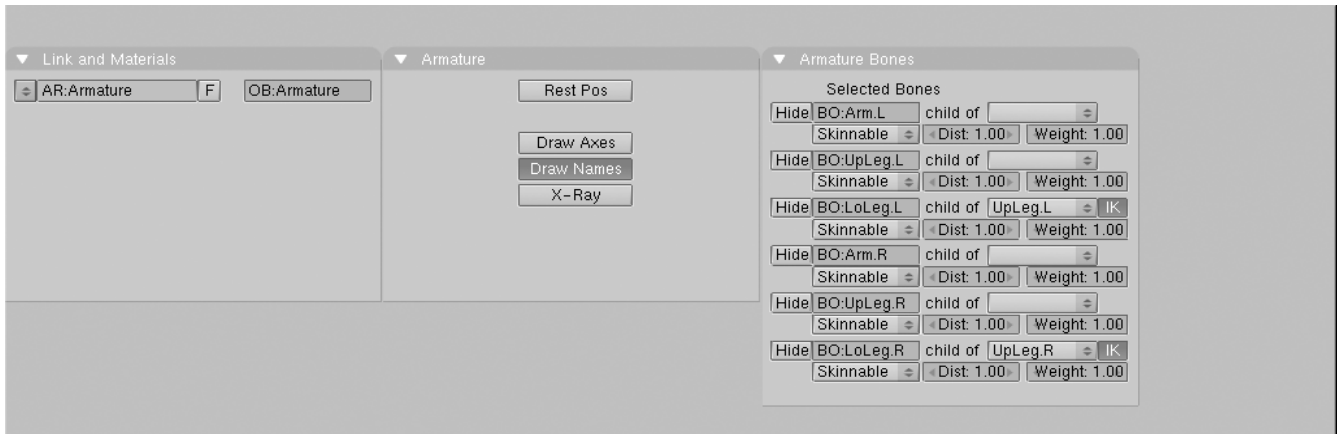
Posición de los huesos: Los huesos que está añadiendo deformarán la malla del cuerpo de Gus. Para producir un resultado correcto, intente situar las uniones de los huesos como se muestra en las ilustraciones.

- Ahora sitúe el cursor en el centro y seleccione todos los huesos con la Tecla A. Duplíquelos con Shift-D salga de modo de movimiento con la Tecla ESC entonces voltee los con la Tecla M relativamente al cursor y al eje X Global como hizo con las mallas (*El esqueleto completo después de duplicarlo y voltearlo.*).



El esqueleto completo después de duplicarlo y voltearlo.

Una vez que haya seleccionado todos los huesos (Tecla A), la ventana de botones de edición debería mostrar un panel *Armature Bones* (huesos del esqueleto) el cual debería mostrar los botones de la armadura (*La ventana de botones de edición para una armadura.*).



La ventana de botones de edición para una armadura.

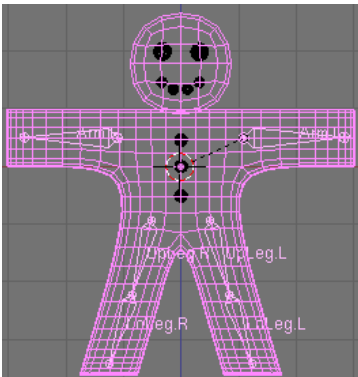
Presione el botón *Draw Names* (Dibujar nombres) para ver los nombres de los huesos, entonces **SHIFT-LMB** en los nombres en la ventana de botones de edición (*La ventana de botones de edición para una armadura.*) para cambiarlos a algo más apropiado tal como Arm.R, Arm.L, UpLeg.R, LoLeg.R, UpLeg.L, y LoLeg.L. Salga del modo Edición con (TAB).

Renombrando los huesos: Es muy Importante que llame a sus huesos con un sufijo '.L' or '.R' para distinguir los que pertenecen a la izquierda (left) y los que pertenecen a la derecha (right), así el editor de acciones (Action editor) será capaz de voltear automáticamente sus poses.

Asignación del esqueleto (Skinning)

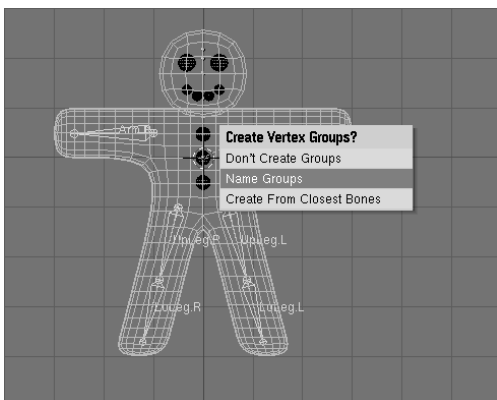
Ahora debemos conseguir que una deformación en el esqueleto provoque una deformación proporcional en el cuerpo. Haremos esto con la técnica Skinning, la cual asigna los vértices a los huesos de tal forma que los primeros están sujetos al movimiento de los últimos.

- Selecciona el cuerpo de Gus, entonces presionando la tecla **SHIFT** selecciona el esqueleto de tal forma que el cuerpo este morado y el esqueleto está de color rosa claro.
- Presiona **CTRL-P** para encajar el cuerpo al esqueleto. Un dialogo emergente aparecerá (*La ventana emerge cuando encajamos un Objeto a una esqueleto.*). Selecciona la entrada *Use Armature* (Usar esqueleto).



La ventana emerge cuando encajamos un Objeto a una esqueleto.

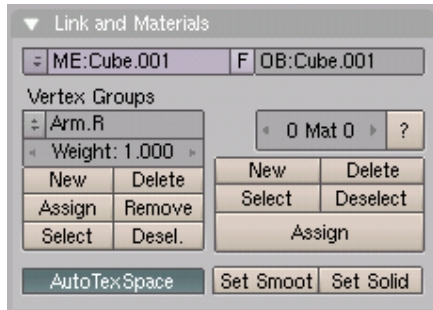
- Un nuevo menú aparecerá, preguntándole si quiere que Blender no haga nada, cree grupos de vértices vacíos o grupos de vértices poblados. (*Opciones de Skinning automático.*).



Opciones de Skinning automático.

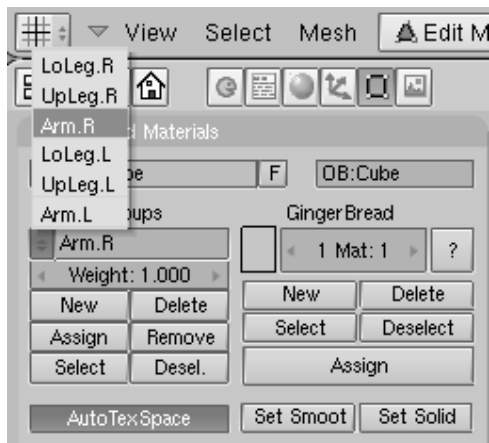
- Usaremos la opción de skinning automático. Vaya y seleccione *Create From Closest Bones* (Crear a partir de los huesos más cercanos).

Ahora seleccione únicamente el cuerpo de Gus y entre en modo Edición (TAB). Notará la presencia en la ventana de botones de edición (Edit buttons) (F9) y en el panel *Herramientas de malla 1* (Mesh Tools 1), de un menú y botones para grupos de vértices (Vertex Group) (*Los botones del grupo de vértices en la ventana de botones de edición de una malla.*).



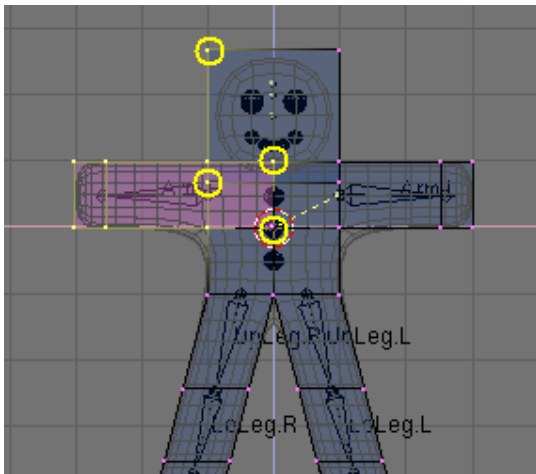
Los botones del grupo de vértices en la ventana de botones de edición de una malla.

Presionando en el menú de botones un menú con todos los grupos de vértices disponibles emergerá (seis en nuestro caso, pero con un verdadero personaje, con manos y pies completamente equipados de huesos, ¡puedes tener docenas de ellos!) *El menú con los grupos de vértices automáticamente creados en el proceso de skinning.* Los botones *Select* (Seleccionar) y *Deselect* (Deseleccionar) le mostrarán que vértices pertenecen a cada grupo.



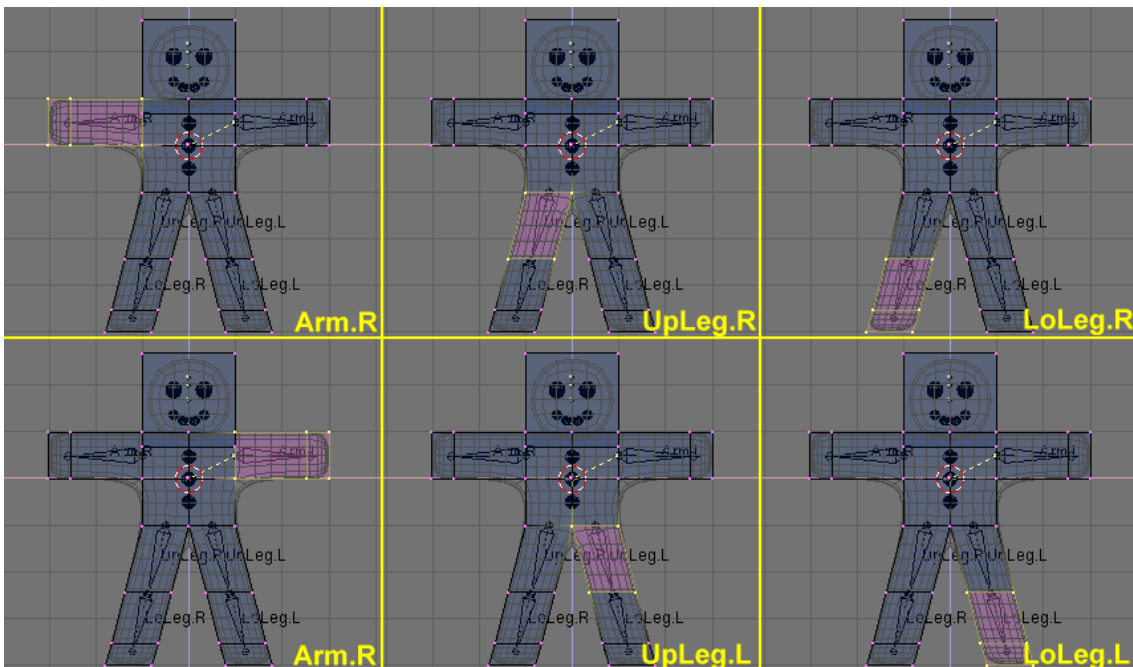
El menú con los grupos de vértices automáticamente creados en el proceso de skinning.

Seleccione el grupo del brazo derecho (Arm.R) y, con todos los vértices deseleccionados (Tecla A, si es necesario) presione *Select*. Debería ver algo como *Gus en Modo Edición con todos los vértices del grupo Arm.R seleccionados.*



Gus en Modo Edición con todos los vértices del grupo Arm.R seleccionados.

Los vértices marcados con círculos amarillos en *Gus en Modo Edición con todos los vértices del grupo Arm.R seleccionados*, pertenecen al grupo de deformación, pero no deberían pertenecer. El proceso de autoskinning encontró que estaban muy cerca del hueso y los añadió al grupo de deformación. No les queremos en este grupo porque, dado que algunos son de la cabeza y otros son del pecho, añadirlos al grupo de deformación, deformaría esas partes del cuerpo. Para eliminarlos del grupo, deselecciona todos los otros vértices, aquellos que deberían *permanecer* en el grupo usando la caja de selección (Tecla B), pero usa MMB, no LMB, para definir la caja, así todos los vértices que estén en la caja se deseleccionarán. Una vez que solo los vértices 'indeseables' estén seleccionados, presione el botón *Remove* (Los botones del grupo de vértices en la ventana de botones de edición de una malla.) para eliminarlos del grupo Arm.R. Deselecciona todos (Tecla A) entonces compruebe otro grupo. Compruébelos todos y asegúrese de que se parecen a estos *Los seis grupos de vértices*..



Los seis grupos de vértices.

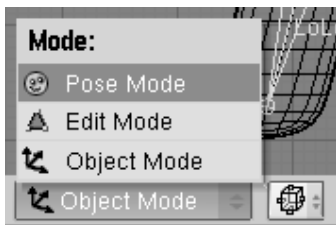
Grupos de vértices: Sea cuidadoso cuando asigne o elimine vértices de los grupos de vértices. Si observa deformaciones inesperadas, puede haber olvidado algunos vértices, o puede haber situado demasiados vértices en el grupo. Puede modificar sus grupos de vértices en cualquier momento.

Otros detalles: Nuestras deformaciones solo afectaran al cuerpo de Gus, no a sus ojos, boca o botones, los cuales son objetos independientes. Mientras que esto no es un problema a considerar en esta simple animación, debe ser tenido en cuenta en proyectos más complejos, por ejemplo emparentando o uniendo las distintas partes del cuerpo para crear una malla única (describiremos en detalle esas opciones en los próximos capítulos).

Posando

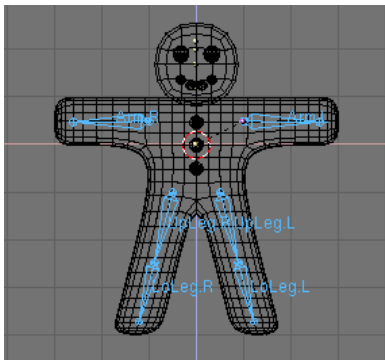
Una vez que haya dado creado el esqueleto y se lo haya asignado a Gus puede comenzar a jugar con él como si fuera un muñeco, moviendo sus huesos y viendo los resultados.

- Selecciona únicamente la armadura, entonces selecciona *Modo Posado* (Pose Mode) en el menú de los modos ("Mode" Menu) (*El botón para cambiar al modo posado en la barra de herramientas de la ventana 3D.*). Esta opción solo aparecerá si una armadura está seleccionada.



El botón para cambiar al modo posado en la barra de herramientas de la ventana 3D.

- La armadura se volverá azul. Está ahora en Modo Posado. Si selecciona un hueso se volverá azul claro, no rosa, y si lo mueve (TECLA G), o lo rota (TECLA R), ¡el cuerpo se deformará!



¡Ahora está en modo posado!

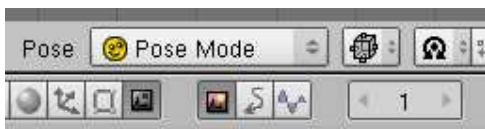
Posición inicial: Blender recuerda la posición original de los huesos. Puede volver a poner el esqueleto (armature) en dicha posición simplemente presionando el botón RestPos en el panel de botones de edición de esqueletos (Armature Edit Buttons) (*La ventana Editar Botones para una armadura.*).

Cinemática directa (FK) e inversa (IK):

Mientras manejamos los huesos en modo posado notarás que se comportan como cuerpos rígidos, cuerpos inextensibles con juntas esféricas al final. Solo puedes mover el primer hueso de la cadena y todos los otros le seguirán. Todos los huesos subsecuentes en la cadena no podrán moverse, solo pueden ser rotados, así que si es rotado respecto al hueso anterior de la cadena todos los huesos subsecuentes de la cadena seguirán su rotación. Este procedimiento, llamado Cinemática directa (Forward Kinematics) es fácil de seguir, pero precisar la localización hasta del último hueso de la cadena es difícil. Podemos utilizar otro método, Cinemática Inversa (Inverse Kinematics) donde solo tenemos que definir la posición del último hueso de la cadena, y el resto asume una posición, automáticamente calculada por Blender, para mantener la cadena sin huecos. De este modo posicionar las manos y los pies es mucho más sencillo.

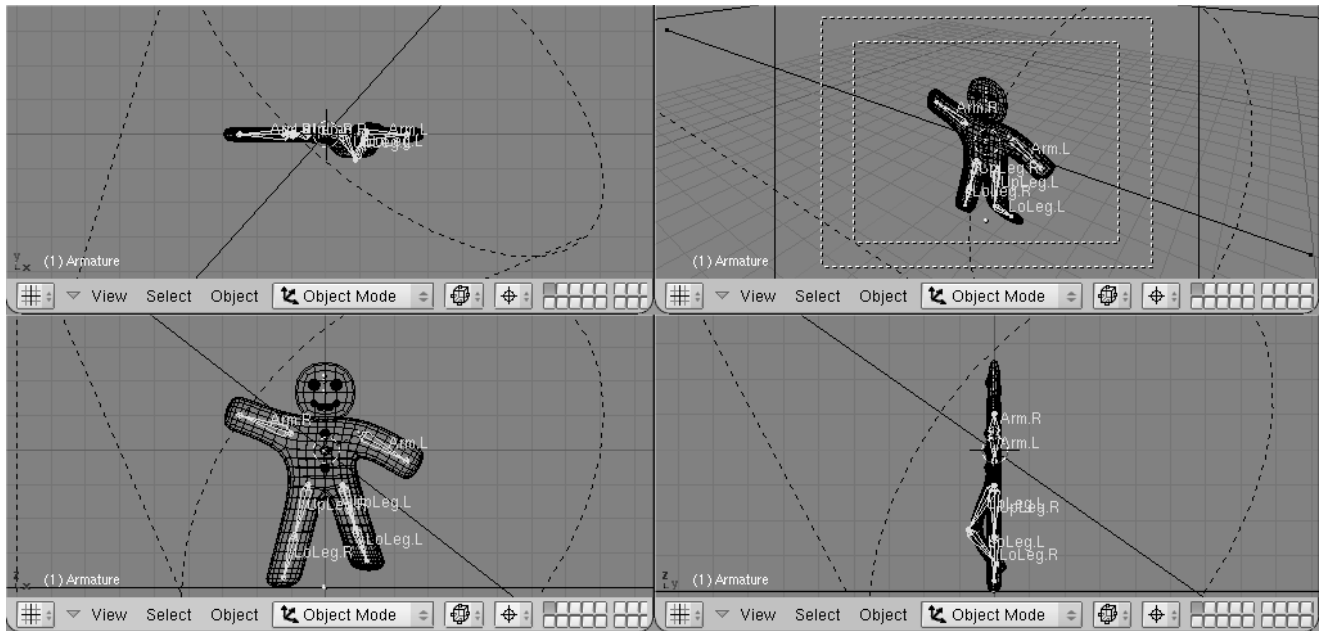
Haremos que Gus ande definiendo cuatro poses diferentes relativas a cuatro diferentes estados de una zancada. Blender hará el trabajo de crear una animación fluida.

- Primero, verifique que está en el frame (imagen) 1 de la línea temporal. El número de frames aparece en un botón a la derecha de la barra de herramientas de la ventana de botones (*El frame actual en la barra de herramientas de la ventana de botones.*). Si no está puesto a 1, póngalo ahora.



El frame actual en la barra de herramientas de la ventana de botones.

- Ahora, rotando solo un hueso al mismo tiempo (Tecla R), levantaremos UpLeg.L y doblaremos LoLeg.L hacia atrás mientras levantamos un poco Arm.R y bajamos otro poco Arm.L, tal como se muestra en *Nuestra primera pose.*



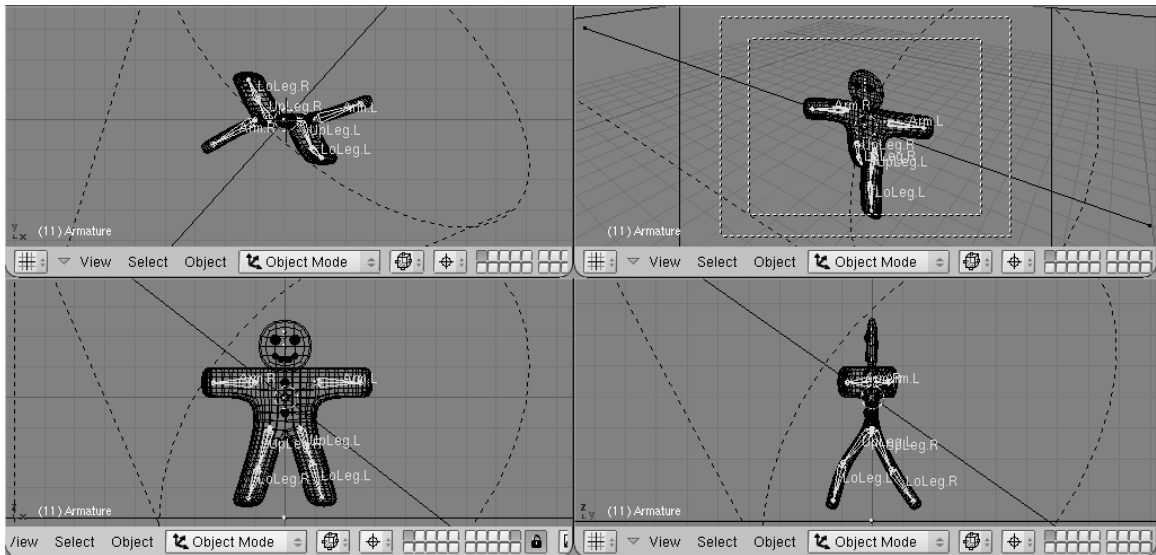
Nuestra primera pose.

- Seleccione todos los huesos con la Tecla A. Con el puntero del ratón en la ventana 3D, presione la Tecla I. Un menú emergerá *Asignando la pose al frame..* Seleccione *LocRot* en este menú. Esto obtendrá la posición y la orientación de todos los huesos y los guardará en una pose en el frame 1. Esta pose representa a Gus en la mitad de la zancada, mientras esta moviendo su pierna izquierda hacia delante y sobre el suelo.



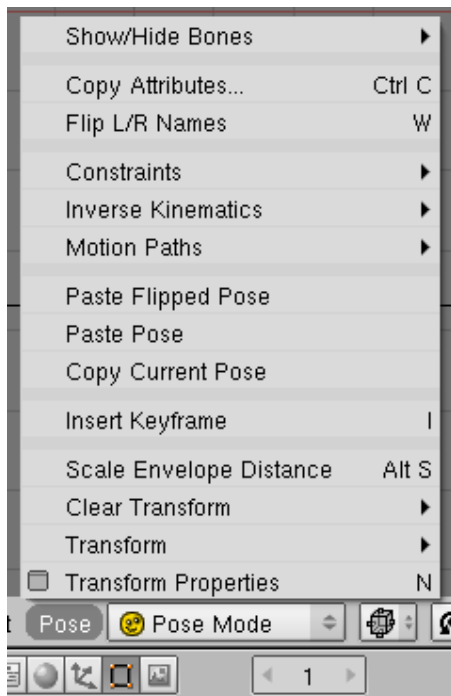
Asignando la pose al frame.

- Ahora muévase al frame 11 introduciéndolo numéricamente o presionando la Tecla Arriba. Entonces mueva a Gus a una posición diferente , como *Nuestra segunda pose..*, con su pierna izquierda adelantada y la derecha retrasada, ambas ligeramente dobladas. ¡Gus está caminando en el sitio!



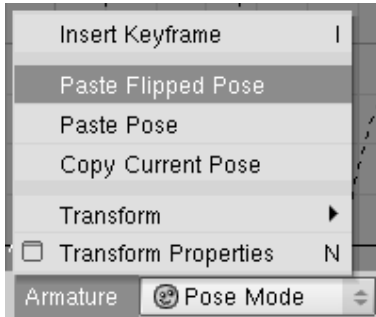
Nuestra segunda pose.

- Seleccione todos los huesos de nuevo y presione la Tecla **I** para guardar la pose en el frame 11.
- Ahora necesitamos una tercera pose en el frame 21, con la pierna derecha levantada, porque estamos en el medio de la otra mitad de la zancada. Esta pose es el reflejo de la pose que hemos definido en el frame 1. Otra forma sería, volver al frame 1 y en el menú *Armature* en la cabecera de la ventana 3D seleccione la entrada *Copy Pose* (Copiar Pose) (*Copiando la pose al buffer.*). Ha copiado la pose actual al buffer.



Copiando la pose al buffer.

- Vaya al frame 21 y pegue la pose con la opción *Paste Flipped Pose* (Pegar pose volteada) en el menú *Armature* (*Pegar la copia como una nueva, y volteada, pose.*). Este botón pegará la pose seleccionada, intercambiando las posiciones de los huesos con el sufijo .L con aquellos huesos que tienen el sufijo .R, efectivamente ¡dándolos la vuelta!



Pegar la copia como una nueva, y volteada, pose.

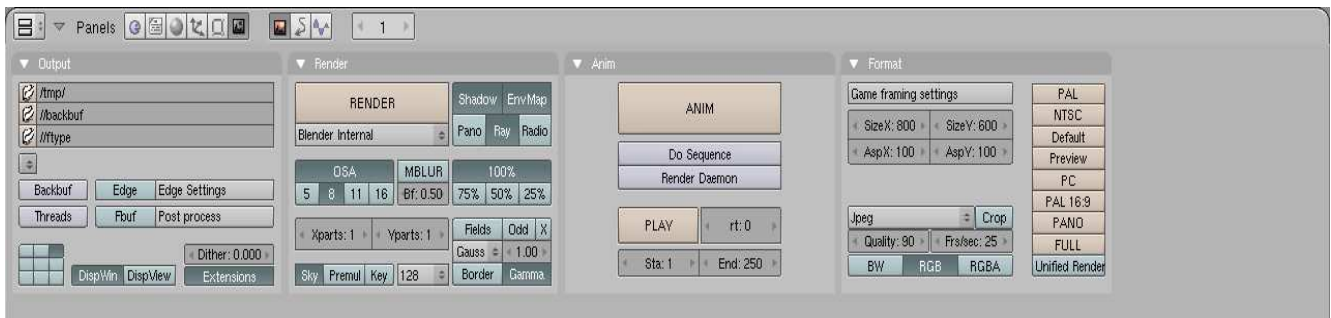
La pose está ahí pero ¡aún no ha sido guardada!. Debe presionar la **Tecla I** con todos los huesos seleccionados.

- Ahora aplique el mismo procedimiento para copiar la pose del frame 11 al frame 31, también volteándola.
- Para completar el ciclo, necesitamos copiar la pose del frame 1 *sin* voltear al frame 41. Cópelo de la manera usual, y péguelo utilizando la entrada *Paste Pose* (Pegar pose). Finalice la secuencia guardando la pose con la **Tecla I**.

¡Gus anda!

El primer paso es el núcleo del proceso de caminar, y una vez que lo tenga definido hay técnicas para hacer que un personaje ande a través de un camino complejo. Pero, para el propósito de este capítulo de iniciación, este primer paso es suficiente.

- Cambie a la ventana de Renderizado con (**F10**) y configure el comienzo (start) y el final (end) de la animación a 1 y a 40 respectivamente (*Configurando los botones de renderizado para una animación.*). Debido a que la imagen (frame) 41 es idéntica a la imagen 1, solo necesitamos renderizar las imágenes que van de la 1 a la 40 para producir el ciclo completo.



Configurando los botones de renderizado para una animación.

- Seleccione AVI Raw como el tipo de fichero que vamos a utilizar en el panel *Formato* (Format) (*Configurando los botones de renderizado para una animación.*). Mientras que esto no es una buena elección generalmente, principalmente por problemas de tamaño de archivo (como se explicará más tarde), tiene la ventaja de ser rápido y funcionar en cualquier computadora, así que cumple nuestras necesidades. (Puede también seleccionar el formato AVI Jpeg para obtener un fichero de un tamaño más reducido, pero utilizando la compresión por pérdidas Jpeg y obteniendo una película que algún reproductor externo puede no ser capaz de reproducir).
- Finalmente, presione el botón *Animar* (ANIM) en el panel *Animación* (Anim). Recuerde que todas las capas que quiera usar en la animación ¡deben ser seleccionadas!. En nuestro caso, las capas seleccionadas serán la 1 y la 10.

Nuestra escena es muy simple, y Blender probablemente renderizará las 40 imágenes en unos pocos segundos. Mire como van apareciendo en pantalla.

Una vez que el renderizado se haya completado debería obtener un fichero llamado *0001_0040.avi* en un subdirectorio *render* de su actual directorio - el mismo que contiene el fichero *.blend*. Puede reproducir este fichero directamente en Blender presionando el botón *Reproducir* (Play) que está debajo del botón *Animar* (ANIM) (*Configurando los botones de renderizado para una animación.*). La animación se reproducirá una y otra vez. Para detenerla presione **ESC**. Solo hemos producido un ciclo muy básico para poder andar . Hay muchas más cosas en Blender, ¡pronto las descubrirá!.

PARTE II - MODELADO

Como has podido ver en el capítulo Introducción Rápida, la creación de una escena en 3D necesita al menos tres cosas: Modelos, Materiales e Iluminación. En este capítulo analizaremos en profundidad cada una de ellas. La Parte III tratará sobre Materiales, la Parte IV sobre Texturas y la Parte V sobre Iluminación.

Una vez que hayas terminado con esta Parte puedes elegir Renderizado si quieres sacar partido de los motores de renderizado de imágenes que tiene Blender, o también puedes optar por leer la Parte VII para aprender todo sobre la animación en Blender.

Capítulos

- Modo Objeto
- Booleanos
- Objetos Básicos
- Modo Edición
- Modelado Básico de Malla
- Subdivisión de Superficies
- Herramientas Arista y Cara
- Modelado Avanzado de Malla
- Curvas
- Superficies
- Texto
- Meta Objetos

Modo Objeto

La geometría de una escena de Blender se construye a partir de uno o más objetos: luces, curvas, superficies, cámaras, mallas y los objetos básicos descritos en *Objetos Básicos*. Cada objeto puede ser movido, rotado y cambiado de tamaño en ObjectMode. Para cambios más detallados en la geometría, se puede trabajar sobre la malla de un Objeto en EditMode (véase *Modo Edición*). Una vez se ha añadido un objeto básico mediante el menú **SPACE>>Add**, Blender cambia a modo EditMode por defecto si el objeto es una Malla (Mesh), una Curva (Curve) o una Superficie (Surface). Se puede cambiar a ObjectMode presionando **TAB**. La estructura de alambre del objeto, si la hay, debería aparecer en rosa, lo que significa que el objeto está ahora seleccionado y activo.

Selección de objetos

Para seleccionar un objeto, pulse sobre él en **RMB**. Para seleccionar múltiples objetos, mantenga pulsado **SHIFT** y pulse en **RMB**. Generalmente, el último objeto para seleccionar es el que está activo: Aparece en rosa pálido, mientras que los objetos seleccionados no activos aparecen en color morado. La definición de objeto activo es importante por varias razones, incluyendo la "paternidad" de los mismos. Para deseleccionar el objeto activo, haga click sobre él de nuevo con **RMB**, si hay varios objetos seleccionados pulse la tecla **SHIFT** para mantener los otros como están. Pulse **AKEY** para seleccionar todos los objetos de la escena (si no hay ninguno seleccionado en ese momento) o para deseleccionarlos todos (si uno o más están seleccionados). **BKEY** activa Border select. Use la opción Border select para seleccionar un grupo de objetos dibujando un rectángulo mientras mantiene pulsada **LMB**. Se seleccionarán todos los objetos dentro del rectángulo o que estén en contacto con él.

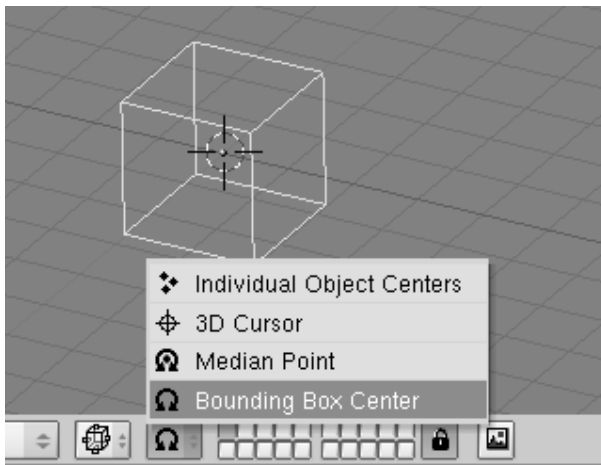
Movimiento (traslación) de objetos

Para mover grupos de objetos, pulse **GKEY** para activar el *modo Grab* (modo Agarrar) para todos los objetos seleccionados. Dichos objetos se verán como estructuras de malla de alambre que pueden moverse con el ratón (sin pulsar ningún botón del mismo). Para confirmar la posición nueva, pulse **LMB** o **ENTER**; para salir del modo Grab pulse **RMB** o **ESC**. La cabecera de la ventana 3D muestra la distancia que se está moviendo el objeto. Para restringir el movimiento a un solo eje del sistema global de coordenadas, entre en el modo Grab, mueva el objeto en el eje deseado, y pulse **MMB**. Para desactivar esta restricción pulse de nuevo **MMB**. Como nueva característica en la versión 2.3 se puede restringir el movimiento a un eje dado pulsando **XKEY**, **YKEY** o **ZKEY**. Cada tecla restringe el movimiento al eje correspondiente *global00*, tal como hace **MMB**. Una segunda pulsación de la misma tecla restringe el movimiento al eje correspondiente local del objeto. Una tercera pulsación de la misma tecla quita las restricciones. Se dibujan unas líneas para que se pueda visualizar mejor la restricción. Una vez que el modo Grab está activado se puede introducir la translación del objeto numéricamente simplemente tecleando el número. Esto permitirá introducir la primera coordenada mostrada en la cabecera de la ventana 3D. Se puede cambiar la coordenada con **TAB**, usando **NKEY** para salir/volver a entrar al modo de introducción numérica, **ENTER** para finalizar y **ESC** para salir. **BACKSPACE** dejará los valores originales. Nótese que debe usarse la tecla **.KEY** y no la **NUM**. para los decimales. Si mantiene pulsada la tecla **CTRL** mientras mueve el objeto activará el modo snap (movimiento por bloque), y el objeto se moverá en un número completo de unidades de Blender (cuadros

de la rejilla). El modo Snap termina cuando se suelta la tecla **CTRL**, por lo que debe asegurarse de confirmar la posición antes de hacerlo. La ubicación de objetos seleccionados puede dejarse al valor por defecto pulsando **ALT-G**.

Rotación de objetos

Para rotar objetos, active el modo Rotación (Rotate) pulsando **RKEY**. Al igual que en el modo Grab, puede cambiarse la rotación moviendo el ratón, confirmar con **LMB** o **ENTER**, cancelar con **RMB** o **ESC**. La rotación en el espacio 3D se produce alrededor de un eje, y hay varias formas de definir dicho eje. Blender define un eje mediante una dirección y un punto por el que pasa. Por ejemplo, por defecto la dirección de un eje es ortogonal a su pantalla. Si está viendo la escena desde delante desde un lado o desde arriba el eje de rotación será paralelo a una de las coordenadas globales del sistema de ejes. Si está viendo la escena desde un ángulo el eje de rotación estará en ángulo también, lo que puede originar una rotación muy extraña de su objeto. En ese caso necesitará mantener el eje de rotación paralelo a los ejes del sistema de coordenadas. Active y desactive este modo pulsando **MMB** durante el modo Rotación y vea el indicador de ángulo en la cabecera de la ventana. De igual modo, una vez está en modo Rotación, puede pulsar **XKEY**, **YKEY** o **ZKEY** para restringir la rotación a uno de los ejes de *referencia globales*. Pulsando **XKEY-XKEY** (dos veces **XKEY**) se restringe la rotación alrededor del eje x de la *referencia local* del objeto. Esto mismo es válido para la doble pulsación de **YKEY** y **ZKEY**. Como en el modo Grab, una tercera pulsación elimina las restricciones. Es posible introducir números para una rotación exacta, al igual que con las translaciones. Seleccione el punto por el que pasa el eje de rotación con el menú pertinente en la cabecera de la ventana 3D, como se describe más abajo. (*Los botones de selección de punto de rotación.*).



Los botones de selección de punto de rotación.

- *Centro del cuadro limitador (Bounding Box)* - el eje pasa por el centro del cuadro limitador de la selección. (Si sólo hay un objeto seleccionado, el punto usado es el punto central del objeto, que puede no ser necesariamente el centro geométrico. En *Los botones de selección de punto de rotación.* está en el medio del borde de la derecha, marcado por un punto granate. Para más información sobre este punto véase *mesh_modelling_editmode.*)

- *Punto de la Mediana* - el eje pasa por el punto de la mediana de la selección. Esta diferencia sólo es importante en el modo Edición, y el punto 'Mediana' es el baricentro de todos los vértices.
- *Cursor 3D* - el eje pasa por el cursor 3D. El cursor puede colocarse en cualquier sitio que se desee antes de rotar. Puede usarse esta opción para realizar fácilmente ciertas translaciones al mismo tiempo que se rota el objeto.
- *Centros de objetos individuales* - cada objeto seleccionado recibe su propio eje de rotación, todos paralelos entre sí y que pasan por el punto central de cada objeto, respectivamente. Si selecciona sólo un objeto, conseguirá el mismo efecto que con el primer botón.

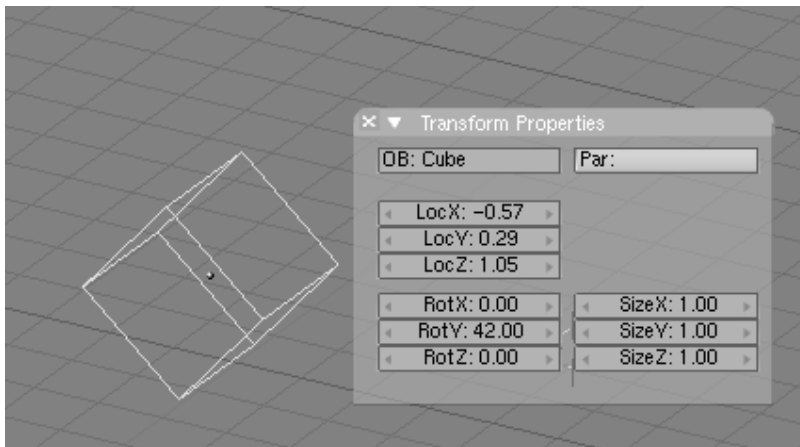
Si acaba de empezar a usar la rotación, no se preocupe demasiado por los detalles precedentes. Practique con las herramientas de Blender y se irá haciendo una idea de cómo trabajar con ellas. Manteniendo pulsada **CTRL** se pasa al modo Snap. En el modo Snap las rotaciones están restringidas a 5 pasos. Manteniendo pulsada **SHIFT** se puede hacer un ajuste fino. La rotación de los objetos seleccionados puede dejarse al valor por defecto pulsando **ALT-R**.

Cambio de tamaño/imagen especular de objetos

Para cambiar el tamaño de los objetos pulse **SKEY**. Al igual que en el modo Agarrar y Rotación cambie el tamaño moviendo el ratón, confirme con **LMB** o **ENTER**, y cancele con **RMB** o **ESC**. El cambio de tamaño en 3D requiere un punto central. Este punto se define con los mismos botones que para el punto de soporte del eje en la rotación (*Los botones de selección de punto de rotación.*). Si aumenta el tamaño del objeto, todos los puntos del mismo se alejarán del punto central seleccionado; si lo disminuye, los puntos se moverán hacia él. Por defecto, los objetos seleccionados cambian de tamaño uniformemente en todas direcciones. Para cambiar las proporciones (hacer el objeto más largo, ancho, etc.) puede bloquear el proceso de escalado a uno de los ejes globales de coordenadas al igual que haría para mover objetos. Para hacer esto entre el modo de cambio de tamaño, mueva el ratón un poco en la dirección del eje sobre el que quiere ejercer la acción y pulse **MMB**. Para volver al cambio de tamaño uniforme pulse **MMB** de nuevo. Podrá ver los factores de escala en la cabecera de la ventana 3D. De nuevo todas las consideraciones referentes a restricciones con respecto a un eje dado hechas en el modo Agarrar se mantienen, así como las de entrada numérica de datos. De nuevo **CTRL** activa el modo "Por Bloque" o Snap, con un escalado discreto a 0,1 pasos. Pulse **SHIFT** para un ajuste fino. El escalado de objetos seleccionados puede ponerse al valor por defecto pulsando **ALT-S**. La imagen especular se hace con una aplicación diferente de la herramienta de escalado. Dicha imagen no es más que escalar con un factor negativo en una dirección. Para hacer una imagen especular en los ejes X o Y pulse **SKEY** para ir al modo escalado, y luego **NKEY** para pasar a introducción numérica de datos. Seleccione las coordenadas deseadas e indique '-1' como factor de escala.

Panel de propiedades de transformación

Supongamos que quisiera que se mostrasen la posición/rotación/escalado de su objeto en números. O que quisiera introducir los valores de dichos parámetros de su objeto directamente. Para hacerlo, seleccione el objeto que quiere editar y pulse **NKEY**. El panel *Transform Properties* (Propiedades de transformación) (*El cuadro de diálogo numérico*) se muestra en pantalla. Pulse **SHIFT-LMB** y un número para introducir un valor, luego pulse **OK** para confirmar los cambios o mueva el ratón fuera de la ventana para cancelar.



El cuadro de diálogo numérico

El panel también muestra el nombre del objeto en el botón *OB*:. Puede editarlo desde este lugar.

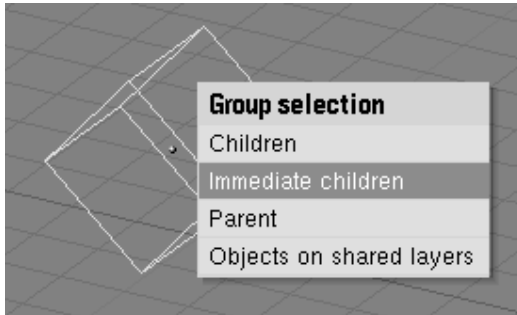
Duplicación

Para duplicar un objeto pulse **SHIFT-D** y se creará una copia idéntica de los objetos seleccionados. La copia se crea en la misma posición, en modo Agarrar. Será un objeto nuevo a excepción de que comparte Material, Textura e IPO con el original. Estos atributos se asocian a las dos copias, y el cambio de un material de uno de los objetos también afecta al otro. (Puede hacer materiales distintos para cada uno, como se describe en el capítulo de Materiales). Puede crear un *Duplicado Asociado* en vez de un duplicado real pulsando **ALT-D**. Esto creará un objeto nuevo con todos sus datos asociados al objeto original. Si se ha modificado uno de los objetos asociados en el Modo Edición todas las copias también serán modificadas.

Emparentar (Agrupar)

Para crear un grupo de objetos primero debe convertir a uno de ellos en padre de los otros. Para hacer esto, seleccione al menos dos objetos, pulse **CTRL-P**, y confirme la pregunta *¿Hacer padre?*. El objeto activo se convertirá en padre de todos los demás. El centro de todos los hijos estará ahora asociado al centro del padre por una línea de puntos. En este momento, el arrastre, la rotación y el escalado del padre tendrán el mismo efecto en los hijos. El Emparentado es una herramienta muy importante con muchas aplicaciones avanzadas, como se verá en capítulos posteriores. Pulse **SHIFT-G** con un objeto activo para ver el menú de Selección de Grupo (*Selección de grupo*). Este menú contiene:

- *Children* - Selecciona todos los hijos del objeto activo y los hijos de los hijos hasta la última generación.
- *Immediate Children* - Seleccionar todos los hijos del objeto activo pero no los hijos posteriores.
- *Parent* - Selecciona el padre del objeto activo.
- *Objects on shared layers* - En realidad esto no tiene nada que ver con padres e hijos. Selecciona todos los objetos de la(s) misma(s) capa(s) del objeto activo.



Selección de grupo


Mueva el hijo al padre borrando su origen (selecciónelo y pulse **ALT-O**). Borre una relación parental con **ALT-P**. Puede (*Liberación de hijos*):

- *Clear parent* - Libera los hijos, que vuelven a su ubicación, rotación y tamaño originales.
- *Clear parent...and keep transform* - Libera los hijos y mantiene la ubicación, rotación y tamaño que les dio el padre.
- *Clear parent inverse* - Coloca a los hijos con respecto al padre como si se hubieran colocado en la referencia Global. Esto borra definitivamente la transformación del padre sobre los hijos.

Figure 5-4. Liberación de hijos

Seguimiento (Tracking)

Para rotar un objeto de manera que se oriente hacia otro y mantener esta orientación si uno de los objetos se mueve seleccione al menos dos objetos y pulse **CTRL-T**. Aparecerá un cuadro de diálogo preguntándole si quiere usar una *restricción* de Seguimiento o el sistema de pista antiguo (Pre-2.30). La restricción de Pista se analizará en *character_constrains* y es el método recomendable. Se explicará aquí brevemente el sistema antiguo de seguimiento, de manera que se asume que tiene seleccionado *Old Track* en el cuadro de diálogo. Por defecto el objeto inactivo sigue al objeto activo de manera que los puntos de su eje local Y apuntan al objeto al que se hace el seguimiento. Sin embargo, esto puede no ocurrir si el objeto "seguidor" tiene ya una rotación. Se puede hacer un seguimiento con corrección cancelando la rotación (**ALT-R**) de este objeto. La orientación del objeto "seguidor" también se puede fijar con el eje Z

hacia arriba. Para cambiar esto, seleccione dicho objeto, cambie la ventana de Botones a Contexto de Objeto (, o F7) y seleccione el eje de seguimiento en la primera fila de seis botones, así como el eje que mirará hacia arriba en la segunda, que está en el panel *Anim Setting*. (*Configuración del eje de seguimiento*).



Configuración del eje de seguimiento

Para eliminar una restricción de seguimiento, seleccione el objeto "seguidor" y pulse **ALT-T**. Al igual que en la eliminación de las restricciones entre padres e hijos, debe elegir si quiere conservar o perder la rotación impuesta por el seguimiento.

Otras Acciones

Borrado

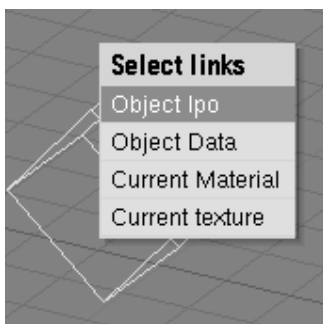
Pulse **XKEY** o **DEL** para borrar los objetos seleccionados. El uso de XKEY es más práctico para la mayoría de la gente, porque se puede pulsar fácilmente con la mano izquierda sobre el teclado.

Unión

Pulse **CTRL-J** para unir todos los objetos seleccionados en un único objeto. (Los objetos deben ser del mismo tipo). El punto central del objeto resultante se obtiene del objeto *activo* anterior.

Seleccionar enlaces

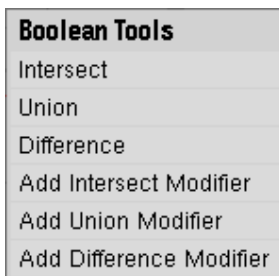
Pulse **SHIFT-L** para seleccionar todos los objetos que compartan un enlace con el objeto activo. Puede seleccionar objetos que compartan enlace IPO, de datos, de material, o de textura (*Selección de enlaces*).



Selección de enlaces.

Booleanos

Las operaciones booleanas son acciones concretas que pueden usarse únicamente con objetos de tipo malla. Mientras que funcionan para todos los objetos malla, están especialmente indicadas para usarse con objetos cerrados sólidos con una región interna y externa bien definidas. Así, es muy importante definir consistentemente las normales en cada objeto, es decir, todas las normales de cada objeto deben apuntar hacia afuera. Vea *chapter_mesh_modeling_A* para más información sobre normales y sobre por qué pueden producirse normales apuntando parcialmente hacia fuera y parcialmente hacia dentro. En el caso de objetos abiertos, el interior es definido matemáticamente extendiendo las caras del borde del objeto hasta el infinito. Como tal, puede encontrar que se produzcan resultados inesperados en estos objetos. Una operación booleana nunca afecta a los operandos originales; el resultado es siempre un nuevo objeto de Blender. Las operaciones booleanas se invocan seleccionando exactamente dos mallas y presionando **WKEY**. Hay tres tipos de operaciones booleanas para escoger en el menú emergente: *Intersect* (intersección), *Union* (unión) y *Difference* (diferencia). Las operaciones booleanas también toman en consideración materiales y texturas UV, produciendo objetos con índices de materiales u objetos multimapeados UV.

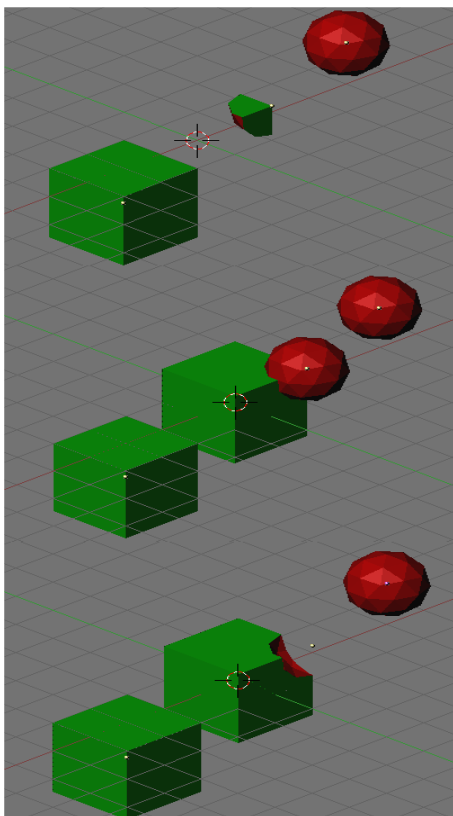


Opciones para operaciones booleanas.

Considere el objeto *Opciones para operaciones booleanas*.

- La operación *Intersect* crea un nuevo objeto cuya superficie encierra el volumen común a ambos objetos originales.
- La operación *Union* crea un nuevo objeto cuya superficie encierra el volumen de ambos objetos originales.
- La operación *Difference* es la única en la que el orden de selección es importante. El objeto activo (púrpura claro en vista de alambre) es restado del objeto seleccionado. Es decir, la superficie del objeto resultante encierra un volumen que corresponde al volumen del objeto seleccionado e inactivo, pero no al objeto seleccionado y activo.

Objetos resultantes para: intersección, unión, diferencia (de arriba a abajo). muestra el resultado de las tres operaciones.



Objetos resultantes para: intersección, unión, diferencia (de arriba a abajo).

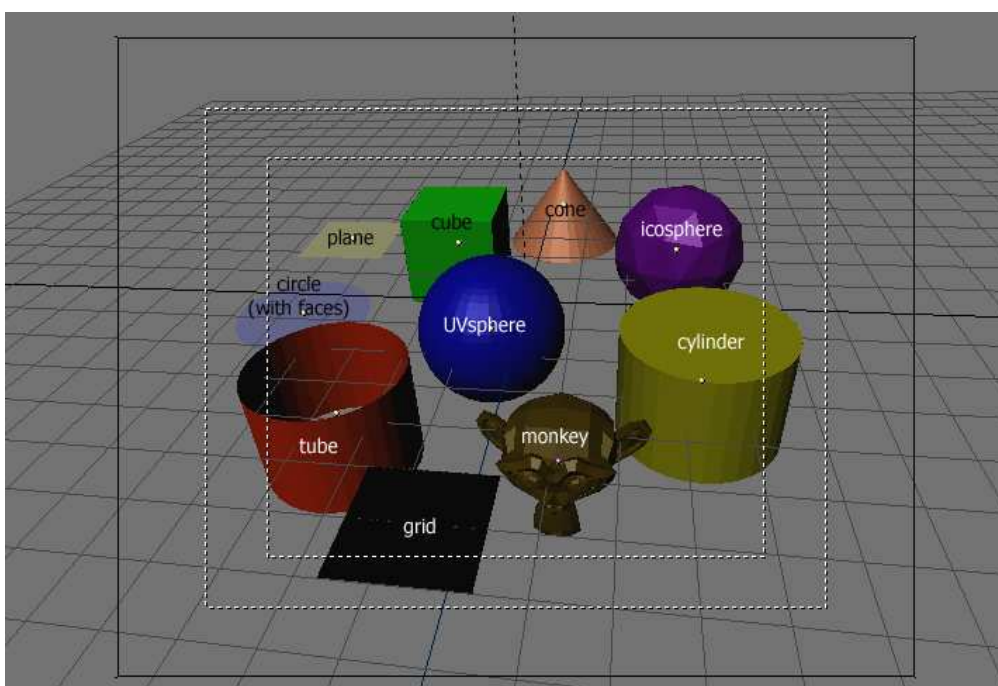
El número de polígonos generados puede ser muy grande comparado con las mallas originales, especialmente al usar objetos cóncavos complejos. Además, los polígonos generados pueden ser generalmente de calidad pobre: muy largos y delgados y en ocasiones muy pequeños. Intente usar el Decimator de mallas (botones EditButtons F9) para reparar este problema. Los vértices en la malla resultante que caen en la frontera de los dos objetos originales, frecuentemente no encajan, y los vértices del borde son duplicados. Esto es bueno en algunas circunstancias puesto que significa que puede seleccionar partes de la malla original seleccionando un vértice en el resultado y presionando el botón Select Linked (Seleccionar vinculados, **LKEY**). Ello es útil si desea asignar materiales y similares al resultado.

Nota: En ocasiones la operación booleana puede fallar con un mensaje diciendo "An internal error occurred -- sorry" ("Ocurrió un problema interno"). Si ello ocurre, intente mover o rotar los objetos ligeramente.

Objetos Básicos

El objeto principal de una escena en 3D es normalmente una *Malla* (Mesh). En este capítulo primero vamos a enumerar los objetos básicos de malla, o *primitivas*, para después seguir con una larga serie de secciones describiendo en detalle las acciones que se pueden realizar sobre objetos de malla.

Para crear un Objeto Básico pulse **SPACE** y seleccione "ADD>>Mesh", o acceda al menú 'add' pulsando **SHIFT-A** o simplemente mantenga presionado **LMB** sobre la ventana 3D por más de medio segundo. Seleccione del menú el objeto básico que le gustaría crear. En las líneas que siguen describiremos cada objeto básico o *primitive* que se puede crear en Blender. *Objetos Básicos o Primitivas* también muestra la variedad de objetos básicos que pueden crearse.



Objetos Basicos o Primitivas.

Plano (Plane)

Un plano estándar que contiene cuatro vértices, cuatro aristas y una cara. Es como una hoja de papel sobre una mesa; no es realmente un objeto tridimensional ya que es totalmente plano y no tiene espesor. Entre los objetos que pueden crearse a partir de planos se incluyen pisos, espejos y manteles.

Cube (Cubo)

Un cubo estándar que contiene ocho vértices, doce aristas, y seis caras, es un objeto tridimensional real. Entre los objetos que pueden crearse a partir de cubos podemos citar dados, cajas y cofres.

Circle (Circunferencia)

Una circunferencia estándar compuesta de n vertices. El número de vertices puede indicarse en la ventana que aparece cuando se crea la circunferencia. Mientras más vertices posea, más suave será el contorno. Ejemplos de objetos creados a partir de circunferencias son discos, platos o cualquier tipo objeto plano y redondo.

UVSphere (EsferaUV)

Una esferaUV estándar está hecha de n segmentos y m anillos. El nivel de detalle puede indicarse en la ventana emergente que aparece cuando se crea la esferaUV. Aumentando el número de segmentos y anillos, la superficie de la esferaUV se suaviza. Los segmentos son como meridianos terrestres, van de un polo al otro, en cambio los anillos son como paralelos terrestres. Ejemplos de objetos que pueden crearse a partir de esferasUV son pelotas, cabezas o las perlas de un collar.

Nota: Si crea una esferaUV de seis segmentos y seis anillos, obtendrá algo que en la VistaSuperior es un hexágono (seis segmentos), con cinco anillos más dos puntos en los polos. O sea, un anillo menos que los esperados, o dos más si cuenta los polos como anillos de radio 0.

Icosphere (Icoesfera)

Una icoesfera está hecha de triángulos. El número de subdivisiones puede especificarse en la ventana que aparece cuando se crea la Icoesfera; aumentando el número de subdivisiones se hace más pulida la superficie de la icoesfera. Una Icoesfera de nivel 1 es un icosaedro, un sólido con 20 caras triangulares equiláteras. Cualquier nivel de subdivisión mayor subdividirá cada cara triangular en cuatro triángulos, resultando en una apariencia más esférica cada vez. Las icoesferas se utilizan normalmente para obtener una disposición de los vertices más isotrópica y económica que en una esferaUV.

Cylinder (Cilindro)

Un cilindro estándar hecho de a partir de una sección circular de n vertices. El número de vertices de sección transversal circular puede indicarse en la ventana emergente que aparece cuando se crea el objeto; mientras mayor sea el número de vertices, más suave será la sección circular transversal. Algunos objetos que pueden crearse a partir de cilindros incluyen barras y manijas.

Tube (Tubo)

Un tubo estándar hecho de n vertices. El número de vertices en la sección transversal circular hueca puede especificarse en la ventana emergente que aparece cuando se crea el objeto; a mayor número de vertices más suave se vuelve la sección circular hueca. Entre los objetos que pueden crearse a partir de tubos, podemos citar caños o vasos. (La diferencia básica entre un cilindro y un tubo es que el primero posee sus extremos cerrados.)

Cone (Cono)

Un cono estándar hecho de n vertices. El número de vertices en la base circular puede especificarse en la ventana emergente que aparece cuando se crea el objeto; a mayor número de vertices más suave se vuelve la base circular. Entre los objetos que pueden crearse a partir de conos, podemos citar púas o sombreros de punta.

Grid (Rejilla o cuadrícula)

Una cuadrícula estándar hecha de n por m vertices. La resolución del eje X y el eje Y puede especificarse en la ventana emergente que aparece cuando se crea el objeto; cuanto más alta sea la resolución, mayor será el número de vertices creados. Entre los objetos que pueden crearse a partir de cuadrículas se incluyen paisajes (con la herramienta de edición proporcional o PET) y otras superficies orgánicas.

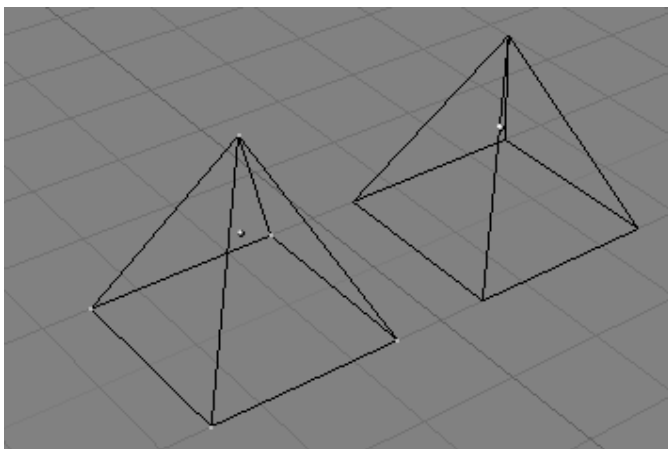
Monkey (Mono)

Es un regalo de la vieja NaN a la comunidad y es vista como una broma de los programadores o "huevo de Pascuas". Crea una cabeza de mono (de hecho, de "mona") una vez que se pulsa sobre el botón *Oooh Oooh Oooh*. El nombre de la mona es Suzanne y es la mascota de Blender.

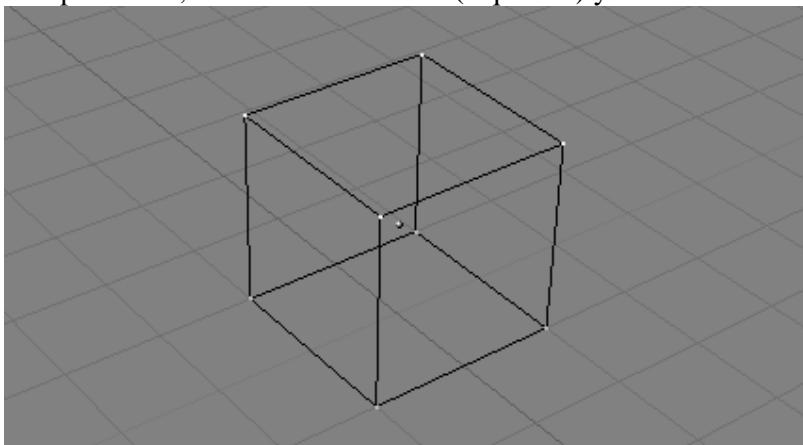
Modo Edición

Cuando se trabaja con objetos geométricos en Blender, se lo puede hacer en dos modos: Modo Objeto y Modo Edición. Básicamente, como se vio en la sección anterior, las operaciones en Modo Objeto afectan a los objetos completos, y las operaciones en Modo Edición solo afectan a su geometría pero no a sus propiedades globales como la Posición o la Rotación. En Blender puedes pasar de un modo al otro mediante **TAB**.

El Modo Edición funciona solamente con un objeto a la vez: el objeto activo. Fuera de Modo Edición los objetos se dibujan en color rosado en la Ventana 3D (en modo alambre) cuando son seleccionados de lo contrario aparecen en color negro. El objeto Activo en Modo Edición se dibuja en color negro, pero cada vértice es resaltado en color rosado (*Dos piramides, una en Modo Edición (izquierda) y una en Modo Objeto (derecha).*). Los vértices seleccionados son dibujados en color amarillo (*Cubo con los vértices seleccionados en amarillo.*) y, si se encuentran presionados los botones apropiados (*Draw Faces* y *Draw Edges*) en el Panel *Mesh Tools 1* del Contexto Edición (*F9*) también se resaltarán las caras y las aristas.



Dos piramides, una en ModoEdición (izquierda) y una en Modo Objeto (derecha).



Cubo con los vértices seleccionados en amarillo.

Estructuras: Vértices, Aristas y Caras

En las mallas básicas, todo se construye a partir de tres estructuras: *Vértices*, *Aristas* y *Caras*. (Aquí no estamos incluyendo a Curvas, NURBS, etc.) Pero no hay razón para desanimarse: esta simplicidad nos provee un cúmulo de posibilidades que serán la base para todos nuestros modelos.

Vértices

Un vértice es elementalmente un punto o una posición en el espacio 3D. Normalmente es invisible al renderizar y en Modo Objeto. (No confundir el punto central de un objeto con un vértice, es similar pero es más grande y además no puedes seleccionarlo.) Para crear un nuevo vértice, cambia a ModoEdición, mantén presionada **CTRL**, y pulsa con el **BIR**. Por supuesto, como la pantalla de la computadora es bidimensional, Blender no puede determinar las tres coordenadas del vértice a partir de un click de ratón, por lo que el nuevo vértice se ubica sobre el plano de profundidad del cursor 3D "dentro" de la pantalla. Cualquier vértice seleccionado previamente será conectado automáticamente al nuevo por medio de una arista.

Aristas

Una arista siempre conecta dos vértices con una línea recta. Las aristas son los "alambres" que ves cuando una malla se ve en la vista de alambre. Normalmente son invisibles en la imagen renderizada. Se usan para construir caras. Crea una arista seleccionando dos vértices y pulsando **TECLAF**.

Caras

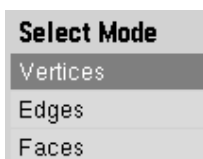
Una Cara es la estructura de más alto nivel en una malla. Las caras se usan para construir la superficie del objeto propiamente dicha. Las caras son lo que ves cuando renderizas una malla. Se la define como el área entre tres o cuatro vértices, con una arista en cada lado. Los triángulos siempre funcionan bien porque siempre son planos y fáciles de calcular. Ten cuidado al crear caras de cuatro lados, porque internamente son separados en dos triángulos. Las caras de cuatro lados solo funcionan bien si la cara es prácticamente plana (todos sus puntos descansan en un plano imaginario) y convexa (en ninguna esquina el ángulo es mayor o igual a 180 grados). Este es el caso con las caras de un cubo por ejemplo. (Esta es la razón por la que no pueden verse ninguna diagonal en modo alambre, porque dividiría cada cara cuadrada en dos triángulos. Aunque tu puedes crear un cubo con caras triangulares, solo se vería más confuso en Modo Edición.) Sin embargo, el área entre tres o cuatro vértices, delineada por aristas, no tiene que ser una cara necesariamente. Si esta área no tiene una cara, será simplemente transparente o inexistente en la imagen renderizada. Para crear una cara, selecciona tres o cuatro vértices y pulsa **TECLAF**.

Modos Vértice, Arista y Cara

En *Modo Edición* hay tres modos diferentes de selección:

- **Modo Vértices.** Pulse **TAB** y seleccione *Vértices* del menú. Los vértices seleccionados se dibujan en amarillo y los deseleccionados en color rosa.
- **Modo Arista.** Pulse **CTRL-TAB** y seleccione *Aristas* del menú. En este modo, los vértices no son dibujados. En su lugar, las aristas seleccionadas son dibujadas en amarillo y las deseleccionadas tienen un color negro.
- **Modo Caras.** Pulse **CTRL-TAB** y seleccione *Caras* del menú. En este modo, las caras se dibujan con un punto de selección en medio de la misma. Éste se usa para seleccionar la cara. Las caras seleccionadas se dibujan en amarillo con los puntos de selección en naranja; y las deseleccionadas se dibujan en negro.

Vea *Menú de selección de Modo Edición*. Casi todas las herramientas de modificación están disponibles en los tres modos. Por lo que puedes *Rotar*, *Escalar* y *Extrudir* etc. en todos los modos. Obviamente, rotar y escalar un vértice simple no hará nada útil, por lo que algunas herramientas son más o menos aplicables en algunos modos.



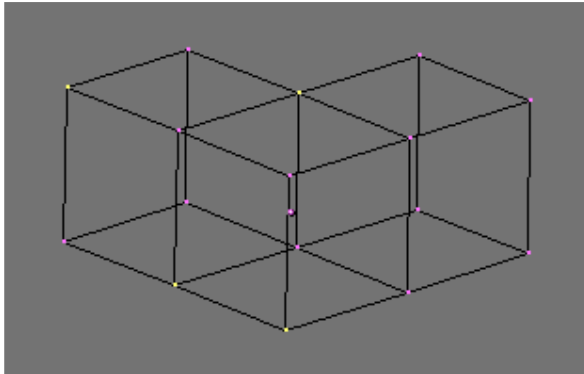
Menú de selección de Modo Edición.

Puede también acceder a los diferentes modos seleccionando uno de los tres botones en la barra de herramientas. Vea *Botones de selección de Modo Edición. Modos Vértices, Aristas y Caras de izquierda a derecha.* Usando los botones, puede también entrar en modos mixtos al hacer clic **SHIFT-BIR** en los botones.

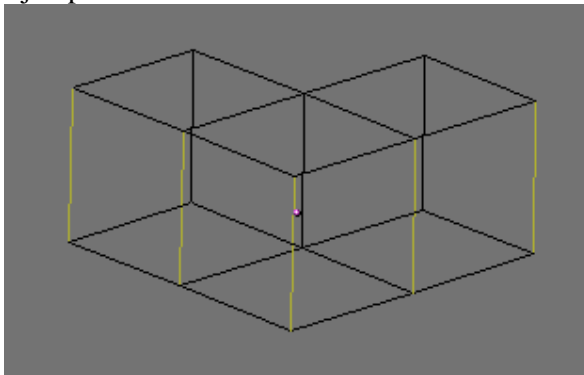
Cuando se cambia de modo, desde *Vértices* a *Aristas* y desde *Aristas* a *Caras*, las partes seleccionadas seguirán estándolo si forman un conjunto completo en el nuevo modo. Por ejemplo, si las cuatro aristas de una cara están seleccionadas, cuando se cambie del modo *Aristas* al de *Caras*, éste mantendrá la cara seleccionada. Todas las partes seleccionadas que no formen un conjunto completo en el nuevo modo, serán deseleccionadas. Vea *Ejemplo Modo Vértice*, *Ejemplo Modo Arista*, *Ejemplo Modo Cara* y *Ejemplo Modo Mixto* para ejemplos de los diferentes modos.



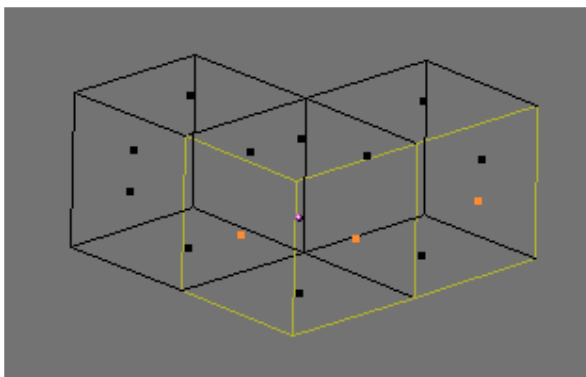
Botones de selección de Modo Edición. Modos Vértices, Aristas y Caras de izquierda a derecha.



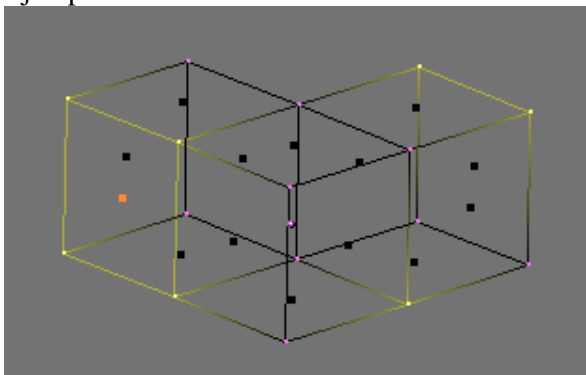
Ejemplo Modo Vértice



Ejemplo Modo Arista



Ejemplo Modo Cara

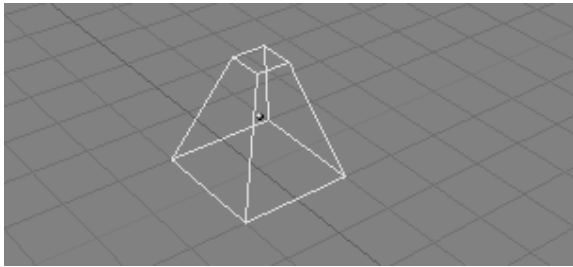


Ejemplo Modo Mixto


Edición Básica

La mayoría de las operaciones simples de Modo Objeto (como seleccionar, mover, girar y escalar) funcionan de manera idéntica sobre vértices a como lo hacen sobre objetos. Por lo tanto puedes aprender muy rápido cómo utilizar las operaciones básicas de Modo Edición. La única diferencia notable es una nueva opción para escalar **ALT-S** que escala los vértices seleccionados en la dirección de las normales (achica-engorda). Por ejemplo, la pirámide truncada en *Pirámide truncada*, fue creada con los siguientes pasos:

- Agrega un cubo en una escena vacía. Entra en Modo Edición.
- Asegurate que todos los vértices estén deseleccionados (rosa). Usa la selección por área (**TECLAB**) para seleccionar los cuatro vértices de la parte superior.
- Verifica que el centro de escalado esté seteado como cualquier cosa *excepto* el cursor 3D (ver *BSG.EDT.F.MAK.OBI*), luego cambia a Modo Escalar (**SKEY**), reduce el tamaño y confirma con **BIR**.
- Sal de Modo Edición pulsando **TAB**.



Pirámide truncada.

Una característica adicional para Modo Edición es la herramienta de Simetría. Si tienes seleccionados algunos vértices y pulsas **TECLAM** aparecerá un menú con nueve opciones. Puedes seleccionar una para rebatir el vértice seleccionado simétricamente respecto de cualquiera de los ejes X, Y o Z de las referencias Globales, local o según la vista. En ModoEdición también se encuentra disponible el modo SelecciónCircular. Se lo invoca pulsando dos veces la tecla **TECLAB** en lugar de una sola vez como harías para la Selección por Áreas. Aparecerá alrededor del cursor un círculo gris y con un click **BIR** se seleccionarán todos los vértices dentro de este círculo. Con **NUM+** y **NUM-** o la **RR**, si la hubiera, se agranda o reduce el círculo. Todas las operaciones en ModoEdición son ejecutadas sobre los vértices; las caras y aristas se adaptan automáticamente ya que dependen de la posición de los vértices. Para seleccionar una arista debes seleccionar los dos puntos de los extremos o bien ubicar el ratón sobre la arista y pulsar **CTRL-ALT-BMR**. Para seleccionar una cara, se debe seleccionar cada esquina. Las operaciones en Modo Edición son muchas y la mayoría se encuentran resumidas en la ventana de Botones de Contexto Edición, que se accede a través del botón de la cabecera () o por medio de **F9** (*Botones de Edición*). Observa el grupo de botones en el Panel *Mesh Tools 1*:



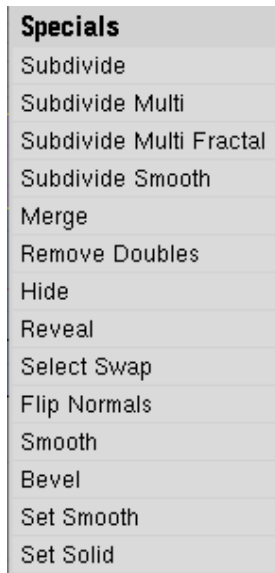
Botones de Edición

- *NSize*: - Determina la longitud, en Unidades de Blender, de las normales a las caras, si son dibujadas.
- *Draw Normals* - Activa/desactiva la representación de Normales. Si se encuentra activada, las normales de las caras se dibujarán como segmentos de color cian.
- *Draw Faces* - Si está activada, las caras se dibujan con un azul traslúcido, o rosado traslúcido si están seleccionadas. Si está en OFF, las caras son invisibles.
- *Draw Edges* - Las aristas se dibujan en negro, pero si este botón está activado, las aristas seleccionadas se dibujarán en amarillo. Las aristas que unan un nodo seleccionado y uno no seleccionado, tendrán un gradiente amarillo-negro.
- *Draw Creases* - If this is selected the edges, with creasing other than zero, a part of the edge will be drawn with a thicker line. See mesh_modeling_creases

- *Draw Seams* - If this is selected the edges, with a marked seam, will be drawn with an orange color. See link: TBD
- *All Edges* - En Modo Objeto se muestran solamente las aristas indispensables para definir la forma de un objeto. Con este botón puedes forzar a Blender a mostrar todas las aristas.
- *Edge Length* - If this is selected the length, in Blender units, will be drawn near the center of the selected edge.
- *Edge Angles* - If this is selected the angle, in degrees, will drawn near the corner of two selected edges. The edges must be connected to same corner.
- *Face Area* - If this is selected the area, in Blender units, will be drawn near the center of the selected face.

Nota: Por supuesto, todos estos colores son personalizables en el Editor de Temas.

Con **TECLAW** puedes invocar el menú "Specials" (*Menú Specials*). Con este menú puedes acceder rápidamente a muchas funciones de uso habitual en el modelado de polígonos.



Menú Specials.

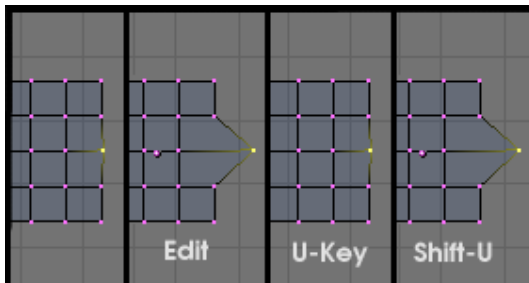
- *Subdivide* - Cada arista seleccionada se divide en dos, en el punto medio se crean vértices nuevos, y las caras se subdividen también si es necesario.
- *Subdivide Fractal* - Igual que la anterior, pero los vértices nuevos son desplazados aleatoriamente dentro de un rango definido por el usuario.
- *Subdivide Smooth* - Como la anterior, pero los nuevos vértices son desplazados hacia el baricentro (centro de masa) de los vértices conectados.

- *Merge* - Funde los vértices seleccionados en un único vértice ubicado en el baricentro o en la posición del cursor.
- *Remove Doubles* - Funde todos los vértices seleccionados cuya distancia relativa sea menor que un umbral especificado (0.001 por omisión).
- *Hide* - Oculta los vértices seleccionados.
- *Reveal* - Muestra los vértices ocultos.
- *Select Swap* - Todos los vértices seleccionados se deselectan y viceversa.
- *Flip Normals* - Cambia la dirección de las normales de las caras seleccionadas.
- *Smooth* - Suaviza una malla moviendo los vértices hacia el baricentro de los vértices enlazados.
- *Bevel* - Bevels the entire object regardless of the selected vertices, edges or faces. See mesh_modelling_bevel

Muchas de estas acciones tienen asignado un botón propio en el Panel *Mesh Tools* de la Ventana de Botones de Edición (*Botones de Edición*). El umbral para *Remove Doubles* se puede especificar allí también.

Deshacer para Mallas

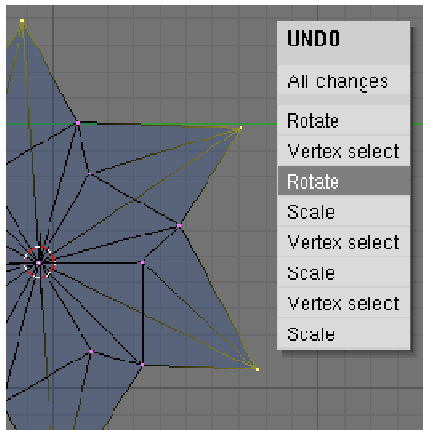
Blender tiene un sistema de deshacer global, dándole múltiples niveles de deshacer en todas las áreas de Blender; a excepción de: Modo Edición Armadura y ventanas Selección de Archivos, Audio y Oops. Las nuevas teclas rápidas globales para deshacer es **CTRL-Z** y **CTRL-SHIFT-Z** para rehacer. El comando Deshacer para mallas trabaja en segundo plano almacenando en memoria copias de tu malla a medida que es modificada. Pulsando **UKEY** en Modo Edición volvemos a la malla salvada deshaciendo la última operación de edición (*Deshacer* y *Rehacer*). Las operaciones de Deshacer solo se almacenan para una malla a la vez. Puedes salir e ingresar a la misma malla sin perder información de Deshacer, pero una vez que editas otra malla, se perderá la información de Deshacer para la primera.



Deshacer y Rehacer

Pulsando **SHIFT-U** re-hace la última operación deshacer (*Deshacer* y *Rehacer*). Pulsando **ALT-U** invoca la menú Deshacer (*Menú deshacer*). Allí se listan todos los pasos de Deshacer por nombre para que puedas encontrar rápidamente el camino hasta un punto conocido

aceptable de tu trabajo. El menú ALT-U contiene también la opción *All Changes*. Esta opción es más poderosa que simplemente pulsar varias veces **TECLAU** y recuperará la malla a como estaba al comienzo de la sesión de edición, incluso si has agotado todos los pasos de Deshacer.

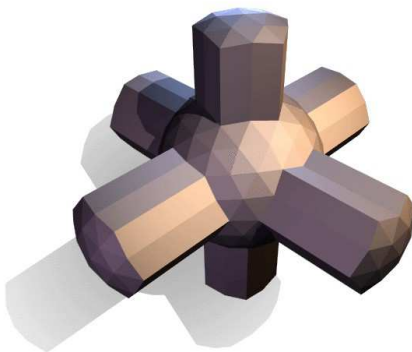


Menú Deshacer

El comando Deshacer de edición puede consumir mucha memoria. Una malla de 64.000 caras y vértices puede usar más de 3MB de RAM por cada paso. Si estás trabajando en una máquina con poca memoria RAM puedes ajustar el número máximo de pasos de Deshacer en la Ventana de Preferencias de Usuario, bajo *Edit Methods*. El rango permitido va de 1 a 64. El valor por omisión es 32.

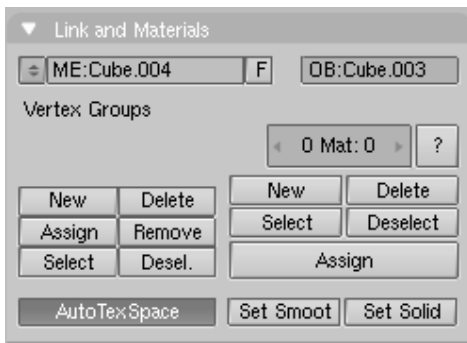
Suavizando (Smoothing)

Como se vió en las secciones anteriores, los polígonos son una parte fundamental de Blender. La mayoría de los objetos en Blender son representados a través de polígonos, y objetos realmente curvos a menudo son aproximados mediante mallas poligonales. Al renderizar imágenes, quizás observe que estos polígonos aparecen como una serie de pequeñas caras planas. (*Objeto de prueba simple sin suavizar*). A veces este es el efecto deseado, pero muchas veces queremos que nuestros objetos se vean lisos y suaves. Esta sección describe cómo suavizar un objeto, y como aplicar el filtro de AutoSuavizado (AutoSmooth) para combinar rápidamente polígonos suavizados y facetados en el mismo objeto.

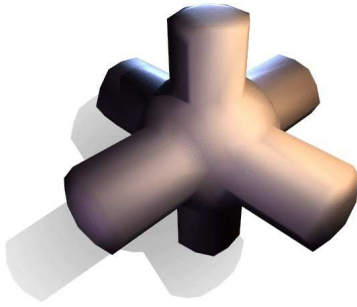


Objeto de prueba simple sin suavizar

Existen dos maneras de activar las características de suavizado de caras de Blender. La más fácil es suavizar o facetar un objeto entero seleccionando un objeto malla, en ModoObjeto, cambiar a los botones de Contexto de Edición (F9), y hacer click sobre el botón *Set Smooth* en el Panel *Link and Materials* (Botones *Set Smooth* y *Set Solid* de la ventana de botones de edición). El botón no se queda presionado, pero fuerza a Blender a asignar el atributo de "suavizado" a cada cara de la malla. Ahora al renderizar la imagen con F12 debería obtenerse la imagen que se ve en *El mismo objeto pero completamente suavizado con 'Set Smooth'*. Observe que el contorno del objeto aún se encuentra notoriamente facetado. Activar el suavizado no modifica realmente la geometría del objeto; solo cambia la forma en que se calcula el sombreado de las superficies creando la ilusión de una superficie suave y pulida. Haga click sobre el botón *Set Solid* en el mismo Panel para recuperar el sombreado al que se ve en el *Objeto de prueba simple sin suavizar*.

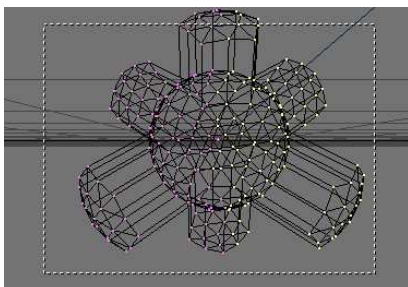


Botones Set Smooth y Set Solid de la ventana de botones de edición



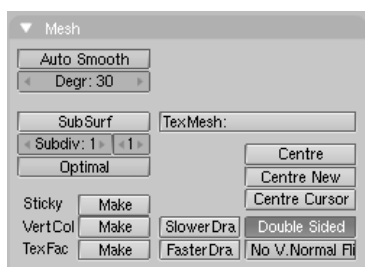
El mismo objeto pero completamente suavizado con 'Set Smooth'

Opcionalmente puede elegir qué caras suavizar ingresando en Modo Edición para el objeto con TAB, seleccionando las caras y haciendo click sobre el botón Set Smooth (*Objeto en ModoEdición con algunas caras seleccionadas.*). Cuando la malla está en ModoEdición, solo las caras seleccionadas recibirán el atributo "suavizado". Puede facetar caras (eliminando el atributo "suavizar") de la misma manera: seleccionando caras y haciendo click sobre el botón Set Solid.

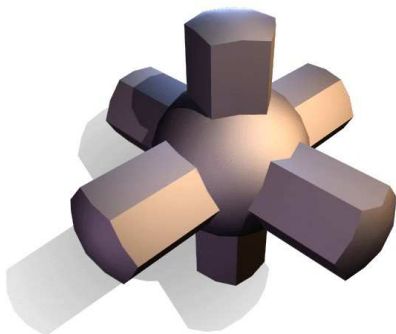


Objeto en ModoEdición con algunas caras seleccionadas.

Puede ser complicado crear algunas combinaciones de caras pulidas y facetadas empenado solo las técnicas descritas. Aunque hay remedios (tales como separar grupos de caras seleccionandolos y pulsando **TECLAY**), existe una manera más sencilla de combinar caras suaves y sólidas: utilizando AutoSmooth (AutoSuavizado). Presione el botón *AutoSmooth* en el Panel *Mesh* de los botones de edición (*Grupo de botones AutoSmooth en la ventana de Botones de Edición.*) para decirle a Blender qué caras deberán suavizarse basado en el ángulo entre caras (*La misma prueba con AutoSuavizado activado*). Los ángulos en un modelo que sean más agudos que el ángulo especificado en el botón numérico *Degr* no serán suavizados. Valores mayores produzcan caras suavizadas mientras que valores menores harán que se vea igual que una malla que fue facetada completamente. Solo las caras que han sido puestas como suaves, serán afectadas por el AutoSuavizado. Una malla, o cualesquiera caras que hayan sido puestas como Sólidas, no cambiarán su sombreado cuando se active el AutoSuavizado. Esto permite un control adicional sobre que caras serán suavizadas y cuales no, soslayando las decisiones tomadas por el algoritmo de AutoSuavizado.



Grupo de botones AutoSmooth en la ventana de Botones de Edición.



La misma prueba con AutoSuavizado activado

Extrudir

El comando "Extrudir"(**EKEY**) es una herramienta de importancia capital cuando se trabaja con Mallas. Este comando permite crear cubos a partir de rectángulos y cilindros a partir de círculos, así como crear muy fácilmente las ramas de un árbol. Aunque el proceso es bastante intuitivo, los principios detrás la Extrusión son bastante complejos como se describe líneas abajo.

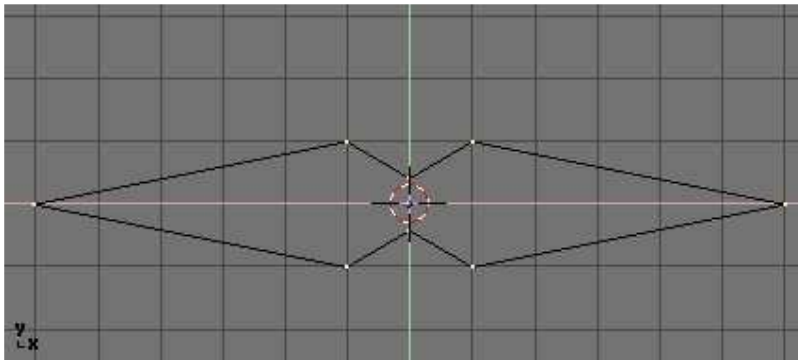
- Primero, el algoritmo determina el contorno exterior a extrudar; es decir, cuales de las aristas seleccionadas serán convertidas en caras. Por omisión, el algoritmo considera a las aristas que pertenecen a 2 o más de las caras seleccionadas como internas, y por ende no partes del contorno.

- Las aristas del contorno se convierten entonces en caras.
- Si las aristas en el contorno pertenecen solo a una cara de la malla, entonces todas las caras seleccionadas son duplicadas y conectadas a las caras recién creadas. Por ejemplo, los rectángulos se convertirán en cubos en esta etapa.
- En otros casos las caras seleccionadas son conectadas a las nuevas caras pero no duplicadas. esto evita que aparezcan caras indeseadas aprisionadas "dentro" de la malla resultante. Esta distinción es extremadamente importante ya que nos asegura la creación de volúmenes consistentemente coherentes todas las veces que se use Extrudar.
- Las aristas que no pertenezcan a ninguna cara seleccionada, es decir que formen un contorno "abierto", son duplicadas y se crea una nueva cara entre la nueva arista y la original.
- Los vértices aislados seleccionados que no pertenezcan a ninguna arista seleccionada son duplicados y una nueva arista se crean entre el vértice duplicado y el original.

El modo Mover se activa automáticamente una vez finalizado el algoritmo Extrudar, de modo que las caras, aristas y vértices recién creados puedan moverse utilizando el ratón. Extrudar es una de las herramientas de modelado más usadas en Blender. Es simple, directa y fácil de usar, pero aun así muy poderosa. La siguiente lección breve muestra cómo modelar una espada usando Extrudar.

La Hoja

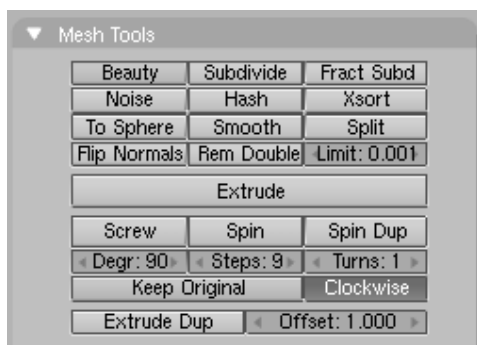
- Inicia Blender y borra el plano por defecto. En la vista superior añade un círculo malla con ocho vértices. Mueve los vértices para ajustarlos a la configuración mostrada en *Circulo deformado para generar la sección transversal de la hoja..*



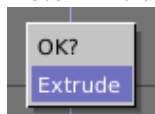
Circulo deformado para generar la sección transversal de la hoja.

- Selecciona todos los vértices y redúcelos con **S**KEY para que la figura entre en dos unidades de rejilla. Cambia a la vista frontal con **N**UM1.
- La figura que hemos creado es la base de la hoja. Usando Extrudar, crearemos la hoja en unos pocos pasos sencillos. Con todos los vértices seleccionados pulsa **E**KEY, o pulsa en el botón *Extrude* en el Panel *Mesh Tools* del Contexto de Edición (**F9** - *Botón Extrude en la ventana de Botones de Edición*). Un diálogo emergente nos preguntará

Ok? Extrude (Caja de confirmación de Extrude.). Pulsa sobre el texto o pulsa **ENTER** para confirmar, de lo contrario puedes mover el ratón fuera o presionar **ESC** para salir. Si ahora mueves el ratón verás que Blender ha duplicado los vértices, conectados a los originales con aristas y caras y que has entrado en modo Mover.

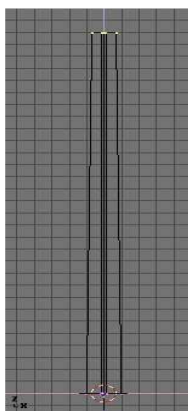


Botón Extrude en la ventana de Botones de Edición



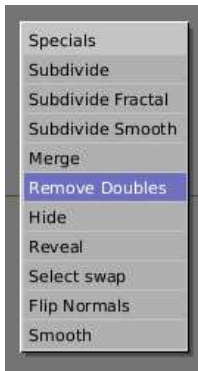
Caja de confirmación de Extrude.

- Mueve los vértices nuevos hasta 30 unidades hacia arriba, restringiendo el movimiento con **CTRL**, luego pulsa **LMB** para confirmar su nueva posición y achicalos un poco con **SKEY** (La Hoja).

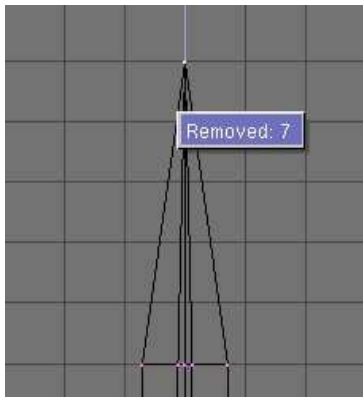


La Hoja

- Presiona **EKEY** otra vez para extrudar la punta de la hoja, luego mueve los vértices 5 unidades hacia arriba. Para hacer que la hoja termine en un vértice, escala los vértices superiores hasta 0.000 (mantén pulsada **CTRL** para esto) y pulsa **WKEY**>Remove Doubles (Menú Edición de Malla) o pulsa en el botón *Rem Doubles* en los Botones de Edición (F9). Blender informará que has eliminado siete u ocho vértices y quedará uno solo. ¡ La hoja está terminada ! (Menú Edición de Malla)



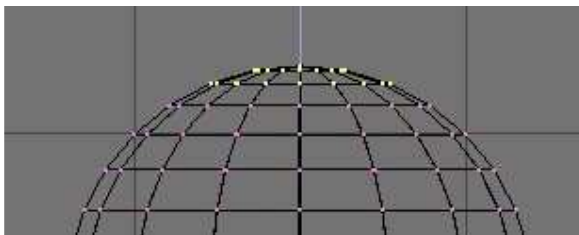
Menú Edición de Malla



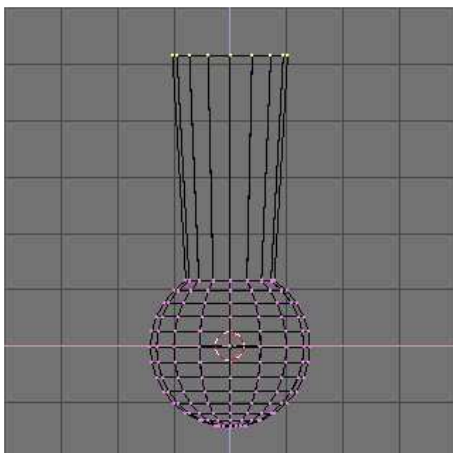
La hoja completa

La Empuñadura

- Sal de Modo Edición y mueve la hoja a un lado. Agrega una UVSphere con 16 segmentos y anillos, y deselecciona todos los vértices con **AKEY**.
- Selecciona los tres anillos de vértices superiores con **BKEY** y borralos con **XKEY>>Vértices** (La UV sphere para la empuñadura: los vértices se eliminan).

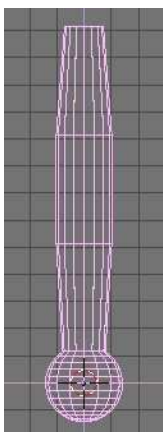


La UV sphere para la empuñadura: los vértices se eliminan



Primera extrusión de la empuñadura

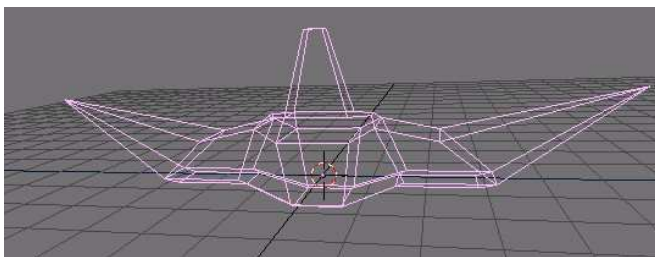
- Selecciona el anillo superior y extrudalo. Mueve hacia arriba cuatro unidades y agrandalo un poco (*Primera extrusión de la empuñadura*), luego extruda dos veces y mueve cada vez otras cuatro unidades hacia arriba. Achica el ultimo anillo un poco (*Empuñadura completa*).
- Sal de ModoEdición y escala la empuñadura completa para darle proporciones acordes a la hoja. Ubicalo debajo de la hoja.



Empuñadura completa

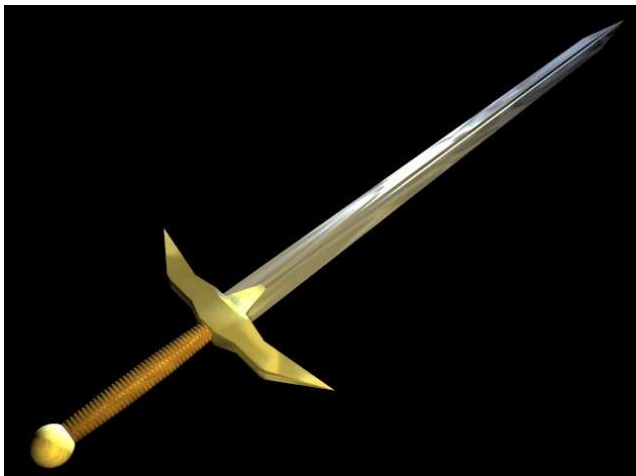
La Guarnición

Para este momento ya te deberías haber acostumbrado a la secuencia 'extrudar>mover>escalar', así que intenta modelar una linda guarnición siguiéndola. Comienza con un cubo y extruda distintos lados un par de veces, escalando cuando haga falta. Deberías ser capaz de obtener algo similar a *Guarnición Completa*.



Guarnición Completa

Después de texturizar, la espada se ve como en *Espada terminada, con texturas y materiales*



Espada terminada, con texturas y materiales

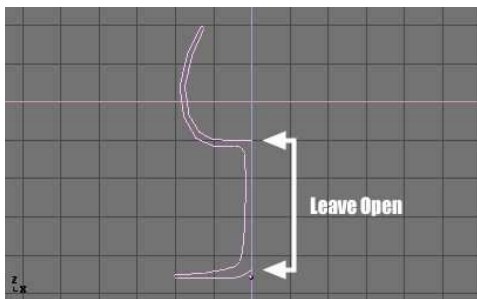
Como puedes ver, Extrudar es una herramienta muy potente que te permite modelar objetos relativamente complejos rápidamente (la espada completa fue modelada en menos de media hora). Sacando provecho de extrudar>mover>escalar hará tu experiencia como modelador en Blender mucho más agradable.

Giro (Spin) y Giro duplicado (SpinDup)

Spin y spin dup son otras dos poderosas herramientas de modelado que permiten crear fácilmente sólidos de revolución o estructuras axialmente periódicas..

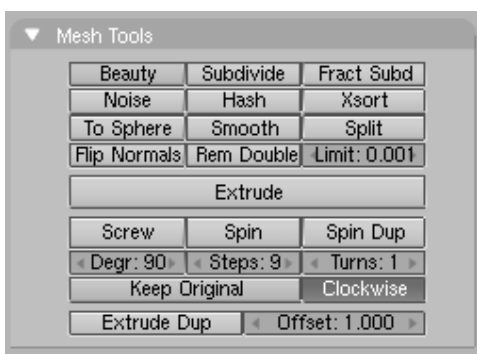
Giro

Use la herramienta Spin para crear el tipo de objetos que generaría por medio de un lathe. (Por esta razón, esta herramienta es llamada a menudo una herramienta "lathe" o "barrido" en los textos n.) Primero, cree una malla representando el perfil de su objeto. Si está modelando un objeto hueco, es una buena idea darle un espesor al contorno. La *Perfil del vaso* muestra el perfil para un vaso de vino que modelaremos como demostración.



Perfil del vaso

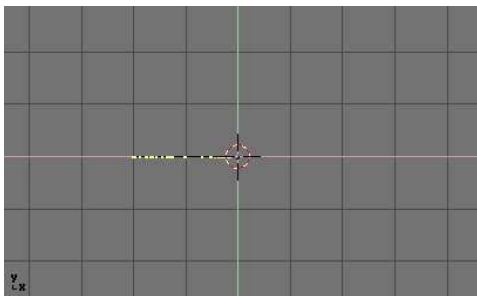
En ModoEdición, con todos los vertices seleccionados, acceda al Contexto Edición (F9). El botón *Degr* en el Panel *Mesh Tools* indica el número de grados que barrerá el objeto (en este caso queremos hacer un barrido de 360° completo). El botón *Steps* especifica cuantos perfiles se generarán en el barrido (Figure 6-27).



Botones de Spin

Al igual que Spin Duplicate (que se discute en la próxima sección), los efectos de Spin dependen de la posición del cursor y de qué ventana (vista) es la activa. Rotaremos el objeto alrededor del cursor en la Vista Superior. Cambie a la Vista Superior con NUM7.

- Ubique el cursor en el centro del perfil seleccionando algun vertice central y ajustando el cursor esa posición con **SHIFT-S>>Curs->Sel.** *Perfil del vaso, vista sueprior en ModoEdición, justo antes de realizar el barrido.* muestra el perfil del vaso de vino en la Vista Superior con el cursor ubicado correctamente.

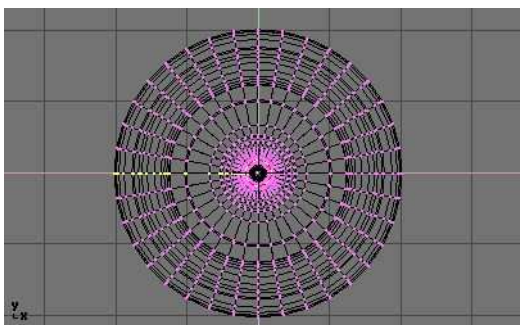


Perfil del vaso, vista sueprior en ModoEdición, justo antes de realizar el barrido.

Antes de continuar, observe el número de vertices del perfil. Encontrará estos datos en la barra de información en la parte superior de la interfaz de Blender (*Datos de malla - numero de caras y vertices.*).

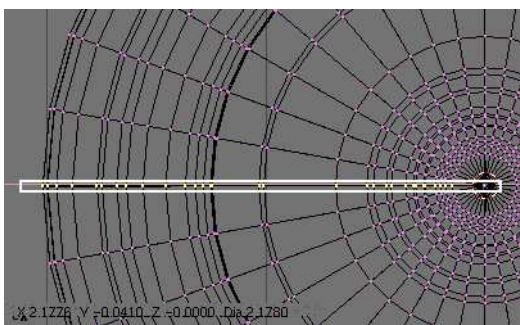
Datos de malla - numero de caras y vertices.

- Haga click en el botón "Spin". Si tiene más de una ventana abierta, el cursor cambiará a un flecha con un signo de interrogación, y deberá hacer click sobre la ventana que tenga la Vista Superior antes de seguir. Si tiene solo una ventana abierta, el barrido se realizará inmediatamente. *Perfil barrido* muestra el resultado de un barrido exitoso.



Perfil barrido

- La operación de spin deja vertices duplicados en el perfil. Puede seleccionar todos los vertices en la zona de la costura con la selección por área (BKEY) (*Selección de vertices en la costura*) y ejecutar un comando Remove Doubles (Borra Dobles).



Selección de vertices en la costura

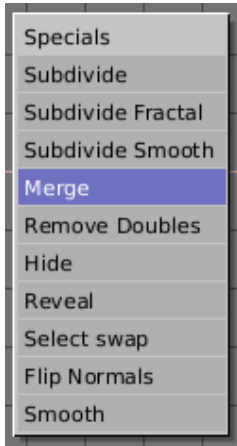
Observe la cuenta de vertices seleccionados antes y después de la operación Remove Doubles (*Cuenta de vertices después de eliminar dobles.*). Si todo va bien, la cuenta final de vertices (38 en este ejemplo) debería coincidir con el numero del perfil original observado en *Datos de Malla - Números de Vértices y Caras*. De no ser así, algunos vertices fueron olvidados y deberá fundirlos manualmente. O aún peor, se han fundido demasiados vertices.



Cuenta de vertices después de eliminar dobles.

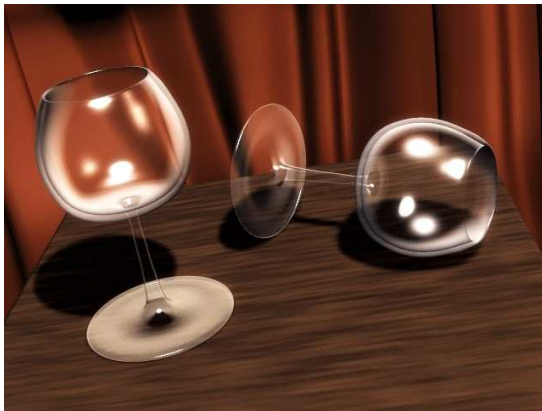
{{Tip|Fundiendo dos vertices en uno| Para fundir (soldar) dos vertices juntos, seleccione ambos vertices manteniendo pulsada **SHIFT** y haciendo clic con **RMB** sobre ellos. Presione **SKEY** para empezar a escalar y mantenga presionada **CONTROL** mientras escala, para reducir los vertices a 0 unidades en los ejes X, Y y Z. Haga click con **LMB** para finalizar la

operación de escalado y luego presione el botón *Remove Doubles* en la ventana de Botones de Edición. Alternativamente, puede presionar **W** y seleccionar *Merge* del menú emergente (*El menú Merge*). Luego, en el siguiente menú, elija si el nuevo vertice se ubicará en el centro de los vertices seleccionados o en la posición del cursor. En nuestro caso, es mejor la primer opción.



El menú Merge

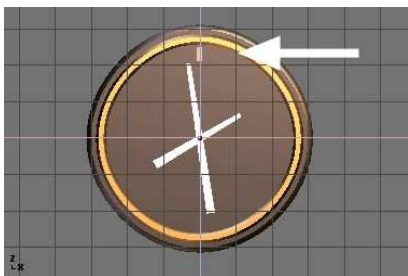
Todo lo que resta ahora es recalcular las normales, seleccionando todos los vertices y pulsando **CTRL-N** >> *Recalc Normals Outside*. En este momento puede abandonar ModoEdición y plaicar materiales o suavizar, poner algunas luces, un acámara y hacer un render. El *Render final de los vasos*. muestra nuestro vaso de vino terminado.



Render final de los vasos.

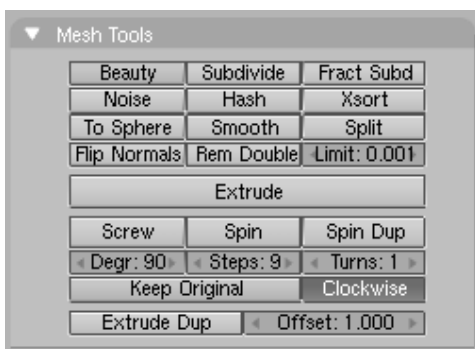
Giro duplicado

La herramienta Spin Dup es una excelente manera de obtener rápidamente una serie de copias de un objeto sobre una circunferencia. Por ejemplo, si hubiese modelado un reloj, y quisiera agregarle las marcas de las horas.



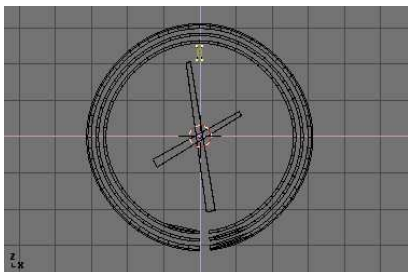
Marca de la hora indicada por la flecha

Modele solo una marca, en la posición de las 12 del reloj (*Marca de la hora indicada por la flecha*). Seleccione la marca y cambie a Contexto Edición con F9. Ponga 360 como el número de grados en el botón numérico *Degr:* en el Panel *Mesh Tools*. Queremos hacer 12 copias de nuestro objeto, así que ponga 12 en el parámetro *Steps* (*Botones Spin Dup*).



Botones Spin Dup

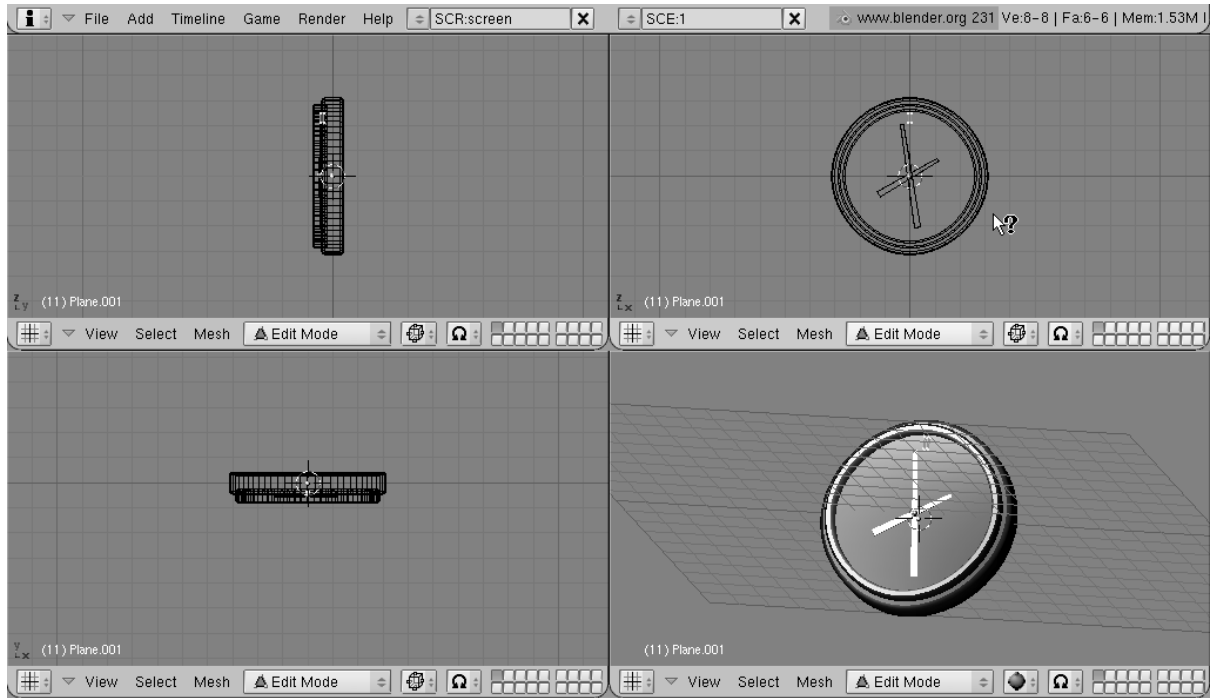
- Cambie la vista a una en la que desee rotar el objeto utilizando el teclado numérico. Tenga en cuenta que el resultado del comando *Spin Dup* depende de la vista que esté usando al momento de presionar el botón.
- Ubique el cursor en el centro de rotación. Los objetos serán rotados alrededor de este punto. Seleccione el objeto que desea duplicar y entre en ModoEdición con TAB.
- En ModoEdición, seleccione los vertices que quiera duplicar (recuerde que puede seleccionar todos los vertices con AKEY o todos los vertices conectados con el vertice bajo el puntero del ratón con LKEY) Ver *Malla seleccionada y lista para ser duplicada*.



Malla seleccionada y lista para ser duplicada

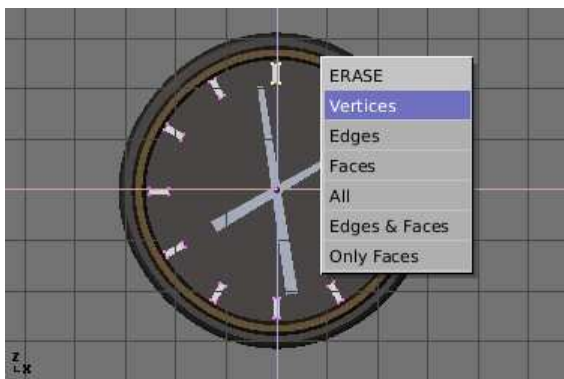
- Pulse el botón Spin Dup. Si tiene más de una Ventana3D abierta, notará que el puntero del ratón cambia a una flecha con un signo de interrogación. Haga click en la ventana en la que quiera realizar su rotación. En este caso, queremos emplear la vista frontal (*Selección de la Vista para Spin Dup.*).

Si la vista que desea no es visible, puede descartar la flecha/signo de interrogación con **ESC** hasta cambiar a una ventana con la vista apropiada con el teclado numérico



Selección de la Vista para Spin Dup.

Cuando se duplica con revolución un objeto barriendo 360 grados, un duplicado del objeto aparece en la misma posición que el objeto original, generando geometría doble. Notará que después de hacer click sobre el botón *Spin Dup*, la geometría original permanece seleccionada. Para borrarla simplemente presione **XKEY>>Vertices**. El objeto original será eliminado, pero la versión duplicada debajo del mismo, quedará intacta (*Eliminación de objetos duplicados*).



Eliminación de objetos duplicados

Evitando duplicados: Si gusta de un poco de matemática, no necesitará preocuparse de los duplicados pues podrá evitarlos desde el comienzo. Solo haga 11 duplicados, no 12, y no sobre un giro completo de 360° , sino en un barrido de 330° (o sea $360 \cdot 11/12$). De esta manera ningún duplicado aparecerá sobre el objeto original. En general, para hacer n duplicados en 360 grados sin superposiciones, gire un objeto menos sobre $360 \cdot (n-1)/n$ grados.

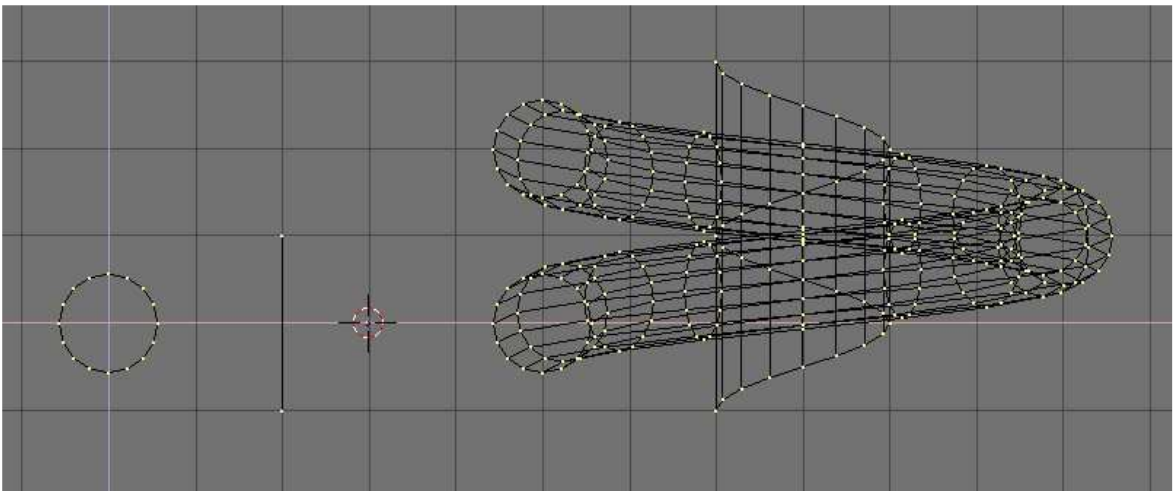
Render Final del Reloj. muestra el render final del reloj.



Render Final del Reloj.

Rosca (Screw)

La herramienta "Screw" o "Rosca", combina un "Spin" o "Giro" repetitivo con una traslación, para generar un objeto similar a una rosca o espiral (una helicoide). Emplea esta herramienta para crear roscas, resortes o estructuras en forma de concha.



El modo de uso de la función "Screw" es estricto:

- Pon la Ventana3DSet en Vista Frontal (NUM1).
- Pon el cursor3D en la posición en la que debe pasar el eje de rotación. Este eje debe ser vertical.

- Asegurate de disponer de una *poligonal abierta*. Esto puede ser tanto una arista única, como se ve en la figura, un semi círculo o cualquier otra.

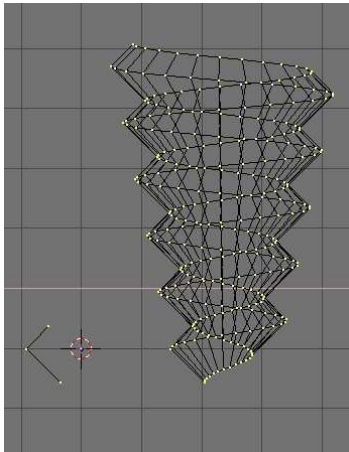
Solo necesitas asegurarte que haya dos extremos "libres"; dos vertices pertenecientes a una única arista enlazados luego a otro vertice. La función "Screw" localiza estos dos puntos y los usa para calcular el vector traslación que se suma al "Spin" en cada rotación (*Como hacer un resorte: antes (izquierda) y después (derecha) de aplicar el comando Screw.*). Si estos dos vertices están en el mismo lugar se crea un Spin normal. De otro modo, ¡ pasan cosas interesantes!

- Selecciona todos los vertices que participarán en el "Screw".

Asigna un valor a los Botones Numericos *Steps*: (Pasos:) y *Turns*: (Vueltas:) en el Panel *Mesh Tools*. *Steps*: determina cuantas veces se repite el perfil en cada giro de 360°, mientras que *Turns*: establece el número de rotaciones de 360° completas que tendrán lugar.

- Pulsa *Screw*!

Si hubiera multiples ventanas3D, el puntero del ratón cambia a un signo de interrogación. Cliquea sobre la Ventana3D en la que deberá ejecutarse el comando "Screw". Si los dos extremos "libres" están alineados verticalmente el resultado es el visto más arriba. Si no, el vector traslación se mantiene vertical, igual a la componente vertical de vector que une estos dos vertices "libres", mientras que la componente horizontal genera un agrandamiento (o una reducción) de la rosca como se ve en *Rosca creciente (derecha) obtenida con el perfil de la izquierda..*

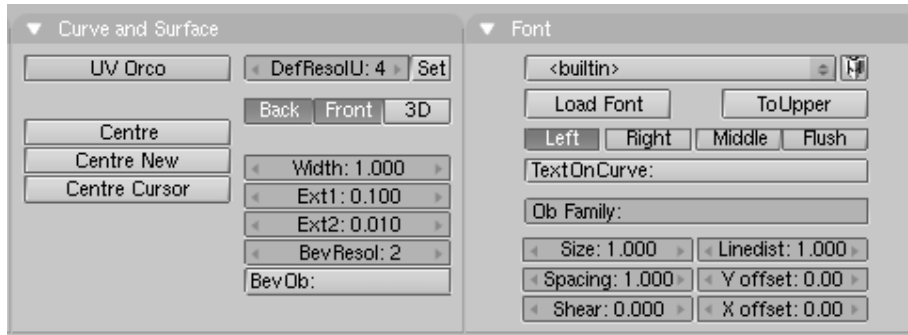


Rosca creciente (derecha) obtenida con el perfil de la izquierda.

Herramienta Curvar (Warp)

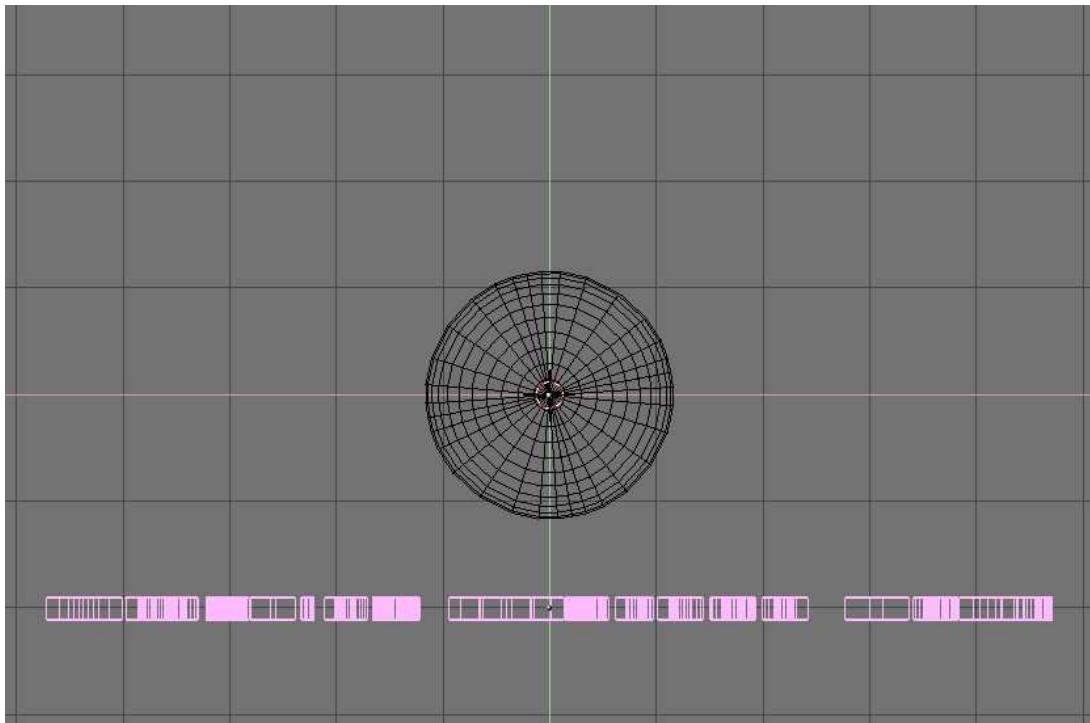
La herramienta Curvar (Warp) es una herramienta poco conocida en Blender, en parte debido a que no se la encuentra en la Ventana de Botones de Edición, y en parte porque solo es útil en casos muy específicos. No es algo que el usuario promedio de Blender vaya a utilizar todos los días. Un bloque de texto curvado alrededor de un anillo es útil cuando se crean logotipos voladores, pero sería muy difícil de modelar sin el uso de la herramienta Curvar. Para nuestro ejemplo, curvaremos la frase "Amazingly Warped Text" alrededor de una esfera.

- Primero cree la esfera.
- Luego cree el texto en VistaFrontal, en el Panel *Curve and Surface* del Contexto de Edición, ponga *Ext1* a 0.1 - haciendo el texto 3D, y ponga *Ext2* a 0.01, añadiendo un bonito chanfle al borde. Ponga *BevResol* a 1 o 2 para tener un chanfle suave y baje la resolución para que la cuenta de vertices no sea demasiado alta cuando luego subdividamos el objeto (*Parámetros del Texto* - y ver *chapter_text*). Convierta el objeto a curvas, y luego a malla, (**ALT-C** dos veces) porque la herramienta Curvar no funciona sobre texto ni curvas. Subdivida la malla dos veces, de modo que la geometría cambie de forma limpiamente, sin defectos o artefactos.



Parámetros del Texto.

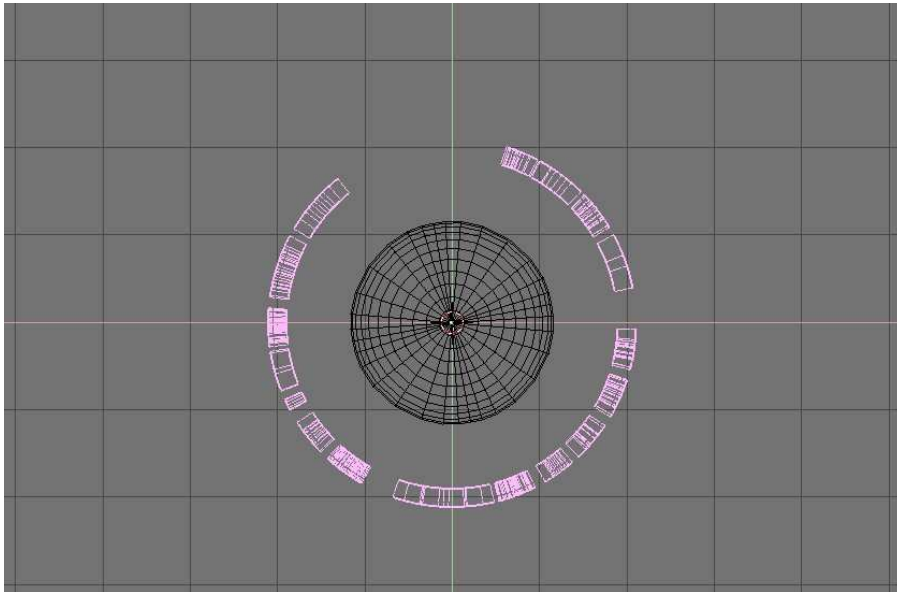
Cambie a Vista Superior y aleje la malla del cursor 3D. Esta distancia define el radio de curvatura a utilizar por Curvar. (Ver *Vista Superior del texto y de la esfera..*)



Vista Superior del texto y de la esfera.

Ingresa en ModoEdición para la malla (**TAB**) y pulse **AKEY** para seleccionar todos los vertices. Presione **SHIFT-W** para activar la herramienta Curvar. Mueva el ratón arriba o abajo

interactivamente para definir la cantidad de curvado. (*Texto curvado*). Manteniendo pulsada **CTRL** haremos que el curvado se haga en intervalos de cinco grados.



Texto curvado.

Ahora puede cambiar a vista de cámara, agregar materiales, luces y renderizar (*Render Final*).

Subdivisión de Superficies

Blender proporciona varias características de modelado avanzado de malla, mayormente orientadas a un manejo más sencillo de mallas complejas o a un manejo más económico del modelado de superficies suaves con un bajo número de vértices.

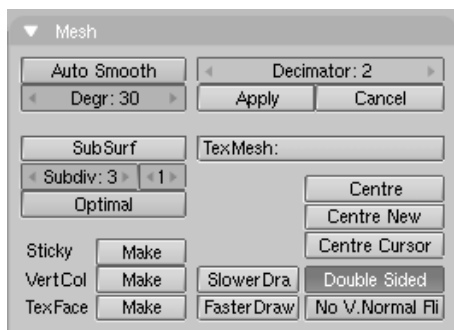
Subdivisión de Superficies Catmull-Clark

Con cualquier malla regular como base, Blender puede calcular una superficie suavizada en el momento, mientras se modela o al hacer el render, usando Catmull-Clark Subdivision Surfaces o, abreviando, *SubSurf*. SubSurf es un algoritmo matemático para calcular una subdivisión "suavizada" de la malla. Esto permite un modelado de Malla de alta resolución sin la necesidad de guardar y mantener enormes cantidades de datos. Esta técnica también permite darle una apariencia más "orgánica" a los modelos. En realidad, una malla con SubSurf y una superficie NURBS tienen muchos puntos en común, puesto que ambos se basan en una "malla" de bajos polígonos para definir una superficie de "alta resolución". Pero también hay notables diferencias:

- NURBS permite un control más preciso de la superficie, ya que se pueden definir "pesos" independientes para cada punto de control de la malla de control. En una malla con SubSurf no se pueden controlar los "pesos".
- SubSurfs posee una forma de modelado mas flexible. Como una SubSurf es una operación matemática funcionando sobre una malla, se pueden usar todas las técnicas

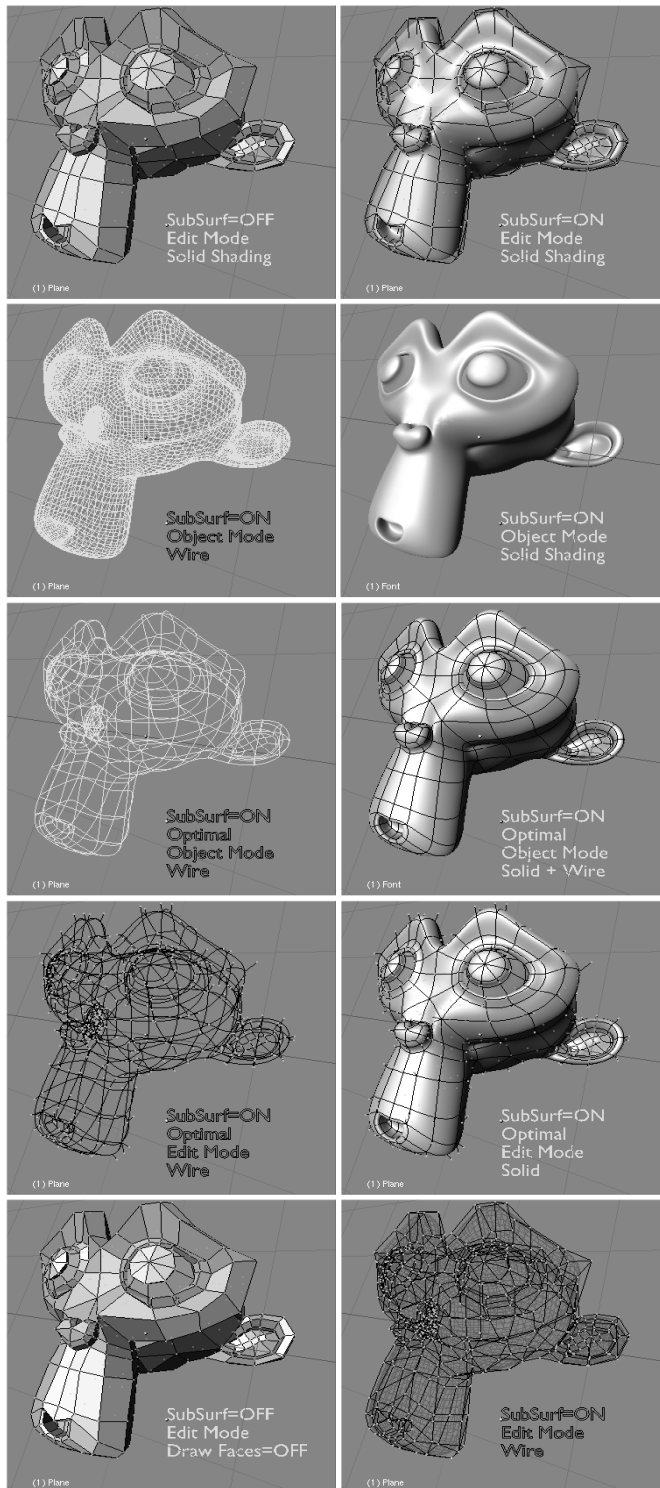
de modelado descritas en este capítulo sobre la malla. Existen más técnicas, que son mucho más flexibles, que las disponibles para NURBS control polygons.

SubSurf es una opción de la malla, que se activa en el Editing Context *Mesh* Panel (F9 - *Botones SubSurf*). Los Num Buttons debajo de este definen, a la izquierda, la resolución (o nivel) de subdivisión para propósitos de visualización 3D; el de la derecha, la resolución al momento del render. También se puede usar **SHIFT-O** estando en ModoObjeto. Esto activa o desactiva SubSurf. El nivel de SubSurf también puede ser controlado mediante **CTRL-1** a **CTRL-4**, pero esto sólo afecta el nivel de subdivisiones en la visualización. Ya que los cálculos de SubSurf son realizados en tiempo real, mientras se modela y al momento del render, y como utilizan el CPU de manera intensiva, es, en general, una buena costumbre mantener el nivel de SubSurf bajo (distinto de cero) mientras se modela, y alto al momento del render.



Botones SubSurf

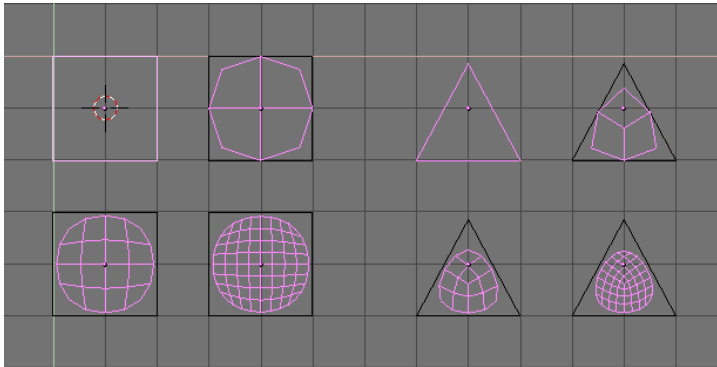
A partir de la versión 2.3, Blender tiene un nuevo botón relacionado a SubSurf: *Optimal*. Este botón cambia la forma en que las mallas con SubSurf son dibujadas y puede ser de gran ayuda al momento de modelar. *SubSurfed Suzanne*. muestra una serie de imágenes de diferentes combinaciones en Suzanne's Mesh.



SubSurfed Suzanne.

SubSurf de un cuadrado simple y caras rectangulares. muestra los niveles 0, 1, 2 y 3 de SubSurf aplicados a una cara cuadrada o a una cara triangular. Este tipo de subdivisión es realizada, en una malla genérica, en cada cara cuadrada o triangular.

Es evidente como cada cara cuadrada simple produce 4^n caras en la superficie subdividida. n es el nivel de SubSurf, o su resolución, mientras que cada cara triangular produce $3 \cdot 4^{(n-1)}$ caras nuevas (*SubSurf de un cuadrado simple y caras rectangulares.*). Este aumento dramático en la cantidad de caras (y vértices) produce una baja en la velocidad de todas las acciones de Edición y Render, y hace necesario usar un nivel de SubSurf más bajo durante el proceso de Edición, que durante el Render.



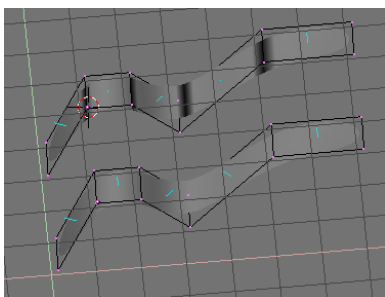
SubSurf de un cuadrado simple y caras rectangulares.

El sistema de subdivisión de Blender está basado en el algoritmo Catmull-Clarke. Esto produce una malla suavizada agradable, pero cada cara "subdividida", o sea, cada pequeña cara creada por el algoritmo a partir de una cara simple de la malla original, comparte la orientación de la normal de la correspondiente cara original.

Esto no es un problema para la forma en sí, como muestra *Vista lateral de mallas subdivididas. Con normales al azar (arriba) y con normales coherentes (coherent normals) (abajo)*, pero sí lo es en la etapa de render y en el modo de visualización sólido, donde cambios abruptos de las normales pueden producir desagradables líneas negras (Vista de Sólido de mallas Subdivididas con normales inconsistentes (inconsistent normals) (arriba) y con normales consistentes (consistent normals) (abajo).).

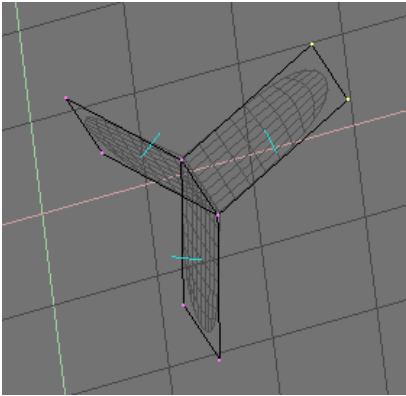
[[Image:Manual-Part-II-SubSurf05a.png|frame|none|Vista lateral de mallas subdivididas. Con normales al azar (arriba) y con normales coherentes (coherent normals) (abajo)

Utilizando el comando **CTRL-N** en ModoEdición, con todos los vértices seleccionados, para hacer que Blender recalcule las normales.



Vista de Sólido de mallas Subdivididas con normales inconsistentes (inconsistent normals) (arriba) y con normales consistentes (consistent normals) (abajo).

En estas imágenes las normales de las caras están dibujadas en azul claro. Se puede activar el dibujo de las normales en el menú BotonesEdición (F9). Nótese que Blender no puede recalcular las normales correctamente si la malla no es "Manifold". Una malla "Non-Manifold" es una malla para la cual un "out" no puede ser calculado inequívocamente. Básicamente, desde el punto de vista de Blender, es una malla donde existe bordes que pertenecen a más de dos caras. *Una malla "Non-Manifold"* muestra un ejemplo simple de una malla "Non-Manifold". En general, una malla "Non-Manifold" ocurre cuando se tienen caras internas y por el estilo.



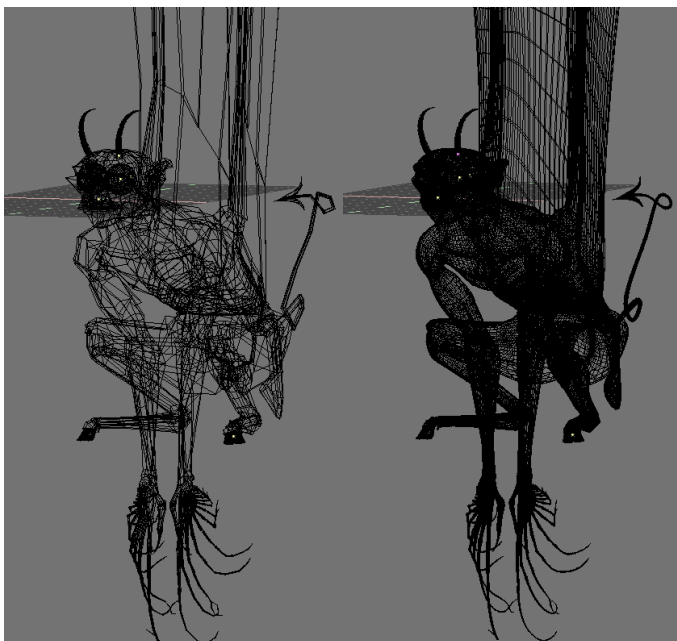
Una malla "Non-Manifold"

Una malla "Non-Manifold" no es un problema para mallas convencionales, pero puede producir artefactos extraños en mallas SubSurf. Además, no permite bajar el número de polígonos (decimation), por lo que es mejor evitarlo lo más posible. Dos pistas para saber cuando una malla es "Non Manifold":

- Al recalcular las normales, quedan líneas negras en alguna parte
- La herramienta para bajar el número de polígonos (Decimator) en el panel *Malla* no funciona, argumentando que la malla es "No Manifold"

Las herramientas SubSurf permiten la creación de buenos modelos "orgánicos", pero hay que tener en cuenta que una malla regular con caras cuadradas, en lugar de caras triangulares, da mejores resultados.

Malla base de una Gárgola (izquierda) y su correspondiente malla SubSurf con nivel 2 (derecha). y *Vista de sólido (izquierda) y render final (derecha) de la Gárgola.* muestran ejemplos de lo que se puede hacer con el SubSurf de Blender.



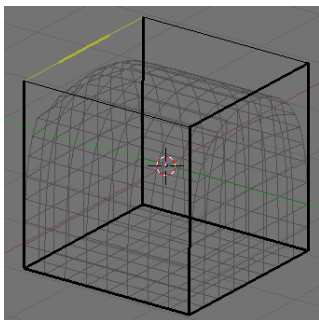
Malla base de una Gárgola (izquierda) y su correspondiente malla SubSurf con nivel 2 (derecha).

[[Image:Manual-Part-II-SubSurf08.png|frame|none|Vista de sólido (izquierda) y render final (derecha) de la Gárgola.

Pliegues pesados para superficies de subdivision

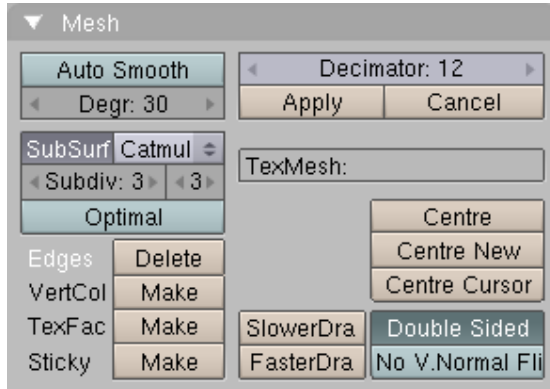
Los pliegues pesados para superficies de subdivision permiten ajustar lo afilado de los borde. Los pliegues son una propiedad de los bodes de una malla, y pueden ser editados en una malla *Modo Edicion* cuando la malla tiene subdivision. Selecciona los bordes que quieres afilar, y presiona **SHIFT-E** para cambiar la cantidad de filo del borde. Puedes activar una indicacion del filo de tu borde al habilitar **Draw Creases**. Ver *Herramientas de Mallas 1 panel*.

El valor de filo de un borde es indicado por una parte mas ancha en el borde. Si el borde tiene un filo de 1.0, el borde se vera bastante ancho, y si el valor de afilado es de 0.0, el borde sera delgado. Si el valor de filo esta entre 0.0 y 1.0, solo una parte del borde se vera mas ancha. Ver *Afilado de un borde alrededor de 0.5*.



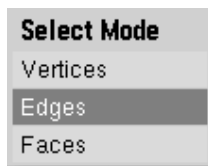
Afilado de un borde alrededor de 0.5

Para usar el afilado es necesario activar subdivision. Selecciona el cubo por default, si no tienes uno, añadelo! Ve al panel de Edicion (F9) y presiona el boton de *SubSurf*. Asegurate que el tipo de subdivision sea *Catmull-Clark* en el menu desplegable, ahora sube el nivel de subdivision a 3 en ambos, en el de desplegado y en el de render. Ver *Panel de Mallas*. El cubo se vera como una esfera.



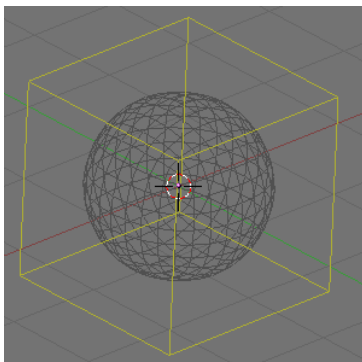
Panel de Mallas

Enta a *Modo Edicion* (TAB), con el cubo seleccionado. Por default estas en el modo de *Seleccion de Vertices*, ahora presiona CTRL-TAB para obtener un menu de *Seleccion de Modo*, *Menu Seleccion de Modo*, selecciona Bordes.



Menu Seleccion de Modo

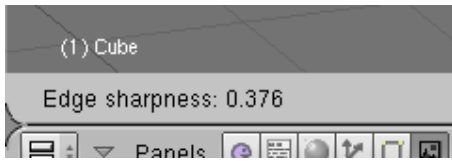
Ahora selecciona todos los bordes presionando **AKEY**. Todos los bordes deberian obtener un color amarillento, *Todos los bordes estan seleccionados*. Si todos los bordes son negros, entonces algo estaba previamente seleccionado. Presiona **AKEY** de nuevo para seleccionar todos los bordes.



Todos los bordes estan seleccionados

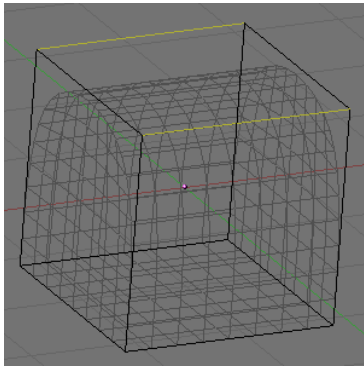
Ahora presiona **SHIFT-E** para editar el borde de filo del borde. El valor de filo se vera en tiempo real en la barra de herramientas, *Afilado de Bordes*. Mueve el cursor mas cerca o mas

lejos del borde para afectar su valor de afilado. Asigna el valor de afilado de todos los bordes en 1.0. El cubo regresara a su aspecto normal de cubo.



Afilado de Bordes

Ahora selecciona 2 bordes opuestos en la parte de arriba del cubo. Presiona **SHIFT-E** para editar el valor de afilado de los bordes. Asigna el valor de afilado a 0.0, *Dos bordes seleccionados*.



Dos bordes seleccionados

Si renderizas obtendras un bello cubo con una parte superior redondeada, *Resultado*. Una mejora muy buena en el volumen de trabajo para lograr esta clase de malla!



Resultado

Herramientas de Aristas y Caras

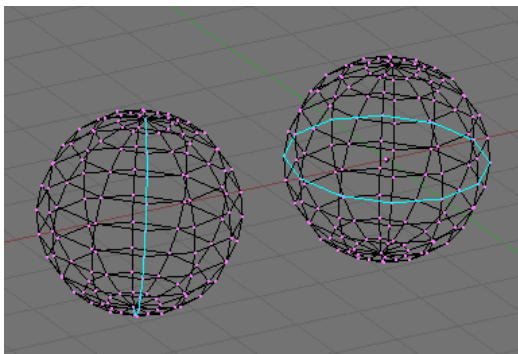
Algunas herramientas de modelado nuevas se incorporaron en Blender 2.30. Las mismas están focalizadas en modelado de Aristas en lugar de modelado de vertices. Un tema clave en Modelado es a menudo la necesidad de añadir vertices en ciertas zonas de una malla, y esto muchas veces tiene que ver con agregar o dividir aristas en una dada región. Blender nos ofrece ahora dos herramientas para ello, una *Herramienta Cuchillo* (Knife Tool) capaz de dividir aristas en los lugares deseados, y una herramienta *Bucle de Caras* (Face Loop) capaz de seleccionar hileras de caras y dividir las consistentemente. Las herramientas de Aristas están agrupadas en un menú enlazado a la **KKEY**, pero cada herramienta individual tiene también su propia combinación de teclas.

Selección de Arista/Cara

In EditMode, by pressing **ALT-B** one activates the edge/face select tool. If **ALT-B** is pressed once, then Blender is in edge select mode. The edge under the cursor is highlighted cyan. For each end point in the edge the following operations are performed:

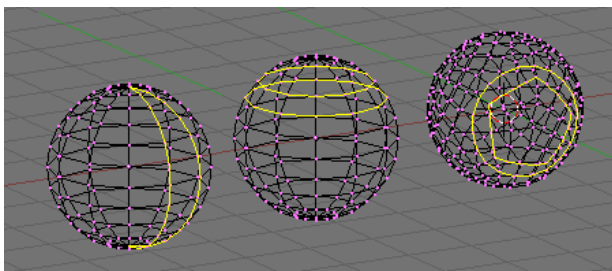
- It checks to see if it connects to only 3 other edges.
- If the edge in question has already been added to the list, the selection ends.
- Of the 3 edges that connect to the current edge, the ones that share a face with the current edge are eliminated and the remaining edge is added to the list and is made the current edge.

This way a loop of edges is highlighted (*One open (left) and one closed (right) Edgeloop.*). By pressing **LMB** such a highlighted loop is converted into a set of selected vertices. Any previously selected vertices become unselected. If you wish to add the highlighted loop to the current selection use **SHIFT-LMB**, while if you want to subtract the highlighted loop from the current selection use **ALT-LMB**.



One open (left) and one closed (right) Edgeloop.

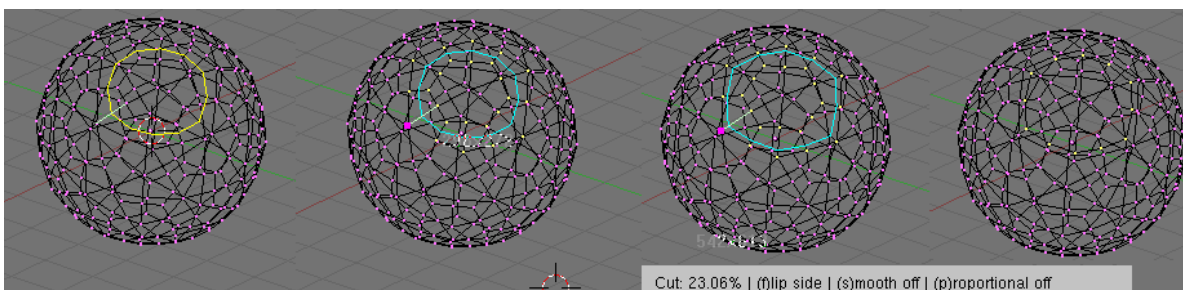
If **ALT-B** is pressed twice a Face Loop, rather than an Edge Loop, is highlighted. A face loop is made by two neighbouring edge loops and extends only to quadrilateral faces, ending when a triangular face is met (and the two bounding edgeloops merge into one). The same mouse actions apply as for the edge loops (*One open (left) and two closed (center and right) Faceloops.*). Face loop selection is also invoked with **SHIFT-R** in EditMode.



One open (left) and two closed (center and right) Faceloops.

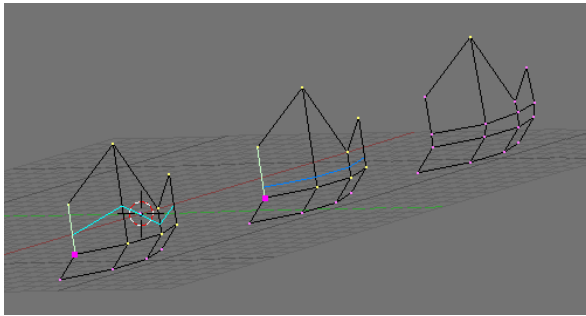
División de Bucles de Caras

The Loop tool allows you, eventually, to split, a *loop* of faces. This loop is defined as described in the previous section. In EditMode press **CTRL-R** rather than **SHIFT-R**. The edge under the cursor is aquamarine, the median line of the corresponding face loop is highlighted yellow (*Splitting a Faceloop.*, left). Once the face loop selection is performed via **LMB** a cyan line is highlighted between the two edgeloops defining the faceloop. One of the two vertices pertaining to the edge under the mouse pointer defining the edgeloop is highlighted via a big magenta dot (*Splitting a Faceloop.*, center left). Now by moving the mouse the cyan edge loop moves towards or away from the magenta dot. In the 3D Window header the distance of the edge loop from the reference magenta point is given as a percentage of the edge length. You can force the edge to move in 10% steps by keeping **CTRL** pressed. You can flip the reference vertex of the reference edge (the magenta point) with **F** (*Splitting a Faceloop.*, center right). By clicking **LMB** the edge loop is created, all faces and internal edges of the face loop are split in half at the points highlighted by the cyan edge loop. (*Splitting a Faceloop.*, right).



Splitting a Faceloop.

This is a really useful way to refine a mesh in a SubSurface-friendly way. By default the new, cyan, edge loop is created so that each edge is divided into two parts which are proportional one to the other and the proportionality ratio is the percentage given on the header (*Proportional and Smooth face cuts.*, left). You can force the new edge loop to stay at a given, fixed, distance from the edge loop to which the reference vertex belongs by switching proportional mode off with **P**. This turns the highlighted edgeloop blue too (*Proportional and Smooth face cuts.*, center). **P** acts as an on/off switch.

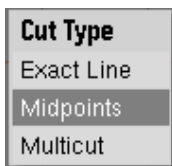


Proportional and Smooth face cuts.

Furthermore, by default, new vertices for the new edge loop are placed exactly on the pre-existing edges. This keeps subdivided faces flat. If a smoother result is desired **S** can be used, prior to finalizing the split, to set smooth mode on/off. If smooth mode is on then new vertices are not on the previous edge any more but displaced in the direction of the normal to the edge by a given percentage. A pop up asks for the percentage after **LMB** is pressed to finalize the split (*Proportional and Smooth face cuts.*, right).

Herramienta Cuchillo

La Herramienta Cuchillo funciona subdividiendo aristas si sus dos vertices están seleccionados y la arista es intersectada por una linea "cuchillo" trazada manualmente por el usuario. Por ejemplo, si deseas cortar un agujero en solo una cara en el frente de una esfera, puedes seleccionar solamente los vertices del frente y luego trazar la linea con el ratón. Para probar la herramienta, añade una Malla Rejilla (Grid). Estarás en Modo Edición con todos los vertices seleccionados. Pulsa **SHIFT-K** para invocar a la herramienta Cuchillo. Se te dará a elegir el tipo de corte: *Exact Line* dividirá las aristas exactamente por donde cruce el cuchillo, *Midpoints* dividirá las aristas intersectadas por su punto medio. Para este corte, elegimos *Midpoints*. Ahora puedes clicar con **LMB** y comenzar a trazar. Si mueves y haces click con **LMB** trazarás segmentos rectos entre los puntos cliqueados; si mantienes presionado el **LMB** mientras arrastras el ratón, trazarás lineas a mano alzada. Las poligonales se pueden trazar con un número arbitrario de segmentos, pero la rutina de intersección solo detectará un cruce por arista. Si cruzamos otra vez una arista no se le realizarán cortes adicionales. El **MMB** restringe el trazado a un eje como se espera. El ajuste a la rejilla no se encuentra implementado actualmente pero se espera incorporarlo en futuras versiones. Cuando hayas terminado de trazar tu linea, pulsa **ENTER** para confirmar el corte. **ESC** siempre cancelará la operación. *Cuchillo "Center" con poligonal (arriba); Cuchillo "Exact" con un único segmento (medio) and Cuchillo "Exact" en modo mano alzada (abajo).* muestra algunos ejemplos.

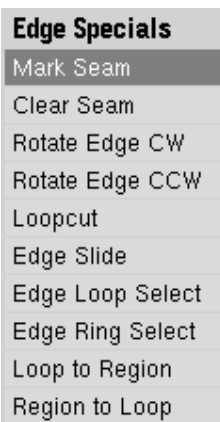


Cuchillo "Center" con poligonal (arriba); Cuchillo "Exact" con un único segmento (medio) and Cuchillo "Exact" en modo mano alzada (abajo).

Nota: Con mallas grandes, será más rápido seleccionar un numero pequeño de vertices, aquellos que definan las aristas que desees dividir, ya que el Cuchillo se ahorrará tiempo en verificar los vertices seleccionados contra los cruces del trazo de cuchillo.

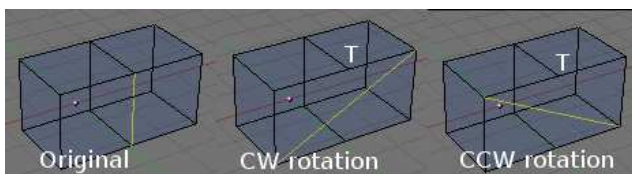
Herramientas Especiales de Aristas

La edición de aristas incluye algunas herramientas especiales. Éstas son *Rotación de Arista* en ambas direcciones para edición de mallas y *Marcar Costura* y *Limpiar Costura* que se usan para el mapeado UV. Para ver más sobre *Costuras* (seam) mira *materials_uv_textures_lscm*. Para acceder a *Rotación de Arista* pulsa CTRL-E y elige *Rotate Edge CW* o *Rotate Edge CCW* en el menu *Edge Specials*. Ver *Edge Specials menu*..

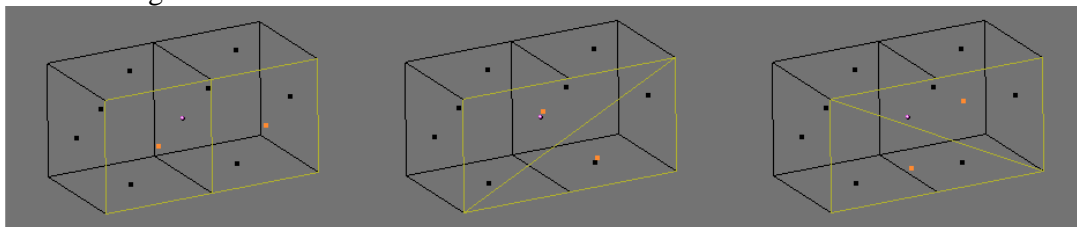


Edge Specials menu.

Para rotar una arista se necesita seleccionar una arista o dos caras adyacentes. Esto funciona en todo el three (NT:árbol?) *Elección de Modos de Edición*. Ver 'mesh_modelling_editmode_modes para aprender más sobre modos de selección. Seleccionando *Rotate Edge CW* or *Rotate Edge CCW* la arista seleccionada rotará en el sentido de las agujas del reloj o en sentido contrario. Ver *Selected Edge Rotated CW and CCW*. y *Common Edge rotated CW and CCW*..



Selected Edge Rotated CW and CCW.

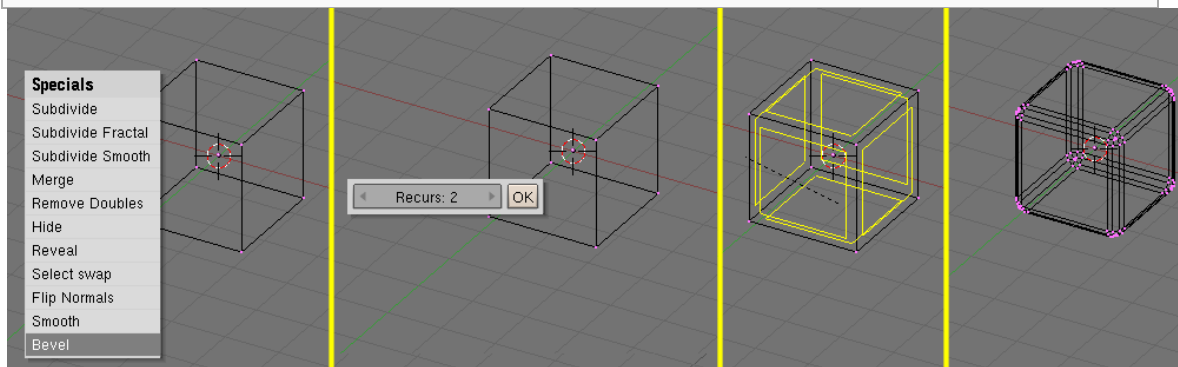


Common Edge rotated CW and CCW.

Herramienta de Biselado

Blender tiene, desde la versión 2.32, una herramienta para hacer solevados. Un solevado es algo que suaviza una esquina o arista afilada. En el mundo real las aristas de los objetos no son muy afiladas. Incluso el filo de un cuchillo no puede ser considerado una arista perfectamente afilada, si lo observas detenidamente, y la mayoría de las aristas están solevadas intencionadamente por razones prácticas o mecánicas. La herramienta de solevado de Blender está aún bajo un profundo desarrollo y la actual implementación está lejos de ser perfecta ya que soleva todos las aristas de la malla. Por tanto no hay control sobre los vértices que se quiere mantener afilados o sobre las aristas que se encuentran en superficies que no son necesarias solevarlas. La herramienta de solevado se puede usar en modo Edición, accediendo a través del menú de la **TECLA-W**, en la entrada *Bevel* (*Solevando un cubo.*, izquierda). Una vez seleccionada, una ventana emergente preguntará sobre el número de recursiones en el solevado (*Solevando un cubo.*, centro izquierda). Si es una recursión, entonces se reduce el tamaño de cara cara y de cada arista se crea una nueva cara. Caras de tres o cuatro vértices se crean según la necesidad. Si el número de recursiones es mayor que uno, entonces el procedimiento anterior es aplicado recursivamente ese número de veces, como por ejemplo si *Recur: 2* cada arista es transformada en 4 aristas, y tres nuevas caras aparecerán en la arista, suavizando la original. En general el número de nuevas aristas es 2 elevado al número de recurrencias *Recur*.

Número de vértices: Recuerda que por cada nueva arista se crean dos vértices, y algunos más que se crean en la intersección de las aristas, así que el número de vértices puede crecer muy rápidamente si soleva con un número de recursión elevado!!



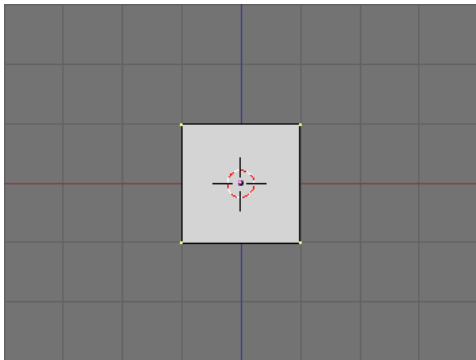
Solevando un cubo.

Una vez que el número de recurrencias *Recur* es seleccionado cada cara de la malla muestra un resaltado amarillo (*Solevando un cubo.*, centro derecha). Moviendo el cursor del ratón, crecerá o disminuirá el resaltado amarillo, y su correspondiente factor de disminución aparecerá en la cabecera de la ventana. Presionando **CTRL** el factor va en pasos de valor 0.1, presionando **SHIFT** puedes afinar dicho factor mucho más. Presionando **SPACE** una ventana emergente aparecerá, preguntándole cual valor de solevado desea introducir. **LMB** finaliza la operación y **RMB** o **ESC** la cancela. El resultado final se puede ver en (*Solevando un cubo.*, derecha).

Modelado Avanzado de Malla

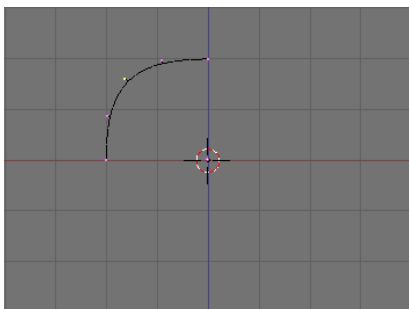
Modelado Simétrico

Con frecuencia es necesario modelar objetos que poseen algún tipo de simetría. Para simetría radial, rotacional o múltiple, la mejor aproximación es modelar cuidadosamente una estructura base y luego, como paso final, duplicar la célula básica utilizando SpinDup o cualquier comando apropiado. Para los objetos con simetría bilateral, es decir aquellos con un plano de simetría como la mayoría de los seres vivos (incluyendo al hombre) y muchas máquinas, el método supone el modelado de la mitad del objeto, y luego rebatir especularmente un duplicado de la primera mitad para obtener el objeto entero. Como normalmente es difícil mantener las proporciones correctas modelando solo una mitad, es posible duplicar la mitad antes de que esté totalmente modelada, y actuar solo sobre ésta actualizando automáticamente la otra.



Un plano.

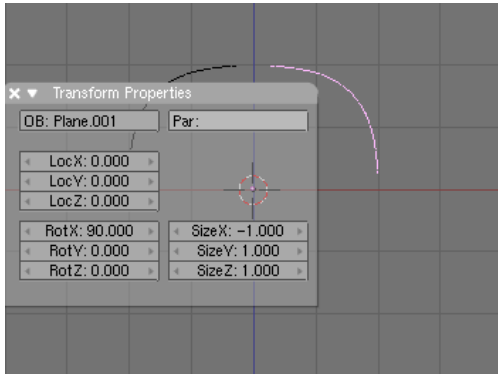
En la Vista Frontal, agregue un plano o cualquier otro objeto (*Un plano.*). Considérelolo como el punto inicial para la mitad del objeto. Digamos la mitad derecha del objeto, la cual para nosotros en la vista frontal, se encontrará a la izquierda de la pantalla. El plano de simetría es el plano YZ. Mueva la malla en Modo Edición, de manera que quede totalmente a la izquierda del centro. Elimine algunos vértices, y añada otros para darle la forma general como se ve en la *Mitad derecha.*



Mitad derecha.

Ahora cambie a Modo Objeto y con la mitad seleccionada, haga un duplicado vinculado con ALT-D. Pulse ESC para salir del Modo Mover, y presione NKEY. En el panel de entrada numérica que aparece, ponga *SizeX* a -1 (*Rebatiendo especularmente el duplicado*

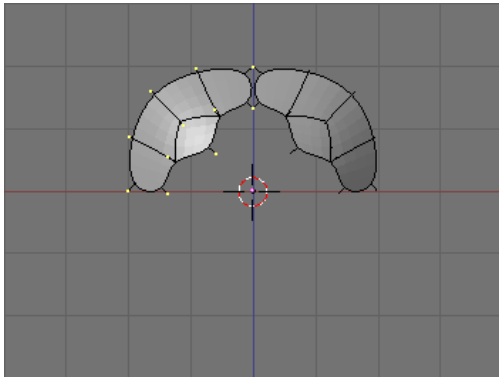
vinculado). esto rebatirá especularmente la copia vinculada con respecto al centro del Objeto, por ello la importancia de mantener el centro sobre un plano de simetría.



Rebatiendo especularmente el duplicado vinculado.

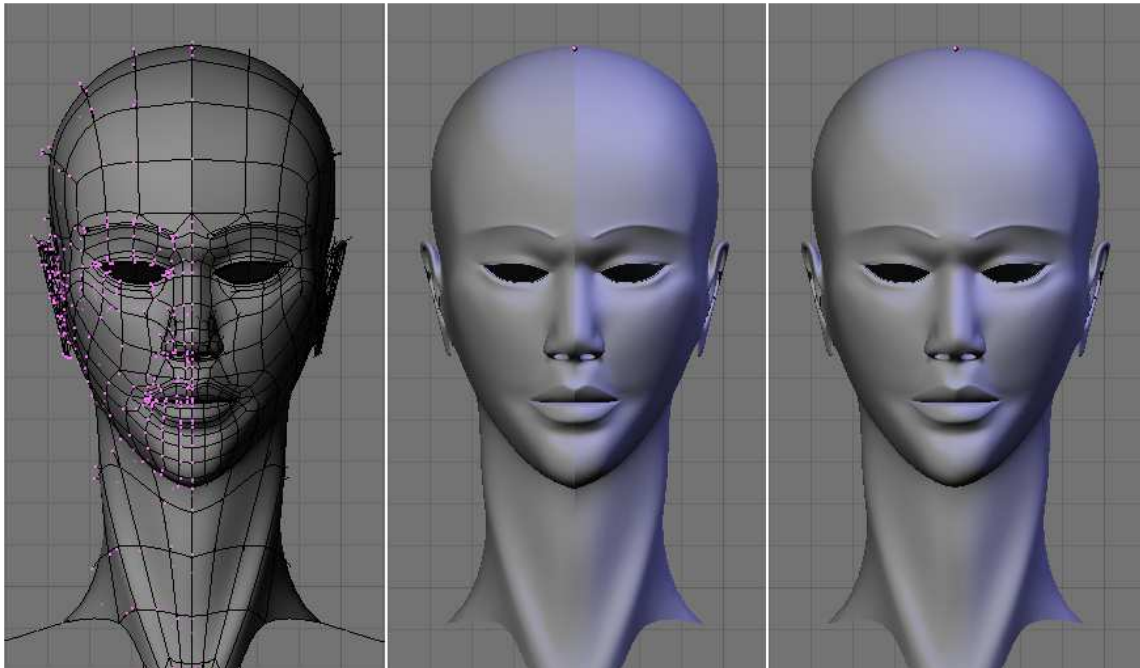
Haber duplicado el Objeto como un duplicado vinculado, significa que los dos objetos comparten los mismos datos de malla, los que están implícitamente rebatidos, por el escalado negativo unitario sobre el eje X, que es perpendicular al plano de simetría.

Ahora puede editar cualquier de las dos mitades. Como comparten los datos de malla, cualquier cambio, sea un extruido, borrado, corte en bucle de caras, etc, se reflejará inmediatamente en la otra mitad (*Editando una mitad*).



Editando una mitad.

Editando cuidadosamente una mitad, y posiblemente utilizando un esquema o diagrama como fondo que sirva de guía, se pueden lograr resultados muy interesantes (*Una cabeza. Izquierda: ModoEdición; Centro: ModoObjeto; Derecha: Unida.*, izquierda).



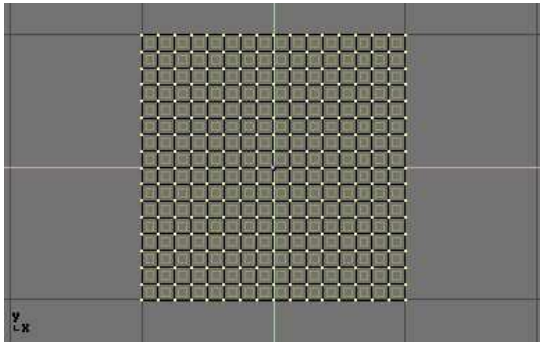
Una cabeza. Izquierda: ModoEdición; Centro: ModoObjeto; Derecha: Unida.

Como último paso, cuando esté completo el modelado simétrico, se deben seleccionar las dos mitades y unir las en un único objeto (**CTRL-J**). Esto hace desaparecer la junta (muy visible en la *Una cabeza. Izquierda: ModoEdición; Centro: ModoObjeto; Derecha: Unida.*, centro). Una vez unidas en un objeto único, (*Una cabeza. Izquierda: ModoEdición; Centro: ModoObjeto; Derecha: Unida.*, derecha), puede comenzar a modelar las sutiles asimetrías que toda persona tiene.

Nota: En Blender 2.33 y versiones anteriores, la implementación OpenGL provocaba que los duplicados vinculados rebatidos, tuviesen normales erróneas, de modo que una de las mitades era negra. Esto fue corregido en Blender 2.34, pero versiones viejas pueden aprovechar esta técnica igualmente, haciendo que la malla sea de lado simple cuando se usa el modelado simétrico.

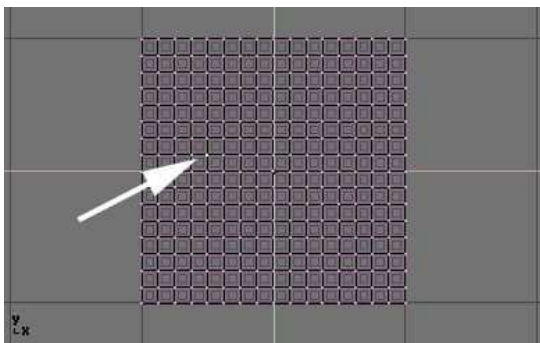
Herramienta de Modelado Proporcional (PET)

Cuando se trabaja con mallas densas puede ser difícil hacer ajustes finos a los vértices sin provocar desagradables pliegues y pellizcos en la superficie del modelo. Cuando te encuentres con situaciones como estas, utiliza la herramienta de edición proporcional. Actúa como un imán para deformar suavemente la superficie del modelo sin crear pliegues ni bultos. En la Vista Superior, agrega un plano a la escena con **SPACE>>Add>>Mesh>>Plane**. Subdivídelo un par de veces con **WKEY>>Subdivide** (o pulsando en el botón *Subdivide* del Panel *Mesh Tools* en el Contexto Edición para obtener una malla relativamente densa (*Una malla plana densa.*). O añade una rejilla directamente por medio de **SPACE>>Add>>Mesh>>Grid**, especificando el número de vértices en cada dirección. Cuando termines deselecciona todos los vértices con **AKEY**.



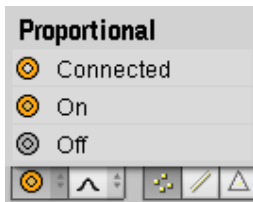
Una malla plana densa.

Selecciona un vertice en la malla pulsando **RMB** (*Una malla plana densa con solo un vértice seleccionado.*).



Una malla plana densa con solo un vértice seleccionado.

Mientras estés aún en modo Edición, activa la herramienta de edición proporcional pulsando la **Tecla O** o usando el icono de la barra de herramientas y seleccionando tanto a *On* como a *Connected*. (*Icono de Edición proporcional*). Cuando la edición proporcional esté activa el icono se pondrá de color naranja.



Icono de Edición proporcional

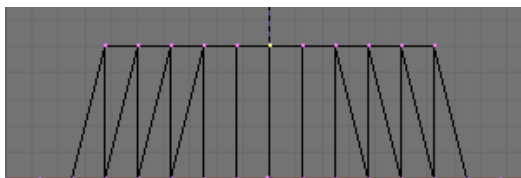
- *Connected* - Únicamente los vértices que estén conectados a otros en la malla, podrán verse afectados.
- *On* - Todos los vértices pueden verse afectados.
- *Off* - ¡La edición proporcional está desactivada!

Cambia el tipo de curva guía que se va a usar con el submenú *Mesh>>Proportional Falloff*, con el icono de la barra de herramientas *Menú Falloff*. o pulsando **SHIFT-O** para cambiar entre las opciones { {Literal|Constant{ {Literal| (Constante), *Linear* (Lineal), *Sharp* (Afilado),

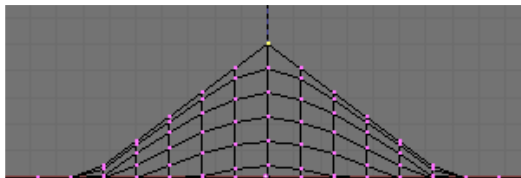
Root (Cuadrático), *Sphere* (Esférico) y *Smooth* (Suave). (*Constante* - Sin Falloff., *Falloff lineal.*, *Falloff Afilado.*, *Falloff Cuadrático.*, *Falloff Esférico.* y *Falloff suave.*).



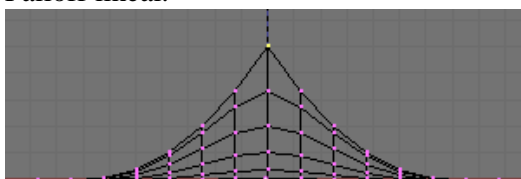
Menú Falloff.



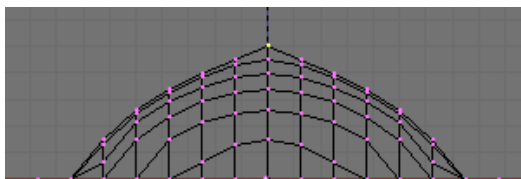
Constante - Sin Falloff.



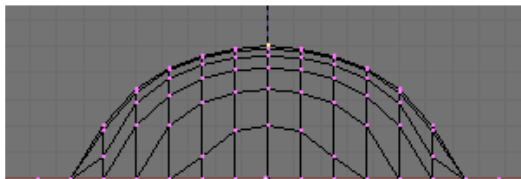
Falloff lineal.



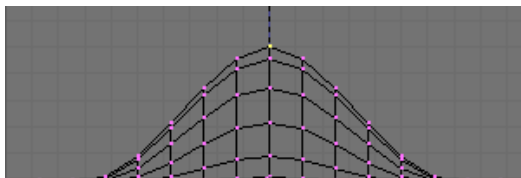
Falloff Afilado.



Falloff Cuadrático.

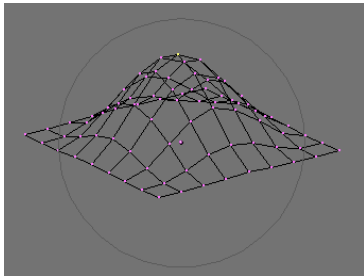


Falloff Esférico.



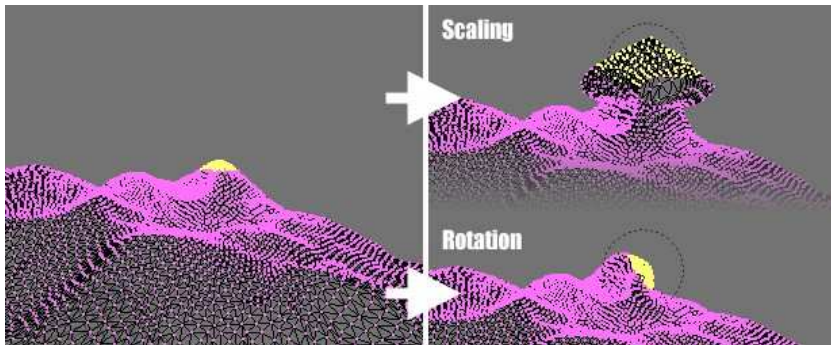
Falloff suave.

Cambia a la vista frontal (NUM 1) y activa la herramienta mover con **GKEY**. A medida que arrastres el puntero hacia arriba notarás que los vértices cercanos también son arrastrados con él. Cuando estés satisfecho con la ubicación del vértice, presiona **LMB** para fijar su posición. Si no estás conforme, anula la operación y vuelve la malla a su estado inicial con **RMB** o **ESC**. Mientras estás editando puedes aumentar o reducir el radio de influencia (mostrado como el círculo punteado en *Círculo de influencia*.) pulsando **NUM+** y **NUM-** respectivamente. A medida que modificas el radio, los puntos alrededor de la selección ajustaran sus posiciones correspondientemente. También puedes utilizar **MW** para agrandar o achicar el círculo.



Círculo de influencia.

La herramienta de edición proporcional puede utilizarse para producir grandes efectos con las herramientas de escalado (**SKEY**) y rotación (**RKEY**) como se ve en *Un paisaje obtenido con Edición Proporcional*.



Un paisaje obtenido con Edición Proporcional

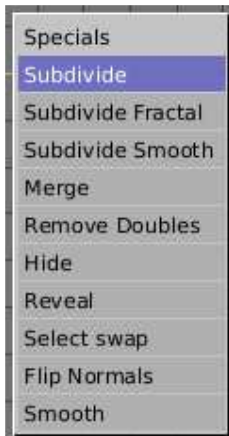
Combina estas técnicas con pintura de vértices para crear paisajes fantásticos. *Paisaje final renderizado* muestra los resultados de la edición proporcional luego de aplicar texturas e iluminación.



Paisaje final renderizado

Ruido

La función Ruido permite desplazar los vertices en las mallas basado en los valores de gris de una textura aplicada a las mismas. De esta manera se pueden generar grandes paisajes o texto esculpido en las mallas.



Herramienta Subdivide (Subdividir)

Agrega un plano y subdivídelo al menos cinco veces con el menú **Specials** **WKEY>>Subdivide** (*Herramienta Subdivide (Subdividir)*). Ahora añade un material y asignale una textura Clouds. Ajusta el parametro *NoiseSize*: a 0.500. Elige blanco como color para el material y negro como el color de la textura, para dar suficiente contraste para el comando Ruido.



Botón Ruido en Botones de Edición

Verifica que estés en Modo Edición y que todos los vertices estén seleccionados, luego cambia al Contexto Edición **F9**. Pulsa el botón *Noise* en el Panel *Mesh Tools* (*Botón Ruido en Botones de Edición*) varias veces hasta que el paisaje se vea bien. *El proceso de aplicación de Ruido* muestra el plano original - texturizado - y también lo que ocurre cuando pulsas *Noise*. Ahora borra la textura del paisaje porque arruinará el aspecto. Agrega algunos luces, algo de agua, activa el suavizado y el Subsurf en el terreno. (*Paisaje generado con Ruido*).

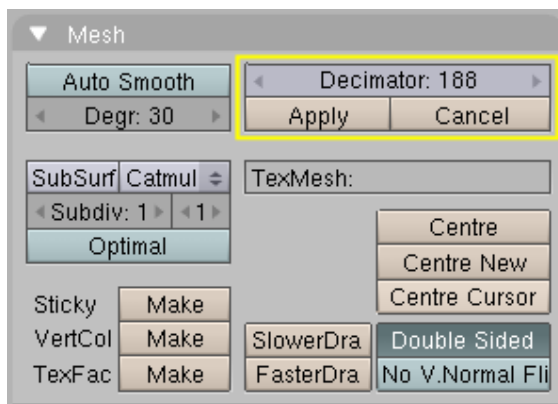
Figure 7-41. El proceso de aplicación de Ruido

Figure 7-42. Paisaje generado con Ruido

Nota: El desplazamiento de ruido, siempre tiene lugar a lo largo de la coordenada z de la malla, que coincide con la orientación del eje z del objeto en referencia local.

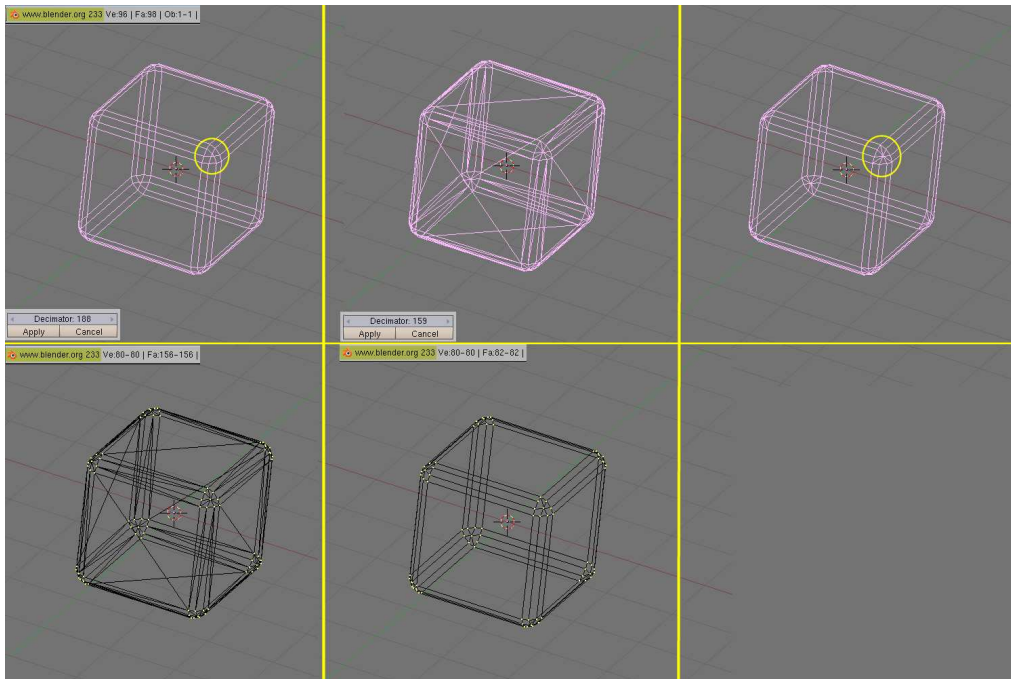
Herramienta Diezmador (Decimator)

La herramienta Diezmador, es una característica a menudo pasada por alto que permite reducir la cantidad de caras/vertices de una malla con un mínimo cambio de forma. Esto no es aplicable a mallas que han sido creadas modelando cuidadosa y económicamente, donde todos los vertices y caras son necesarios para definir correctamente la forma, pero si la malla es el resultado de un modelado complejo, con edición proporcional, reinamientos sucesivos, incluso alguna conversión de malla Subsurf a malla sin Subsurf, podría muy bien terminar con una malla con montones de vertices que no sean realmente necesarios. Un ejemplo simple es un plano, y un objeto Cuadrícula 4x4 sin deformar. Ambos renders son exactamente iguales, pero el plano tiene una cara y cuatro vertices, mientras que la cuadrícula tiene nueve caras y 16 vertices, es decir montones de vertices y caras innecesarios. La Herramienta Diezmador (*Botones del Diezmador*) permite eliminar estas caras innecesarias. Su deslizador numérico informa la cantidad de caras de la malla seleccionada en Modo Objeto. El diezmador solo manipula triangulos, por lo que cada cara cuadrilátera es dividida implícitamente en dos triangulos para ser "diezmada".



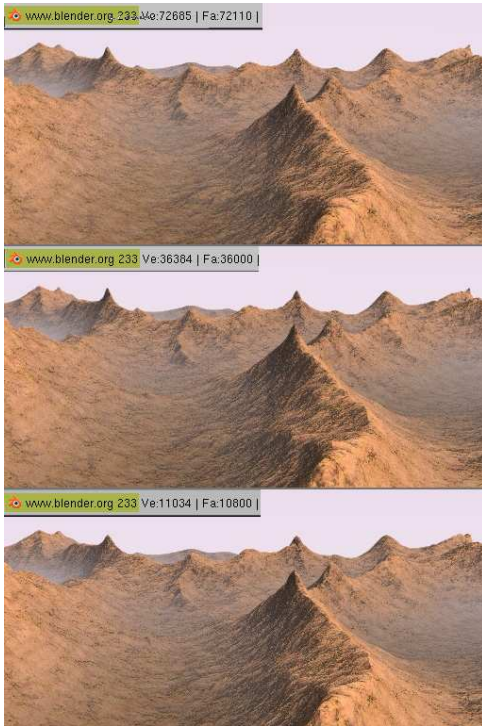
Botones del Diezmador.

Consideremos el ejemplo utilizado en la sección Bisel. Como quizás observe, hay una pequeña cara triangular sobre cada vertice del cubo que muy probablemente sea innecesaria (*El Diezmador "diezmando"*, arriba a la izquierda). La cabecera dice que el cubo tiene 98 caras y 96 vertices. El botón Diezmador dice que el cubo tiene 188 caras triangulares, es decir 90 cuadrilateros (que serían 180 triangulos) y 8 triangulos.



El Diezmador "diezmando".

Cambiando el número en el botón numérico del diezmador, pulsando con el ratón o bien tipeando en él, la malla cambia inmediatamente a "sólo triángulos". A medida que el número disminuye, las caras van desapareciendo una tras otra. Blender hace que las caras coplanares y los vertices sobre aristas alineadas desaparezcan primero. Esto tiende a preservar la forma de la malla. A medida que más y más caras son eliminadas, menos vertices coplanares y colineales son soldados, por lo tanto podría ocurrir un cambio de forma apreciable (*El Diezmador "diezmando"*., arriba al centro). En este caso particular, si solo deseamos que el triangulo central de cada vertice del cubo desaparezca, esperamos que la malla final sera de $2 \times 6 = 12$ caras por cada cara del cubo, $2 \times 3 \times 12 = 72$ caras por cara arista biselada, y $9 \times 8 = 72$ caras por cada vertice biselado, totalizando 156 caras. Es muy raro saber de antemano cuantas caras puede tener la malla final, comúnmente debe verificar cuidadosamente la malla en la Ventana3D para constatar que la forma aún es aceptable. Los dos botones inferiores del Diezmador, finalizan o cancelan el diezmado. Una vez terminado, los triangulos ya no se mostrarán más (*El Diezmador "diezmando"*., arriba a la derecha) pero la malla no obstante está hecha solamente de triangulos (*El Diezmador "diezmando"*., abajo a la izquierda). Puede volverla a cuadriláteros si así lo desea seleccionando todos los vertices y pulsando **ALT-J** (*El Diezmador "diezmando"*., abajo al centro). De esta manera redujimos la cuenta de vertices a 80 y la de caras a 82, sin ninguna pérdida apreciable en la forma. Podrá parecer una ganancia exigua, pero si este cubo va a ser duplicado con "Dupliverts" en un sistema de partículas con 1000 partículas, seguramente habrá valido la pena.



Paisaje diezmado, arriba: original; medio: ligeramente diezmado; abajo: fuertemente diezmado.

El *Paisaje diezmado*, arriba: original; medio: ligeramente diezmado; abajo: fuertemente diezmado. muestra un paisaje generado por medio de la cuidadosa aplicación de la técnica de Ruido descrita anteriormente, sobre una cuadrícula bastante extensa. En la parte superior, el resultado de la malla original y debajo dos niveles de diezmado distintos. Para el ojo, la diferencia es efectivamente apenas apreciable, pero la reducción en la cuenta de vértices aquí representa una gran ganancia.

Las curvas y las superficies son objetos como las mallas, pero se diferencian en el punto de que son expresadas en terminos de funciones matemáticas, como una serie de puntos. Blender implementa curvas y superficies Bezier y Non uniform Rational B-Splines (NURBS). Ambas, aunque siguen diferentes leyes matemáticas, son definidas en términos de un conjunto de "vértices de control" que definen un "polígono de control". La manera en que las curvas y las superficies son interpoladas (Bezier) o atraídas (NURBS) por los puntos de control pueden parecer similares, en primera instancia, a la subdivisión de superficies Catmull-Clark. Cuando se las compara con las mallas, las curvas y las superficies tienen ventajas y desventajas. Como las curvas están definidas por una menor cantidad de datos, producen buenos resultados usando menos memoria durante el modelado, mientras que la demanda de memoria se incrementa cuando renderizamos. Algunas técnicas de modelado, tales como la extrusión de un perfil "a lo largo de un camino", son solo posibles con las curvas. Pero por otro lado el gran control disponible a través de los vértices en una malla, no es posible con las curvas. Hay veces que es más apropiado utilizar las curvas y superficies en lugar de mallas y otras veces es al contrario. Si ha leído el capítulo previo y lee este será capaz de elegir cuando usar mallas y cuando usar curvas.

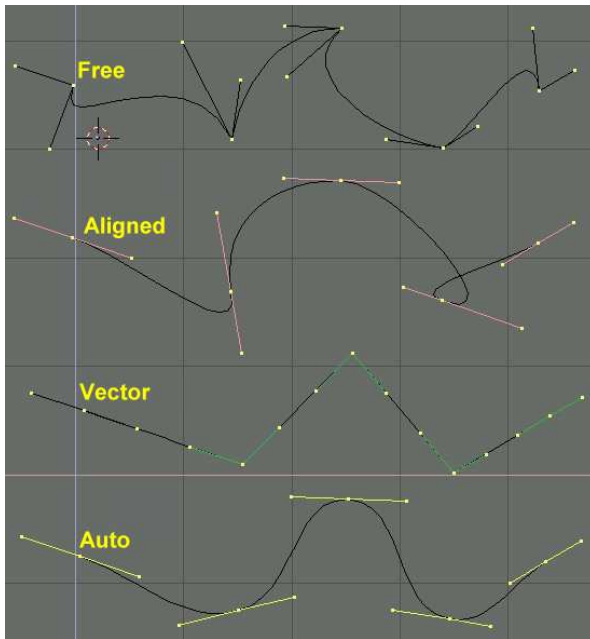
Curvas

Esta sección describe las curvas Bèizer y las curvas NURBS, mostrando de ambas un ejemplo de como funcionan.

Béziars

Las curvas Bézier son posiblemente las más usadas para el diseño de letras o de logotipos. Tambien son usadas frecuentemente en animación, como caminos para el movimiento de los objetos o como curvas IPO para variar las propiedades de los objetos en función del tiempo. Un punto de control (vértice) de una curva Bézier consta de un punto y dos agarraderas. El punto, en el medio, es usado para mover el punto de control en su totalidad; Al seleccionar dicho punto también seleccionamos las dos agarraderas, y le permitirá mover el vértice completamente. Seleccionando y moviendo una o las dos agarraderas podrá variar la forma de la curva. Una curva Bézier es tangente al segmento de línea que va desde el punto a la agarradera. La suavidad (steepness) de la curva es controlada con la longitud de las agarraderas. Hay cuatro tipos de agarraderas (*Tipos de agarraderas para curvas Bézier*):

- Free Handle (Agarradera libre) (en negro). Esto puede usarse de la manera en la que desee. Hotkey: la Tecla H (cambia entre libre y alineada);
- Aligned Handle (Agarradera alineada) (en morado). Las agarraderas siempre estan en linea recta. Hotkey: la Tecla H (cambia entre libre y alineada);
- Vector Handle (Agarradera vector) (en verde). Ambas partes de una agarradera siempre apuntan a la agarradera previa o a la siguiente. Hotkey: la Tecla V;
- Auto Handle (Agarradera automática) (en amarillo). Esta agarradera tiene automaticamente asignada por Blender una longitud y dirección para asegurar el resultado más suave. Hotkey: SHIFT-H.

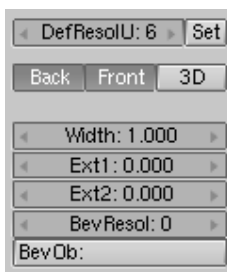


Tipos de agarraderas para curvas Bézier

Las agarraderas pueden ser *movidas*, *rotadas* y *escaladas* exactamente como se haría con un vértice en una malla. Tan pronto como las agarraderas se mueven, el tipo se modifica automáticamente:

- Las agarraderas automáticas se convierten en alineadas;
- Las agarraderas vector se convierten en libres.

Aunque la curva Bézier es un objeto matemáticamente continuo tiene que ser representado de una forma discreta desde el punto de vista del renderizado. Esto se consigue configurando una propiedad, la *resolución*, la cual define el número de puntos que son computados entre cada par de puntos de control. La *resolución* puede ser configurada separadamente para cada curva Bézier (*Configurando la resolución de la curva Bézier.*).

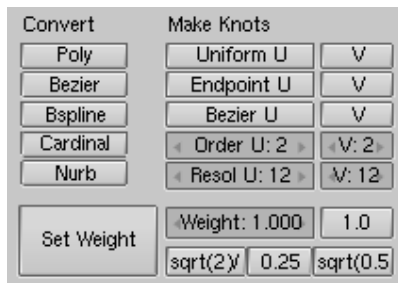


Configurando la resolución de la curva Bézier.

NURBS

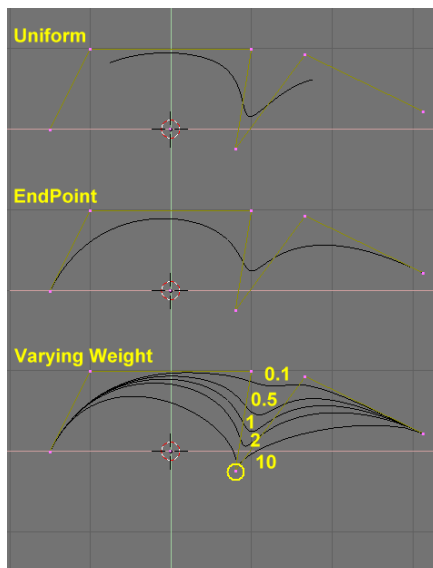
Las curvas NURBS son definidas como polinomios racionales, y son más generales, estrictamente hablando, que las curvas Bézier y las B-Splines ya que son capaces de seguir exactamente cualquier contorno. Por ejemplo un círculo Bézier es una *aproximación* polinomial de un círculo, y esta aproximación es evidente, mientras que un círculo NURBS es

exactamente un círculo. Las curvas NURBS tienen un gran conjunto de variables, que le permitirán crear formas matemáticamente puras (*Botones de control de las curvas Nurbs.*). Por otro lado, trabajar con ellas requiere un poco más de teoría:



Botones de control de las curvas Nurbs.

- **Knots (Nudos).** Las curvas Nurbs tienen un vector nudo, es decir, una fila de números que especifica la definición paramétrica de la curva. Dos parámetros son importantes para esto. *Uniform* (Uniforme) produce una división uniforme en las curvas cerradas, pero cuando es usado con curvas abiertas se obtiene finales de curva aleatorios, que son bastante difíciles de predecir donde estarán. *Endpoint* (Punto final) configura el nudo de tal manera que el primero y el último vértice son siempre parte de la curva, lo que hace mucho más fácil situarla;
- **Order (Orden).** El *orden* es el grado de la ecuación de la curva. Orden '1' es un punto, orden '2' es lineal, orden '3' es cuadrática, y así. Siempre use un orden 5 para curvas "camino" ya que se mostrará bastante fluido bajo todas las circunstancias, sin producir irritantes discontinuidades en el movimiento. Matemáticamente hablando esto es el orden del numerador y denominador del polinomio que define la NURBS;
- **Weight (Peso).** Las curvas Nurbs tienen un 'peso' en cada vértice - el grado en el que un vértice participa en la tensión de la curva.



Configuración de las Nurbs para controlar el polígono y los pesos.

La *Configuración de las Nurbs para controlar el polígono y los pesos*. muestra la configuración del vector Nudo así como el efecto de variar el peso de un simple nudo. Como con las curvas Béziérs, la resolución puede ser configurada independientemente en cada curva.

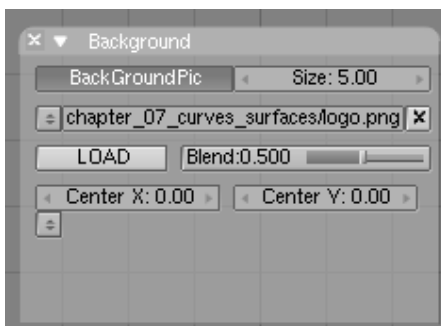
Un ejemplo funcional

Las herramientas de Blender para las curvas proporcionan una rápida y sencilla manera de construir asombrosos logotipos y textos extruidos. Usaremos esas herramientas para transformar un dibujo del logotipo en dos dimensiones en un modelo en 3D. *El logotipo* muestra el diseño del logotipo que estamos construyendo.



El logotipo

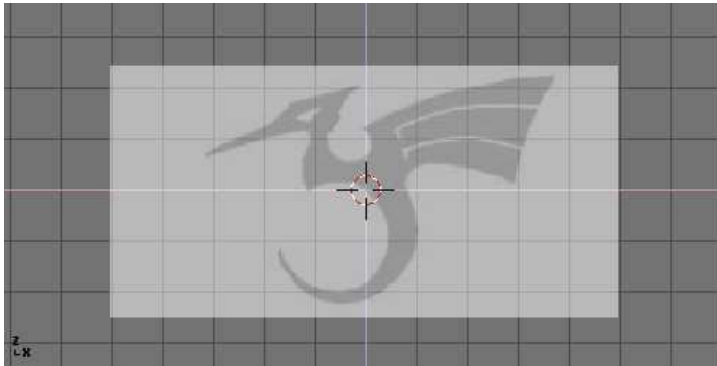
En primer lugar, importaremos nuestro dibujo original para poder utilizarlo como guía. Blender soporta los formatos de imagen TGA, PNG, and JPG entre otros. Para cargar la imagen, seleccionaremos la entrada de menú *View>>Background Image...* de la ventana 3D que esté usando. Una ventana transparente aparecerá, permitiéndole seleccionar una imagen para usarla como fondo. Active el botón *BackGroundPic* (Imagen de fondo) y use el botón *LOAD* (CARGAR) para localizar la imagen que quiere usar como guía (*Configuración de la ventana 3D*). Puede seleccionar el grado de transparencia de la imagen de fondo con el deslizador *Blend*.



Configuración de la ventana 3D.

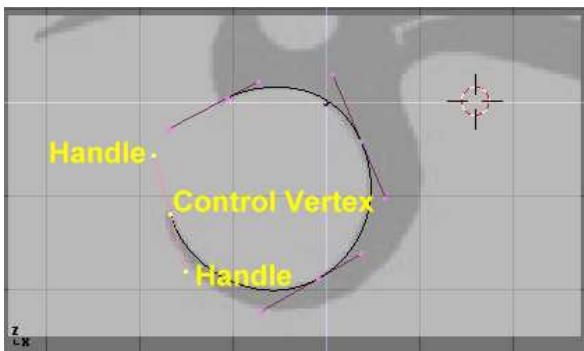
Haga desaparecer el panel emergente con la **Tecla ESC** o presionando el botón *X* en la cabecera del panel (*Dibujo del logo cargado de fondo*). Puede ocultar la imagen de fondo

cuando haya terminado volviendo a sacar el panel y deseleccionando el botón *BackGroundPic*.



Dibujo del logo cargado de fondo

Añada una nueva curva con **SPACE**>>Curve>>Bezier Curve. Un segmento de curva aparecerá y Blender entrará en modo edición. Moveremos y añadiremos puntos para crear una forma cerrada que describa el logotipo que está intentando trazar. Puede añadir puntos a la curva seleccionando uno de los dos puntos finales de curva, y entonces manteniendo pulsado **CTRL** y pinchando en **BIR**. Note que el nuevo punto estará conectado al punto seleccionado anteriormente. Una vez que el punto haya sido añadido, puede moverse seleccionando un vértice de control y presionando la tecla **G**. Puede cambiar el ángulo de la curva arrastrando y moviendo las agarraderas asociadas a cada vértice (*Agarraderas Bézier*).



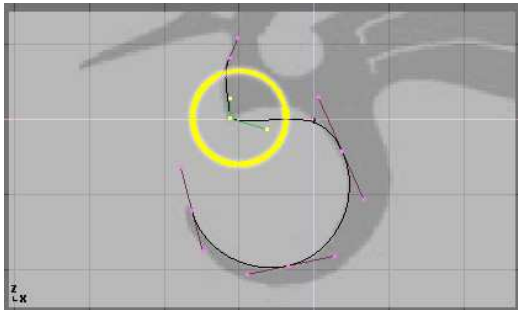
Agarraderas Bézier.

Puede añadir un nuevo punto entre dos puntos existentes seleccionando ambos y presionando la Tecla **W**>>Subdivide (*Añadiendo un punto de control*).

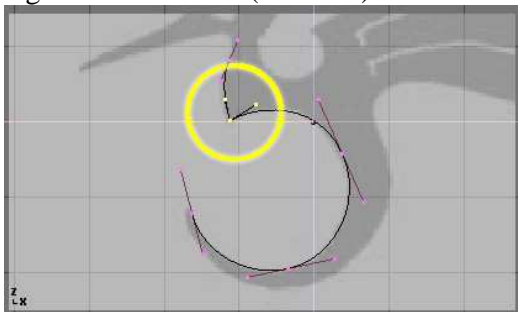


Añadiendo un punto de control.

Los puntos pueden ser eliminados seleccionandolos y presionando la Tecla X>>Selected. Para cortar una curva en dos, seleccione dos vértices de control adyacentes y presione la Tecla X>>Segment. Para hacer las esquinas afiladas, seleccione un vértice de control y presione la Tecla V. Notará que el color de las agarraderas cambia de morado a verde (*Agarraderas Vector (en verde).*). En este punto, puede mover las agarraderas para ajustar la forma en la que la curva entra y sale del vértice de control (*Agarraderas libres (en negro).*).

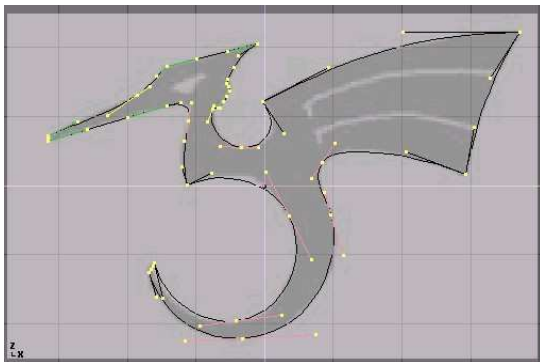


Agarraderas Vector (en verde).



Agarraderas libres (en negro).

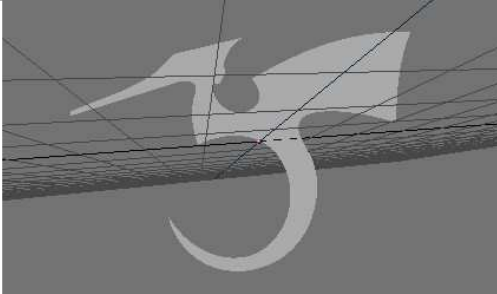
Para cerrar la curva , seleccione al menos uno de los vértices de control de la curva y presione la Tecla C. Esto conectará el último punto de la curva con el primero (*El final de línea acabado.*). Puede ser necesario manipular las agarraderas para obtener la forma que desee.



El final de línea acabado.

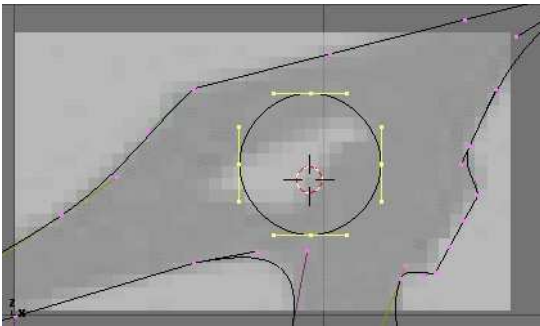
Saliendo del modo Edición con TAB y entrando en el modo sombreado con la Tecla Z debería mostrar que la curva se renderiza como una forma sólida (*Logotipo sombreado.*). Nosotros queremos poner algunos agujeros dentro para representar los ojos y las alas del dragon.

Superficies y agujeros: Cuando trabajamos con curvas, Blender automáticamente detecta agujeros en la superficie y los maneja de acuerdo a las siguientes reglas. Una curva cerrada siempre es considerada el límite de una superficie y será renderizado como una superficie sólida. Si una curva cerrada es incluida completamente dentro de otra, la curva interior se sustrae de la exterior, creando efectivamente un agujero.



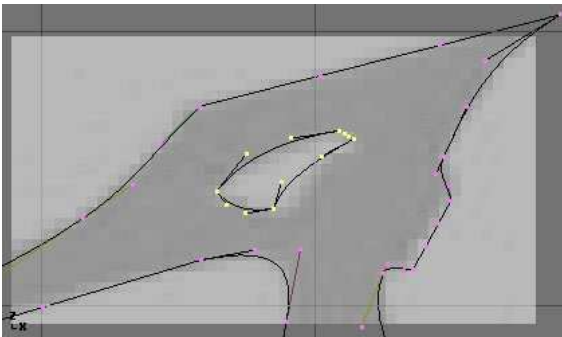
Logotipo sombreado.

Regrese al modo de alambre con la Tecla **Z** y entre de nuevo al modo Edición con **TAB**. Mientras permanezca en el modo Edición, añada una curva 'círculo' con **SPACE>>Curve>>Bezier Circle** (*Añadiendo un círculo.*). Escale negativamente hasta un tamaño adecuado con la tecla **S** y muevalo con la tecla **G**.



Añadiendo un círculo.

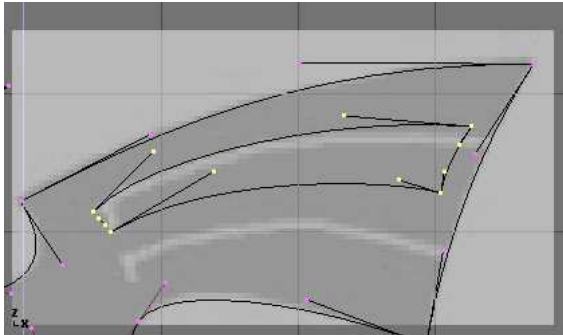
Dé forma al círculo usando las técnicas que hemos aprendido (*Definiendo un ojo.*). Recuerda añadir los vértices al círculo con la Tecla **W>>Subdivide**.



Definiendo un ojo.

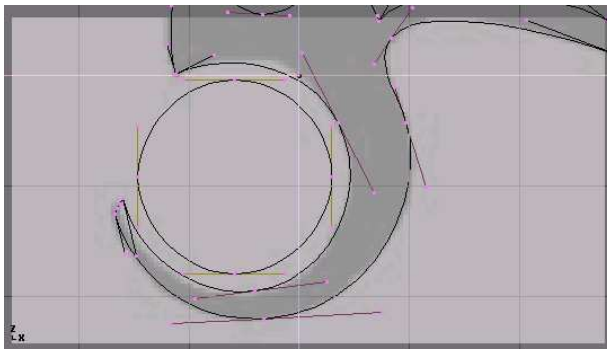
Cree un ala añadiendo un círculo Bézier, convirtiendo todos los puntos a esquinas afiladas, y ajustandolos tanto como sea necesario. Puede duplicar esta figura para crear la segunda ala y

de esta forma salvar algo de tiempo. Para hacer algo así, asegurese que ningún punto esté seleccionado, entonces mueva el cursor sobre uno de los vértices en la primera ala y seleccione los puntos que estén enlazados con la **Tecla L** (*Definiendo las alas.*). Duplique la selección con **SHIFT-D** y mueva los nuevos puntos a su posición.



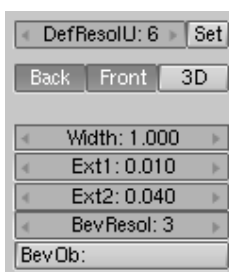
Definiendo las alas.

Para añadir más geometría que no esté conectada con el cuerpo principal (situando un círculo en la cola curvada del dragón, por ejemplo), use la combinación **SHIFT-A** para añadir más curvas como se muestra en *Colocación del círculo en la cola..*



Colocación del círculo en la cola.

Ahora que tenemos la curva, necesitamos configurar su biselado (bevel) y su espesor. Con la curva seleccionada, vaya a los botones de Edición (F9) y localice el panel *Curves and Surface* (curvas y superficies). El parámetro *Ext1* configura el espesor de la extrusión mientras que el parámetro *Ext2* configura el tamaño del biselado. *BevResol* configura como de afilado o curvado será el biselado. *Configuración del Biselado* muestra la configuración utilizada para extruir esta curva.



Configuración del Biselado

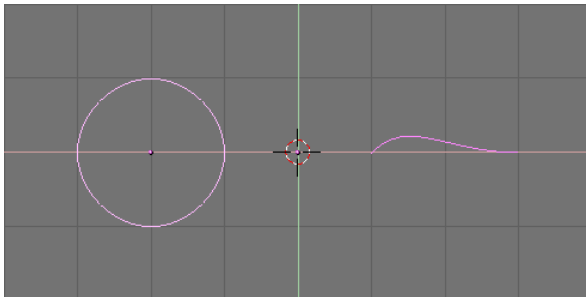
Cuando su logotipo esté completo, puede añadir materiales y luces y hacer un bonito render (*Renderizado final.*).



Renderizado final.

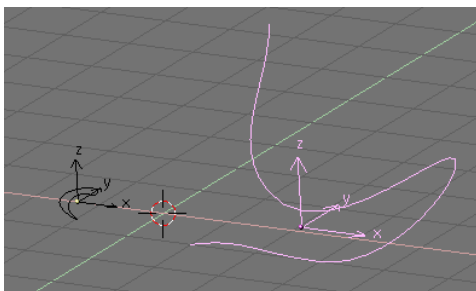
Extruir a lo largo de una trayectoria

La técnica de extrusión a lo largo de una trayectoria (*Extrude along path*) es una poderosa herramienta de modelado. Consiste en crear una superficie moviendo un perfil dado a lo largo de una trayectoria determinada. Tanto el perfil como la trayectoria pueden ser una curva Bézier o una curva NURBS. Asumiremos que ha añadido una curva Bézier y un círculo Bézier como objetos separados a su escena (*Perfil (a la izquierda) y trayectoria (a la derecha).*).

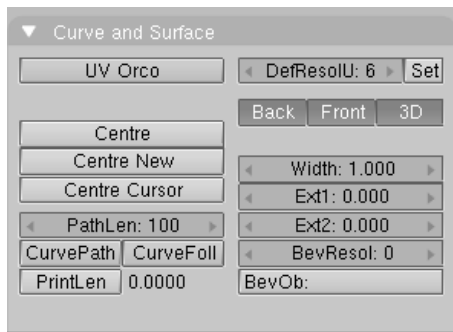


Perfil (a la izquierda) y trayectoria (a la derecha).

Juegue un poco con ambos para obtener un bonito perfil alado y una sugerente trayectoria (*Perfil modificado (a la izquierda) y trayectoria (a la derecha).*). Por defecto, las curvas Bézier solo existen en un plano y son objetos de dos dimensiones. Para conseguir que la trayectoria abarque las tres dimensiones, como en el ejemplo, presione el botón *3D* en los botones de Edición de curva (F9) en el panel *Curve and Surface* (*Botón 3D.*).

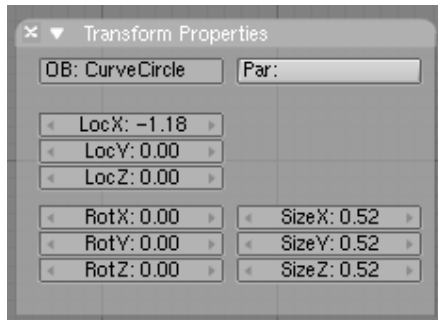


Perfil modificado (a la izquierda) y trayectoria (a la derecha).



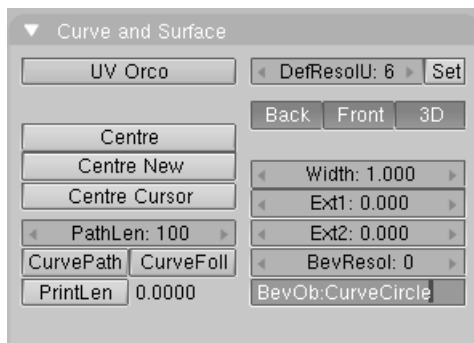
Botón 3D.

Ahora mire el nombre del objeto "perfil". Por defecto este nombre será "CurveCircle" y es mostrado en el panel emergente que aparece cuando se presiona la Tecla **N** cuando está seleccionado. Puede cambiar dicho nombre con **SHIFT-LMB** si así lo desea (*Nombre del perfil*).



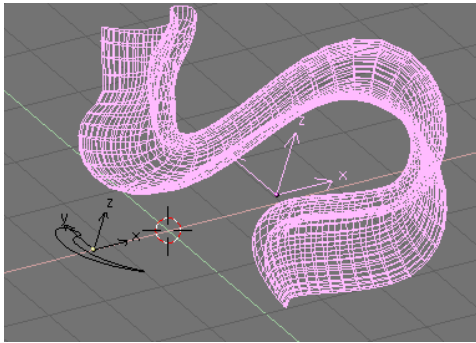
Nombre del perfil.

Ahora seleccione la trayectoria. En los botones de Edición localice el campo de texto *BevOb:* en el panel *Curve and Surface* y escriba allí el nombre del perfil. En nuestro caso "CurveCircle" (*Especifique el perfil que se utilizará en la trayectoria*).



Especifique el perfil que se utilizará en la trayectoria.

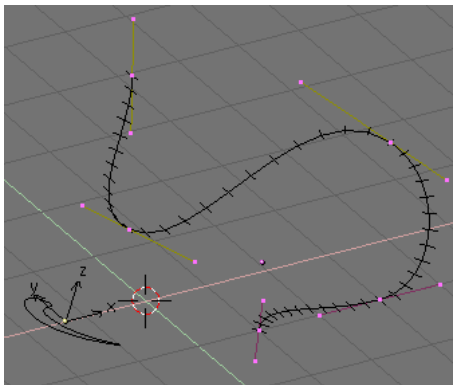
El resultado es una superficie definida por el perfil, que cubre todo el camino (*Resultado de la extrusión*).



Resultado de la extrusión.

Para comprender los resultados, y obtener los efectos deseados es importante comprender los siguientes puntos:

- El perfil esta orientado de tal manera que su eje z es tangente (en este caso dirigido hacia delante) a la trayectoria y su eje x está en el plano de la trayectoria; consecuentemente el eje y es ortogonal al plano de la trayectoria;
- Si la trayectoria es tridimensional el "plano de la trayectoria" esta definido localmente en vez de globalmente y es mostrado, en modo Edición, por varios segmentos cortos perpendiculares a la trayectoria (*Plano local de la trayectoria.*);
- El eje y del perfil siempre apunta hacia arriba. Esto es a menudo una fuente de resultados inesperados y de problemas, como explicaremos más tarde.

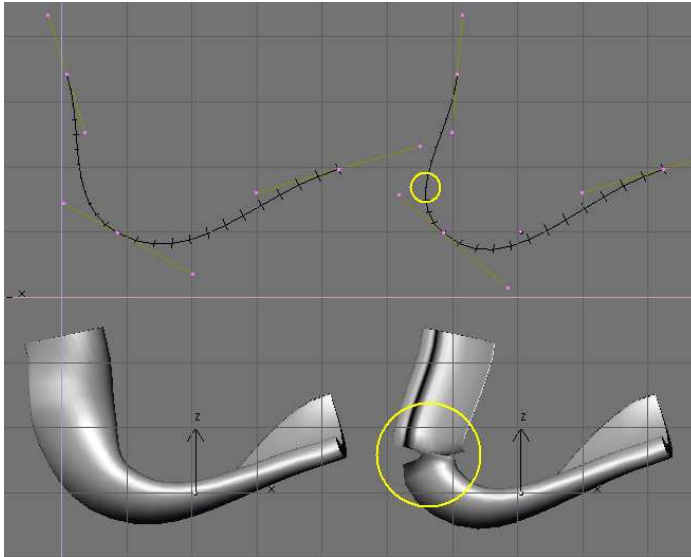


Plano local de la trayectoria.

{{Tip|Inclinación (Tilting)| Para modificar la orientación del plano local de la trayectoria seleccione un punto de control y presione la Tecla T. Entonces mueva el ratón para cambiar la orientación de los segmentos suavemente en las proximidades del punto de control. LMB fija la posición, y ESC vuelve al estado anterior. }}

Con el eje y obligado a apuntar hacia arriba, ocurrirán resultados inesperados cuando la trayectoria es tridimensional y el perfil que está siendo extruido llega a un punto donde la trayectoria es exactamente vertical. De hecho si la trayectoria es vertical y entonces continua doblandose hay un punto donde el eje y del perfil comenzaría a apuntar hacia abajo. Si esto ocurre, como el eje y está forzado a apuntar hacia arriba hay una rotación muy brusca, 180 grados, del perfil, para que de nuevo el eje y apunte hacia arriba. *Problemas con la extrusión*

debido a la restricción del eje y. muestra el problema. A la izquierda hay una trayectoria cuya inclinación es tal que la normal al plano local de la trayectoria está siempre apuntando hacia arriba. A la derecha vemos una trayectoria donde, en el punto rodeado de amarillo, la normal comienza a apuntar hacia abajo. el resultado de la extrusión presenta un giro brusco allí.



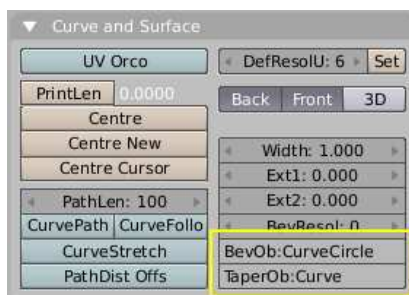
Problemas con la extrusión debido a la restricción del eje y.

Las únicas soluciones a este problema son: Usar multiples trayectorias o cuidadosamente colocar la trayectoria de tal forma que la normal siempre apunte hacia arriba.

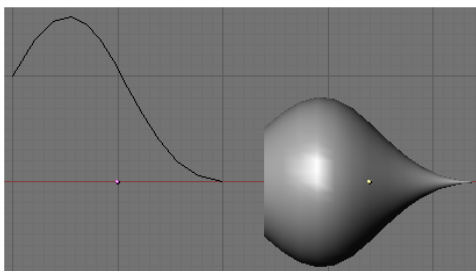
Cambiando la orientación del perfil: Si la orientación del perfil a lo largo de la curva no es como esperabas, y quieres rotarla para la longitud total de la trayectoria, hay una forma mejor que inclinar todos los puntos de control de la trayectoria. Simplemente puede rotar el perfil en modo Edición en su plano. De esta manera el perfil cambiará pero su referencia local no.

Curva Taper

Taper es una herramienta para curvar objetos. En el panel de Edición (F9) tiene un campo *TaperOb* donde tiene que poner el nombre de la curva que definirá la anchura del objeto curvado.

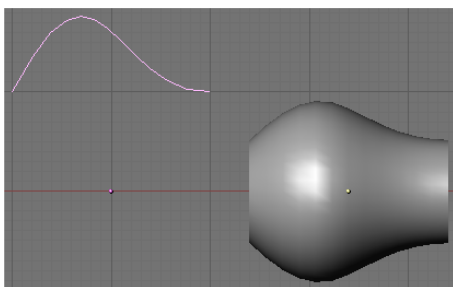


Panel de curvas y superficies



Ejemplo 1 de Taper

En *Ejemplo 1 de Taper* puede claramente ver el efecto que la curva taper de la izquierda tiene sobre el objeto curvado de la derecha. Aquí la curva taper de la izquierda está más cercana al centro del objeto lo que produce una curvatura más pequeña en el objeto de la derecha.

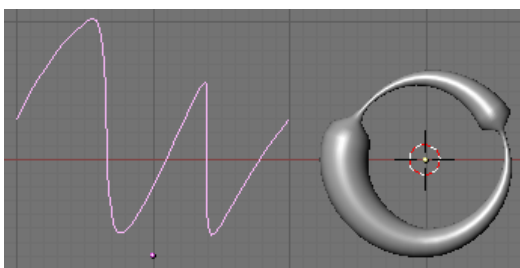


Ejemplo 2 de Taper.

En *Ejemplo 2 de Taper* un punto de control en la curva taper de la izquierda se saca del centro lo que da un resultado más ancho al objeto curvado de la derecha.

Nota: El objeto curvado es extruido con un círculo. (Vea *chapter_extrude_along_path* para más sobre extrusión de curvas).

En *Ejemplo 3 de Taper*, vemos el uso de una curva taper irregular añadida a un círculo.

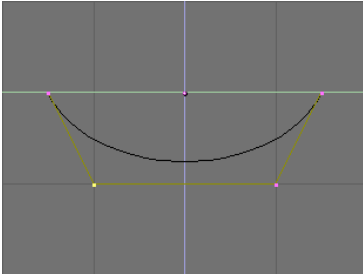


Ejemplo 3 de Taper

Skinning

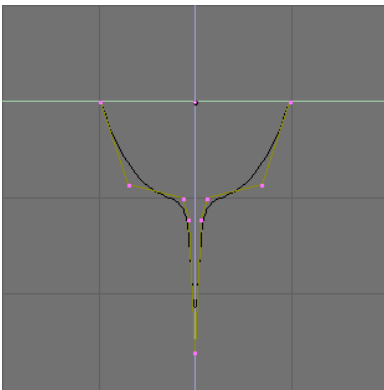
Skinning es el arte de definir una superficie usando 2 ó más perfiles. En Blender puede hacer eso preparando algunas curvas con la forma de la superficie deseada para después convertirlas a una única superficie NURBS. Como ejemplo crearemos un bote de pesca. La primera cosa que haremos, en la vista lateral (NUM3), es añadir una "Surface Curve". Asegurate de que

realmente sea una curva Surface y no una curva Beizer o NURBS, o el truco no funcionará (Una "curva superficie" para realizar skinning.).



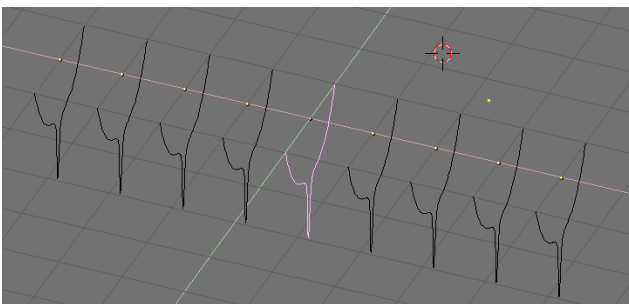
Una "curva superficie" para realizar skinning.

Dé a la curva la forma de la sección transversal de la mitad del barco, añadiendo tantos vértices como sean necesarios con el botón "Split" (partir) y, posiblemente, configurando el 'Endpoint' (punto final) de la curva a 'U' y/o a 'V'(Perfil del barco.).

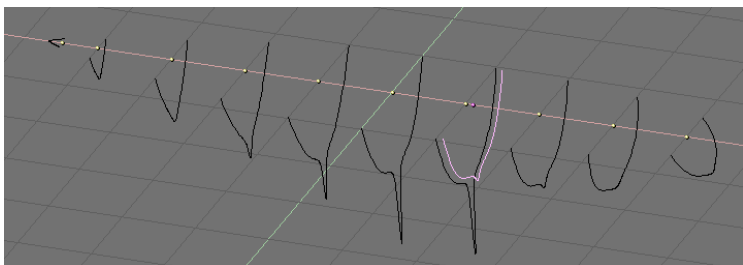


Perfil del barco.

Ahora duplique (SHIFT-D) la curva, tantas veces como sea necesario, hacia la izquierda y hacia la derecha (*Varios perfiles a lo largo del eje del barco.*). Ajuste las curvas para que coincidan con las diferentes secciones transversales que tiene el barco a lo largo de su eslora. Para llevar esto acabo, unos blueprints (planos técnicos con el perfil) ayudarán bastante. Puede cargar un blueprint de fondo (como hicimos para el diseño del logo en este mismo capítulo) para ajustar los perfiles de las secciones transversales (*Varios perfiles de la forma correcta.*). Note que la superficie que hemos creado pasa de un perfil a otro suavemente. Para crear cambios bruscos necesitará situar los perfiles muy cerca los unos a los otros, como en el caso del perfil seleccionado en *Varios perfiles de la forma correcta.*.

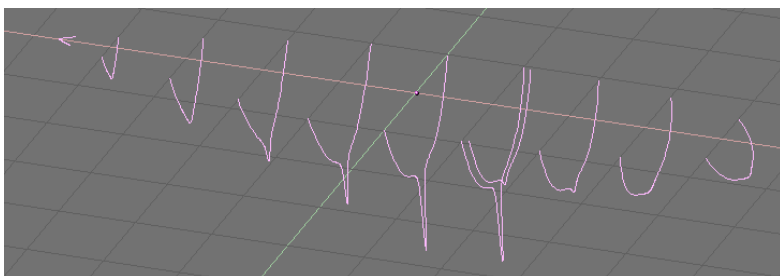


Varios perfiles a lo largo del eje del barco.



Varios perfiles de la forma correcta.

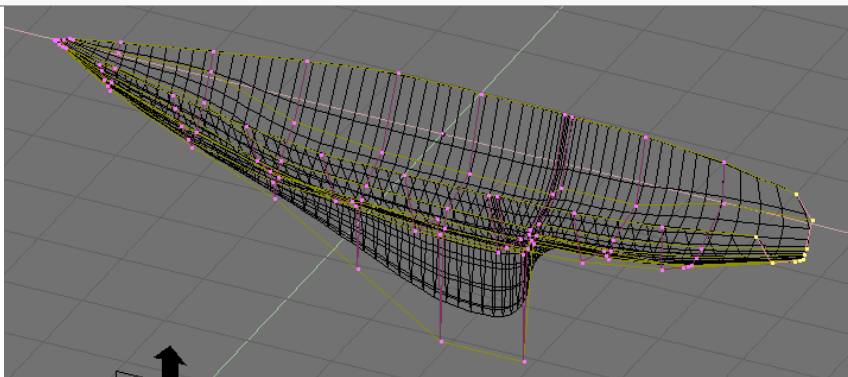
Ahora seleccione todas las curvas (con la Tecla A o la Tecla B), y unalas presionando CTRL-J y contestando 'Yes' a la cuestión 'Join selected NURBS?' (¿Unir NURBS seleccionados?). Se resaltarán todos los perfiles *Perfiles unidos*.



Perfiles unidos.

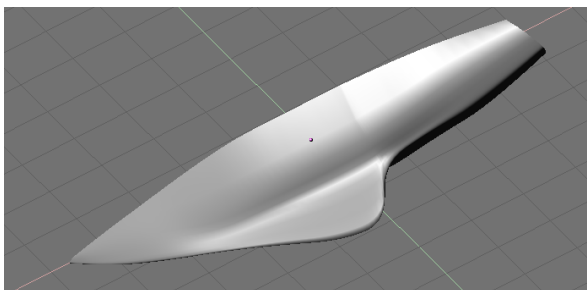
Ahora entre en el modo edición (TAB) y seleccione todos los puntos de control con la Tecla A; entonces presione la tecla F. Los perfiles se convertirán en una superficie (*Superficie en modo edición*).

Nota: Como debería ser evidente desde el primer al último perfil en este ejemplo, las secciones transversales no necesitan ser definidas en un grupo de planos ortogonales.



Superficie en modo edición.

Modifique la superficie, si fuese necesario, moviendo los puntos de control. *El casco finalizado*. muestra una vista en modo sombreado. Probablemente necesitará incrementar *ResolU* y *RelolV* para obtener una forma más óptima.



El casco finalizado.

Configuración de los perfiles: La única limitación de esta poderosa técnica es que todos los perfiles tienen que tener el mismo número de puntos de control. Esto es por lo que es bastante buena idea modelar primero la sección transversal más compleja y después duplicarla, moviendo los puntos de control como sea necesario, sin añadir o quitar dichos puntos, como se muestra en el ejemplo.

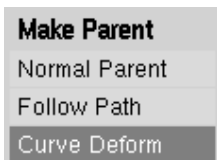
Curva Deformar

La *Curva Deformar* proporciona un método sencillo pero eficiente de definir una deformación en una malla. Emparentando un objeto malla a una curva, puede deformar la malla estirando o encogiéndola a lo largo, u ortogonalmente, al eje dominante. La *Curva Deformar* funciona en un eje dominante, X, Y, o Z. Esto significa que cuando mueva su malla en la dirección dominante, la malla seguirá el movimiento de la curva. Moviéndola en una dirección ortogonal moverá la malla más cerca o más lejos de la curva. La configuración por defecto de Blender hace que el eje dominante sea el eje Y. Cuando mueva el objeto más allá del final de la curva, el objeto continuará deformándose en la dirección del vector del final de la curva.

Un consejo: Intente posicionar su objeto sobre la curva y mueva esta a su alrededor. Esto le proporcionará un mejor control sobre cómo funciona la deformación.

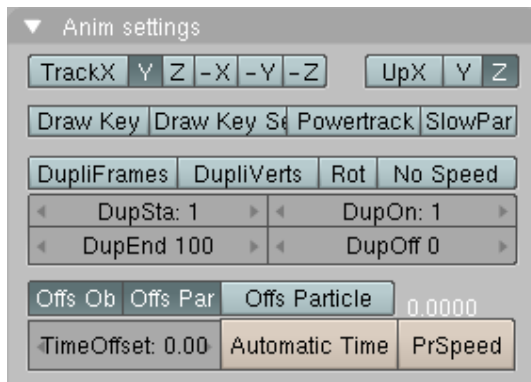
La Interfaz

Cuando emparente una malla a una curva (CTRL-P), se le mostrará un menú, *El menú que aparece cuando se emparenta..* Seleccionando **Curve Deform** activará la función *Curva Deformar* en el objeto malla.



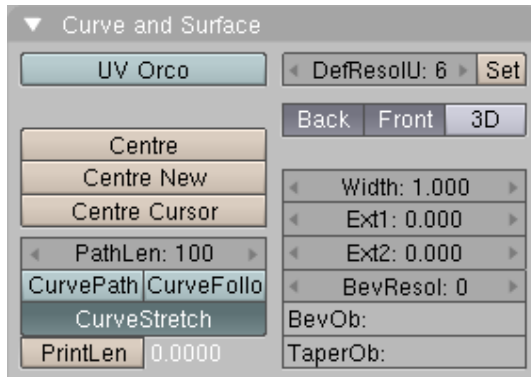
El menú que aparece cuando se emparenta.

La propiedad del eje dominante se encuentra activa en el objeto malla. Por defecto el eje dominante en Blender es Y. Esto puede cambiarse seleccionando uno de los botones **Track X**, **Y** o **Z** en el *Panel Anim*, *La configuración del Panel Anim.*, en contexto Objeto (F7).



La configuración del Panel Anim.

Las curvas cíclicas trabajan tal y como se espera en las deformaciones de un objeto a lo largo de una trayectoria. **CurveStretch** proporciona una opción que permite que la malla del objeto se extire, o se encoja, sobre la curva. Esta opción está en el contexto *Edición* (F9) de la curva. Ver *Panel de Curvas y superficies*.

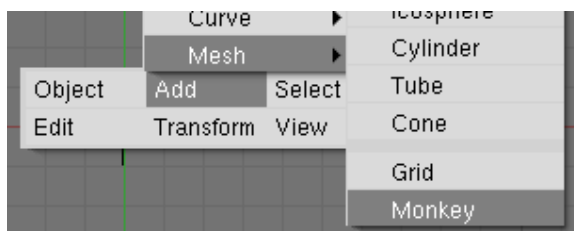


Panel de Curvas y superficies.

Ejemplo

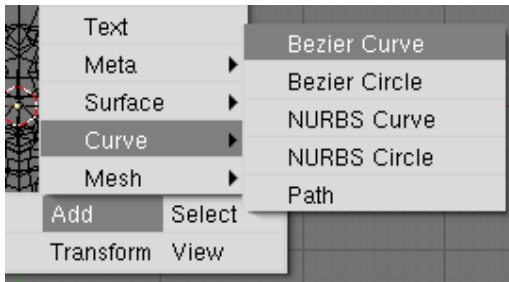
Vamos a realizar un sencillo ejemplo.

- Elimine el cubo que hay por defecto en la escena y añada un Mono (Monkey)!! (SHIFT-A -> Add -> Mesh -> Monkey, ¡Añada un mono!).



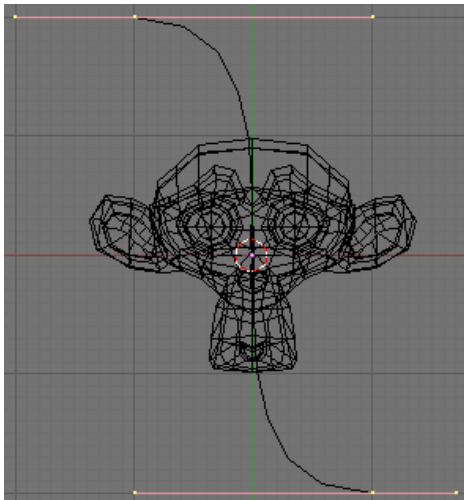
¡Añada un mono!

- Ahora pulse TAB para salir del *Modo de Edición*. A continuación añada una curva. (SHIFT-A -> Add -> Curve -> Bezier Curve, *Añada una curva.*).



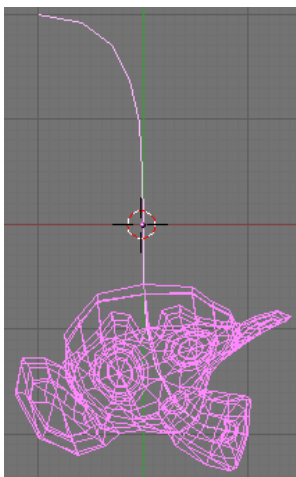
Añada una curva.

- Mientras estemos en el Modo edición, mueva los puntos de control de la curva como se muestra en *Editando la curva.*, y entonces salga del *Modo Edición*, (TAB).



Editando la curva.

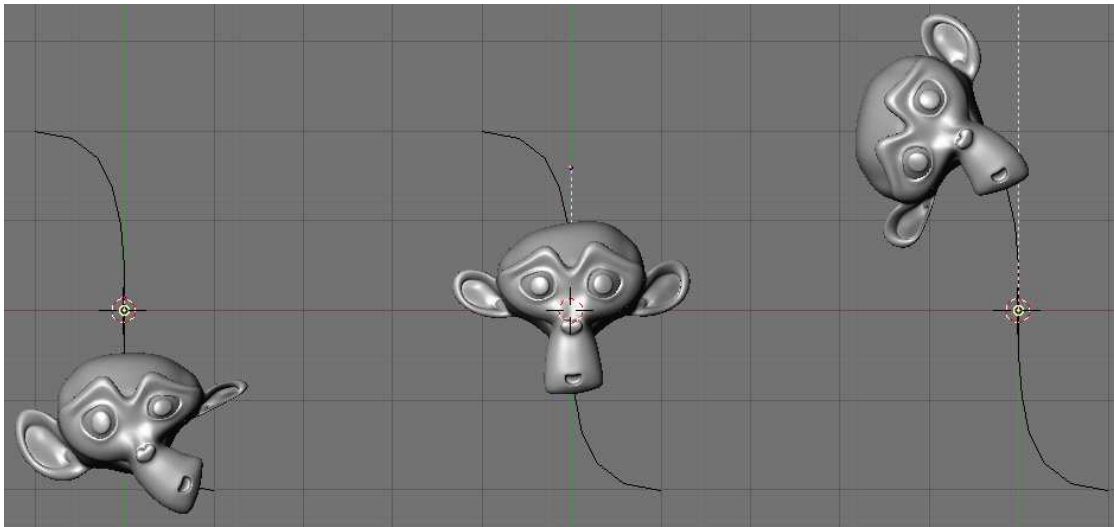
- Seleccione el Mono, (RMB), y con shift (mayúsculas) pulsado seleccione la curva, (SHIFT-RMB). Presione CTRL-P para abrir el menú *Make Parent*. Seleccione la opción **Curve Deform**. (El menú que aparece cuando se emparenta.). El mono adoptará la posición de la curva tal y como se muestra en *Mono en la curva.*.



Mono en la curva.

- Ahora si seleccionas el mono, (RMB), y lo mueves, (G), en la dirección Y, (por defecto, el eje dominante), el mono se deformará a lo largo de la curva.
- En *Deformaciones del mono.*, puedes observar el mono en diferentes posiciones a lo largo de la curva. Para obtener una visión más clara de la deformación se ha activado *SubSurf* con *Subdiv 2* y *Set Smooth* en la malla del mono. (F9 para mostrar las opciones de *Edición*).

Un consejo: Cuando se mueva el mono en cualquier dirección distinta a la del eje dominante crearás algunas deformaciones raras. Algunas veces es lo que querrás conseguir, así que necesitarás experimentar un poco para dominarlo.



Deformaciones del mono.

Superficies

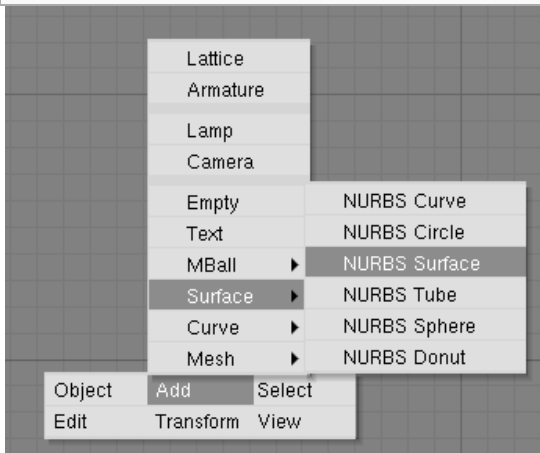
Las Superficies actualmente son una extensión de las curvas NURBS. En Blender son consideradas como un tipo ObData separado. Mientras una curva produce una interpolación unidimensional, las Superficies presentan una segunda dimensión adicional. La primera dimensión es U, como para las curvas, y la segunda es V. Una rejilla bidimensional de puntos de control define la forma de esas superficies NURBS. Utilice las Superficies para crear y modificar fluidamente las superficies curvadas. Las Superficies pueden ser cíclicas en ambas direcciones, permitiéndole fácilmente crear una forma de 'donut', y pueden ser dibujadas como 'sólidos' en Modo Edición (zbuffered, con iluminación OpenGL). Esto hace que trabajar con Superficies sea bastante sencillo.

Nota: Actualmente Blender tiene un conjunto de herramientas básicas para superficies, que permiten crear agujeros y superficies blandas. Se incrementará las funciones de esta área en futuras versiones.

Puede tomar una de las 'primitivas' de Superficie del menú ADD (Añadir) como punto de inicio (*El menú añadir superficie.*). Observe que puede elegir Curve (Curva) y Circle

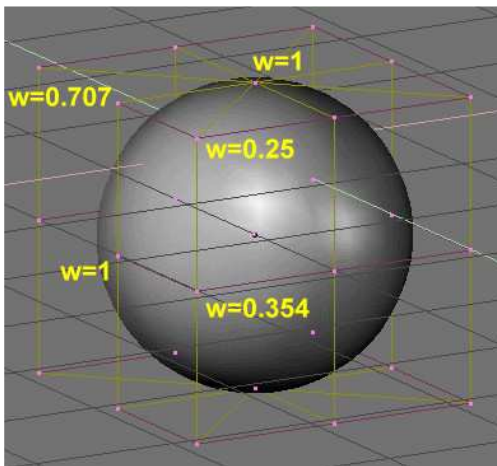
(Círculo) en el menú Superficie. Esto es posible porque las curvas NURBS son intrínsecamente superficies NURBS, simplemente tienen una dimensión restringida.

Nota: Una verdadera curva NURBS y una curva 'superficie' NURBS no son intercambiables, como se dará cuenta al realizar el proceso de extrusión en la sección 'skinning' que verá más adelante.



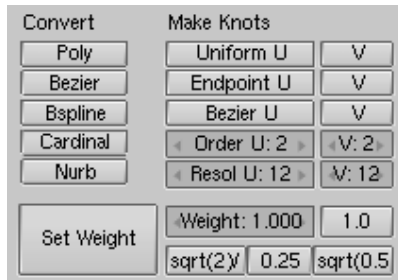
El menú añadir superficie.

Cuando añada una curva 'superficie' puede crear una verdadera superficie simplemente extruyendo la curva (Tecla E). Cada arista de la superficie puede entonces extruirse como desee para dar forma al modelo. Utilice la Tecla C para conseguir que sea ciclica la dirección U o la V. Asegúrese de configurar los nudos a *Uniform* (Uniforme) o *Endpoint* (Punto final) con uno de los botones del panel *Herramientas de curva* (Curve Tools) de los botones de Edición. Cuando trabajamos con superficies, es práctico trabajar con una columna o fila de vértices. Blender proporciona una herramienta de selección para esto: **SHIFT-R**, "Select Row (seleccionar columna)". Comenzando por el último vértice seleccionado, una completa columna de vértices se extiende seleccionada en la dirección 'U' o 'V'. Elija "seleccionar columna" de nuevo con el mismo vértice para intercambiar entre la dirección 'U' y 'V'.



Las NURBS pueden crear formas puras tales como círculos, cilindros y esferas (pero note que un círculo Bèzier no es un círculo puro). Para crear círculos puros, esferas o cilindros, debe

configurar los pesos de los vértices. Esto no es intuitivo y debería leer más sobre las NURBS antes de intentar esto. Basicamente, para producir un arco circular a partir de una curva con tres puntos de control, los puntos finales deben de tener un peso unitario, mientras el peso del punto de control central debe ser igual a la mitad del coseno de la mitad del ángulo entre los segmentos creados al ensamblar los puntos. Figure 9-21 muestra esto para una esfera. Los tres números estándar son incluidos por defecto en el panel *Curve Tools* (Herramientas de curva) en los botones de edición (*Pesos preconfigurados*).



Pesos preconfigurados

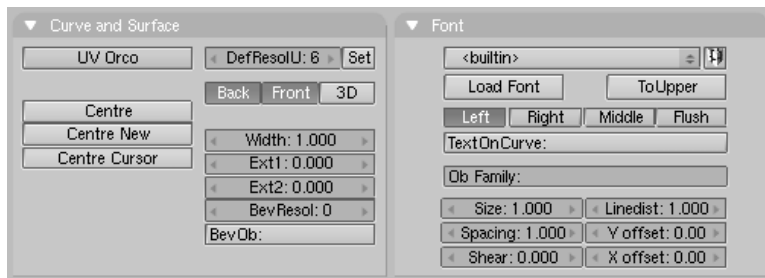
Texto

El texto es un tipo especial de curva en Blender. Blender tiene su propio tipo de fuente interna pero también puede usar fuentes externas, incluyendo fuentes PostScript Type 1 y fuentes True Type (*Ejemplos de texto.*).



Ejemplos de texto.

Abra Blender o comience una escena nueva presionando **CTRL-X**. Añada un objeto "Texto" con la caja de herramientas (**SPACE>>Add>>Text**). Puede editar el texto con el teclado en modo Edición; un cursor le mostrará su posición en el texto. Cuando deje el modo Edición con **TAB**, Blender rellenará el "texto-curva", produciendo un objeto sólido que es renderizable inmediatamente. Ahora vaya a los botones de Edición (**F9**) *Botones de edición de texto.*



Botones de edición de texto.

Como puedes ver en el panel *Font*, Blender, por defecto, utiliza su propia fuente *interna* cuando crea un objeto "texto". Ahora pinche en *Load Font* (cargar fuente). Navegue en la ventana de archivos para encontrar un directorio que contenga fuentes PostScript Type 1 o True Type y cargue una nueva fuente. (Puede descargarse varias fuentes PostScript de internet, y además Microsoft Windows incluye muchas fuentes True Type de su propiedad, aunque en este caso tenga cuidado ya que algunas de ellas ¡tienen copyright!). Pruebe con algunas fuentes. Una vez que haya cargado una, puede usar el menú de botones para cambiar la fuente de un objeto "texto". Hasta ahora lo que tenemos es un objeto plano. Para añadir algo de profundidad, podemos usar los botones *Ext1:* y *Ext2:* en el panel *Curve and Surface* tal y como hicimos con las curvas. Use la opción *TextOnCurve:* (Texto en curva) para conseguir que el texto siga una curva bidimensional. Use los botones de alineación, que se encuentran encima del campo de texto *TextOnCurve:* en el panel *Font*, para alinear el texto con la curva. Una función de Blender particularmente poderosa es la posibilidad de que un objeto "texto" sea convertido con **ALT-C** a una curva Bézier, lo cual permite le permitirá editar la forma de cada carácter en la curva. Esto es especialmente útil cuando se crean logotipos o cuando se producen letras a nuestro gusto. La transformación de texto en curva es irreversible y, por supuesto, una posterior transformación de curva en malla también es posible.

Caracteres especiales

Normalmente, un objeto fuente comienza con la palabra "Text", la cual puede ser borrada fácilmente con **SHIFT-BACKSPACE**. En modo Edición, el objeto texto se modifica al meter algo por teclado. Casi todas las hotkeys están deshabilitadas. El cursor puede ser movido con las flechas de dirección. Use **SHIFT-FLECHAIZQUIERDA** y **SHIFT-FLECHADERECHA** para mover el cursor al final de la línea o al principio o al final del texto.

Casi todos los caracteres especiales están disponibles. Este es un resumen de esos caracteres:

- **ALT-c:** Copyright
- **ALT-f:** Florin Holandés
- **ALT-g:** Grados
- **ALT-l:** Libra Británica
- **ALT-r:** Marca registrada
- **ALT-s:** S Alemana
- **ALT-x:** Símbolo de multiplicar
- **ALT-y:** Yen Japonés
- **ALT-1:** Un pequeño 1
- **ALT-2:** Un pequeño 2
- **ALT-3:** Un pequeño 3
- **ALT-?** Signo de interrogación Español

- ALT-!: Signo de exclamación Español
- ALT->: Un doble >>
- ALT-<: Un doble <<

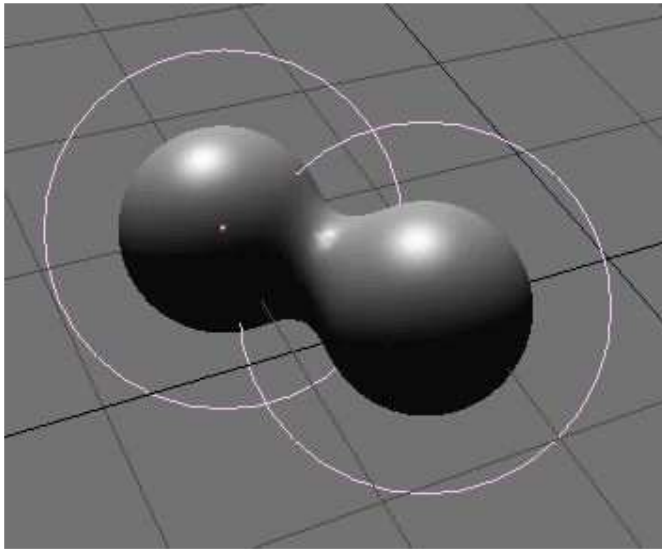
Todos los caracteres de su teclado deberían funcionar, incluyendo vocales unidas y demás. Si necesita caracteres especiales (tales como letras acentuadas, que no están en los teclados estadounidenses) puede generar muchos de ellos usando una combinación de 2 caracteres. Para hacer esto, presione ALT-BACKSPACE junto con la deseada combinación para producir un carácter especial. Algunos ejemplos son dados a continuación:

- Tecla A, ALT-BACKSPACE, TILDE: ã
- Tecla A, ALT-BACKSPACE, COMA: à
- Tecla A, ALT-BACKSPACE, ACENTO: á
- Tecla A, ALT-BACKSPACE, Tecla O: â
- Tecla E, ALT-BACKSPACE, COMILLAS: ë
- Tecla O, ALT-BACKSPACE, BARRA: ø

También puede añadir ficheros ASCII a un objeto "texto". Guarde el fichero como /tmp/.cutbuffer y presione ALT-V. Por otro lado puede escribir su texto desde una ventana de texto de Blender, cargando el texto en dicha ventana, o pegandolo desde el portapapeles y presionando ALT-M. Esto crea un nuevo objeto "texto" con el contenido del buffer (hasta 1000 caracteres)

Meta Objetos

Los Meta Objects consisten en elementos esféricos, tubulares y cubicos que pueden afectar a otra forma. Puedes solo crear redondeados y líquidos 'mercuriales', o 'arcillosos', formas que existen *procesalmente*, que son computados dinamicamente. Usa los Meta Objects para crear efectos especiales o como base para modelar. Los Meta Objects son también llamados superficies *implicitas*, para precisar que no son explícitamente definidos por vértices (aclopamientos) o puntos de control (como son las superficies). Los Meta Objects definen una *estructura dirigida* que se pueda considerar como la fuente de un campo estático. El campo puede ser positivo o negativo y por lo tanto el campo generado por las estructuras vecinas que dirigen se pueden atraer o rechazar. La superficie implícita se define como la superficie donde el campo 3D por todas las estructuras directas que asumen un valor dado. Por ejemplo un Meta Ball, es un punto que dirige la estructura, genera un campo isotrópico alrededor de él y los valores constantes de la superficie de campo son esferas centradas en el punto que dirige. Dos bolas vecinas del Meta interactúan y, si están lo suficientemente cerca, las dos superficies se combinan en una sola superficie (*Dos Metaballs*).



Dos Metaballs.

De hecho, los Meta Objects no son nada mas que formulas matematicas que realizan operaciones logicas unas con otras (Y, O), y que pueden ser añadidas o subtraídas una con otra. Este metodo es tambien llamado CSG, Geometria Solida Constructiva. Porque de esta naturaleza matematica, CSG usa una pequeña parte de memoria, pero requiere de lotes de CPU para computar. Para optimizar esto las superficies implicitas son *poligonizadas*. El área completa de CSG esta dividida en una rejilla 3D, y para cada *borde (edge)* de la rejilla se crea un cálculo, y (donde es mas importante) la formula tiene un momento crucial, una 'cima (vertex)' para que la *poligonización* sea creada. Para crear un Meta Object presiona **ESPACIO** y selecciona **Add>>MBall**. Tu puedes seleccionar las formas de la base: Esfera, Tubo, Plano, Elipsoide y Cubo. Las MetaBalls tienen un punto que dirige la estructura, los MetaTubes tiene un segmento que dirige la estructura, los MetaPlanes un plano y los MetaCubes un cubo. La estructura subyacente llega a ser evidente como bajas los valores *Wiresize* y aumentes el *Threshold* en el Panel de *Meta Ball*. Cuando estas en modo edición, tu puedes mover y escalar los Meta Objects como tu desees. Esta es la mejor manera de construir formas estaticas - en oposición a las animadas- . Los Meta Objects pueden tambien influenciar *fuera* del Modo de Edición. Cuando estas fuera del Modo de Edición tu tienes mucha más libertad; las bolas pueden rotar o moverse y consiguen cada *transformación* de los Objetos Padre. Este metodo requiere más tiempo de calculo y debido a esto es algo lento. Las siguientes reglas describen la relación entre los Meta Objects:

- Todos los Meta Objects con el mismo nombre de 'familia' (el nombre sin el numero) influencia con los otros. Por ejemplo "MBall", "MBall.001", "MBall.002", "MBall.135". Observa aquí que no estamos hablando del nombre del bloque de MetaBall ObData.
- Los objetos con el nombre de familia sin un numero determinado la base, la resolución, y la transformación del poligono. Tambien tiene el area del material y la textura que será referido como base del Meta Object.
- Solo un material puede ser usado por un sistema Meta Object . Ademas, los Meta Objects serán separados del area de la textura; esto normaliza las coordenadas de los vertices. Normalmente el area de la textura es identica a los *limites* de todos los

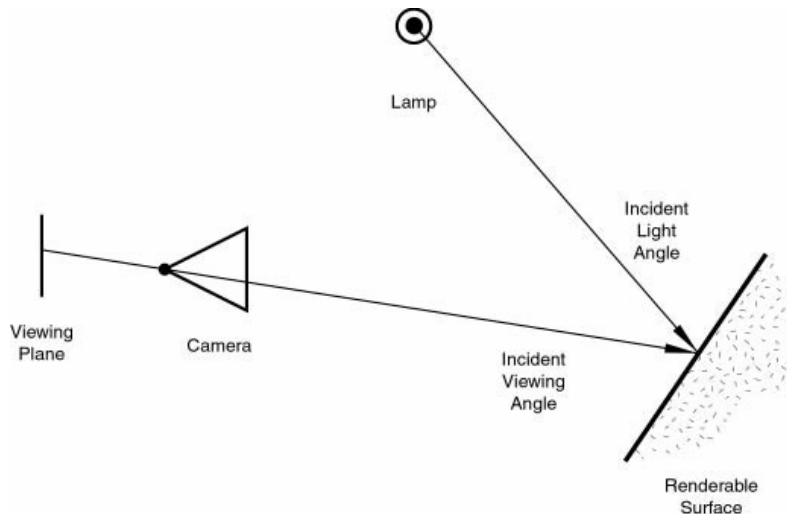
vertices. El usuario puede forzar un area de textura con el comando **TKEY** (fuera del modo edición).

- De hecho como la base dicta la poligonización del objeto implica eso, por lo que si tenemos dos Meta Objects y movemos uno de ellos veremos la poligonización de un objeto *non-base* durante el movimiento, sin importar cual de los dos objetos se está moviendo realmente.

El Panel *Meta Ball* ofrece pocas posibilidades. Si está en Modo Objeto, solo estará el Panel ahí presente. Puedes definir la dimensión media de ambas poligonizaciones en la vista 3D via Panel *Wiresize*, y la representación del tiempo via Panel *Rendersize*. Cuanto más bajo sea el suavizado del Meta Object más lento será la computación. El Num Button *Threshold* es un ajuste importante para MetaObjects. Controla el 'nivel de campo' en el cual se computa la superficie. Para tener un control mas fino, cuando estamos en modo edición, el Num Button *Stiffness* de *Meta Ball Tools* permite que aumentes o reduzcas el campo de influencia del MetaObject. En este ultimo panel puedes cambiar el tipo de Meta Object y fijarlo negativo (que es substractivo, más bien que añadido) con otro Meta Objects de las mismas características.

PARTE III - MATERIALES

Antes de que pueda entender como diseñar de forma eficiente con materiales, debe entender como interactúan con el motor de render de Blender las luces simuladas y las superficies, y como los parámetros del material controlan estas interacciones. Una profundización en el entendimiento del motor le ayudará a entender todo ello. La imagen renderizada que usted crea con Blender es una proyección de la escena en una superficie imaginaria llamada *plano de proyección* (*viewing plane*). El plano de proyección es análogo a la película en una cámara tradicional, o a la córnea en un ojo humano, excepto que recibe luz simulada en lugar de luz real. Para renderizar una imagen de una escena primero debe determinar que luz de la escena está llegando a cada punto del plano de proyección. La mejor manera de responder a esta pregunta es seguir una línea recta (el rayo de la luz simulada) desde dicho punto en el plano de proyección y el punto focal (la posición de la cámara) hasta que impacte sobre una superficie representable en la escena, en este punto podremos determinar que luz debería impactar ese punto. Las propiedades de la superficie y el ángulo de incidencia de la luz nos dirán qué cantidad de luz debería ser reflejada por el ángulo de visión incidente (*Principio básico del motor de render.*).



Principio básico del motor de render.

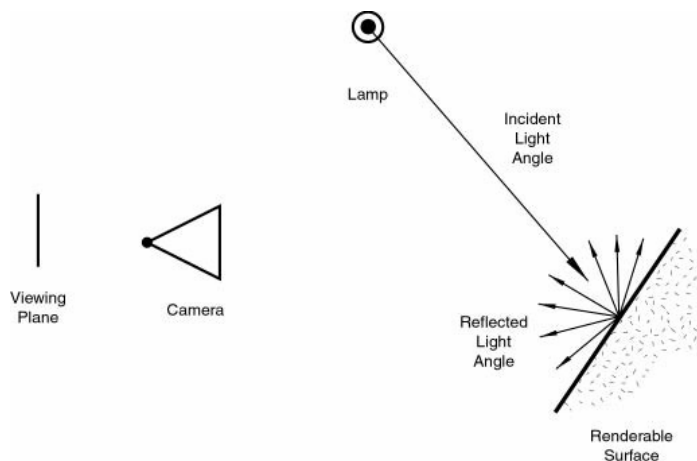
Cuando un rayo de luz impacta sobre cualquier punto de una superficie, se dan lugar dos tipos básicos de fenómenos: difusión y reflexión especular. La difusión y la reflexión especular se distinguen la una de la otra principalmente por la relación entre el ángulo de luz incidente y el ángulo de la luz reflejada.

Capítulos

- Sombreadores de Difusión
- Sombreadores de Especificidad
- Materiales en la práctica
- Sombreadores de Rampa
- Reflejos y Transparencias
- Materiales Múltiples
- Materiales Especiales

Sombreadores de Difusión de Luz

El impacto de la luz sobre una superficie y la posterior reirradiación por fenómeno de difusión puede ser muy dispersa, por ejemplo, la re-irradiación en todas las direcciones de forma isotrópica. Esto significa que la cámara verá la misma cantidad de luz desde ese punto de superficie sin importar cual sea su *ángulo de visión incidente*. Es esta cualidad la que hace a la luz difusa *independiente del punto de visión*. Por supuesto la cantidad de luz que impacta una superficie depende del ángulo de luz incidente. Si la mayoría de la luz que golpea una superficie es reflejada de forma difusa, la superficie tendrá una apariencia mate (*Luz re-irradiada con fenómeno de difusión.*).



Luz re-irradiada con fenómeno de difusión.

Desde la versión 2.28, Blender ha implementado tres fórmulas matemáticas diferentes para calcular la difusión. Y, de forma más notable, el fenómeno de difusión y reflexión especular, que suelen ir limitados en un solo tipo de material, y han sido separados para que sea posible elegir la implementación de la difusión y reflexión especular de forma separada. Las tres implementaciones de difusión, o *sombreadores (shaders)*, usan dos o tres parámetros cada una. Los dos primeros parámetros son compartidos por todos los sombreadores de difusión y son los *Colors de Difusión* o simplemente *color*, del material, y la cantidad de energía de la luz incidente que es difundida en realidad. Esta última cantidad, dada en un rango de $[0,1]$, es normalmente llamada *Refl* en el interfaz. Los sombreadores implementados son:

- *Lambert* -

Éste fué el sombreador de difusión por defecto de Blender hasta la versión 2.27. Y como tal, todos los tutoriales antiguos se refieren a él, y todas las imágenes pre-2.28 han sido creadas con él. Este sombreador tiene únicamente los parámetros por defecto.

- *Oren-Nayar* -

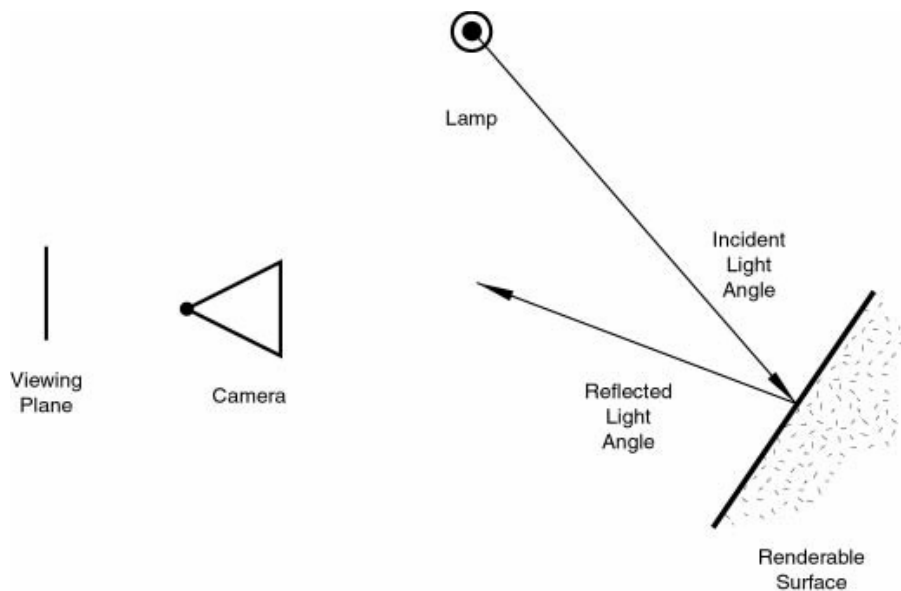
Este sombreador introducido por primera vez en Blender 2.28. Toma una especie de aproximación 'física' al fenómeno de la difusión. Además de los dos parámetros por defecto, tiene un tercero que es usado para determinar la cantidad de rugosidad microscópica de la superficie.

- *Toon* -

Este sombreador fue introducido por primera vez en Blender 2.28. Es un sombreador muy poco 'físico' ya que no está pensado para imitar la realidad, sino para producir renders de 'dibujos animados', con límites muy claros de luz-sombra y regiones uniformes de iluminadas-en sombra. Aunque es relativamente simple, también requiere dos parámetros más que definen el tamaño de las áreas iluminadas y el suavizado de los límites de sombra. Una sección siguiente, dedicada a la implementación actual del material, analizará todo esto y sus parámetros relativos.

Reflexión Especular

Al contrario que la difusión, la reflexión especular es *dependiente del punto de visión*. De acuerdo con la ley de Snell, la luz que impacta sobre una superficie especular se verá reflejada con un ángulo espejado al ángulo de luz incidente (NdT: Un ángulo igual a ángulo incidente +90 grados), lo cual hace que el ángulo de visión cobre mucha importancia. La reflexión especular crea finos y brillantes reflejos, haciendo que la superficie parezca pulida (*Reflexión especular*).



Reflexión especular.

En realidad, la Difusión y la reflexión especular son generadas por el mismo proceso exacto de dispersión de la luz. La difusión es dominante en una superficie que tiene una pequeña escala de rugosidad en ella, con respecto a la longitud de onda, de forma que la luz se ve reflejada en muchas direcciones por cada pequeño fragmento de superficie, con cambios muy pequeños en el ángulo de la superficie. Por otro lado, la reflexión especular, predomina en una superficie que es suavizada, con respecto a la longitud de onda. Esto implica que la dispersión de los rayos de cada punto de la superficie serán direccionados en su mayoría en la misma dirección, más que al ser dispersada de forma difusa. Es simplemente un problema de la escala de detalle. Si la rugosidad de la superficie es mucho menor que la longitud de onda de la luz incidente parecerá plano y actuará como un espejo.

Note: Es importante incidir especialmente en que el fenómeno de la reflexión especular discutido aquí no es la reflexión que deberíamos ver en un espejo, pero sí los destellos de luz que deberíamos ver en una superficie pulida. Para conseguir reflexiones como las de un espejo debería usar un trazado de rayos como tal (raytracer), pero ésto puede producir superficies estilo espejo convincentes, mediante una aplicación cuidadosa de texturas, como veremos más tarde.

Como la difusión, la reflexión especular tiene bastantes implementaciones diferentes, o *sombreadores especulares*. De nuevo, cada una de dichas implementaciones comparte dos parámetros comunes: el *Color Especular* y la energía de la especularidad, en un rango de [0,2]. Esto permite que se pueda drenar más energía con eficiencia como reflexión especular como si fuera energía incidente. Como resultado, un material tiene al menos dos colores diferentes, uno difuso y uno especular. El color especular suele ser blanco puro, pero pueden ser cambiados sus valores para conseguir efectos interesantes. Los cuatro sombreadores especulares son:

- *CookTorr* -

Éste fue el único sombreador especular en Blender hasta la versión 2.27. De hecho, hasta dicha versión no era posible aplicar de forma separada sombreadores de difusión y especulares, y no había más que una implementación de un material llano. Además de los dos parámetros estándar, este sombreador usa un tercero, *dureza (hardness)*, que regula la anchura de los resaltes especulares. Cuanto menor dureza tenga, más anchos serán los reflejos.

- *Phong* -

Éste es un algoritmo matemático diferente, usado para calcular los reflejos especulares. No es muy diferente de CookTor, y está configurado por los mismos tres parámetros.

- *Blinn* -

Éste es un sombreador especular más 'físico', pensado para combinarlo con el difuso Oren-Nayar. Es más físico debido a que añade un cuarto parámetro, un *índice de refracción (IDR - IOR)* a los tres anteriores. Este parámetro en realidad no se usa para calcular la refracción de los rayos (para ello se necesita un trazado de rayos), pero para calcular correctamente en la reflexión especular la intensidad y extensión por la ley de Snell. La dureza y los parámetros especulares dan más grados de libertad.

- *Toon* -

Este sombreador coincide con el de difusión Toon. Está diseñado para producir la agudeza y reflejos uniformes de los dibujos. No tiene dureza pero añade un par de parámetros. Tamaño y suavizado, que dictaminan la extensión y rugosidad de los reflejos especulares. Gracias a esta flexible implementación, la cual trata de separar los fenómenos de difusión y de reflexión especular, nos permite tener un sencillo control de blender de cuánta luz incidente que impacta sobre un punto en una superficie se dispersa por difusión, cuanta se refleja de forma especular, y cuánta es absorbida. Esto, alternativamente, determinar en qué direcciones (y en qué cantidades) la luz es reflejada por un punto de luz dado; esto es, desde qué fuentes (y en qué cantidades) la luz es reflejada hacia un punto dado en el plano de proyección. Es muy


importante recordar que el color del material es solo un elemento en el proceso de render. El color normalmente es el producto del color de la luz y el color del material.

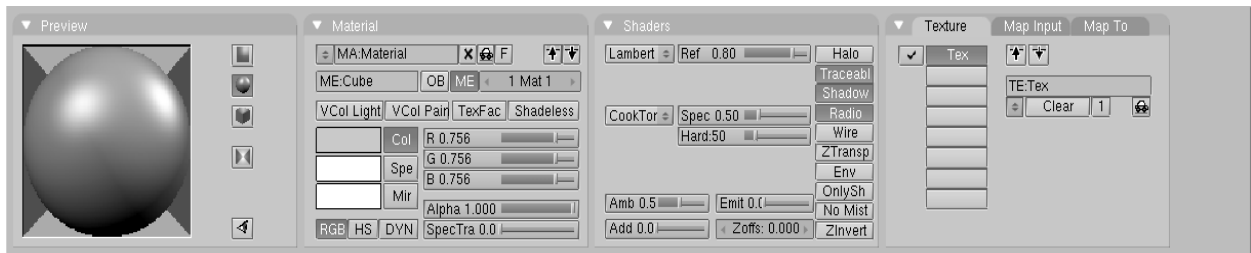
Materiales en la práctica

En esta sección veremos como configurar los distintos parámetros de materiales en Blender, y qué esperar como resultado.



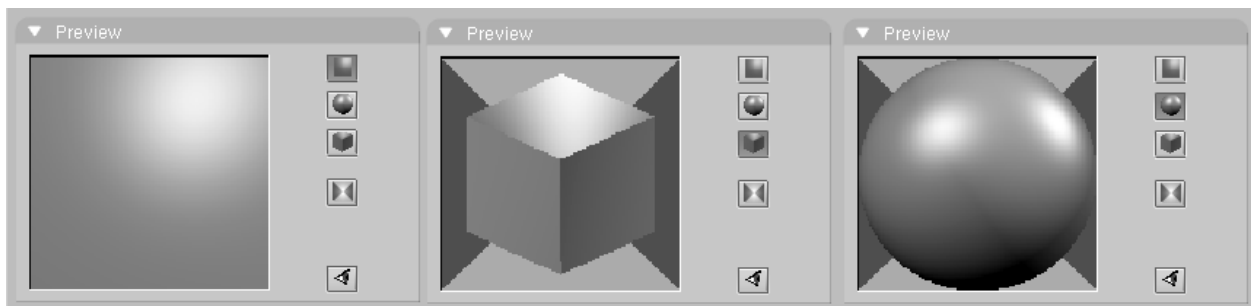
Agregar nuevo material.

Una vez que un objeto es seleccionado, presionando la tecla F5 o , se cambia al contexto de Sombreado y aparece la ventana Botones de Material. Esta ventana aparecerá completamente vacía, a menos que el Objeto seleccionado ya tenga un material asignado. Si no hay un material asignado, agregar uno nuevo con el botón (*Agregar nuevo material.*). Una vez agregado un material, los botones aparecerán como se muestra en *Botones de Material.*. Existen cuatro paneles. De izquierda a derecha: un panel de *Vista Previa (Preview)*, un panel de *Material*, un panel de *Shader* y un panel de *Textura*. Por ahora nos concentraremos en los primeros tres.



Botones de Material.

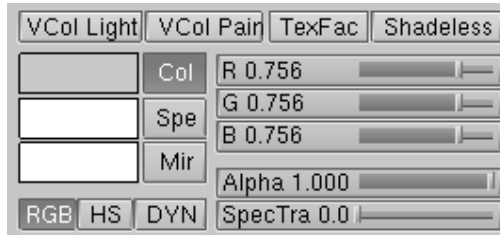
El panel *Vista Previa (Preview)* muestra una previsualización del material. Por defecto muestra un plano visto desde arriba, pero puede ser cambiado a una esfera o un cubo con los botones que se encuentran a la derecha del panel (*Previsualización del Material, plano (izquierda), esfera (centro) y cubo (derecha).*).



Previsualización del Material, plano (izquierda), esfera (centro) y cubo (derecha).

Los Colores del Material

El panel de *Material* (*Botones de Color de Material.*) permite, entre otras cosas, elegir los colores del material.



Botones de Color de Material.

Cada material puede mostrar hasta tres colores:

- *El color básico del material* o Color de Difusión (Diffuse

color), o, en resumen, el *Color* (botón *Col* en la interfase), que es el color usado por el sombreador de difusión (diffuse shader).

- *El color Especular,*

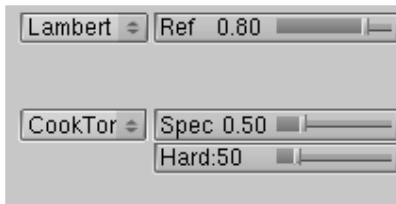
indicado por el botón *Spe* en la interfase, es el color usado por el sombreador especular (specular shader).

- *El color de Espejo (The Mirror color),*

indicado por el botón *Mir* en la interfase, es el color usado por texturas especiales, para simular los reflejos especulares (mirror reflections). (Mas información sobre este tema en la sección de Mapeado de Ambiente. Los botones ya mencionados seleccionan el color pertinente, que es mostrado en la vista previa inmediatamente a la izquierda de cada uno. Los tres deslizadores (sliders) a la derecha, permiten cambiar los valores para el color *activo* mediante un esquema RGB o HSV. Se pueden seleccionar estos esquemas con los botones *RGB* y *HSV* que se encuentran abajo. El botón *DYN* se usa para configurar las propiedades Dinámicas del Objeto dentro del motor de Tiempo Real (que queda fuera del alcance de este libro), mientras que los cuatro botones de arriba se relacionan con *Pintura de Vértices (Vertex Paint)* y *Textura UV (UV Texture)*.

Los Sombreadores (The Shaders)

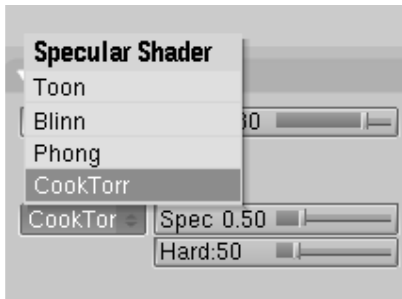
El panel *Sombreador (Shader)* (*Botones del Sombreador de Material (Material Shader buttons).*) muestra dos Botones de Menú que permiten seleccionar un sombreador de difusión (diffuse shader) (*Sombreadores de Difusión del Material (Material Diffuse shaders).*) y un sombreador de especularidad (specular shader) (*Sombreadores de Especularidad del Material (Material Specular shaders).*).



Botones del Sombreador de Material (Material Shader buttons).

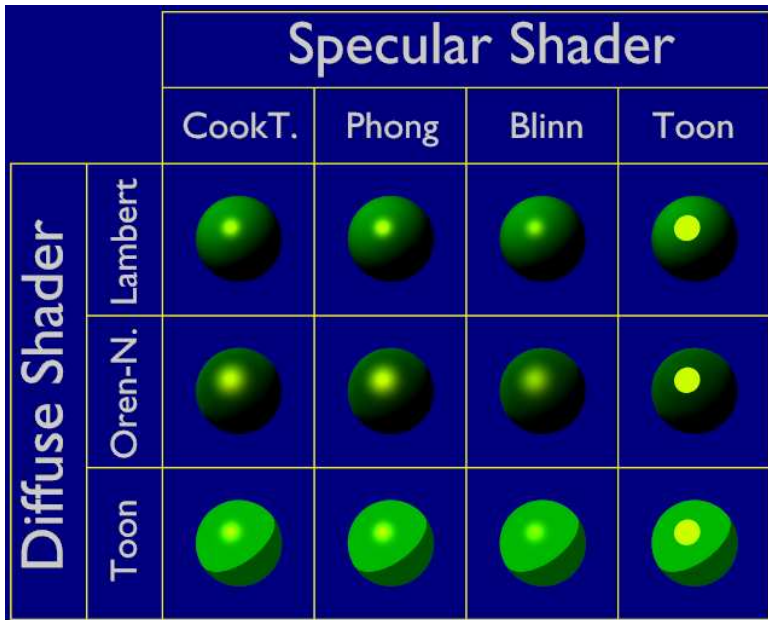


Sombreadores de Difusión del Material (Material Diffuse shaders).



Sombreadores de Especificidad del Material (Material Specular shaders).

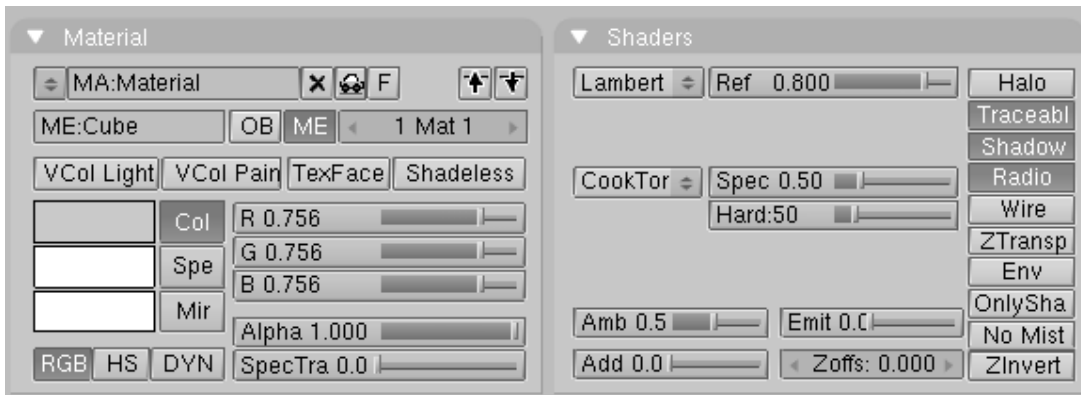
Los dos deslizadores (sliders) al costado, valido para todos los sombreadores, determinan la intensidad de los fenómenos de Difusión y Especificidad. El deslizador (slider) *Ref* tiene un rango de 0 a 1, mientras que *Spec* posee un rango de 0 a 2. Hablando en términos puramente físicos, si *A* es la energía lumínica que afectando al objeto, *Ref* multiplicado por *A* es la energía difundida (diffused) y *Spec* multiplicado por *A* es la energía reflejada especularmente. Para que sea físicamente correcto, esto debe ser $Ref + Spec < 1$, o el objeto irradiaría más energía de la que recibe. Pero esto es CG, así que no es necesario ser tan estricto sobre cuestiones físicas. Dependiendo del sombreador seleccionado, pueden aparecer otros deslizadores (sliders), permitiendo configurar los variados parámetros discutidos en la introducción. Con la intención de ser completo, *Descripción de Sombreadores (Shader overview)*. muestra todas las combinaciones posibles. Por supuesto, ya que existen tantos parametros, esto es sólo una pequeña muestra.



Descripción de Sombreadores (Shader overview).

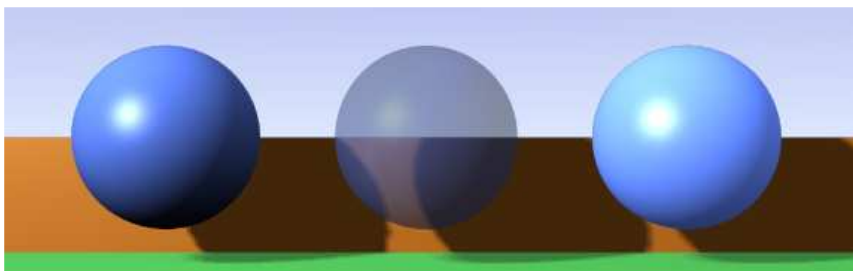
Refinando los Materiales (Tweaking Materials)

Los botones de material restantes, ambos en los paneles de *Material* y *Sombreadores* (shaders), producen algunos efectos interesantes.



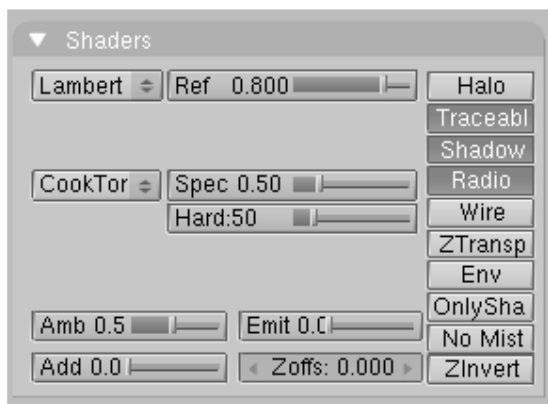
Deslizadores adicionales de Material (Additional material sliders).

Deslizadores adicionales de Material (Additional material sliders). muestra algunos deslizadores (sliders) interesantes. *Alpha* maneja la opacidad del material; 1 es completamente opaco y 0 es completamente transparente. *SpecTra* obliga a los brillos especulares (specularity highlights) en objetos transparentes a ser opacos. *Shadeless* hace al material insensible a su sombreado (shading), dándole un color de difusión uniforme. En el panel *Sombreadores* (Shaders), el deslizador (slider) *Emit*, da como resultado, si es distinto de cero, un carácter emisor al material. Esto hace visible al material, aún sin luces, y puede ser él mismo una fuente de luz si se utiliza el motor de Radiosidad (Radiosity) (*Material común* (izquierda), *material con Alpha < 1* (centro) y material con *Emit > 0* (derecha).).



Material común (izquierda), material con $\text{Alpha} < 1$ (centro) y material con $\text{Emit} > 0$ (derecha).

La columna de botones restante (*Botones especiales de Material.*) activa algunas funciones especiales. El botón de más arriba, *Halo*, convierte al material en un material 'Halo', que será explicado más adelante. *Traceable*, *Shadows* y *Radio* están activos por defecto. El primero permite al material *proyectar* sombras, mientras que el segundo le permite *recibirlas*; el tercer botón permite al material ser tenido en cuenta si se hace un render con Radiosidad (Radiosity).



Botones especiales de Material.

Wire renderiza el Objeto como una estructura de alambre (wireframe). *ZTransp* es necesario para activar el efecto de transparencia por Alpha. Los otros botones no son usados muy seguido y se describen en la sección de referencia al final de este libro.

Sombreadores de Rampa

Introducción

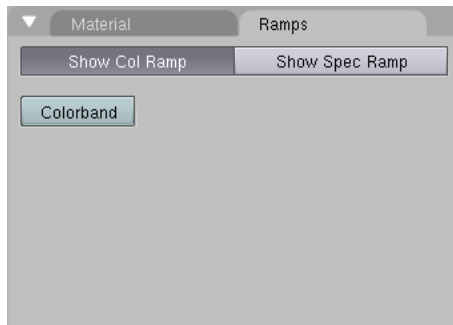
En muchas situaciones de la vida real - como en la piel o los metales - el color de las reflexiones especulares y difusas pueden variar levemente, basados en la cantidad de energía que recibe una superficie o del ángulo de incidencia de la luz. Las nuevas opciones de *Ramp Shader* (*Sombreador de Rampa*) en Blender, permiten establecer un rango de colores para un *Material*, y definen de qué manera variará este rango sobre la superficie y como se mezclará con el "color real" (comúnmente desde el material o como salida de textura).

Como el cálculo de texturas en Blender, tiene lugar antes que el sombreado, el *Ramp Shader* puede reemplazar completamente las texturas o el color del material. Pero mediante el uso de

las opciones de mezcla y valores Alfa, es posible crear una capa adicional de sombreado en los materiales de Blender.

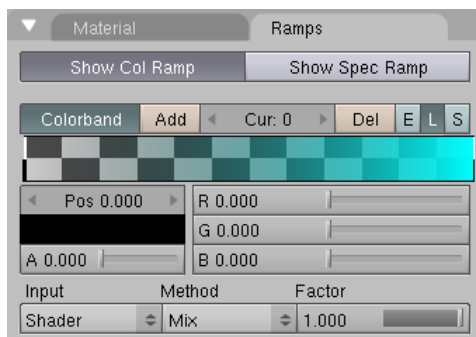
Interfaz

El panel *Ramps (Rampas)* está localizado en el Contexto de *Materiales (F5)*. Aquí puede usar los dos botones superiores para mostrar tanto la configuración de las rampas difusas pulsando **Show Col Ramp** o de las rampas especulares con **Show Spec Ramp**. (*Panel Ramps.*).



Panel Ramps.

Presionando el botón **Colorband** se habilitan los *Sombreadores de Rampa*. Por omisión, se abre con dos colores, el primero con $\text{Alpha} = 0$, y sin color y el segundo con $\text{Alpha} = 1$ y un color cian. (*Panel Colorband de las Rampas.*).



Panel Colorband de las Rampas.

- *Add* - Agrega un nuevo color en el medio de la colorband en gris neutro

por omisión.

- *Cur* - Muestra el número de color de la colorband seleccionado actualmente
- *Del* - Borra la posición actual.
- *E/L/S* - Define el tipo de interpolación entre el color del *Ramp Shader* y el color del *Material*. *E* - Ease o Cubico, *L* - Lineal y *S* - B-Spline.
- *Alpha* - Define en qué medida se verá el efecto del *Ramp Shader*. Un valor

para *Alpha* de 0 significa que el *Ramp Shader* es totalmente transparente y no será visible en el material final. Un valor de 1 hace que el *Ramp Shader* sea opaco. Si definió colores con diferentes valores de *Alpha*, se los interpolará entre sí para lograr una transición suave entre los distintos valores de transparencia. Puede previsualizar los valores de *Alpha* sobre la

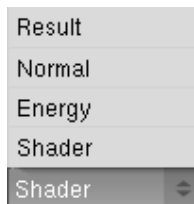
colorband con el patrón en forma de damero detrás de la colorband. Si el patrón es visible entonces la transparencia es menos que 1.

- *R/G/B* - Los valores RGB del color actual. Puede pulsar con **LMB** sobre el campo color bajo el campo *Pos* para elegir un color usando el *Selector de Colores*.

El color actual es indicado con una barra ligeramente más gruesa en la colorband. Para seleccionar una posición de color puede pulsar con **LMB** sobre la posición de color deseada o avanzar el número del color actual en el campo *Cur* arriba y abajo con las flechas izquierda y derecha. También puede pulsar **SHIFT-LMB** en el campo e ingresar el número de color requerido manualmente. Puede mover la posición de un color pulsando simplemente con **LMB** y arrastrandolo sobre la colorband. También dispone de un campo *Pos* donde puede mover la posición con las flechas de cursor izquierda y derecha o pulsar **SHIFT-LMB** en el campo e ingresar la posición manualmente.

Los dos botones emergentes y el deslizador en el fondo del panel definen como funcionan los *Sombreadores de Rampa*:

Input



Menú emergente Input.

- *Shader* - El valor como sale de la fórmula de sombreado básico (como

Lambert o Phong) define el color. Aquí sólo importa la dirección de la luz y no la cantidad de la misma.

- *Energy* - Como *Shader*, pero ahora también se consideran la energía, el

color y la distancia de las luces. esto hace que el material cambie su color a medida que más luz brille sobre él.

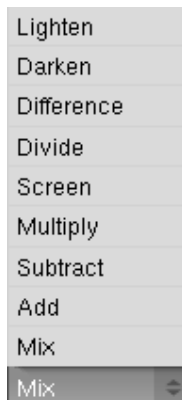
- *Normal* - La normal de la superficie, relativa a la cámara, es empleada por el

Sombreador de Rampa. Esto también es posible lograrlo por medio de una textura, pero se agrega por conveniencia.

- *Result* - Las tres opciones anteriores funcionan por la luz, esta opción lo hace al final de todos los cálculos de sombreado. Esto permite control total sobre el sombreado completo, incluyendo

resultados estilo "Caricatura". Utilizar valores de Alpha aquí es fundamentalmente útil para ajustar un retoque final del *Material*.

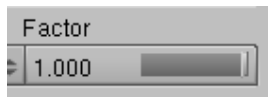
Method (Método)



Menú emergente Método.

El menú *Method* tiene varias opciones para el tipo de mezclado: *Mix* (Mezclar), *Add* (Sumar), *Subtract* (Restar), *Multiply* (Multiplicar), *Screen* (Pantalla), *Divide* (Dividir), *Difference* (Diferencia), *Darken* (Oscurecer) y *Lighten* (Aclarar). Las opciones de *Method* ofrecen una opción para elegir cómo se realizará el mezclado entre el color de *Input* y los *Sombreadores de Rampa*.

Factor



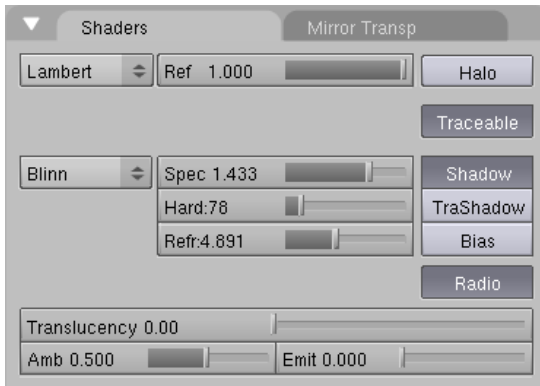
Deslizador Factor.

El deslizador *Factor* denota el factor global del efecto del *Sombreador de Rampa*: 0 significa ningún efecto y 1.0 significa un efecto total.

Un ejemplo

Veamos un sencillo ejemplo de uso de los *Sombreadores de Rampa*.

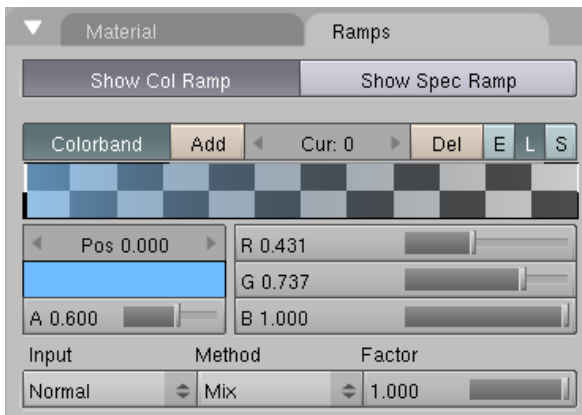
- Elimine el objeto cubo inicial de la escena y cree una malla Mono ! (SHIFT-A -> Add -> Mesh -> Monkey). Presione **Subsurf** y ponga el nivel de *Subdivisión Subsurf* para pantalla y para render a 2. Presione **Set Smooth** para obtener un Mono agradablemente suavizado! Todos estos botones están en el Contexto *Edición*. (F9).
- Ahora pulse **TAB** para salir de *ModoEdición*. Presione **F5** para entrar al contexto *Materiales*. En el panel *Material* presione **Add New** para crear un nuevo material. Modifique los parámetros en la pestaña *Shaders* como en *Valores del Sombreador*.



Valores del Sombreador.

- Pulse sobre la pestaña *Ramps* para abrir el panel *Ramp Shader*. Presione el botón *Colorband* para activar el efecto *Sombreador de*

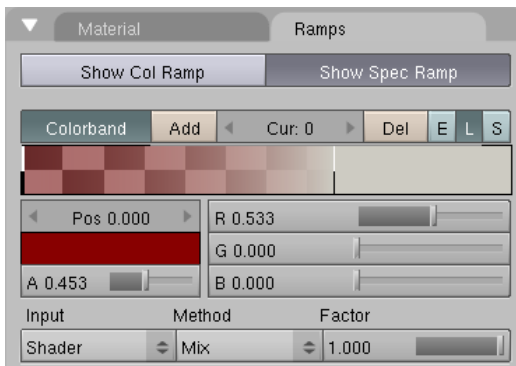
Rampa. Ahora ponga los valores de los parámetros lo más parecido posible a la *Valores del Sombreador de Rampa*. Recuerde poner *Input* a Normal. El segundo color a la derecha tiene puesto *Alpha* = 0 y el color es negro puro.



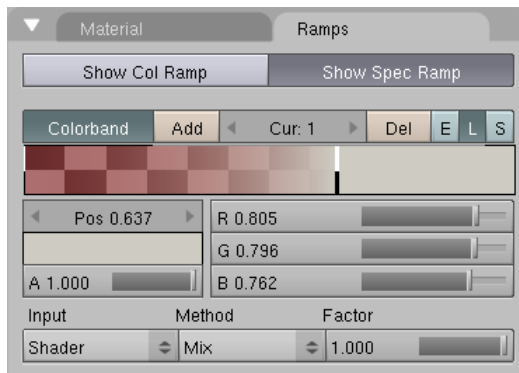
Valores del Sombreador de Rampa.

- En la pestaña *Ramps* presione *Show Spec Ramp* y ajuste los parámetros lo

más fielmente posible a *Color 0 del Sombreador de Rampa Especular*. y a *Color 1 del Sombreador de Rampa Especular*.



Color 0 del Sombreador de Rampa Especular.



Color 1 del Sombreador de Rampa Especular.

Aquí se ve el resultado renderizado con los valores que acabamos de ingresar. En la imagen *Ningún Sombreador de Rampa*, no hay ningún *Sombreador de Rampa* activo. En la *Rampa de Color*, está *activada la Rampa de Color* y *finalmente en la Rampas de Color y Especular*, *ambas rampas, la Rampa de Color y la Rampa Especular están activadas*.

Tenga presente que acabamos de demostrar solo un efecto de los *Sombreadores de Rampa*. Hay mucho más para explorar, pruebe cambiando los parámetros *Input* y *Method* para ver resultados totalmente diferentes de los que hemos visto recién en el ejemplo.



Ningún
Rampa.



Sombreador
de Rampa de Color.



Rampas de Color y Especular.

Reflejos por Raytracing

Antiguamente Blender utilizaba mapas de ambiente para poder simular las reflexiones de objetos. Esta manera de resolver el problema siempre fue dificultosa puesto que necesitaba del uso de *Empties* (objetos nulos) y *capas* para obtener un simple reflejo plano. Afortunadamente, al incluirse *Raytracing* al motor de render, se simplificó enormemente el proceso permitiendonos lograr un mayor realismo en las imagenes.

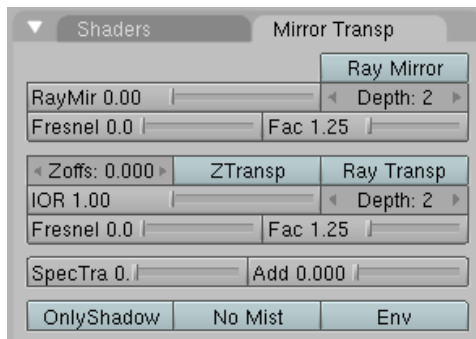
Comprendiendo los fundamentos

No es demasiado complejo, el principio de los reflejos por raytracing es muy sencillo: un rayo es disparado desde la cámara y viaja a través de la escena hasta que encuentra un objeto. Si el primer objeto encontrado por el rayo no es reflejante, entonces el rayo toma el color del objeto. Si el objeto ES reflejante, entonces el rayo rebota y viaja hasta chocar con otro objeto,

y así sucesivamente, hasta que se topa con un objeto no reflejante y éste le transmite a la cadena completa de rayos su color. Eventualmente, el primer objeto reflejante hereda los colores de sus alrededores en forma proporcional a su valor *RayMir*. Obviamente, si solo hay en la escena objetos reflejantes, el render puede durar eternamente. Por esta razón se ha implementado un sistema limitante para el recorrido de un único rayo a través del valor *Depth* : este parámetro establece el máximo número de rebotes permitidos por rayo.

Los botones Ray Mirror

Para activar los reflejos, necesita activar primeramente el botón *Ray Mirror*. El *Panel Mirror Transp.* muestra los parámetros disponibles para los reflejos por raytracing en el primer bloque de valores.



El Panel Mirror Transp.

RayMir Define la cantidad de reflectibilidad del objeto. Utilice un valor de 1.00 si necesita un espejo perfecto, o ponga *RayMir* a cero si no necesita reflejos.

Depth Define el número máximo de recursiones por rayo a ser relejado. Típicamente un valor por omisión de 2 es apropiado. Si sus escena contiene muchos objetos reflejantes o si la cámara se acerca demasiado a un objeto reflejante, necesitará aumentar este valor si desea que los reflejos de objetos aledaños se reflejen en su objeto (!). En ese caso, un valor de *Depth* de 4 o 5 es normalmente suficiente.

Fresnel Define la intensidad del efecto Fresnel. El efecto Fresnel controla cuán reflectivo es el material dependiendo del ángulo entre la superficie normal y la dirección de observación. Comúnmente a mayor ángulo, más reflejante se vuelve un material (el reflejo generalmente tiene lugar en el contorno del objeto).

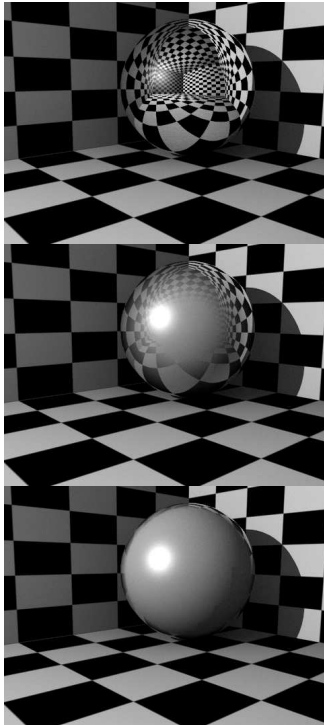
Fac Un 'factor' de control para ajustar como se realiza la mezcla (entre áreas reflejantes y no reflejantes).

Nota: Observe que para lograr un efecto Fresnel *real* con el algoritmo actual, debe poner los valores *Fresnel* a 0.5 y *Fac* a 1.25. De cualquier manera, puede jugar con estos parámetros siguiendo una licencia artística si lo cree necesario.

¿ Aún con problemas para entender qué es el efecto Fresnel ? Hagamos un pequeño experimento para entender de qué se trata. Luego de un día lluvioso, salga y deténgase frente a un charco de agua. Podrá ver el suelo a través del charco. Ahora arrodílese delante del

charco con la cara cerca del suelo y mire de nuevo hacia el punto más lejano sobre la superficie del charco. La superficie líquida cercana a usted le permitirá distinguir el suelo aún, pero si mueve la mirada hacia el otro extremo del charco, el suelo se irá enmascarando gradualmente hasta que lo único visible será el reflejo del cielo. Este es el efecto Fresnel: una superficie comparte propiedades reflexivas y no-reflexivas entre el ángulo de observación y la normal de la superficie.

En la siguiente figura *Demostración del efecto Fresnel con valores igual a (de arriba hacia abajo) 0.0, 2.5 y 5.0*, se demuestra perfectamente este comportamiento en un material perfectamente reflexivo (RayMir 1.0).



Demostración del efecto Fresnel con valores igual a (de arriba hacia abajo) 0.0, 2.5 y 5.0

Un valor de Fresnel 0.0 representa un material espejado perfecto, mientras que un valor Fresnel 5.0 podría representar un material brillante y reflexivo (por ejemplo madera barnizada ?) . Es apenas notorio, pero en la imagen inferior el material es perfectamente reflexivo.

Reflejos Coloreados

Por omisión, un material perfectamente reflexivo como el Cromo, o un espejo, reflejarán exactamente los colores de su entorno. Pero algunos materiales igualmente reflexivos, colorearán sus reflejos con su propio color. Este es el caso del cobre bien pulido o el oro, por ejemplo. Para poder imitar esto desde Blender, debe definir el Color Mirror consecuentemente (Ver *BSG.MAT.F.S68.204*).

Transparencias por Raytracing

Las transparencias y la refracción de la luz dentro de un material transparente son parámetros fundamentales para lograr un renderizado realista de un material como vidrio macizo.

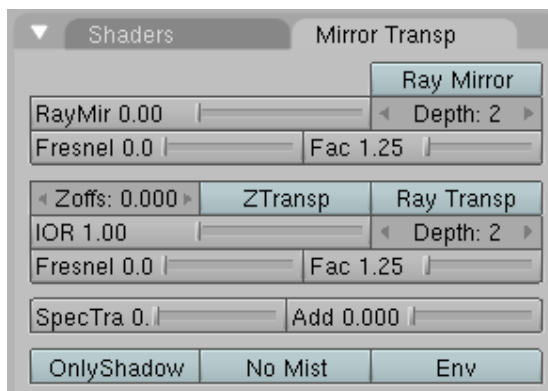
Comprendiendo los fundamentos

Si pudo seguir la explicación para reflejos por raytracing, no tendrá problemas en entender las transparencias por raytracing ya que los fundamentos son los mismos, salvando unos pequeños detalles. Un rayo se dispara desde la cámara y este viaja por la escena hasta que encuentra un objeto. Si el primer objeto tocado por el rayo NO es transparente, entonces el rayo toma su color. Si el objeto ES transparente, el rayo continúa su trayecto a través del mismo hasta toparse con un nuevo objeto y así sucesivamente, hasta que un objeto no transparente (opaco) se alcanza y entonces este objeto da su color a toda la cadena de rayos. Eventualmente el primer objeto hereda los colores de sus alrededores en forma proporcional a su valor de *Alpha* (y del valor de Alpha de cada material transparente atravesado).

Sin embargo, mientras el rayo viaja a través del objeto transparente, puede ser desviado de su ruta dependiendo del Índice de Refracción del material (IOR). Efectivamente, cuando se mira a través de una esfera de vidrio, se observa que el fondo está cabeza abajo y distorsionado: esto es debido al Índice de Refracción del vidrio.

Los botones Ray Transp

Para activar la transparencia, primeramente se debe activar el botón *Ray Transp*. La figura *El Panel Mirror Transp.* muestra los parámetros disponibles para transparencias por raytracing, en el segundo bloque de valores. También debe darle al material un valor de *Alpha* que sea menor de 1.0 (completamente opaco, mientras que 0.0 es totalmente transparente). Por favor, lea *materials_in_practice_tweaking_materials* para aprender más sobre este parámetro y su uso.

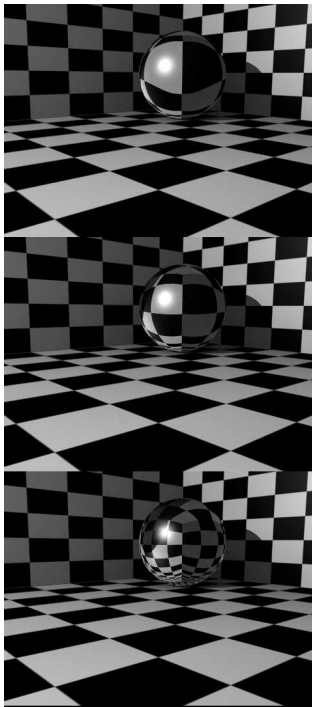


El Panel Mirror Transp.

IOR Define cuanto será refractado un rayo que atravesase el material, produciendo entonces una imagen distorsionada del fondo (ver la figura *Influencia del IOR de un Objeto en la distorsión del fondo: esferas de Agua, Vidrio y Diamante (de arriba hacia abajo).*). Se conocen los valores para distintos materiales comunes: para Aire es 1.000 (sin refracción),

para Alcohol es 1.329, para Vidrio es 1.517, Plástico es 1.460, Agua es 1.333 y Diamante es 2.417.

Depth Define el número máximo de superficies transparentes que un rayo puede atravesar. No hay valores típicos. Los objetos transparentes fuera de este rango de *Depth* se renderizarán totalmente negros si se los observa a través del objeto transparente para el que fue establecido el valor *Depth*. En otras palabras: si observa puntos negros en la superficie del objeto transparente, significa que debe aumentar el valor *Depth* (este es un problema común con objetos con transparencia por raytracing).



Influencia del IOR de un Objeto en la distorsión del fondo: esferas de Agua, Vidrio y Diamante (de arriba hacia abajo).

Fresnel Define la intensidad del efecto Fresnel. El efecto Fresnel controla cuán reflectivo es el material dependiendo del ángulo entre la superficie normal y la dirección de observación. Comúnmente a mayor ángulo, más reflejante se vuelve un material (el reflejo generalmente tiene lugar en el contorno del objeto).

Fac Un 'factor' de control para ajustar como se realiza la mezcla (entre áreas transparentes y opacas).

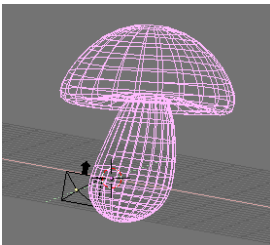
Nota: Observe que para lograr un efecto Fresnel *real* con el algoritmo actual, debe poner los valores Fresnel a 0.5 y Fac a 1.25. De cualquier manera, puede jugar con estos parámetros siguiendo una licencia artística si lo cree necesario. Si necesita más explicaciones del efecto Fresnel, refiérase a *materials_ray_mir_buttons*, Reflejos por Raytracing.

Consejo sobre transparencias por raytracing: proyectando sombras *transparentes*:

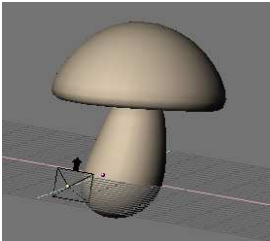
Materiales Múltiples

La mayoría de los objetos están armados de modo que puedan modelarse por partes, con cada una de las mismas hecha de un material diferente. Pero en algunas situaciones, podría ser útil tener un objeto modelado como una malla única y que no obstante posea distintos materiales.

Considere la imagen de la seta como se la ve en la *Malla de una Seta*. Este objeto es una malla única a la que se quieren asignar dos materiales: uno para el tronco y otro para el sombrero. Veremos cómo hacerlo.

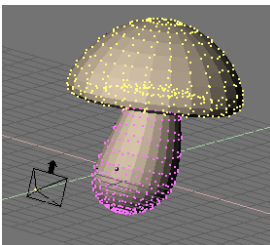


Malla de una Seta



Seta con un material

- Cree un material cremoso apropiado para el tronco de la seta y asígnelo a la seta entera. (*Seta con un material*).

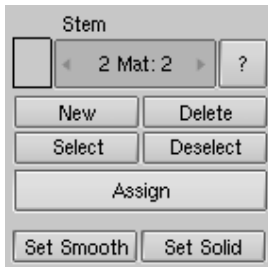


Seta con los vertices del sombrero seleccionados.

- En una Ventana3D, ingrese en Modo Edición para la seta, y seleccione todos los vertices que pertenecen

al sombrero de la misma (*Seta con los vertices del sombrero seleccionados.*).

- En el Panel *Link and Material* de los Botones de Edición de Malla (F9) presione *New* (Añadiendo un nuevo material a la malla.).



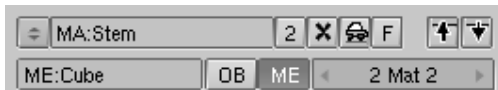
Añadiendo un nuevo material a la malla.

- La malla debería tener ahora dos materiales. En la etiqueta debería leerse *2 Mat: 2* lo

que significa que el material número 2 de los 2 posibles, está activo. Las caras seleccionadas serán asignadas a este nuevo material cuando presione el botón *Assign*; las caras no seleccionadas mantendrán la asignación de material previa. Para ver qué caras pertenecen a qué material, utilice los botones *Select* y *Deselect*. Navegue entre los materiales con el botón numérico *Mat:*. Puede disponer de hasta 16 materiales por malla.

- En cualquier caso, ambos materiales de malla ¡ son instancias del mismo material ! Entonces, teniendo activo el material que quiera cambiar, cambie a los Botones de Materiales (F5) donde

encontrará un botón "2 Mat 2" idéntico al anterior. El material tiene en este momento dos usuarios, como se aprecia por el color azul en el nombre del material, y el botón que luce un "2" (*Material con usuarios múltiples*).

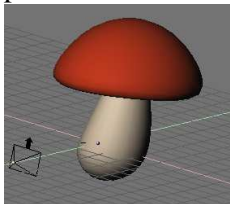


Material con usuarios múltiples

- Pulse sobre el "2" y confirme la pregunta *OK? Single user*. Ahora tiene un duplicado exacto del material. El material original aún se llama "Tallo" ("Stem") y el duplicado es "Tallo.001"

("Stem.001"). Renombre el duplicado a "Sombrero" ("Cap"). Puede editar ahora el material a su gusto

para obtener un sombrero de seta atractivo. (*Seta con dos materiales.*).



Seta con dos materiales.

Texturas: Si su material utiliza texturas, seguirán vinculadas aún después de hacer "usuario-único" al material. Para desvincularlas, de modo que pueda editar las texturas de ambos materiales individualmente, vaya a los botones de Texturas del material, y haga también a la textura "usuario único".

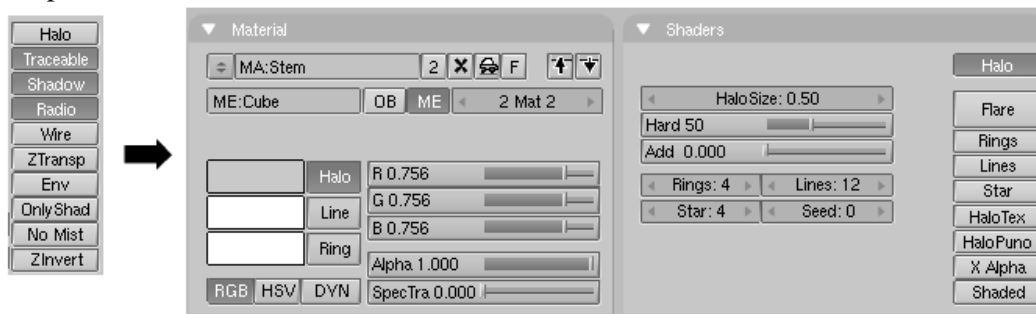
Materiales Especiales

Blender provee una gama de materiales que no obedecen al paradigma del sombreador y que son aplicados por-vértices en lugar de por-caras.

Materiales Halo

Pulsar **F5** para mostrar los botones de Material, y luego pulsar el botón *Halo* en el Panel *Shaders*.

Los paneles cambian, como se muestra en *Botones Halo*.



Botones Halo

Como se ve, las caras de la Malla ya no se renderizan; en cambio, se renderiza un "halo" en cada vértice. Esto es muy útil para sistemas de partículas ya que estos generan vértices libres, pero también pueden ser muy útiles para crear cierto tipo de efectos especiales, al crear un glow de un objeto, o al crear una fuente de luz visible. Como se puede ver en los tres colores que, en un material estándar eran color Difuso (Diffuse), Especular (Specular) y Espejo (Mirror), ahora son relativos a tres características diferentes: el color del halo, el color de algún posible anillo y el color de alguna posible línea que se quiera agregar con los botones correspondientes en *Botones Halo*.



Resultados del Halo

Resultados del Halo muestra el resultado de aplicar un material halo a una malla de un solo vértice. El tamaño del halo, su transparencia y su dureza pueden ser ajustados con los deslizadores correspondientes en *Botones Halo*. Los deslizadores *Añadir (Add)* determinan cuanto de los colores del halo son 'agregados a' (en lugar de mezclados con) los colores de los objetos que se encuentran atrás y junto con otros halos. Para configurar el número de anillos, líneas, y puntos de estrella independientemente, una vez que se han activado los Botones correspondientes, utilizar los Botones Numéricos *Anillos: (Rings:)*, *Líneas: (Lines:)* y *Estrella: (Star:)*. Los anillos y líneas son ubicados y orientados al azar. Para cambiar su patrón se puede modificar el valor del Botón *Seed:*, que configura el generador de números al azar. Usaremos un material halo para crear un display de matriz de puntos.

- Para comenzar, agregar una grilla de 32 x 16 unidades. Luego, agregar una cámara y ajustar la escena de manera de tener una buena vista de la cartelera.
- Usar un programa de 2D para crear algún texto rojo sobre fondo negro, usando una tipografía simple y en negrita. *Textura de la Matriz de Puntos*. Muestra una imagen de 512 píxeles de ancho, por 64 de alto, con algo de espacio en negro a ambos lados.



Textura de la Matriz de Puntos.

- Agregar un material para la cartelera, y convertirlo en tipo *Halo*. Asignar un valor de 0.06 a *HaloSize*. Al renderizar la escena debería verse una grilla de puntos blancos.
- Agregar una Textura, cambiar a Botones de Textura y convertirla en

textura de imagen. Cuando se carga la imagen y se renderiza nuevamente, deberían verse algunos puntos teñidos de rojo sobre la grilla.

- Volver a Botones de Material y ajustar el parámetro *sizeX* a

0.5 aproximadamente, luego renderizar; el texto debería estar centrado en la cartelera.

- Para eliminar los puntos blancos, ajustar el color del material para que sea

de un rojo oscuro y renderizar. Deberían verse solamente los puntos rojos, pero la cartelera aún se ve demasiado oscura. Para corregir esto, entrar en Modo Edición de la cartelera y copiar todos los vértices utilizando el atajo de teclado **SHIFT-D**. Luego, ajustar el brillo mediante el valor de *Add* en los Botones de Material.



Visualización de la Matriz de Puntos (Dot Matrix display)

Se puede animar la textura para que se mueva sobre la cartelera, usando el valor *ofsX* en el panel *Textura (Texture)* de Botones de Material. (Se puede usar una resolución mayor para la grilla, pero habrá que achicar el tamaño de los halos o de otra manera se superpondrán.) (*Visualización de la Matriz de Puntos (Dot Matrix display)*).

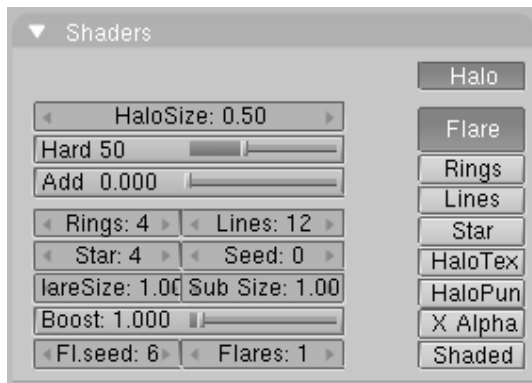
Lens Flares

El ojo humano ha sido entrenado para creer que una imagen es real si muestra artefactos que resultan del proceso mecánico de la fotografía. *Desenfoque de Movimiento (Motion blur)*, *Profundidad de Campo (Depth of Field)*, y *lens flares* son sólo tres ejemplos de estos artefactos. Los dos primeros se explican en *chapter_rendering*; el último se puede producir mediante halos especiales.

Un lens flare simulado le dice al espectador que la imagen fue creada con una cámara, lo que hace que el espectador crea que es auténtica. Los lens flares se crean en Blender a partir de una malla, utilizando primero el botón *Halo* y luego las opciones *Flare* en el Panel *Sombreadores (Shaders)* dentro de la configuración del material.

Prueba activando *Anillos (Rings)* y *Líneas (Lines)*, pero mantiene sutiles sus colores. Juega con los valores de *Flares:* y *Fl.seed:* hasta lograr un resultado que sea agradable a la vista. Puede ser necesario jugar con *FlareBoost:* para obtener un efecto mas fuerte (*Configuración de Lens Flare*).

(Esta herramienta no simula la física de los fotones viajando a través de un lente de vidrio; es sólo para engañar al ojo.)



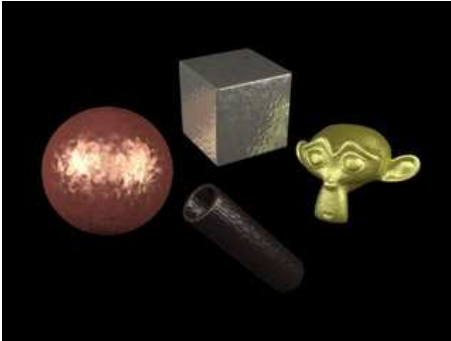
Configuración de Lens Flare

El lens flare de Blender se ve bien en movimiento, y desaparece cuando un objeto oculta la malla que contiene el flare.(*Lens Flare*).



Lens Flare

PARTE IV - TEXTURAS



Algunos metales. Imagen de las notas de versión de la 2.33.

Los ajustes de imagen que hemos visto hasta ahora producen objetos finos, suaves, *uniformes*. Naturalmente estos objetos no son verdaderamente reales, donde la falta de uniformidad es común.

Blender tiene en cuenta esta falta de uniformidad, ya en el color, fuerza especular o reflectiva, rugosidad, y demás, a través de las *texturas*.

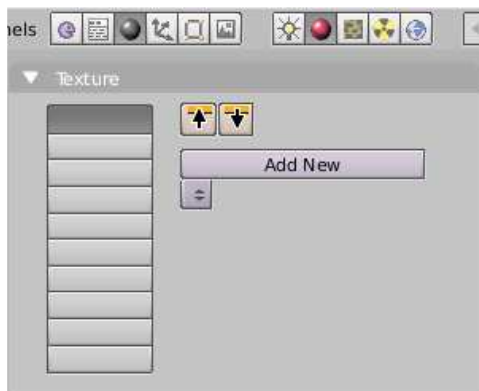
Éstas pueden ser bien texturas procedurales (*Algunos metales*), es decir, texturas que son creadas por una fórmula matemática, imágenes, o mapas de entorno para crear la impresión de reflexiones y refracciones.

Capítulos

- Canales de Textura
- Map Input
- Map To
- Botones de Textura
- Texturas Procedurales
- Texturas Imagen
- Mapeado UV
- Mapas de Relieve y de Normales
- Mapas de Ambiente
- Mapas de Desplazamiento
- Plugins de Textura

Canales de Textura

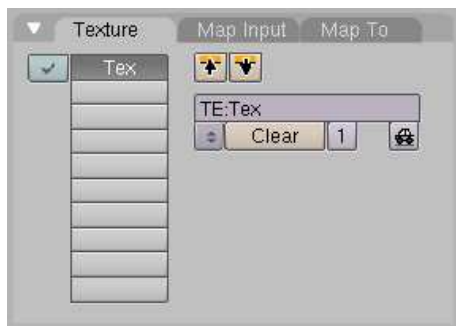
Texturas desde el Punto de Vista del Material



Panel de Textura vacío.

En Blender, los Materiales y las Texturas forman bloques separados con la intención de mantener la interface simple y permitir una integración universal entre bloques de Texturas, Luces y Mundo. Esta relación entre un Material y una Textura, llamado "mapeado", es bidireccional. Primero, la información que es enviada a la Textura debe ser especificada. Luego, se especifica el efecto que la Textura tiene sobre el Material.

El panel *Texture* (textura), a la derecha (y paneles similares existentes para los botones de Luces y Mundo), definen todos estos cálculos. Para un Material sin texturas, el panel muestra una columna de diez *canales* de textura vacíos (*Panel de Textura vacío*). Seleccionando uno de estos canales y presionando *Add New*, o seleccionando una textura existente con el Botón de Menú justo debajo, se agrega una textura y el Panel muestra dos etiquetas mas: *Map Input* y *Map To* (*Panel Texture con una Textura*).



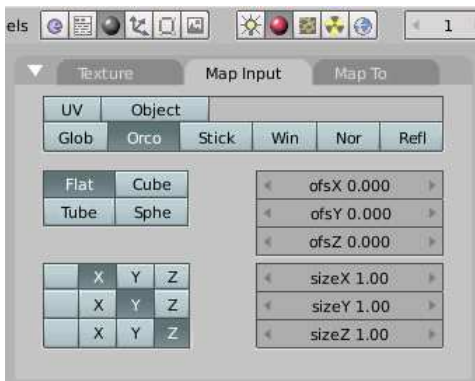
Panel Texture con una Textura.

Los botones de las Etiquetas están organizados en la secuencia en que se realiza el "pipeline de la textura".

Cada canal tiene su propio mapeado individual. Por defecto, las texturas se ejecutan una después de la otra y son luego superpuestas. Como resultado, una textura agregada en el segundo Canal de Textura puede remplazar completamente a una en el primer Canal.

- Al lado de cada Canal de Textura que no está vacío hay un botón que permite activar o desactivar un determinado canal. Canales desactivados simplemente son eliminados de la línea de trabajo (pipeline).
- La Textura en sí es designada por su nombre, que puede ser editado en el Botón de Texto que se encuentra arriba en el Menú de Botones de Textura.
- *Clear*: (borrar) borra la Textura.
- *Botón-Num*: muestra el número de Objetos que usan esa Textura. No puede crearse una copia "de Usuario Único" aquí; ello debe hacerse en los *botones Textura* (F6).
- *Botón-Auto*: intenta asignar un nombre adecuado a la Textura.
- *Correct Nor Map*: (corregir mapa normal) En algunos casos - especialmente en mapas invertidos (XY), en el Panel *Map Input*, un Mapa Bump (relieve) apunta en la dirección equivocada. Conmute este botón para corregir el mapeado bump (el botón pertenece lógicamente al Panel *Map Input*, pero se incluye aquí).

Map Input



Panel *Map Input*

Map Input (mapa de entrada) controla:

- ¿Cuál es la fuente de las Coordenadas de Entrada?
- ¿Dónde está la esquina superior izquierda? ¿Dónde empieza la Textura?
- ¿Cuán grande es la Textura? ¿Con qué frecuencia se repite?
- ¿Tenemos varios puntos de origen de la textura, como en un cubo?
- ¿Debe rotarse la Textura?

Empecemos con las filas de Botones superiores:

Fuente de las Coordenadas de Entrada

- *UV*: Utiliza un tipo especial de mapeado llamado "mapeado UV". Esto es especialmente útil cuando se utilizan imágenes como texturas, como se verá en Mapeado UV. Pueden usarse múltiples texturas con un Mapeado UV.
- *Object*: (objeto) Utiliza un objeto como fuente de las coordenadas, normalmente un Empty. El nombre del Objeto debe ser especificado en el botón de texto de la derecha. Esta es la forma preferida de ubicar una pequeña imagen, como un logo, sobre un punto determinado del objeto. Es interesante el efecto de animar el objeto fuente de las coordenadas de entrada.

- *Glob*: Utiliza las coordenadas 3D Globales de Blender. Es útil para las animaciones. Al mover el objeto, la textura se mueve a través de él. Interesante para hacer que los objetos aparezcan o desaparezcan en un punto concreto del espacio.
- *Orco*: Utiliza las coordenadas locales originales del Objeto.
- *Stick*: Utiliza las coordenadas locales "sticky" (pegadizas) del Objeto. Si se han hecho coordenadas *Sticky* previamente (Botón F9->Panel Mesh->Sticky), la Textura se renderiza en vista de cámara (también llamado "Mapeado de Cámara").
- *Win*: Utiliza las coordenadas de la ventana de la imagen renderizada. Adecuado para mezclar dos objetos.
- *Nor*: Utiliza la dirección de los vectores de normal como coordenadas. Muy útil para crear ciertos efectos especiales (ver más abajo).
- *Refl*: Utiliza la dirección del vector de reflejo como coordenadas. Necesario para los Mapas de Ambiente.

Texturas dependientes del Ángulo de Incidencia

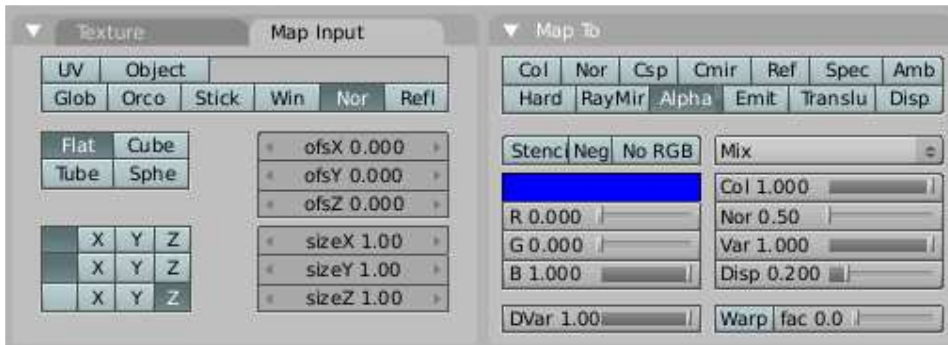
Mencionaremos ahora la limitación de esta técnica: no se trata de un efecto Fresnel real, puesto que es dependiente del ángulo de la cámara - no realmente del ángulo de incidencia. Pero funciona de forma análoga a Fresnel, cuando la cámara no está excesivamente próxima a la superficie o la cara es relativamente pequeña en relación a la cámara. Funciona bien con esferas, y no tan bien con planos grandes.



Ajustes de Material para Alfa dependiente de la incidencia

Como ejemplo simularemos un Alfa Fresnel. Por supuesto puede hacerse mejor mediante el deslizador correspondiente, pero con esta técnica podemos cambiar cada canal 'Map To', no sólo el Alfa. Así usaremos Alfa por motivos de comparación aquí.

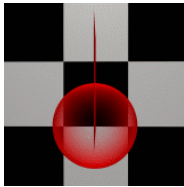
Cree un nuevo material y asigne al valor *Alpha* en el Panel *Material* a 0.04. Active *ZTransp* en el Panel *MirrorTransp* (*Ajustes de Material para Alfa dependiente de la incidencia*).



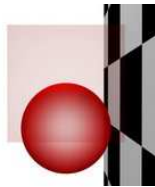
Ajustes de Textura para Alfa dependiente de la incidencia

Añada ahora una nueva textura, ajuste *Map Input* a *Nor*. Coordenadas *-/-/Z*. *Map To Alpha* (*Ajustes de Textura para Alfa dependiente de la incidencia*).

Seleccione el Tipo de Textura *Blend* en los Botones de Textura (F6), Tipo de *BlendSphere* (esfera). Podríamos adaptar la mezcla con *Colorband* en detalle, pero este ejemplo debería mostrar únicamente el principio.



Efecto Fresnel simulado

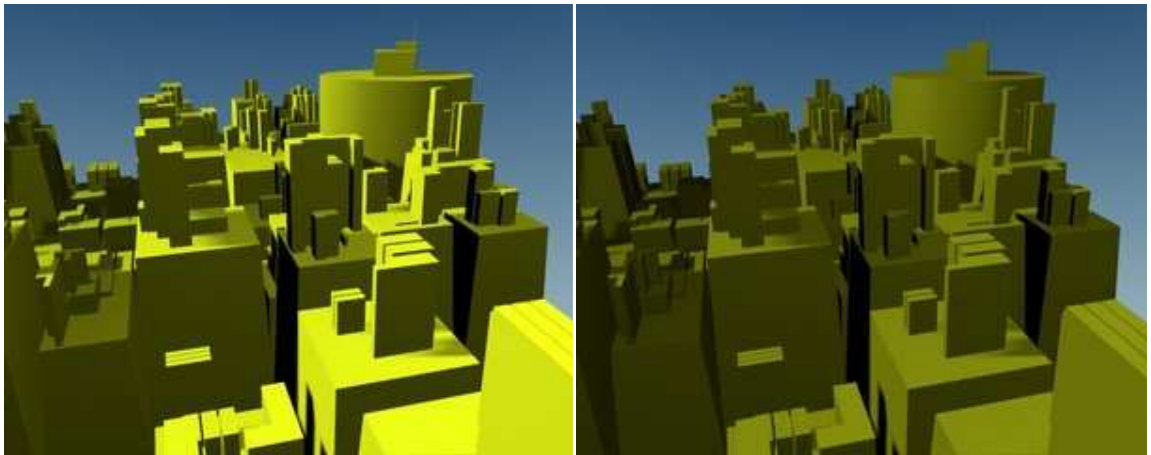


Efecto Fresnel real

Puede verse claramente cómo el plano se pone transparente (96%) si lo observamos directamente perpendicular (*Efecto Fresnel simulado*). Para comparación con Fresnel "real" (*Efecto Fresnel real*).

Profesionalismo Instantáneo

Conseguiremos un bello efecto cambiando ligeramente esta técnica. Active *Map To Ref* y/o algún *Emit*. Ajuste el método de mezcla a *Add* (añadir). Al conseguirse una brillantísima frescura instantáneamente (y es muy fácil de conseguir) se llamó a esto "Profesionalismo Instantáneo" humorísticamente.



Con "Profesionalismo Instantáneo".

Sin "Profesionalismo Instantáneo".

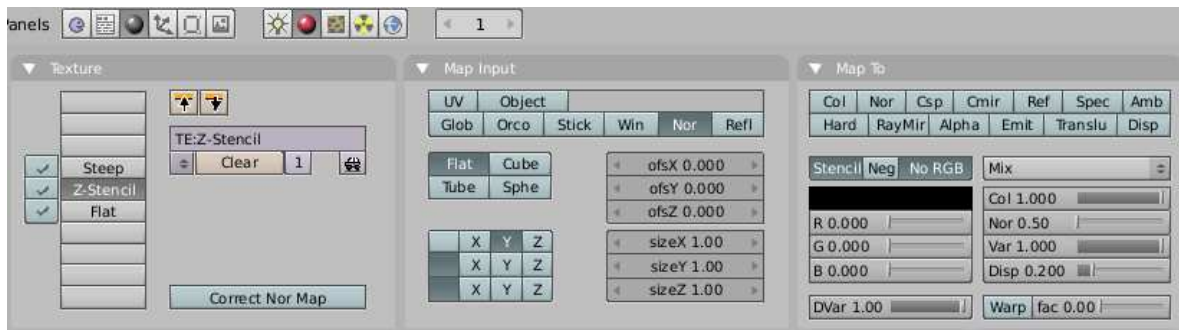
Texturas dependientes de la Pendiente

Una técnica similar puede usarse para crear texturas que son dependientes de la pendiente del terreno. Use una textura Blend lineal con una banda de color (*Ajustes de Textura para texturas dependientes de la pendiente*).



Ajustes de Textura para texturas dependientes de la pendiente.

Use esta textura como un Mapa Stencil (máscara), el cual mantendrá la Textura "Pronunciada" (*Ajustes de Material para para texturas dependientes de la pendiente*). Con los ajustes mostrados el terreno se levanta en la dirección Z.



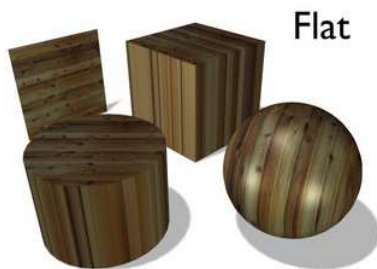
Ajustes de Material para para texturas dependientes de la pendiente.

Mapeado de 2D a 3D

La textura de imagen es la única textura realmente 2D, es la más frecuentemente usada, y una de las texturas más avanzadas que posee Blender. Las características incorporadas, como mapeado de relieves y mapeado MIP con perspectiva corregida, filtrado, y anti-aliasing, garantizan imágenes asombrosas (Para esto hay que tener activado el botón *OSA*, que se encuentra en *DisplayButtons*). Debido a que las imágenes son bi-dimensionales, la forma en que la coordenada 3D es traducida a 2D se especifica en los botones de *mapeado*.

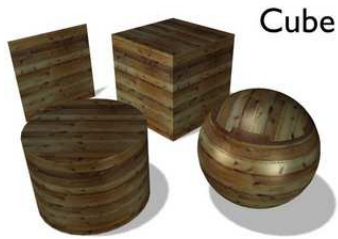
Los cuatro tipos de mapeado estándar son: *Flat* (plano), *(Cube)* (cúbico), *Tube* (tubular) y *Sphere* (esférico).

El tipo más adecuado de mapeado para un modelo determinado, dependerá de la forma general del objeto.



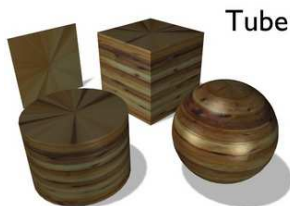
Mapeado Plano.

- El mapeado Plano (*Flat*) (*Mapeado Plano*) da los mejores resultados para caras planas simples. Produce efectos interesantes al aplicarlo sobre una esfera, pero comparado con una esfera mapeada con el mapeado Esférico, el resultado se ve chato. Sobre caras que no están sobre el plano del mapeado, el último píxel de la textura se repite, lo que produce bandas al usarlo sobre un cubo o un cilindro.



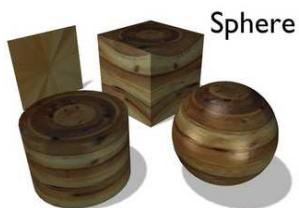
Mapeado Cúbico.

- El mapeado Cúbico (*Mapeado Cúbico*) normalmente da los mejores resultados si se usa sobre objetos que no son demasiado curvos u orgánicos (nótese las "costuras" sobre la esfera).



Mapeado Tubular.

- El mapeado Tubular (*Mapeado Tubular.*), posiciona la textura alrededor del objeto como una etiqueta sobre una botella. Por lo tanto, la textura se ve mas estirada sobre el cilindro. Este tipo de mapeado, obviamente, es muy útil para crear la etiqueta de una botella, o para poner pegatinas sobre objetos redondeados. Sin embargo, este tipo de mapeado no es cilíndrico, por lo que las "tapas" del cilindro no están definidas.



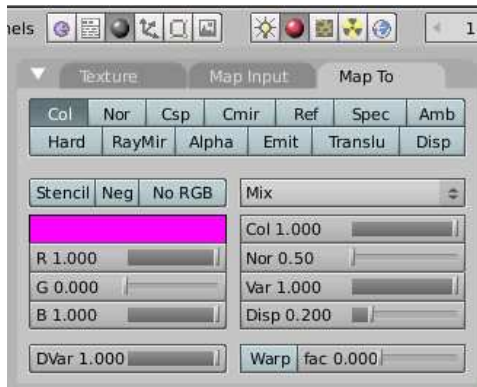
Mapeado Esférico.

- El mapeado Esférico (*Mapeado Esférico*) es el mejor tipo de mapeado para aplicar a una esfera, y es perfecto para la creación de planetas y cosas similares. Es a veces útil para la creación de objetos orgánicos. También produce efectos interesantes al aplicarlo sobre un cilindro.

Desplazamiento de coordenadas, Escalado y Transformación

- *Ofs*: Las coordenadas pueden ser trasladadas asignándoles un desplazamiento (offset). Aumentar *Ofs* mueve la textura arriba a la izquierda.
- *Size*: (tamaño) Escalado de la textura. Ésta es repetida con la frecuencia definida aquí.
- *X,Y,Z*: Cambia las coordenadas X, Y y Z. También permite desactivar la coordenada correspondiente.

Map To



Panel *Map To*.

Las filas superiores del *Panel Map To* muestran los botones conmutadores que determinan qué característica del material se verá afectada por la textura. Algunos de estos botones tienen tres estados, puesto que la textura puede aplicarse de forma positiva o negativa. Todos estos botones son independientes entre sí.

El resultado es también independiente del tipo de valor de entrada. Una textura puede llevar Información de Intensidad (0-255), Color RGB (3 * 0-255), o Vectores Normales (ya sea Relieve o Normales reales: ver sección Mapas de Relieve y de Normales). Algunas de las variantes se muestran a continuación.

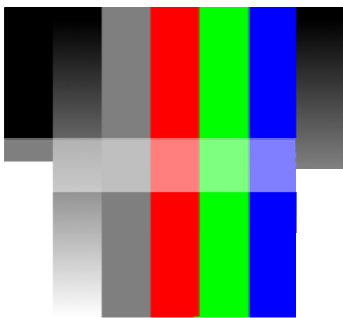
Ajustes *Map To*

- *Col*: (on/off) usa la textura para alterar el color del Material.
- *Nor*: (off/positivo/negativo) usa la textura para alterar la dirección de la normal local. Utilizado para simular imperfecciones de superficie o desuniformidades a través del mapeado bump, o para crear relieves.
- *Csp*: (on/off) usa la textura para alterar el color Especular.
- *Cmir*: (on/off) usa la textura para alterar el color de Espejo. Funciona solamente con mapas de Reflexión.
- *Ref*: (on/off/neg) el valor de la reflexión difusa.
- *Spec*: (on/off/neg) el valor de la reflexión especular.
- *Amb*: (on/off/neg) el parámetro Ambiental.
- *Hard*: (on/off/neg) el parámetro Dureza. Un *DVar* de 1 es equivalente a una Dureza de 130, un *DVar* de 0.5 es equivalente a una Dureza de 65.
- *RayMir*: (on/off/neg) parámetro Ray-Mirror (rayos espejo, sólo para trazado de rayos).
- *Alpha*: (on/off/neg) la Transparencia de material.
- *Emit*: (on/off/neg) el parámetro Emit (emisión).
- *Translu*: (on/off/neg) el parámetro de Translucidez.
- *Disp*: (on/off/neg) para crear Mapas de Desplazamiento.

Ajustes Comunes

- *Stencil*: (máscara) Útil para texturas semitransparentes y "Mapas de Suciedad". La textura se usa como máscara para todas las texturas subsiguientes. Vea el ejemplo a continuación (*Stencil*). Negro define el punto como "intexturizable".
- *Neg*: El efecto de la textura es invertido. Normalmente blanco significa "on" y negro significa "off"; *Neg* invierte esto.
- *No RGB*: Con esta opción, una textura RGB (que afecta al color) es usada como textura de Intensidad (afecta al valor).
- *DVar*: Valor de destino (no para RGB). El valor con el que la textura de Intensidad se mezcla con el valor actual. Dos ejemplos:
 - El valor *Emit* es normalmente 0. Con una textura mapeada a *Emit*, se conseguirá un efecto máximo, porque *DVar* es 1 por defecto. Si ajusta *DVar* a 0, ninguna textura tendrá efecto.
 - Si desea un material transparente, y usa una textura mapeada a *Alpha*, no ocurre nada con los ajustes por defecto, porque el valor *Alpha* en el panel *Material* es 1. Así, debe ajustar *DVar* a 0 para conseguir material transparente (y por supuesto también *ZTransp*). Este es un problema común en los principiantes.
- *Mix*: El Modo de Mezcla de la textura. Funciona de forma similar a los Modos de Capas en Gimp o Photoshop.
- *Col*: El grado con el que la textura afecta al color.
- *Nor*: El grado con el que la textura afecta a la normal. Afecta a los mapas Normal, Bump y Displacement.
- *Var*: El grado con el que la textura afecta a los otros valores.
- *Disp*: El grado con el que una textura cambia el Desplazamiento. Vea la sección Mapas de Desplazamiento.
- *Warp* y *fac* cambian las texturas subsiguientes (*Warp*).

Valores de entrada y resultados



El archivo de entrada. Vertical: Negro, Blanco, mezcla Blanco/Negro, 50% Gris, RGB, mezcla Negro/Alfa. La barra horizontal más clara tiene una transparencia del 50%.

Se ha usado *el archivo de entrada* para demostrar el impacto de diferentes valores de entrada. El color del plano subyacente es magenta (R=1,G=0,B=1). *Map To Col*, color *Map To* de la textura, es amarillo (R=1, G=1, B=0).



Use Alpha (usar Alfa).

Normalmente el color de la textura es opaco. Si se usa Alfa (*Use Alpha*) con la textura imagen, se evalúa la información alfa de la imagen. Ello no hará transparente al material, ¡sólo a la textura! Entonces los píxels muestran el color subyacente (*Use Alpha*).



No RGB.

No RGB muestra el resultado del ajuste correspondiente. La textura RGB se usa como valor de Intensidad. El Blanco produce el color "Map To".



No RGB y Neg.

Neg invierte los valores respectivos.

Texturas Múltiples

Básicamente cada textura es opaca a las texturas sobre ella. Ello no es un problema, si una textura está por ejemplo mapeada al color, la otra mapeada a alfa, etc.

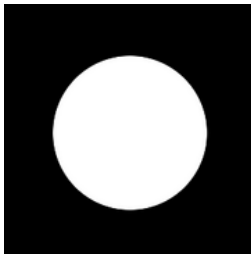
¿Pero cómo confinar una textura a una cierta área, o crear una suave transición entre texturas?

Si quiere darle a una Malla múltiples materiales, vea en lugar de esto, Materiales Múltiples.

Stencil

El modo *Stencil* (máscara) funciona de forma similar a las máscaras de capa en un programa 2D. El efecto de una textura stencil no puede ser atenuado, sólo extendido. Se necesita un mapa de intensidad como entrada.

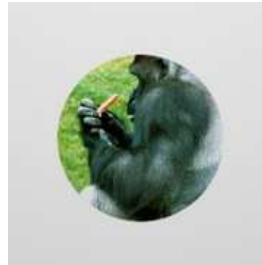
Veamos un ejemplo.



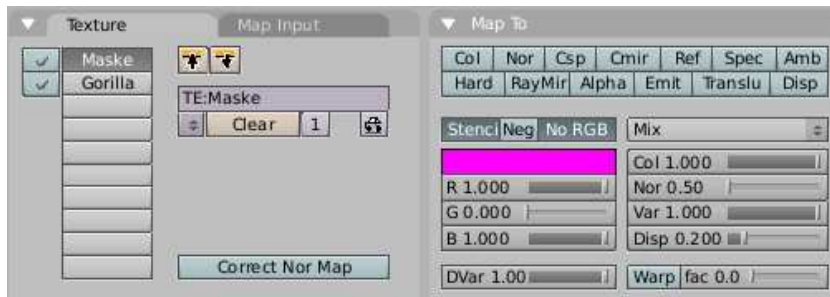
La máscara



La textura



El resultado



Ajustes de Textura

Donde la máscara es negra, las siguientes texturas no tienen efecto. Stencil necesita intensidad como entrada, por lo que debemos usar *No RGB* si queremos utilizar tanto imágenes sin canal Alfa o una textura con banda de color (como las Texturas Blend con banda de color).



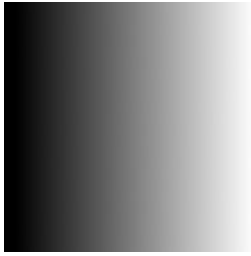
Mapa Stencil con mezcla radial.

Pueden mezclarse dos texturas si se usa una textura blend como mapa stencil (*Mapa Stencil con mezcla radial*).

Warp

La opción *Warp* permite a las texturas influenciar/distorsionar las coordenadas de textura de un canal de textura subsiguiente. El deslizador *fac* define la cantidad de influencia.

La distorsión permanece activa sobre todos los canales subsiguientes, hasta que se establece un nuevo *Warp*. Estableciendo *fac* a cero cancela el efecto.



La Textura Blend



La Textura sobre la que se aplica Warp

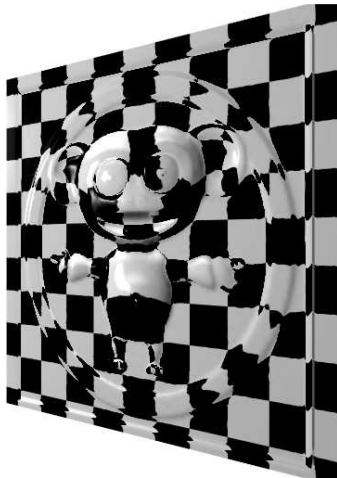


El resultado con Warp




La misma textura para mapeado *Normal* y *Warp*.

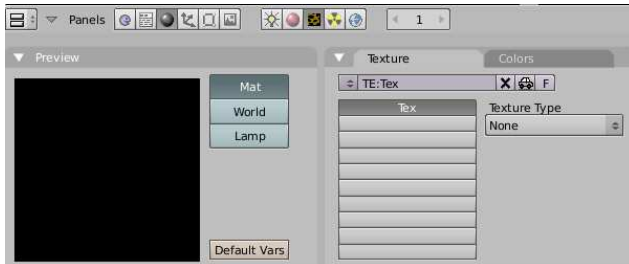
Un ejemplo más interesante extraído de las notas de versión de la 2.36 (*Cornelius como Factor Warp*). El mapa normal de Cornelius (*La misma Textura para mapeado Normal y Warp*) se usó como mapa normal así como textura warp en el canal 1. La textura de cuadros de ajedrez se usa en el canal 2.



Cornelius como Factor Warp.

Botones de Textura

Una vez una textura nueva ha sido añadida a un material, puede ser definida cambiando a Botones de Textura (F6) o al subcontexto  del contexto Sombreado para obtener *Botones de Textura*.



Texture buttons

Una Ventana de Botones de textura nueva y vacía presenta dos paneles:

1. *Preview* (previsualización) de textura
2. *Texture* (textura), con dos pestañas.

En los botones conmutadores del panel *Preview* se define si es un Material, Luz o textura de Mundo, y un botón *Default Var* permite devolver los parámetros de la textura a los valores por defecto.

La pestaña *Texture* replica los canales de textura y el Botón Menú de Textura del Material enlazado. Con la Caja Desplegable podemos seleccionar el *Tipo de Textura*.



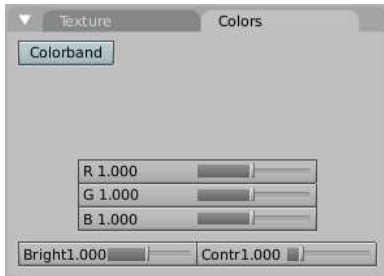
Tipos de Textura.

En cuanto se ha escogido un tipo de textura, aparece un nuevo Panel, con un nombre coincidente con el tipo de textura, donde pueden establecerse los parámetros de la textura.

- *Image*: (imagen) Permite cargar una imagen y usarla como textura. Vea Texturas Imagen.
- *EnvMap*: Para simular Reflexiones (y Refracciones) sin Trazado de Rayos. Vea Mapas de Ambiente.

- *Plugin*: Permite cargar piezas externas de código para definir la textura. Vea Plugins de Textura.
- El resto de botones definen texturas 3D procedurales, que son texturas definidas matemáticamente. Vea Texturas Procedurales.

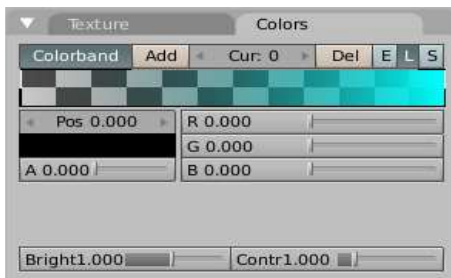
Colores y Banda de Colores



Panel de Textura *Colors* .

Todas las texturas pueden modificarse mediante los botones *Bright* (brillo) y *Contr* en el Panel *Colors* (colores). Todas las texturas cuyos resultados son valores RGB -incluyendo Imágenes y Mapas de Entorno- pueden ser modificadas mediante los deslizadores RGB (*Panel de Textura Colors*).

Si se usan texturas con sólo intensidad el resultado es una textura en blanco y negro, que puede ser ampliamente mejorada por el uso de bandas de color. La banda de color es una herramienta frecuentemente olvidada en la pestaña *Colors* en el Panel *Texture* que proporciona un impresionante nivel de control sobre cómo deben renderizarse las texturas procedurales. En lugar de renderizar simplemente cada textura como una progresión lineal de 0.0 a 1.0, puede usarse la banda de color para crear un gradiente que evoluciona a través de tantas variaciones de color y transparencia (alfa) como se desee (*Banda de color de Textura*).



Banda de color de Textura.

Para usar Bandas de color, seleccione una textura procedural, tal como *Wood* (madera). Haga clic sobre el botón *Colorband* (banda de color). *Colorband* es el editor de gradientes de Blender. Cada punto en la banda puede ser colocado en cualquier posición, y se le puede asignar cualquier color y transparencia. Blender interpola los valores de un punto al siguiente.

Seleccione el punto que quiere editar con el Botón Numérico *Cur*: o haciendo clic izquierdo, luego añada y elimine puntos con los botones *Add* (añadir) y *Del* (borrar). Los valores RGB y Alfa del color actual son mostrados, junto con la posición del punto en la banda. Arrastre con el ratón izquierdo para cambiar la colocación del punto actual.

Texturas Procedurales



La lista *Texture Type* (tipo de textura) en el Panel *Texture* de los Botones de Textura (F6).

Las texturas procedurales son texturas definidas matemáticamente. En general son relativamente fáciles de utilizar, puesto que no necesitan ser mapeadas de ninguna forma específica -lo que no significa que las texturas procedurales no puedan llegar a ser muy complejas.

Este tipo de texturas son 3D 'real'. Con ello queremos decir que encajan perfectamente en las aristas y continúan apareciendo del mismo modo para el que han sido pensadas, incluso cuando se las corta; como si un bloque de madera hubiera sido realmente cortado en dos. Las texturas procedurales no son filtradas ni se les aplica antialias. Eso difícilmente representa un problema: el usuario puede mantener fácilmente las frecuencias especificadas dentro de límites aceptables.

Las texturas Procedurales pueden producir ya sea texturas coloreadas, texturas de sólo intensidad, texturas con valores alfa y texturas de normal. Si se usan las de sólo intensidad, el resultado es una textura en blanco y negro, lo que puede ser ampliamente mejorado por el uso de bandas de color. Por otro lado, si se usan las bandas de color y es necesario tener un valor de intensidad, debe activarse *No RGB* en el panel *Map To*.

Casi todas las texturas procedurales en Blender usan derivadas para calcular las normales para el mapeado de las texturas (a excepción de *Blend* y *Magic*). Esto es importante para los Mapas de Desplazamiento y de Normales. La fuerza del efecto es controlada con el Botón Numérico *Nabla*.

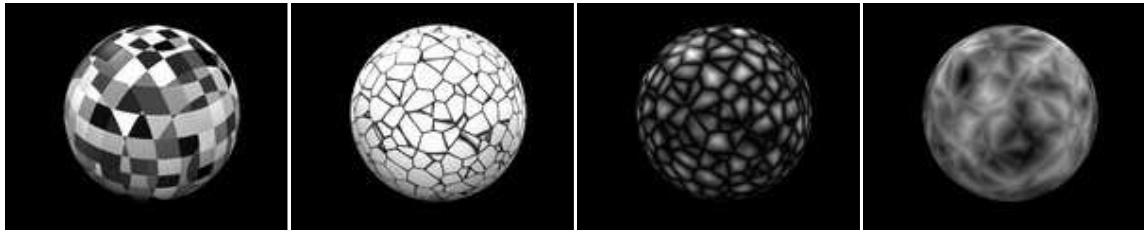
Las texturas no procedurales están remarcadas en gris en *La lista Texture Type*.

Todos los ejemplos a continuación están creados usando varias texturas, en cual caso se ha realizado un aspecto importante con la procedural respectiva.

Noise Basis (ruido base)

Cada textura de Blender basada en ruido (con la excepción de Voronoi y Noise simple) tiene un parámetro *Noise Basis* que permite al usuario seleccionar qué algoritmo debe usarse para generar la textura. Esta lista incluye el algoritmo Blender original. Para dar una idea de los

diferentes tipos, se lista a continuación una imagen de cada tipo como base para una textura *Clouds*. *Noise Basis* gobierna la apariencia estructural de la textura.

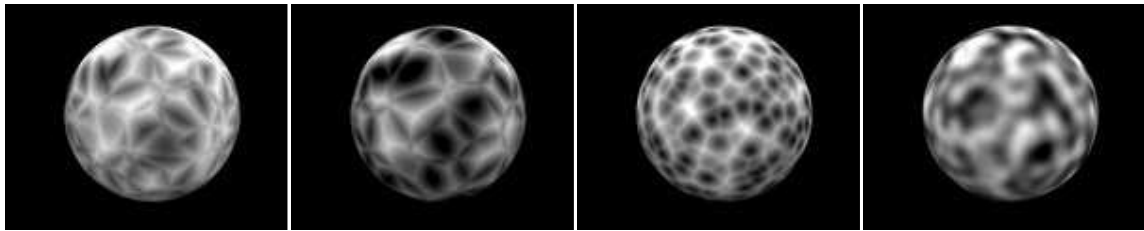


Cellnoise

Voronoi Crackle

Voronoi F2-F1

Voronoi F4



Voronoi F3

Voronoi F2

Voronoi F1

Blender Original

Hay dos posibilidades más para *Noise Basis*, que son relativamente parecidos a *Blender Original*:

- Improved Perlin (Perlin mejorado)
- Original Perlin

Los ajustes *Noise Basis* hacen a las texturas procedurales extremadamente flexibles (especialmente *Musgrave*).

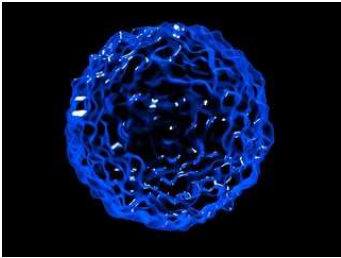
Clouds (nubes)



Paneles de Textura *Clouds*.

- Usado frecuentemente para: Nubes, Fuego, Humo. Encaja bien para ser usado como mapa de Relieve, dando una irregularidad general al material.
- Tipo(s) de Resultado: Intensidad (*Default*) o Color RGB (*Color*).
- Uso:
 - *Default*: El Ruido estándar, da una Intensidad.
 - *Color*: El Ruido da un valor RGB.

- *Soft Noise, Hard Noise:* (ruido suave, ruido duro) Hay dos métodos disponibles para la función Ruido.
 - *NoiseSize:* (tamaño ruido) La dimensión de la tabla de Ruido.
 - *NoiseDepth:* (profundidad ruido) La profundidad del cálculo Cloud. Un número alto resulta en un tiempo de cálculo largo, pero también en detalles más finos.
- Generación: En este caso, se usa una tabla tridimensional con valores pseudo aleatorios, a partir de los cuales un valor de interpolación fluída puede ser calculado para cada coordenada 3D (gracias a Ken Perlin por su magistral artículo "Un Sintetizador de Imagen", de los actos de SIGGRAPH 1985). Este método de cálculo es llamado también Ruido Perlin. Además, cada textura de Blender basada en ruido (con la excepción de Voronoi y Noise simple) tiene un nuevo parámetro "Noise Basis" que permite al usuario seleccionar qué algoritmo se usa para generar la textura.
- Ejemplos:



Una Textura *Clouds* fue usada para desplazar la superficie.

Marble (mármol)



Paneles de Textura *Marble*

- Usado frecuentemente para: Mármol, Fuego, Ruido con una estructura.
- Tipo(s) de Resultado: Valor de Intensidad sólo.
- Uso:
 - *Soft, Sharp, Sharper:* (suave, definido, más definido) Tres preajustes para mármol suave a más claramente definido.
 - *Soft Noise, Hard Noise:* (ruido suave, ruido duro) La función Ruido trabaja con dos métodos.
 - *NoiseSize:* (tamaño ruido) Las dimensiones de la tabla de Ruido.
 - *NoiseDepth:* (profundidad de ruido) La profundidad del cálculo Marble. Un valor más alto resulta en un tiempo mayor de cálculo, pero también en detalles más finos.
 - *Turbulence:* (turbulencia) La turbulencia de las bandas senoidales.

- Generación: En este caso, las bandas son generadas en base a una fórmula senoidal y turbulencia de Ruido.
- Ejemplos:



Mármol de la Librería de Materiales de Blender

Stucci (estuco)



Paneles de Textura *Stucci*

- Usado frecuentemente para: Piedra, Asfalto, Naranjas. Normalmente para Mapeado de Relieve para crear superficies granuladas.
- Tipo(s) de Resultado: Normales.
- Uso:
 - *Plastic*: (plástico) El Stucci estándar.
 - *Wall In*, *Wall out*: (pared adentro, pared afuera) Aquí es donde Stucci recibe su nombre. Es una típica estructura de pared con agujeros o prominencias.
 - *Soft Noise*, *Hard Noise*: (ruido suave, ruido duro) Hay dos métodos disponibles para trabajar con Ruido.
 - *NoiseSize*: (tamaño ruido) La dimensión de la tabla de Ruido.
 - *Turbulence*: (turbulencia) La profundidad de los cálculos Stucci.
- Generación: Basada en Ruido.
- Ejemplos:



Metal oxidado. Se usó *Stucci* para dar un poco de granulosidad a la superficie.

Wood (madera)



Paneles de Textura *Wood*.

- Usado frecuentemente para: Madera.
- Tipo(s) de Resultado: Devuelve únicamente un valor de Intensidad.
- Uso:
 - *Bands*: (bandas) La textura Wood estándar.
 - *Rings*: (anillos) Esto sugiere anillos 'de madera'.
 - *BandNoise*: (ruido de bandas) Aplicar Ruido produce en la textura Wood estándar un cierto grado de turbulencia.
 - *RingNoise*: (ruido de anillos) Aplicar Ruido produce en los anillos un cierto grado de turbulencia.
 - *Soft Noise*, *Hard Noise*: (ruido suave, ruido duro) Hay dos métodos disponibles para la función Ruido.
 - *NoiseSize*: (tamaño ruido) La dimensión de la tabla de Ruido.
 - *Turbulence*: (turbulencia) La turbulencia de los tipos BandNoise y RingNoise.

Sinceramente, no es tan fácil crear madera de apariencia realista mediante texturas procedurales. Es necesario un empty (objeto vacío) como coordenadas de entrada para estirar la textura en la dirección adecuada.

- Generación: En este caso, las bandas son generadas basándose en una fórmula senoidal. Puede añadirse una cierta cantidad de turbulencia mediante la fórmula de Ruido.
- Ejemplos:

Véase la sección Tutorials/Textures/Wood para un método de creación de madera procedural.



"Wenge Wood" por Claas Eike Kuhnen.

Magic (magia)



Paneles de Textura *Magic*.

- Usado frecuentemente para: Esta es difícil; fue complicado encontrarle una aplicación. Podría usarse para "Interferencia de Película Fina", si se ajusta *Map Input* a *Refl* y se usa una *Turbulence* relativamente alta.
- Tipo(s) de Resultado: RGB.
- Uso:
 - *Depth*: (profundidad) La profundidad del cálculo. Un número más alto resulta en un tiempo de cálculo largo, pero también en detalles más finos.
 - *Turbulence*: (turbulencia) La fuerza del diseño.
- Generación: Los componentes RGB son generados independientemente con una fórmula senoidal.
- Ejemplos:



"Interferencia de Película Fina" con Textura *Magic*.

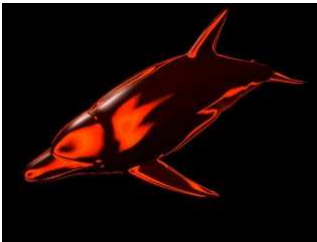
Se han usado dos Texturas *Magic* en "Interferencia de Película Fina" con Textura *Magic*. Ambas usan la misma textura con *Depth* 4, *Turbulence* 12. Ambas tienen *Map Input* establecido en *Refl*. La primera textura es mapeada a *Nor*, la segunda a *Col*.

Blend (mezcla)



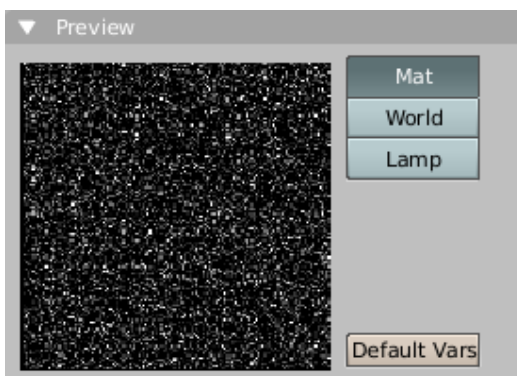
Paneles de Textura *Blend*.

- Usado frecuentemente para: Esta es una de las texturas procedurales más importantes. Las texturas Blend pueden usarse para mezclar otras texturas entre sí (con *Stencil*), o para crear bellos efectos (especialmente con el truco *Map Input: Nor*). Recuerde: si usa una banda de color para crear una mezcla a medida, puede que deba usar *No RGB*, si el valor *Map To* necesita una entrada de intensidad.
- Tipo(s) de Resultado: Intensidad.
- Uso:
 - *Lin*: Una progresión lineal.
 - *Quad*: Una progresión cuadrática.
 - *Ease*: Una progresión no lineal, fluida.
 - *Diag*: Una progresión diagonal.
 - *Sphere*: (esfera) Una progresión con la forma de de una bola tridimensional.
 - *Halo*: Una progresión cuadrática con la forma de de una bola tridimensional.
 - *Flip XY*: (invertir XY) La dirección de la progresión es rotada un cuarto de vuelta.
- Generación: Genera una progresión.
- Ejemplos:



Un blend radial con *Map Input* establecido en *Nor*, y *Map To* a *Ref* y *Emit*.

Noise (ruido)



Vista previa de *Noise*.

- Usado frecuentemente para: Ruido blanco en una animación. No es adecuado si no se quiere una animación. Para rugosidad de material use *Clouds* en su lugar.
- Tipo(s) de resultado: Intensidad.
- Uso: No hay panel ni botón. Simplemente actívelo.
- Generación: Aunque tiene una gran apariencia, no es Ruido Perlin. Es un Ruido verdadero generado aleatoriamente. Da un resultado distinto cada vez, para cada fotograma, para cada píxel.

Musgrave



Paneles de Textura *Musgrave*.

- Usado frecuentemente para: Materiales orgánicos, pero es muy flexible. Puede hacerse prácticamente cualquier cosa con él.
- Tipo(s) de resultado: Intensidad.
- Uso: Esta textura procedural tiene cinco tipos de ruido en los que puede basarse el diseño resultante, y son seleccionables en un menú desplegable en la parte superior del panel. Los cinco tipos son:
 - *fBm*:
 - *Hetero Terrain*:
 - *Hybrid Multifractal*:
 - *Ridged Multifractal*:
 - *Multifractal*:

Estos tipos de ruido determinan el modo en que Blender apila sucesivas copias del mismo diseño uno encima del otro en contrastes y escalas diferentes.

Además de los cinco tipos de ruido, Musgrave tiene un ajuste de ruido base que determina el algoritmo que genera el ruido en sí. Éstas son las mismas opciones de ruido base que se encuentran en las otras texturas procedurales.

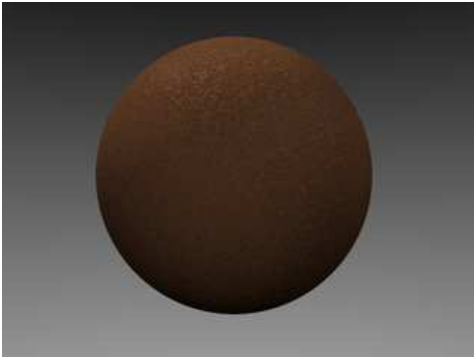
Los tipos de ruido principales tienen cuatro características que pueden establecerse en los botones numéricos debajo de la lista desplegable. Son:

- Dimensión Fractal (botón *H* - Rango 0 a 2): La dimensión fractal controla el contraste de una capa relativo a la capa anterior en la textura. Cuanto más alta es la dimensión, más alto es el contraste entre cada capa, y así mayor detalle se muestra en la textura.
- Lagunaridad (botón *Lacu* - Rango 0 a 6): La lagunaridad controla el escalado de cada capa de la textura Musgrave, de tal modo que cada capa adicional tendrá una escala que es la inversa del valor que se muestra en el botón. Así, si *Lacu*=2 -> Escala=1/2 del original.
- Octavo (*Octs* - Rango 0 a 8): Octavo controla el número de veces que el diseño de ruido original es sobrepuesto sobre sí mismo y escalado/contrastado con la dimensión fractal de los ajustes de lagunaridad.

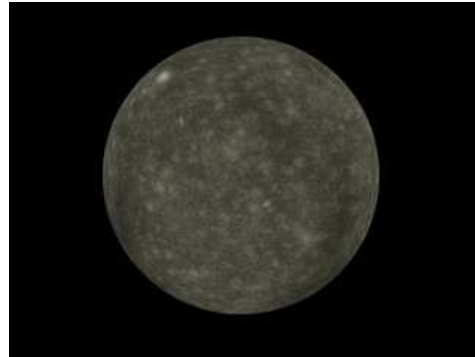
Los tipos *Hybrid Multifractal*, *Ridged Multifractal*, y *Hetero Terrain* tienen ajustes adicionales. Los tres tienen un botón "Fractal Offset" (desplazamiento fractal) etiquetado *Ofst*. Esto sirve como un ajuste de "nivel del mar" e indica la altura base del

mapa de relieve resultante. Los valores de relieve por debajo de este umbral se devolverán como cero. *Hybrid Multifractal* y *Ridged Multifractal* tienen ambos un ajuste *Gain* (ganancia) que determina el rango de valores creados por la función. A más alto el número, más alto el rango. Es un modo rápido de hacer aflorar detalles adicionales en una textura donde los extremos son normalmente recortados.

- Generación: Puede encontrarse más información acerca de estas texturas en la siguiente dirección: Musgrave Documentation
- Ejemplos: Véase la Galería de Ejemplos de las notas de versión para más ejemplos.



Cuero hecho con una única Textura *Musgrave*



Piedra hecha con una combinación de 3 Texturas *Musgrave* diferentes.

Voronoi



Paneles de Textura *Voronoi*.

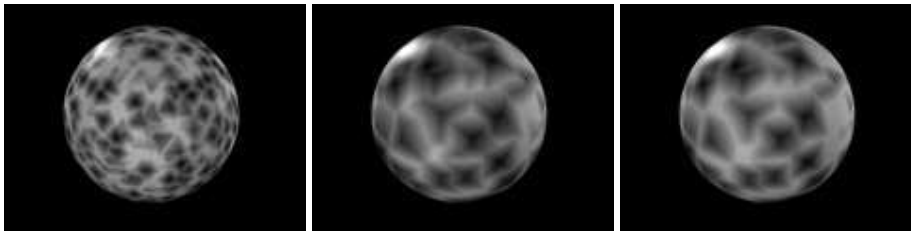
- Usado frecuentemente para: Metal muy convincente, especialmente el efecto "amartillado". Sombreadores orgánicos (como venas en la piel).
- Tipo(s) de Resultado: Intensidad (por defecto), Color.
- Uso: Esta textura procedural tiene siete opciones de Métrica de Distancia. Éstas determinan el algoritmo para encontrar la distancia entre celdas de la textura. Estas opciones son:
 - *Minkovsky*
 - *Minkovsky 4*
 - *Minkovsky 1/2*
 - *Chebychev*

- *Manhattan*
- *Distance Squared* (distancia al cuadrado)
- *Actual Distance* (distancia actual)

El ajuste *Minkovsky* tiene un parámetro configurable (el botón *Exp*) que determina el exponente (e) de la función distancia $(x^e + y^e + z^e)^{1/e}$. Un valor de 1 produce la métrica de distancia Manhattan, un valor inferior a 1, produce estrellas, y valores más altos producen células cuadradas. Así, todos los Ajustes de Distancia son básicamente lo mismo -variaciones de *Minkowsky*.

Pueden conseguirse también células de formas distintas en la textura usando los otros cuatro modos:

- *Chebychev* - células cuadradas de tamaño irregular.
- *Manhattan* - células en forma de diamante posicionadas aleatoriamente.
- *Actual Distance/Distance Squared* - células redondeadas de forma irregular



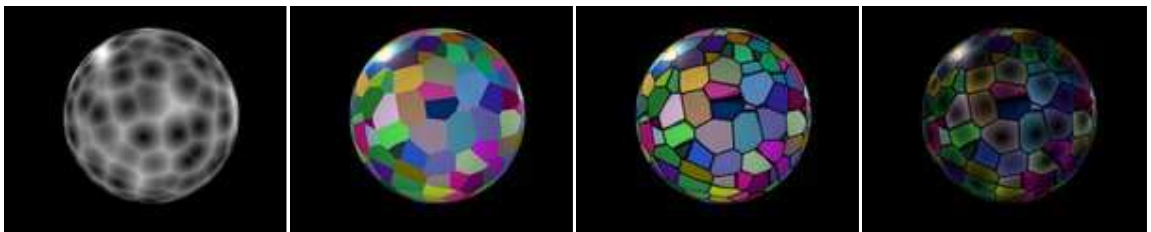
Manhattan

Cebychev

Minkowsky 10

Cuatro deslizadores en la parte inferior del panel Voronoi representan los valores de las cuatro constantes Worley (explicadas un poco en la Documentación Worley), que se usan para calcular las distancias entra cada célula en la textura basadas en la métrica de distancia. Ajustar estos valores puede tener algunos efectos interesantes en el resultado final. Vea la Galería de Ejemplos con algunos ejemplos de estos ajustes y las texturas que producen.

En la parte superior del panel hay cuatro botones de variación que usan cuatro ruidos base diferentes como métodos de calcular el color y la intensidad de la textura de salida. Esto da a la textura Voronoi creada con los "Deslizadores Worley" una apariencia completamente diferente, y es equivalente al ajuste de ruido base que se encuentra en las otras texturas.



Int de Ruido Base.

Col1 de Ruido Base.

Col2 de Ruido Base.

Col3 de Ruido Base.

- Generación: Para una descripción más profunda del algoritmo Worley, vea: [Worley Documentation](#).
- Ejemplos: Vea la Galería de Ejemplos de las notas de versión para más ejemplos.

Distorted Noise (ruido distorsionado)



Paneles de Textura de *Distorted Noise*.

- Usado frecuentemente para: Muy complejo y versátil.
- Tipo(s) de Resultado: Intensidad.
- Uso: Esta textura tiene dos opciones entre las que elegir: *Distortion Noise* (ruido de distorsión), y *Noise Basis* (ruido base). Ambas permiten elegir entre la lista de tipos de ruido de Blender. Tenemos también las opciones en *Distorted Noise* para trabajar con *Noise Size* (tamaño de ruido) y *Distortion Amount* (cantidad de distorsión). Esto permite ajustar el tamaño del ruido generado y la cantidad de *Distortion Noise* afecta a *Noise Basis*.
- Generación: *Distortion Noise* toma la opción que se escoge en *Noise Basis* y lo filtra, para crear un diseño híbrido.
- Ejemplos: Vea la Galería de Ejemplos de las notas de versión para más ejemplos.

Texturas Imagen




El Panel *Image* en los Botones de Textura (F6).

Si selecciona el Tipo de Textura *Image* (imagen) en el Panel *Texture*, aparece el Panel *Image* (imagen). Al hacer clic sobre el botón *Load Image* (cargar imagen), puede cargar un archivo de imagen que esté en uno de los formatos de archivo soportados por Blender.

Los formatos de archivo soportados son BMP, JPG, PNG y TGA. Otros -como PSD- son soportados en parte. Para animaciones puede utilizar secuencias de imagen numeradas;

también se aceptan AVIs sin comprimir o comprimidos en modo JPG. En plataformas Windows todos los vídeos con un codec soportado deberían funcionar.

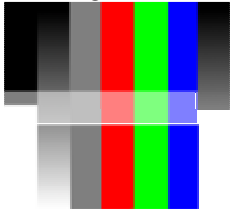
Pueden usarse rutas absolutas y también relativas. La doble barra (//) significa el directorio de trabajo; dos puntos (..) apuntan al directorio padre.

El Botón  guarda la imagen dentro del archivo blend; *Reload* (recarga) carga desde el disco una versión de la imagen que haya podido cambiar.

Varias Opciones de Image




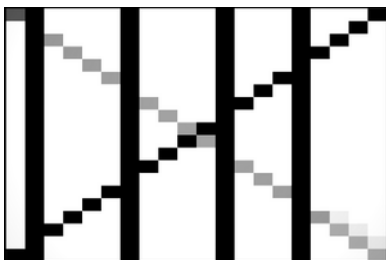
La imagen de fondo.



La imagen en primer plano.

Usaremos dos texturas diferentes para demostrar las distintas opciones de Imagen. *La imagen de fondo* es un archivo JPG convencional; *La imagen en primer plano* es un archivo PNG con varios valores Alfa y de tonos de gris. La barra vertical en el lado derecho de la imagen en primer plano es un degradado Alfa, la barra vertical tiene un Alfa del 50%.

- *InterPol*: Esta opción interpola los píxels de una Imagen. Ello se hace visible al agrandarla. Desactive esta opción para mantener los píxels visibles -se les aplica antialias correctamente. Esta última característica es útil para diseños regulares, tal como líneas y títulos; permanecen 'enfocados' incluso al agrandarlos considerablemente. Cuando agranda esta Imagen de 10x10 píxels , la diferencia sin y con *InterPol* es claramente visible (*Textura Imagen agrandada*).



Textura Imagen agrandada sin *InterPol*.



Textura Imagen agrandada con *InterPol*.

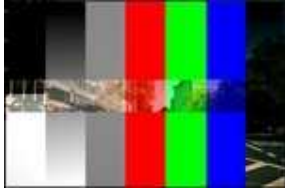


Imagen en primer plano con *UseAlpha*. Se evalúan los valores alfa de los píxels.

- *UseAlpha*: Funciona con archivos PNG y TGA puesto que pueden albergar información de transparencia (*Imagen en primer plano con UseAlpha*).



Imagen en primer plano con *CalcAlpha*.

- *CalcAlpha*: Calcula un alfa basado en los valores RGB de la Imagen. Negro (0,0,0) es transparente, blanco (1,1,1) opaco (*Imagen en primer plano con CalcAlpha*).
- *NegAlpha*: Invierte el valor alfa.
- *MipMap*: Los MipMaps son precalculados: texturas más pequeñas filtradas, para un cierto tamaño. Se genera una serie de imágenes, cada una de la mitad de tamaño que la anterior. Esto optimiza el proceso de filtrado. Cuando esta opción está desactivada, se obtiene generalmente una imagen más enfocada, pero el tiempo de cálculo puede alargarse considerablemente cuando la dimensión del filtro es grande. Sin MipMaps pueden producirse imágenes distintas desde ángulos de cámara ligeramente diferentes, cuando la Textura llega a ser muy pequeña. Ello puede hacerse muy obvio en una animación.
- *Fields*: (campos) Los fotogramas de vídeo consisten en dos imágenes diferentes (campos) que son mezclados por líneas horizontales. Esta opción hace posible el trabajo con campos de imagen. Asegura que cuando se renderizan los Campos, se use el campo correcto de la Imagen en el campo correcto del render. El MipMapping no puede combinarse con *Fields*.
- *Rot90*: Rota la Imagen 90 grados al revés de las agujas del reloj al renderizar.
- *Movie*: (película) Los archivos de Vídeo (AVIs soportados por Blender, SGI) y archivos "anim5" pueden usarse también como textura Imagen. En este caso, el subsiguiente Panel, *Anim and Movie* (animación y película) es poblado con Botones.
- *Anti*: A las imágenes de gráficos tales como cómics e imágenes que consisten únicamente en unos pocos colores con grandes superficies rellenas, puede aplicárseles antialias como preproceso incorporado. Ello no es útil para fotografías y similares. Necesita OSA.

- *StField*: Normalmente, el primer campo en un fotograma de vídeo empieza con la primera línea. Algunos dispositivos de vídeo lo hacen de forma diferente. En ese caso, es necesaria la opción *StField*.
- *Filter*: (filtro) El tamaño del filtro usado por las opciones *MipMap* y *Interpol*.
- *Normal Map*: (mapa de normales) Cuando la textura debería usarse como "Mapa de Normales". Necesita imágenes de entrada especialmente preparadas. Vea la sección Mapas de Relieve y de Normales.

Escalado y Repetición

Una Textura Image es normalmente puesta sobre un objeto de tal modo que cubre al objeto en su totalidad. Con los Botones Numéricos *Min* y *Max* puede establecer el desplazamiento y tamaño de la textura en relación al objeto. Ahora debe decidirse qué hacer con los píxeles que no quedan cubiertos.

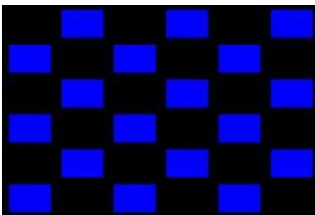


Extend. Fuera de la imagen el color del borde es extendido.

- *Extend*: (extender) Fuera de la imagen el color del borde es extendido (*Extend*). La Imagen fue puesta en el centro del objeto con los ajustes mostrados.

Extend	Clip	ClipCube	Repeat	Checker
MinX -0.600		MaxX 1.600		
MinY -0.600		MaxY 1.600		

- *Clip*: (cortar) Fuera de la Imagen, se devuelve un valor alfa de 0.0. Ello permite 'pegar' un pequeño logo sobre un objeto grande.
- *ClipCube*: (cortar cubo) Lo mismo que *Clip*, pero ahora la coordenada 'Z' se calcula también. Fuera de un área cúbica alrededor de la Imagen, un valor 0.0 es devuelto.
- *Repeat*: (repetir) La Imagen es repetida horizontal y verticalmente con la frecuencia expresada en *Xrepeat* y *Yrepeat*.



Checker genera diseños de tablero de ajedrez. Aquí se usó una imagen azul sobre un fondo negro.

- *Checker*: (ajedrez) Tablas de ajedrez hechas rápidamente. *Mortar* (mortero) gobierna la distancia entre los campos en partes del tamaño de textura.



(Puede usar también la opción *size* (tamaño) en el Panel *Map Input* para crear el número de campos deseado).

Texturas Image animadas

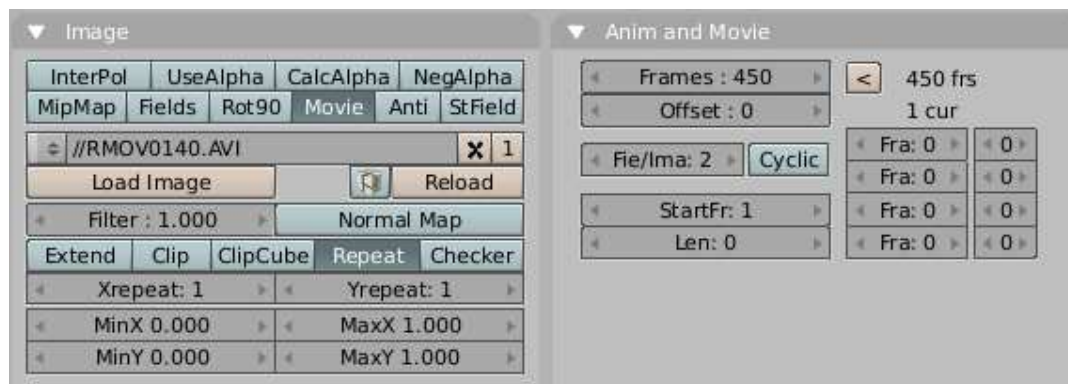
Básicamente pueden animarse diversos atributos de material con animaciones IPO, como los canales de textura. O puede animarse el desplazamiento de la textura (*ofsX/Y/Z*) en el Panel *Map Input*. Puede también usarse otro objeto como fuente de coordenadas de entrada de la textura y animar este objeto.

Pero aquí hablaremos sólo de texturas animadas por sí mismas, como vídeos y secuencias de imágenes.

Vídeos como Texturas

El modo más simple de conseguir una textura animada es usando un archivo de vídeo. El vídeo necesita el mismo número -o una división entera- de fotogramas por segundo (fps) que la animación para correr a la misma velocidad.

Puede usar AVI raso, AVI JPEG y archivos Anim (Formato IFF). En los sistemas operativos Windows todos los AVI cuyos codecs correspondientes estén instalados deberían funcionar.



Un archivo AVI como Textura Imagen.

El número *frs* en el Panel *Anim and Movie* (animación y película) muestra cuántos fotogramas han sido reconocidos. Puede copiarse este número al campo *Frames* (fotogramas) con el botón de flecha. El número *cur* muestra qué fotograma se visualiza en el Panel *Preview* (vista previa).

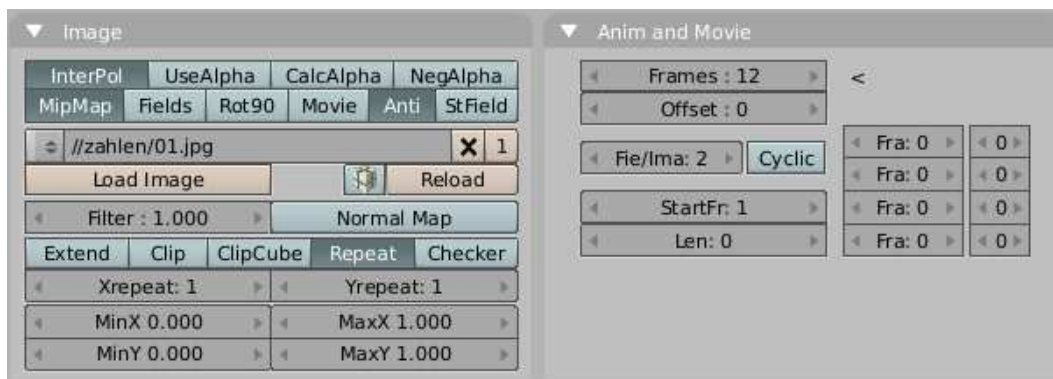
- *Frames*: (fotogramas) Esto activa la opción de animación; otro archivo de imagen (en el mismo bloque Imagen) será leído por cada fotograma renderizado (véase también

Fie/Ima). El número en el campo *Frames* es el número de fotogramas que será usado en la animación. El último fotograma se usará como textura estática para el resto de la animación si no se activa la opción *Cyclic*.

- *Offset*: (desplazamiento) El número de la primera imagen de la animación. El fotograma final se calcula como $Frames + Offset$.
- *Fie/Ima* = "Fields per Image:" (campos por imagen) El número de campos por fotograma renderizado. Si no se renderizan campos, deben introducirse aquí números pares (2 campos = 1 fotograma). Esto establece la velocidad de la animación. Los ajustes correctos dependen de la tasa de fotogramas del vídeo textura, de la tasa de fotogramas de la animación renderizada, de la renderización o no de Campos (en el Panel *Render* del contexto *Scene*), y de si el vídeo textura usa Campos (Botón *Fields* en el Panel *Image* de los Botones de Textura). Algunos ejemplos:
 - El vídeo tiene 24 FpS, la animación tendrá 24 FpS. Estamos renderizando sin *Fields*. Ajuste *Fie/Ima* a 2.
 - El vídeo tiene 12 FpS, la animación tendrá 24 FpS. Estamos renderizando sin *Fields*. Ajuste *Fie/Ima* a 4.
 - El vídeo tiene 16 Fotogramas, la animación tendrá 96 Fotogramas. Estamos renderizando sin *Fields*. Ajuste *Fie/Ima* a 6.
 - El vídeo tiene 24 FpS, la animación tendrá 24 FpS. Estamos renderizando con *Fields* y estamos usando *Fields* en el Panel *Image*. Ajuste *Fie/Ima* a 1.
- *Cyclic*: (cíclico) La Imagen de animación es repetida cíclicamente.
- *StartFr*: (primer fotograma) El momento -en fotogramas Blender- en el cual la Imagen de animación debe empezar. Hasta ese momento se usa la primera Imagen del vídeo como textura.
- *Len*: Este botón determina la longitud de la animación. Un *Len* de 0 significa que la longitud es igual a *Frames*. Asignando a *Len* un valor superior a *Frames*, puede crearse una foto fija al final de la animación cuando se usa *Cyclic*.
- *Fra*: Los botones *Fra* permiten un montaje simple dentro de una Imagen animada. El botón izquierdo, *Fra*, indica el número de fotograma; el botón de la derecha indica por cuánto tiempo debe mostrarse el fotograma (Modo Tartamudeo). A continuación hay un ejemplo. Si usa *Fra* debe ajustar *Frames* y *Len* adecuadamente.

Secuencia numerada de Imágenes como Textura

En lugar de un archivo de vídeo puede usarse también una secuencia numerada de imágenes. El procedimiento más simple será guardar las imágenes en un subdirectorio y cargar una de las imágenes del subdirectorio.



Archivos de Imagen numerados como Textura

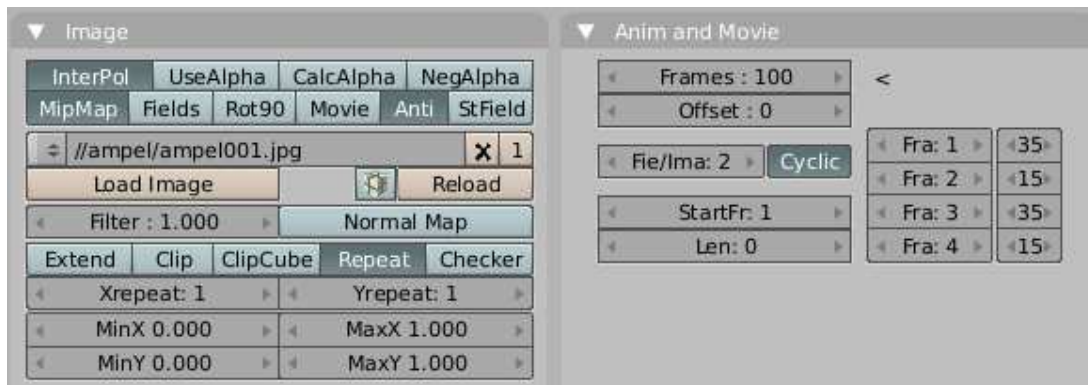
En este ejemplo se han usado 12 archivos de Imagen (01.jpg a 12.jpg). La entrada en el campo *Frames* activa la animación. Ahora Blender trata de encontrar los fotogramas siguientes cambiando un número en el nombre de archivo. ¡No puede usarse también la opción *Movie*!

Números: Debe tener en cuenta algunas cosas para mantener a Blender contando:

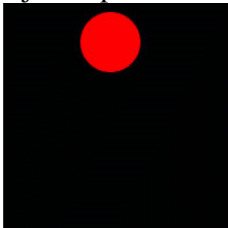
1. Blender intenta encontrar los otros archivos cambiando un número en el nombre de archivo. Sólo el último dígito se interpreta para tal fin. Por ejemplo: 01.ima.099.tga + 1 será 01.ima.100.tga.
2. Los números deben tener la misma longitud. Ponga ceros a la izquierda. Blender cuenta de 1 a 12, pero no hacia atrás hasta 1. En su lugar empieza en 10. Por lo tanto use 01, 02 ...

Todo lo demás funciona igual que con las texturas de vídeo. He aquí un ejemplo de la opción *Fra*.

Asumamos que queremos crear un semáforo animado. Primero creamos cuatro imágenes diferentes: Rojo, Rojo/Ámbar, Verde, Ámbar. Estas cuatro imágenes cambiarán continuamente.



Ajustes para la animación del semáforo con cuatro imágenes. Idea por "tordat" [1]



El semáforo con los ajustes de la figura *Ajustes para la animación*.

Note que son necesarios 100 Fotogramas, aunque sólo se usan cuatro imágenes. El resultado no es sorprendente (*El semáforo*).

Mapeado UV

Introducción

El mapeado UV es una manera de mapear texturas de tipo Imagen sobre modelos tridimensionales. Se puede usar para aplicar texturas a formas arbitrarias y complejas como cabezas humanas o animales. A menudo estas texturas son imágenes pintadas o dibujadas, creadas con programas como Gimp, Photoshop, o cualquiera que sea tu programa de dibujo favorito. Las texturas Procedurales, como hemos visto en los capítulos anteriores, constituyen una buena manera de texturizar un modelo. Lo que es realmente bueno de este tipo de texturas es que siempre "calzan", se ajustan perfectamente al modelo. Las imágenes 2D en cambio, no siempre se ajustarán tan bien. Crear texturas procedurales es relativamente sencillo, y ofrecen una manera rápida de obtener buenos resultados. Sin embargo, hay situaciones en las que este tipo de texturas no es suficiente. Por ejemplo, la piel de una cabeza humana nunca se verá lo suficientemente bien cuando es generada proceduralmente. Las arrugas en un rostro humano, o las saltaduras de pintura en un automóvil no aparecen en lugares al azar sino que dependen de la forma del modelo y de su uso. Las imágenes pintadas manualmente le dan al artista control total sobre el resultado final. En lugar de andar jugando con deslizadores numéricos, los artistas pueden controlar cada píxel sobre la superficie. Esto comúnmente significa más trabajo, pero los resultados valdrán la pena. Un mapa UV es una forma de asignar una parte de una imagen a un polígono en el modelo. Cada vértice del polígono es asignado a un par de coordenadas 2D que definen qué parte de la imagen es mapeada. Estas coordenadas 2D se llaman UVs (comparar con las coordenadas XYZ en 3D). La operación de crear estos mapas UV se conoce también como "despliegue" ("unwrap" en inglés), debido a que todo ocurre como si la malla fuera desenvuelta o desplegada sobre un plano 2D.

A Tip: El mapeado UV es también fundamental en el motor de

juego de Blender como en cualquier otro juego tridimensional existente. Es de hecho el estándar para la aplicación de texturas sobre modelos; casi todo modelo que encuentre en un juego se hallará mapeado UV.

El Editor UV

El mapeado UV se hace en Blender a través de la ventana del Editor UV y de un modo especial de la Ventana 3D llamado el Modo Selección de Caras (Face Select). El Editor UV nos permite mapear las texturas directamente sobre las caras de las mallas. Cada cara puede tener coordenadas de textura individuales y una imagen distinta asignada a la misma, puede combinarse con colores de vértices para hacerla más brillante o más oscura o incluso darle color. Por medio del editor UV, a cada cara se le asignan dos propiedades adicionales:

- *cuatro coordenadas UV*

Estas coordenadas definen la forma en que se mapean las imágenes sobre la cara. Son coordenadas bidimensionales, por lo que se las llama UV, para distinguirlas de las coordenadas XYZ. Estas coordenadas pueden utilizarse para renderizado o para uso en la pantalla OpenGL en tiempo real.

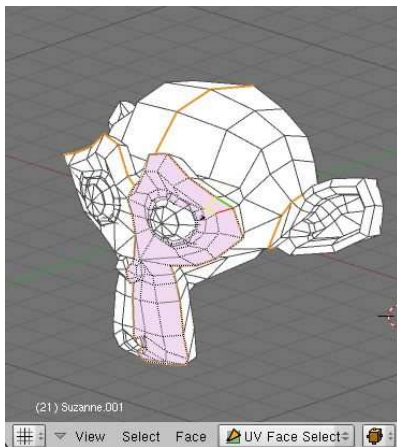
- *un enlace a una Imagen*

Cada cara en Blender puede tener un enlace a una imagen distinta. Las coordenadas UV definen como se mapea esta imagen sobre la cara. La imagen puede renderizarse o ser mostrada en tiempo real. Para poder asignar Imagenes o modificar las coordenadas UV del Objeto Malla activo, la Ventana3D debe estar en el Modo "Selección de Caras".

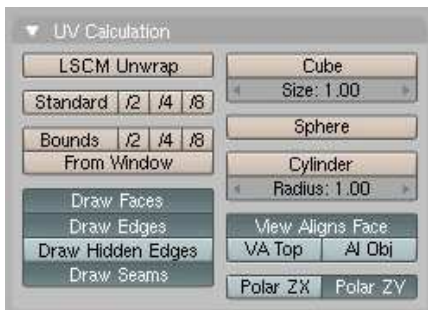


Entrando al Modo Seleccion de Caras.

Agregue a la escena un Objeto Malla, luego ingrese en el Modo Selección de Caras eligiendo la opción "UV Face Select" del menú Modo. La malla será visualizada en modo Z-Buffer. Si entra en el modo de visualizacion Texturizado (**ALT-Z**, tambien llamado "modo papa") verá la malla en color blanco, lo que indica que actualmente no hay ninguna imagen asignada a sus caras. Puede controlar la manera en que se representan las caras usando los botones Draw Edges (Dibujar Aristas) y Draw Faces (Dibujar Caras) en el Panel *UV Calculation* . Si se activa Draw Edges, todas las caras se mostrarán con un delineado. Con Draw Faces activado, todas las caras seleccionadas se visualizarán en un tono rosado claro (o del color del Tema).

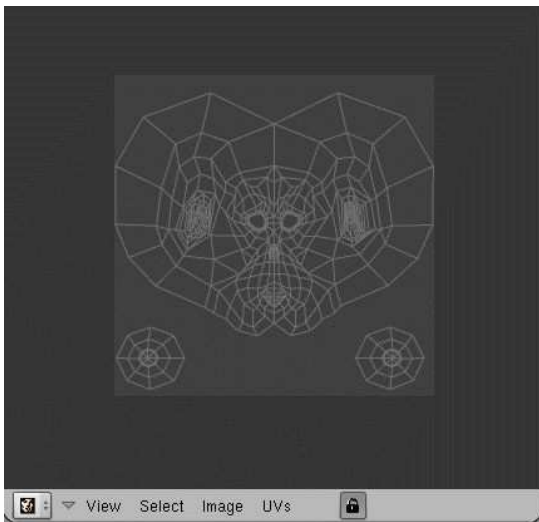


Modo Selección de Caras.



Panel Cálculo de UV.

Presione **A** y todas las caras de la Malla serán seleccionadas y resaltadas con líneas punteadas. En la Ventana 3D puede seleccionar caras con **RMB**, o la Selección por Área (**B**). Si tiene dificultad para seleccionar las caras deseadas, puede ingresar en Modo Edición y seleccionar los vértices que quiera. Luego, al salir de Modo Edición, las caras definidas por los vértices seleccionados deberían también permanecer seleccionadas. Solamente una cara es la activa. O en otras palabras: La Ventana de Imagen, solamente mostrará la imagen de la cara activa. Como se acostumbra en Blender, solamente la última cara seleccionada es la activa y la selección se realiza con **RMB**. Cambie una ventana a la Ventana de Imagen/Editor UV con **SHIFT-F10**. Aquí puede cargar o buscar una imagen con el botón *Load*. Las imágenes deben tener una resolución múltiplo de 64 píxeles (64x64, 128x64 etc.) para poder ser representadas en tiempo real (nota: la mayoría de las tarjetas 3D no soportan imágenes más grandes que 256x256 píxeles). Sin embargo, Blender puede renderizar todas las imágenes asignadas sin importar su tamaño cuando se crean imágenes estáticas o animaciones.



El Editor UV.

Cargar o buscar una imagen en Selección de Caras, asigna automáticamente la imagen a las caras seleccionadas. Esto puede observarse inmediatamente en la Ventana 3D (cuando se encuentra en Modo Vista Texturizada).

Herramientas de Desplegado (Unwrapping)

En la Ventana 3D estando en Modo Selección de Caras, puede presionar **U** para hacer aparecer un menú para calcular las coordenadas UV de las caras seleccionadas. También puede realizar este despliegue utilizando el Panel de Cálculo UV en los Botones de Edición. Dicho panel nos ofrece un mayor control sobre el proceso de despliegue.



UV pre-sets.

Los algoritmos de Despliegue UV disponibles son:

- *Cube (Cúbico)* Determina mapeado cúbico.
- *Cylinder (Cilindrico), Sphere*

(Esférico) *Mapeado ciíndrico/esférico, calculado* desde el centro de las caras seleccionadas.

- *Bounds to 1/8, 1/4, 1/2, 1/1* Las

coordenadas UV se calculan utilizando la proyección de lo mostrado en la Ventana3D, y luego escalado a la fracción de la textura imagen indicada.

- *Standard 1/8, 1/4, 1/2, 1/1* Cada

cara obtiene un conjunto de coordenadas UV cuadradas, que luego son escaladas la fraccion requerida de la textura imagen..

- *From Window (Desde Ventana)* Las

coordenadas UV se calculan utilizando la proyección de lo que se muestra en la Ventana3D.

- *LSCM* Las coordenadas se calculan

empleando el algoritmo Mapa Conforme a Cuadrados Mínimos (Least Squares Conforming Maps). Uselo conjuntamente con las costuras marcadas. En el panel de mapeado UV, puede ajustar la forma en que se realizia el mapeado, y cómo se lo muestra en la Ventana 3D cuando el modelo esté en Modo Seleccion de Caras. Con *View Aligns Face* habilitado, el despliegue Cilíndrico y Esférico es afectado por la vista actual. La vista es considerada como el frente del Cilindro/Esfera, con las tapas arriba y hacia el fodo de la Vista. El cilindor/esfera es *cortado* del lado opuesto de la vista. *Size* y *Radius* definen el escalado del mapa cuando se usan los mapeados Cúbicos y Esférico/Cilindirco repsectivamente. Con el botón *VA Top* (View Aligns Top) habilitado, la Vista debe verse como a través del Cilindro/Esfera. Es cortada en parte superior de la vista. Con esto activado tambien puede definir como se rotará la vista respecto de los polos utilizando las opciones *Polar ZX* y *Polar ZY*. Si se habilita *Al Obj*, el Cilindro/Esfera se gira basado en el giro del Objeto. Los botones *Draw Edges* y *Draw Faces* activan la visualización de aristas u caras en la Ventana3D cuando se está en Modo Selección

de Caras. Las caras seleccionadas en este modo se ven en color púrpura transparente (o el color indicado en el tema), similarmente a ModoEdición. La visualización de costuras en ModoEdición se puede encender y apagar con el botón *Draw Seams*. Los colores de las costuras también pueden modificarse en las opciones del Tema.

Editando las coordenadas UV

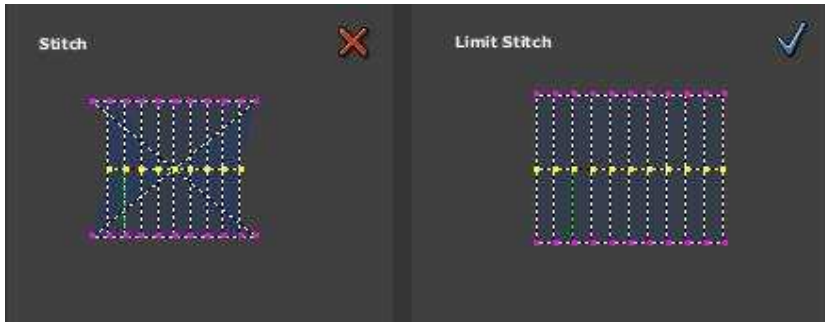
Dentro del Editor UV verá una representación de las caras seleccionadas en la forma de vertices amarillos o rosados, conectados por líneas punteadas. Pueden usarse aquí las mismas técnicas que en ModoEdición de Mallas para seleccionar, mover, rotar, escalar y así sucesivamente. Si pulsa el botón *Lock* puede obtener una vista en 3D en tiempo real de los cambios producidos. Si lo necesita, puede realizar El escalado y traslación de los vertices en los ejes locales X e Y del mapa. Pulse **X** o **Y** una vez iniciado el comando escalar (**S**). También se encuentra disponible la Herramienta de Edición Proporcional (PET) que funciona exactamente igual que lo visto para ModoEdición de Mallas. Al igual que en ese modo, los vertices pueden ocultarse o mostrarse utilizando **H** y **ALT-H** respectivamente.

Grab/Move	G
Rotate	R
Scale	S
Weld/Align	W
Mirror...	M

Menú de transformaciones UV.

Dentro del Editor UV se encuentran disponibles varios modos de selección. Como cada vertice de la malla es dibujado en el Editor por cada cara a la que pertenece, a veces es difícil distinguir si hemos seleccionado el mismo vertice o no. Habilitando la opción *Pegar UV a Vertice de Malla*, al hacer click con **RMB** no solo seleccionará un vertice UV sino todos los vertices UV que correspondan al mismo vertice de la malla. Este modo es accesible incluso si no se lo ha activado desde el menú, simplemente manteniendo pulsada la tecla **CTRL** cuando se selecciona el vertice. La opción *Stick Local UVs to Mesh Vertex (Pegar UV Local a Vertice de Malla)* funciona de la misma manera, pero solo selecciona los UVs que están "conectados", es decir, que están dentro de un rango de 5 pixeles alrededor del primer UV seleccionado. También puede acceder a este modo aún cuando no esté activado por defecto manteniendo pulsada **SHIFT** cuando se seleccione un vertice. Estas dos opciones pueden ser respectivamente activadas o desactivadas pulsando **CTRL-C** y **SHIFT-C**. Con *Active Face Select (Seleccionar Cara Activa)* activada, un click **RMB** seleccionará una cara y la convertirá en la cara activa. Esto puede ser activado y desactivado pulsando **C**. En todos estos tres casos, aparecerá un icono especial en la esquina inferior izquierda del editor UV. Observe que *Active Face Select* y *Stick UVs to Mesh Vertex (Pegar UVs al Vertice de Malla)* pueden también combinarse. La opción *Unlink Selection (Desconectar Selección)* basará la selección actual, es decir solo dejará seleccionados esos vertices, si las caras estuvieran totalmente seleccionadas. Como lo implica el nombre, esto es particularmente útil para desconectar caras y moverlas a otra parte. La combinación de teclas es **ALT-L**. *Select Linked UVs (Seleccionar UVs Conectados)* funciona similarmente a *Select Linked* en la Vista3D. Seleccionará todos los UVs que estén "conectados" a los UVs seleccionados actualmente. La diferencia con la vista 3D es que en el Editor UV, los vertices UV están conectados "implicitamente". Dos vertices UV se consideran conectados si la distancia entre ellos no es mayor de 5 pixeles. La combinación de teclas es **L**. Utilizando el comando "Coser" ("Stitch" **V**), distintas partes de

una mapa UV pueden ser "cosidas" o pegadas entre sí si los vertices UV de los bordes, corresponden a los mismos vertices de malla. El comando Coser funciona uniendo contornos irregulares, seleccione los vertices de la linea fronteriza usando "Stick UVs to Mesh Vertex". *Limit Stitch* (*Limitar Costura*) funciona de manera parecida. La diferencia es que solo pega UVs dentro de un rango dado. El limite por omisión es de 20 pixeles. La ventaja sobre "Coser" es que evita que aquellos UVs que se supone deben permanecer separados, se unan entre sí. En "*Stitch*" and "*Limit Stitch*", puede verse como Limitar Costura evita que las dos partes de un cilindro se superpongan una sobre otra.



"Stitch" and "Limit Stitch".

Puede soldar UVs que no correspondan al mismo vertice de malla usando el comando Weld ("Soldar" **W**). También puede usar este comando para alinear varios vertices en X o Y. Después de pulsar **W** pulse **X** o **Y** para elegir a cual eje quiere alinear. Algunas pistas:

- Pulse **R** en la ventana 3D

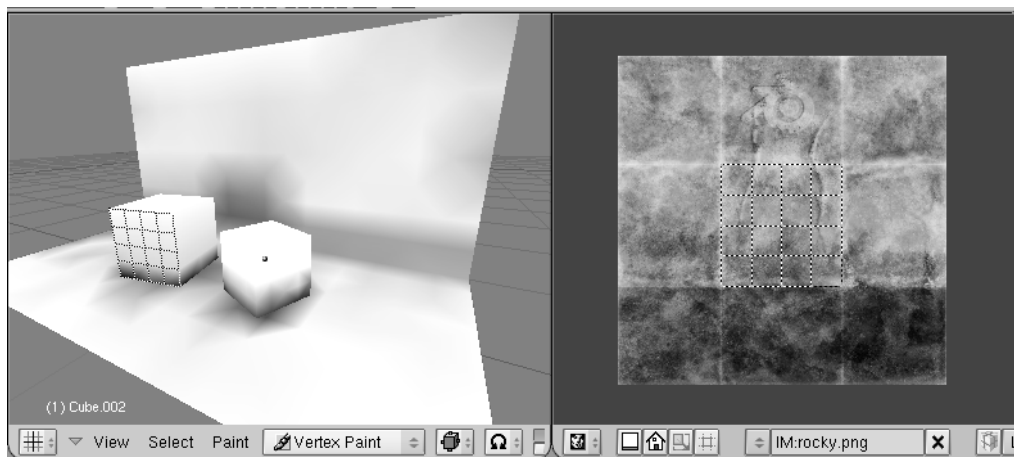
para rotar las coordenadas UV.

- A veces es necesario mover los ficheros de

imagenes a otro lugar en el disco rígido. Presione **N** en la Ventana de Imagen para acceder al menu *Replace Image name*. Puede completar el viejo nombre de directorio y el nuevo. Pulse **OK** para cambiar las direcciones de todas las imagenes utiizadas en Blender que apunten al viejo directorio. (Nota: use como nuevo directorio el código "/" para referirse al directorio donde se encuentra el fichero .blend).

- Puede usar simultaneamente Seleccion

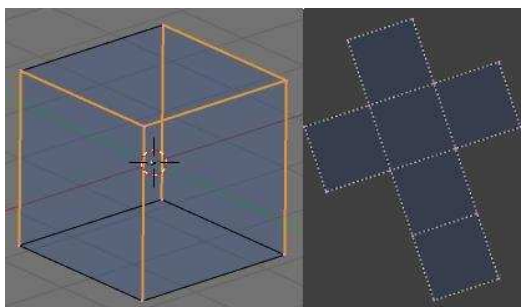
deCaras y Pintura de Vertices (**V**). La Pintura de Vertices solo actuará sobre las caras seleccionadas en este caso. Esta característica es especialmentne util para pintar como si no compartieran vertices. Observe que los colores de vertice se usan para modular el brillo o el color de la textura de imagen aplicada.



Los colores de vertice modulan la textura.

Despliegue LSCM

LSCM es una sigla que significa "Least Squares Conforming Map" o "Mapa Conforme a Cuadrados Mínimos". Este es un método matemático avanzado para crear automáticamente un mapa UV manteniendo al mínimo el estirado y deformación de la textura. Funciona manteniendo los ángulos locales. Como cualquier otro modo de despliegue UV, desplegará las caras seleccionadas en el Modo Selección de Caras UV. Se puede acceder a él pulsando la tecla **U** y eligiendo la opción *LSCM*, o directamente pulsando *LSCM Unwrap* en el Panel *UV Calculation*. Para poder desplegar correctamente una malla empleando LSCM, debe asegurarse de que la malla pueda ser aplanada sin demasiada deformación (en términos matemáticos, debería ser equivalente a un disco). Esto se hace definiendo "costuras" (seams), es decir, lugares por donde será cortada la malla. No necesita definir costuras si la malla puede ser desplegada sobre un plano directamente. En Modo Edición, las aristas seleccionadas pueden ser marcadas como costuras o aristas simples usando las teclas **CTRL-E**. Aquí puede verse un cubo con costuras, y el mapa que resulta luego de aplicar LSCM.



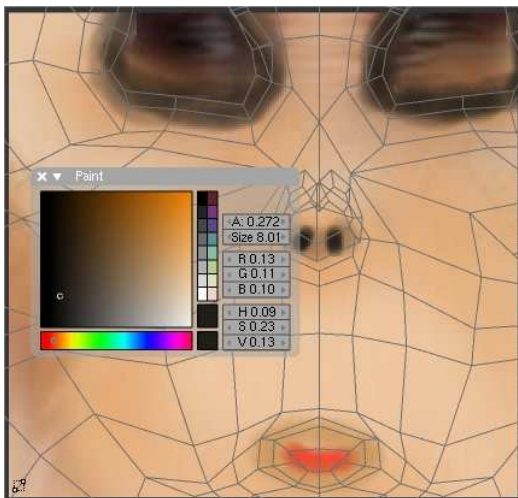
Método de despliegue LSCM.

A menudo una malla no puede desplegarse como un único grupo de caras, sino que debe cortarse en varios grupos. Si las costuras dividen a las caras seleccionadas en varios grupos, el despliegue LSCM los desplegará separadamente ubicandolos en el Editor UV de manera que los grupos de caras no se superpongan. Para facilitar la selección de grupos de caras, el comando **Seleccionar Conectados (L)** en el Modo Selección de Caras UV seleccionará todas las caras conectadas siempre que no haya ninguna costura que las divida. De esta manera puede seleccionar un grupo de caras seleccionando una sola de las caras del grupo y luego pulsando **L**. Para retocar aún más el resultado, los UVs en el Editor UV pueden ser fijados o

"clavados" en una posición determinada. Si se ejecuta LSCM, estos UVs permanecerán en su lugar y el mapa UV resultante se adaptará a los UVs fijos. En el Editor UV, los UVs seleccionados se fijarán o soltarán pulsando **P** o **ALT-P** respectivamente. Pulsando **E** en el Editor UV se iniciará el despliegue LSCM sobre las caras visibles en el Editor. Los UVs fijos son marcados en color rojo.

Pintado de Texturas (Texture Paint)

Una vez que ha cargado una imagen en el Editor UV, puede modificarla utilizando el Modo de Pintado de Texturas. Use la opción *Paint Tool* en el menú *View* para cambiar el tamaño de pincel, Opacidad y Color. Actualmente solo hay un pincel por omisión, pero se está trabajando para ofrecer más pinceles. Todos los cambios realizados se reflejarán inmediatamente en la Vista 3D si el modelo se encuentra en "modo papa". Sin embargo la textura modificada no se guardará hasta que explícitamente sea ordenado. Use la opción *Save Image* en el menú *Image* para guardar su trabajo con un nombre distinto o reemplazar la imagen original.



La herramienta "Paint" en acción.

Observe que la opción *Draw Shadow Mesh* (*Dibujar Sombra de Malla*) se vuelve muy útil para mantener la referencia del mapa UV mientras pinta las texturas.

Renderizado y coordenadas UV

Aún sin una imagen asignada a las caras, puede renderizar usando coordenadas UV. Para ello, utilice el botón *UV* verde en el menú de Botones de Materiales (**F5**). Si también desea renderizar la imagen asignada, deberá pulsar el botón *TexFace* en los botones de Materiales. Combinando esto con la opción *VertexCol* hará que se usen también los colores de vertice.

Mapas de Relieve y de Normales

Los Mapas de Normales y los Mapas de Relieve (Bump) sirven ambos al mismo propósito: simulan la impresión de una Superficie 3D, es decir, de relieve. Pero este relieve no va a proyectar ninguna sombra y no obstruirá a otros objetos. Si el ángulo de cámara es rasante en relación a la superficie, nos daremos cuenta de que la superficie no tiene relieve en realidad.

Los términos "Mapa de Normales" y "Mapa de Relieve" se usan de forma sinónima frecuentemente, pero existen ciertas diferencias:

- Los **Mapas de Relieve** son texturas que almacenan la altura relativa de los pixels desde el punto de vista de la cámara. Los pixels parecen desplazados en la dirección de las normales de las caras, bien en dirección a la cámara o alejándose de ella. Pueden usarse imágenes en escala de grises o bien los valores de intensidad de una textura RGB (incluyendo imágenes).
- Los **Mapas de Normales** son imágenes que almacenan normales directamente en los valores RGB de una imagen. Cambian las normales de tal forma que los pixels parecen movidos de forma arbitraria. Por ello puede crearse cualquier relieve deseado.

Los Mapas de Normales en Blender almacenan una normal de la siguiente forma:

- El color Rojo es mapeado (desde 0 a 255) en X (-1.0 a 1.0)
- El color Verde es mapeado (desde 0 a 255) en Y (-1.0 a 1.0)
- El color Azul es mapeado (desde 0 a 255) en Z (0.0 a 1.0)

Dado que todas las normales apuntan hacia el espectador, no se almacenan valores negativos de Z (serían invisibles de todas formas). En Blender se toma en cuenta el rango completo de azules, aunque otras implementaciones también mapean los colores azules (de 128 a 255), a (0.0 a 1.0). Esta última convención es usada en "Doom 3", por ejemplo.

Ejemplos

Vamos a ver algunos ejemplos. Primero, el render de Suzanne (*Render de Suzanne*). La segunda imagen muestra el Mapa de Normales de Suzanne (*Mapa de Normales de Suzanne*). No se ha usado la cámara en modo *Orto* para hacer más fácil la comparación entre imágenes.

La imagen del extremo derecho en la fila superior muestra el *Mapa de Relieve de Suzanne*. Fue creado con el Plugin ZUtilz. La resolución del Mapa de Relieve es demasiado pequeña para el relativamente grande rango de Valores Z. Podrá darse cuenta de ello observando el render del Mapa de Relieve.



Render de Suzanne

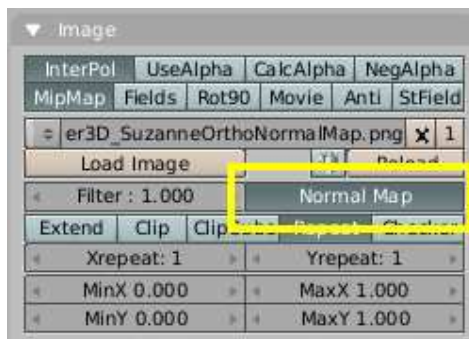


Normal Mapa de Mapa de Relieve de Suzanne. La profundidad de 8 bits del Mapa de Suzanne (no ortogonal) Relieve es demasiado pequeña para cubrir los detalles.

Se han usado ambos mapas como texturas en un plano; primero el Mapa de Normales (*Render del Mapa de Normales*), y luego el Mapa de Relieve (*Render del Mapa de Relieve*). En ambos casos la cámara permaneció en la misma posición en que los mapas fueron hechos (perpendicular al plano).

El *Render del Mapa de Normales* es sólo pseudo 3D. No puede verse el lateral de la cabeza (*Render del Mapa de Normales en vista lateral*). Usando una cámara *Orto* para crear el Mapa de Normales, se consigue menos distorsión de perspectiva (*Render del Mapa de Normales en vista lateral hecho con una cámara Orto*).

Usando Mapas de Normales y de Relieve



Botón *Normal Map* (Mapa de Normales) en el Panel *Image* (Imagen).

Los Mapas de Normales y de Relieve son simples de utilizar. Asegúrese de aplicar la Textura en los botones del panel *Map To* (Mapear A) de *Materials* (Materiales) a *Nor*. La fuerza del efecto está controlada por el Botón Numérico *Nor* del mismo Panel.

Si desea usar un Mapa de Normales, debe activar el botón *Normal Map* (Mapa de Normales) en el panel *Image* en los botones de Textura **F6** (*Botón Normal Map*).

Puesto que sólo las normales son afectadas durante el render, no obtendrá sombras, o Oclusión Ambiental, u otros efectos '3D'. Se trata simplemente de una textura.

Creando Mapas de Normales

Crear Mapas de Normales en Blender es relativamente sencillo.

1. Cree un modelo y alinéelo con las coordenadas del mundo. Ello hará más sencillo posicionar la cámara.

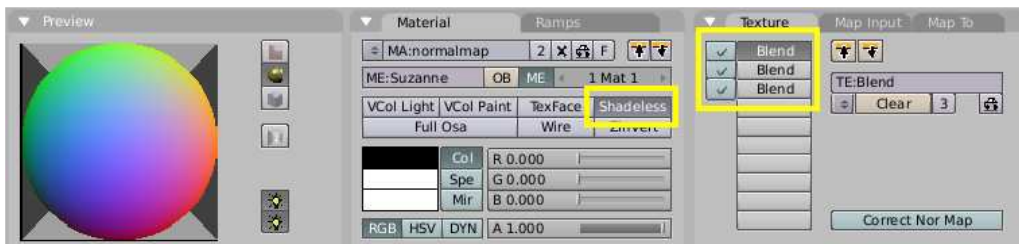


Ajustes de Cámara para crear un Mapa de Normales.

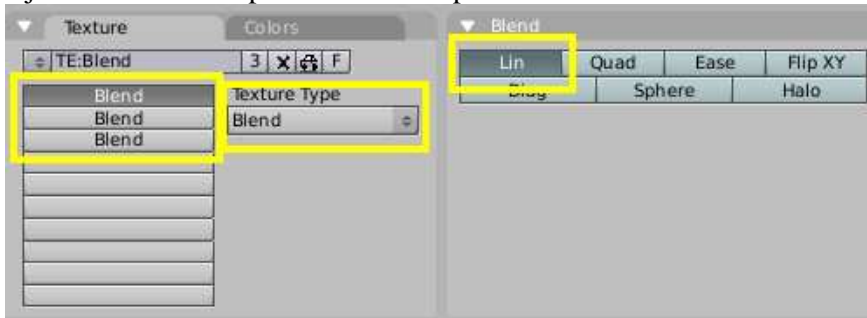
2. Posicione la cámara justo encima del centro de su modelo. Active *Ortho* en el Panel *Camera* (cámara) en los botones *Editing* (Edición, F9). El parámetro *Scale* (escala) gobierna el tamaño del detalle de visualización (*Ajustes de Cámara para crear un Mapa de Normales*).
3. El material del modelo debería ser ajustado a *Shadeless* (sin sombras), así no tenemos que preocuparnos sobre la iluminación (*Ajustes de Material para crear un Mapa de Normales*).
4. Necesitamos tres canales de textura, todos configurados como Texturas *Blend* lineales (*Ajustes de Textura para crear un Mapa de Normales*).
5. Los tres canales de textura usan *Nor* en el Panel *Map Input* (Mapa de Entrada); el Panel *Map To* (Mapear A) Se ajusta a *Col*.
 - La primera textura usa *Map Input* (X, -, -), Color Rojo (1, 0, 0), Modo de Mezcla *Mix* (*Coordenada X*).
 - La segunda textura usa *Map Input* (Y, -, -), Color Verde (0, 1, 0), Modo de Mezcla *Add* (*Coordenada Y*).
 - La tercera textura usa *Map Input* (Z, -, -), ofsX (-0.5), sizeX (2.0), Color Azul (0, 0, 1), Modo de Mezcla *Add* (*Coordenada Z*).

Moviendo y escalando la textura azul conseguimos una resolución más alta en la dirección Z (rango entero).

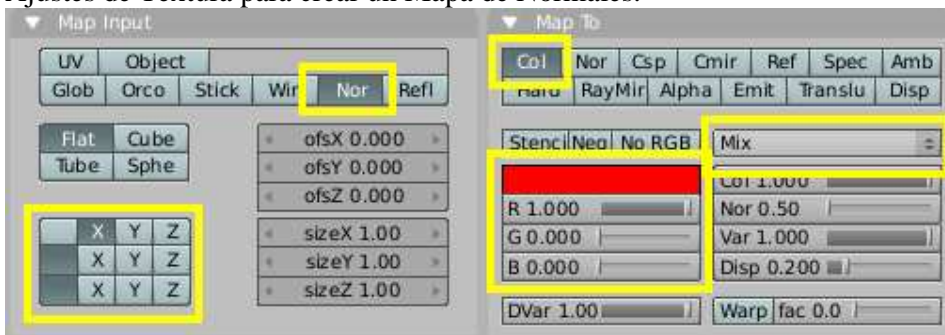
Para una visión general, todos los ajustes de materiales:



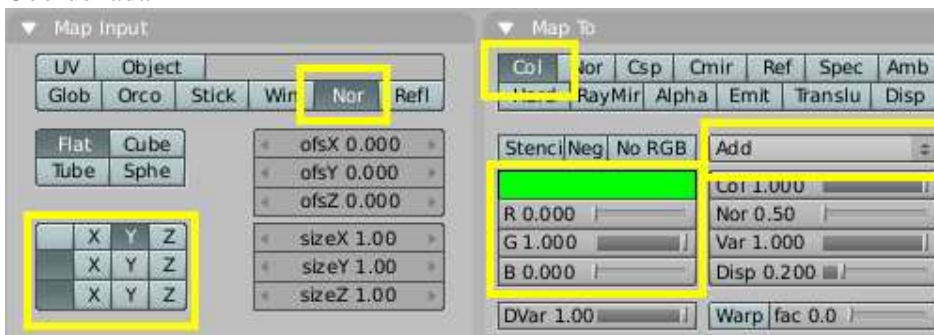
Ajustes de Material para crear un Mapa de Normales.



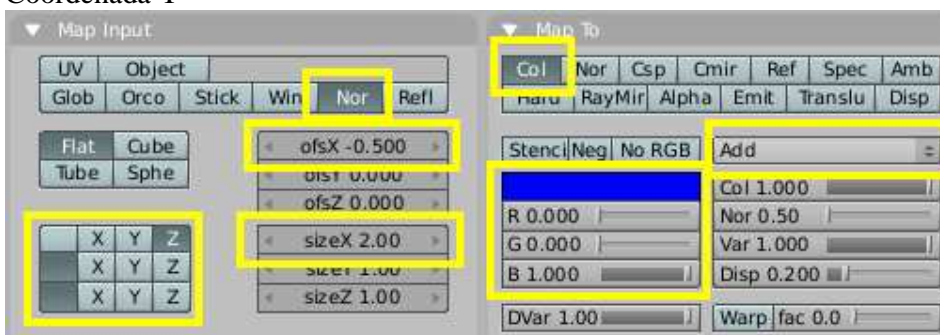
Ajustes de Textura para crear un Mapa de Normales.



Coordenada X



Coordenada Y



Coordenada Z

Puede descargar un archivo de ejemplo con el material, en la página de Notas de Versión. Debe añadir el material "normalmap" del archivo "normalmap_material.blend".



Render del Mapa de Normales, perpendicular a la superficie.



Render del Mapa de Relieve. Haga clic sobre la imagen para ver una versión más grande.



Render del Mapa de Normales en vista lateral.



Render del Mapa de Normales en vista lateral hecho con una cámara *Orto*. Es la misma posición de cámara que en la anterior imagen, con menos distorsión de perspectiva.

Mapas de Normales y superficies curvas: Los Mapas de Normales darán los resultados esperados sólo si se usan con superficies planas.

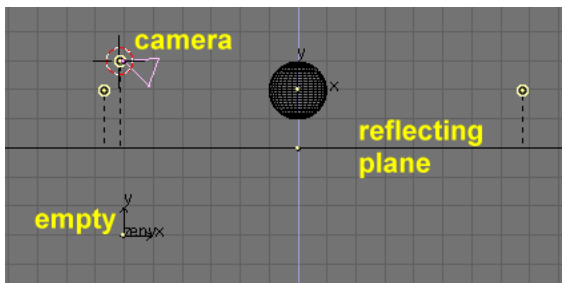
Mapas de Ambiente

Las superficies brillantes que blender genera, muestran reflejos especulares. Lo irónico de estos sombreadores especulares es que son sólo sensibles a las luces (lamps). Específicamente, las superficies con sombreado especular muestran un punto de brillo como si fueran una reflexión de tipo espejo de la luz (lamp). Todo esto tiene sentido excepto que si uno gira la cámara hacia la luz (lamp), ésta no se ve. La cámara ve esta luz sólo si está siendo reflejada por un sombreador especular, no directamente. Por otra parte, los objetos que aparecen muy brillantes en la escena (que reflejan much luz hacia la cámara), pero que no son luces (lamps), no aparecen en estos reflejos. Es fácil hacer una luz (lamp) que este directamente visible para la cámara poniendo algún objeto renderizable en la escena, que se vea como algún tipo apropiado de lámpara, llama, sol, o algo por el estilo. Sin embargo, no hay una solución inmediata para el hecho que, los objetos alrededor no se ven en los reflejos especulares. En una palabra, faltan *reflejos*. Este es el tipo de problema que trataremos utilizando la técnica de mapas de ambiente. Así como se renderiza la luz que alcanza el plano visible utilizando la cámara para definir un punto de vista, se puede renderizar la luz que alcanza la superficie de un objeto (y por lo tanto, la luz que puede ser reflejada hacia la cámara). El mapeo de ambiente de Blender renderiza un mapa cúbico de la escena en los seis puntos cardinales desde algún punto. Cuando los seis bloques de la imagen son mapeados sobre un objeto utilizando

las coordenadas de *Refl*, crean la complejidad visual que el ojo espera ver en reflexiones brillantes.

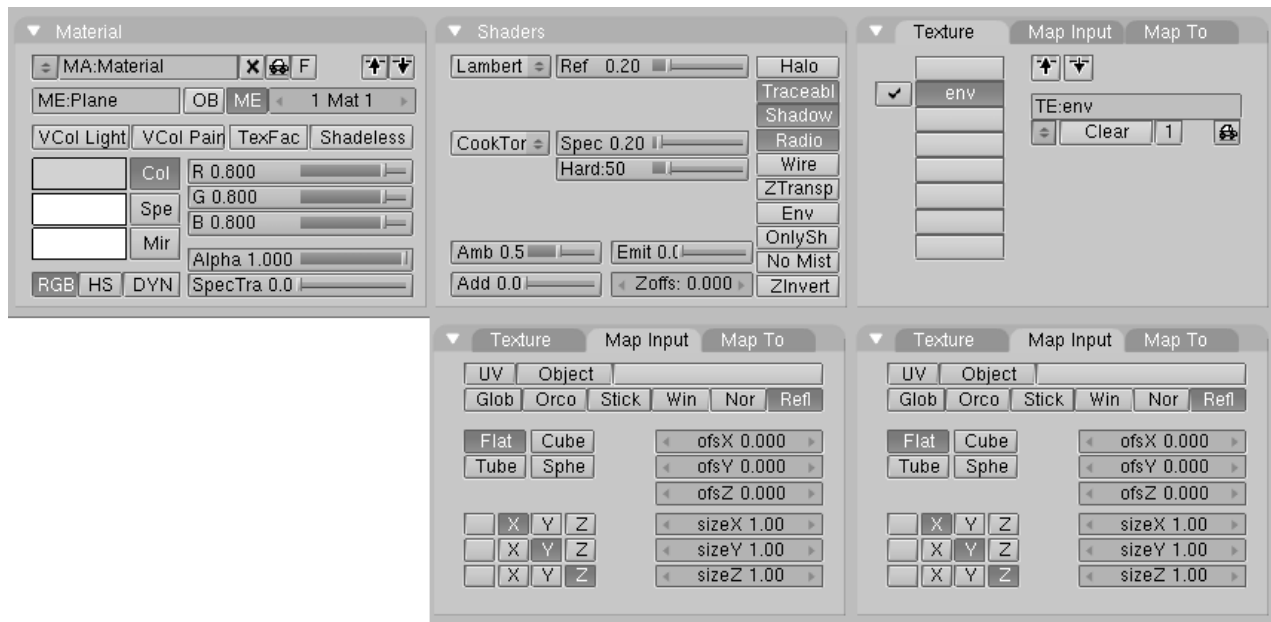
Note: Es útil recordar que el verdadero objetivo de esta técnica es lograr *credibilidad*, no *exactitud*. El ojo no necesita una simulación física exacta de la trayectoria de la luz; solamente necesita creer que la escena es real, viendo la complejidad que espera ver. La cosa menos creíble en la mayoría de las imágenes renderizadas es la esterilidad o pulcritud de la escena, no la exactitud.

El primer paso cuando se crea un mapa de ambiente es definir el punto de vista para el mapa. Para empezar, hay que agregar un empty a la escena y ubicarlo *en la posición del reflejo de la cámara con respecto a la superficie* reflejante. (Esto es posible, estrictamente hablando, solo para superficies reflejantes planas.) En el caso ideal, la posición del empty copiaría inversamente a la posición de la cámara a través del plano del polígono sobre el que está siendo mapeado. Sería realmente difícil crear un mapa de ambiente único para cada polígono de una malla detallada, por lo que hay que aprovechar que el ojo humano es fácil de engañar. En particular, para objetos relativamente pequeños y complejos, se puede solucionar simplemente ubicando el empty cerca del centro. Nombramos al empty *env*, de manera que podamos referirnos a él por su nombre en la configuración del mapa de ambiente. Crearemos una esfera reflejante sobre un plano reflejante, usando la configuración mostrada en *Ejemplo del uso del Mapa de Ambiente*.



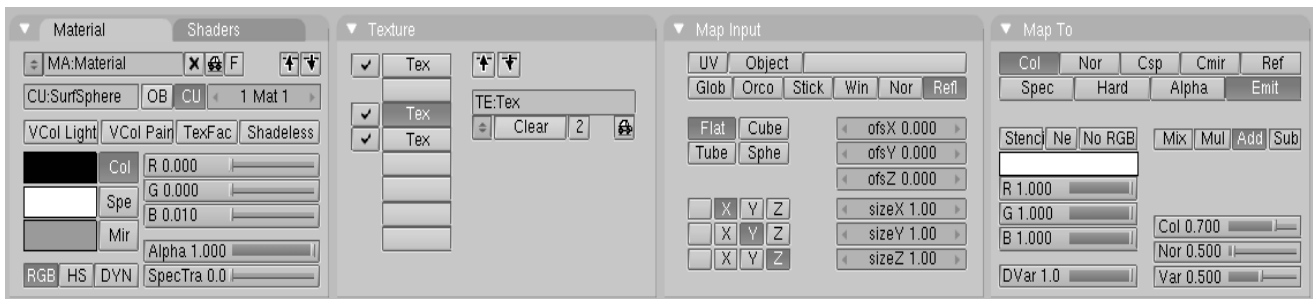
Ejemplo del uso del Mapa de Ambiente

Nótese que el Empty "env" está ubicado exactamente debajo de la cámara, a una distancia de 3 unidades de blender desde el plano reflejante, que es igual a la altura de la cámara sobre el mismo plano. A continuación, pongamos algunas luces, dejemos la esfera sin un material asignado, y movamos el plano a una *capa diferente*. Por ejemplo, digamos que todo está en la capa 1, excepto el plano, que está en la capa 2. Le asignamos al plano un material bajo en *Ref* y *Spec* y agregamos una textura en el canal dos (2) con los parámetros que se muestran en *Material del plano Reflejante*..



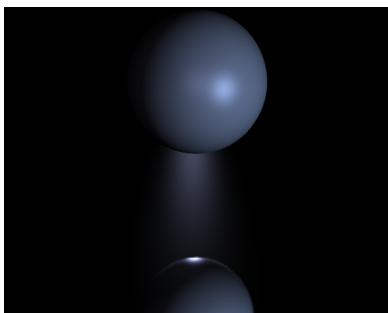
Material del plano Reflejante.

Nótese el mapeado *Refl* y el efecto *Cmir* Utilizamos el canal 2 en lugar del 1, porque necesitaremos el canal 1 más adelante en este ejemplo.



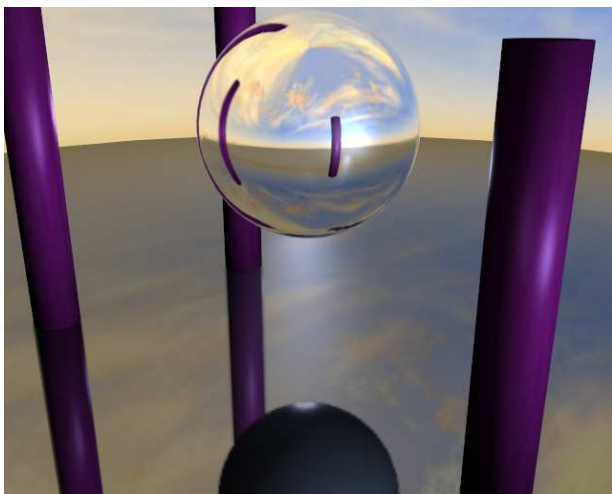
Configuración EnvMap del plano reflejante.

Ahora definimos la nueva textura como *EnvMap*, en los Botones de Textura (F6) (*Configuración EnvMap del plano reflejante.*). En el Panel *Envmap*, observar que el campo *Ob*: contiene el nombre del Empty con respecto al cual vamos a computar el Mapa de Ambiente (EnvMap). Nótese también la resolución del cubo sobre el cual será computado el EnvMap y, lo más importante, los botones *No renderizar capa*: (*Don't render layer:*). Debido a que el EnvMap es computado desde la posición del Empty, debe tener una visión de la escena libre de obstáculos. Ya que el plano reflejante escondería completamente a la esfera, debe estar en su propia capa, que debe ser activada como "No renderizable" para el cálculo del EnvMap. Presionando F12 comienza el proceso de render. Primero se computan las seis imágenes cuadradas correspondientes al EnvMap, a partir de las cuales se produce la imagen final de la esfera reflejada sobre el plano.



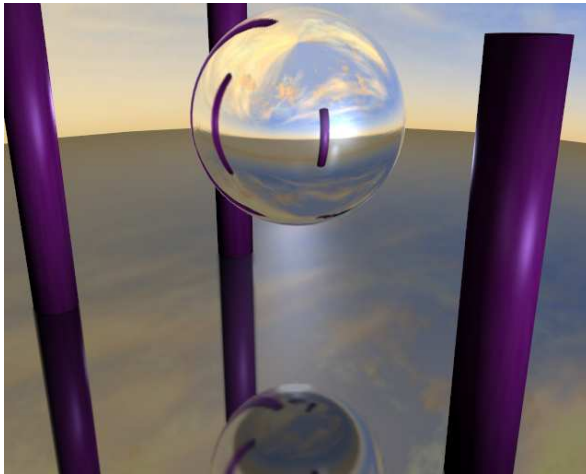
Esfera sobre una superficie reflejante.

Para mejorar la apariencia de la escena, agregamos una gran esfera conteniendo la escena completa y mapeamos una imagen de cielo sobre ésta para simular un mundo nublado. Luego agregamos un nuevo Empty en el centro de la esfera y movemos la esfera a la capa 3. A continuación le asignamos a la nueva esfera un EnvMap, exactamente como hicimos para el plano (pero esta vez, es la capa 3 la que no debe ser renderizada!). Ahora agregamos algunos cilindros, para hacer en entorno aún mas interesante y, antes de pulsar F12, volvemos a la textura del plano y presionamos sobre el botón *Liberar Datos (Free Data)*. Esto obliga a Blender a recalcular el EnvMap para el nuevo entorno. Esta vez, durante el proceso de render, se computan doce imágenes, seis por cada EnvMap. El resultado se ve en *Esfera reflejante sobre una superficie reflejante..* La esfera es más brillante que el plano debido a pequeñas diferencias en la configuración de los materiales.



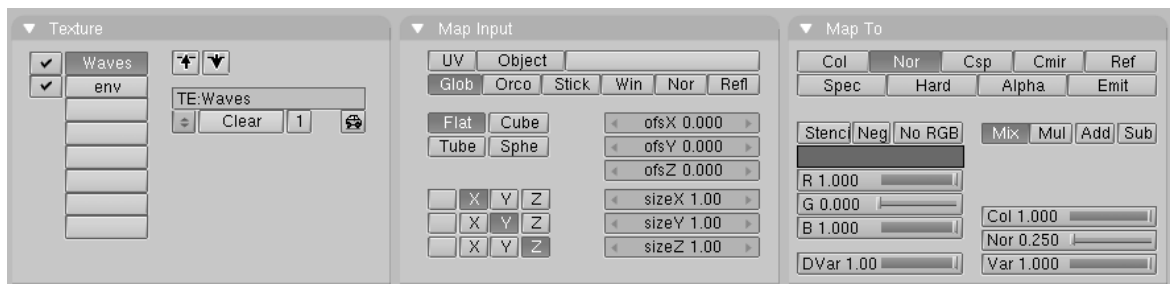
Esfera reflejante sobre una superficie reflejante.

Pero, un momento, hay un problema! La Esfera refleja al Plano, pero el Plano refleja una Esfera gris! Esto pasa porque el EnvMap del Plano es calculado *antes* que el EnvMap de la Esfera. Por lo tanto, cuando lo calcula, la esfera aún es gris, mientras que cuando se calcula el EnvMap de la esfera, el plano ya tiene sus Reflejos. Para corregir esto, localizamos el Botón Numérico *Profundidad (Depth)* en el panel *Envmap*, en los botones de Textura, y le asignamos un valor de 1 tanto para la textura EnvMap del plano como de la esfera. Esto fuerza el cálculo recursivo de EnvMaps. Cada EnvMap es computado y luego es recalculado tantas veces como diga "Profundidad (Depth)", siempre uno luego del otro. El resultado se ve en *Esfera reflejante sobre superficie reflejante, con múltiples reflejos..*



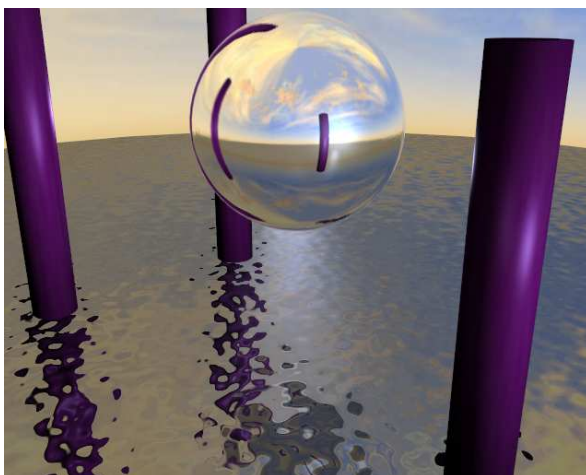
Esfera reflejante sobre superficie reflejante, con múltiples reflejos.

Ahora, si todavía quedan dudas acerca de por qué quedó en blanco el primer canal de textura del material del Plano... Agregamos una nueva textura al primer canal del material del plano. Hacemos que sea *Glob*, afectando *Nor*, con una intensidad de 0,25 (*Configuración adicional de la textura para Mapeado de Relieves (BumpMapping).*).



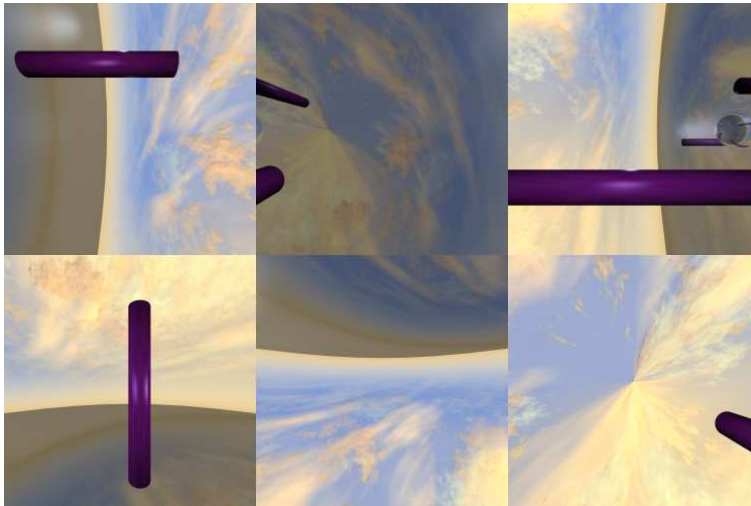
Configuración adicional de la textura para Mapeado de Relieves (BumpMapping).

Esta nueva textura debe ser de tipo *Stucci*; ajustamos el *Tamaño del Ruido (Noise Size)* a 0,15 aproximadamente. Si renderizamos la escena, ahora el plano luce como agua con ondas (*Esfera reflejante sobre agua reflejante, con múltiples reflejos.*).



Esfera reflejante sobre agua reflejante, con múltiples reflejos.

Se debe tener el Mapa de Relieve (BumpMap) en un canal que preceda al EnvMap porque las texturas se aplican secuencialmente. Si se hiciera al revés, los reflejos parecerían estar rotos por las ondas. Se pueden salvar los EnvMaps para ser usados mas tarde y cargarlos con los botones correspondientes en los Botones de Textura. Tambien se pueden construir mapas de ambiente. El estándar es ubicar las seis imágenes mapeadas sobre el cubo en dos filas de tres imágenes cada una, como se muestra en *El EnvMap como queda al ser salvado..*



El EnvMap como queda al ser salvado.

Blender permite tres tipos de mapas de ambiente, como se puede ver en *Configuración EnvMap del plano reflejante.:*

- *Estático (Static)* - El mapa sólo se calcula una sola vez durante la animación, o después de cargar un archivo.

- *Anim* - El mapa es calculado cada vez que se hace un render.

Esto significa que los objetos en movimiento se verán correctamente sobre las superficies reflejantes.

- *Cargar (Load)* - Cuando se salva como un archivo de imagen,

los mapas de ambiente pueden ser cargados desde el disco. Esta opción permite renderizar con mapas de ambiente mucho mas rápidos.

Si la cámara es el único objeto que se mueve, y existe un plano reflejante, el Empty también debe moverse y deben usarse EnvMaps de tipo *Anim*. Si el objeto es pequeño, y el Empty está en su centro, el EnvMap puede ser *Estático (Static)*, aún si el objeto mismo rota, ya que el Empty no lo hace. Si, por otro lado, el Objeto se traslada, el Empty debe seguirlo y el EnvMap debe ser de tipo *Anim*. Otras configuraciones son:

- *Filtro (Filter):* - Con este valor se puede ajustar

la dureza o suavidad de los reflejos.

- *Clipsta*, *ClipEnd* - Estos valores definen los límites de

truncamiento (clipping) cuando se renderizan imágenes con mapas de ambiente.

Mapas de Desplazamiento

El mapeado de Desplazamiento es una poderosa técnica que le permite a una textura de entrada, ya sea procedural o imagen, manipular la posición de las caras renderizadas. A diferencia del mapeado de Relieve o de Normales, donde las normales son torcidas para dar la ilusión de relieve, este crea relieves reales. Proyecta sombras, cubre otros objetos, y hace todo lo que la geometría real puede hacer.

Es útil esencialmente en dos casos:

1. Como Campos de Altitud para generar Paisajes o visualizar Valores en una gráfica 3D.
2. Para crear materiales complejos.

Para manipular suavemente las posiciones de las caras a renderizar, éstas deben ser muy pequeñas. Esto consume memoria y tiempo de CPU.

Uso



Ajustes para un Mapa de Desplazamiento.

El mapeado de Desplazamiento se ha diseñado para comportarse como un canal de textura. Simplemente haga clic en el botón *Disp* en el Panel *Map To*. La fuerza del desplazamiento está controlada por los Botones Numéricos *Disp* y *Nor* (*Ajustes para un Mapa de Desplazamiento*).

Hay actualmente dos modos con los que funciona el desplazamiento:

1. Si la textura sólo da información Normal (como *Stucci*), los vértices se mueven de acuerdo a la textura de entrada. El desplazamiento Normal es controlado por el deslizador *Nor*.
2. Si la textura sólo proporciona información de Intensidad (como *Magic*, derivada del color), los vértices se mueven a lo largo de las normales de vértice (un vértice no tiene

normal, propiamente, es el vector resultante de las caras adyacentes). Los píxels blancos se mueven en la dirección de la normal; los píxels negros se mueven en la dirección opuesta. El desplazamiento de intensidad es controlado con el deslizador *Disp*.

Los dos modos no son excluyentes. Muchas texturas proporcionan ambas informaciones (Cloud, Wood, Marble, Image). La cantidad de cada tipo puede ser combinado usando los deslizadores respectivos. El desplazamiento de intensidad da una superficie más suave y continua, puesto que los vértices son desplazados sólo hacia afuera. El desplazamiento de Normales da una superficie más heterogénea, puesto que los vértices son desplazados en múltiples direcciones.

La profundidad del desplazamiento es escalado con la escala del objeto, pero no con el tamaño relativo de los datos. Ello significa que si doblamos el tamaño del objeto en modo objeto, la profundidad del desplazamiento se dobla también, y así el desplazamiento relativo aparenta ser idéntico. Si escalamos en modo edición, la profundidad de desplazamiento no cambia, y así la profundidad relativa parece más pequeña.

Dónde usar desplazamiento

De mejor a peor, el desplazamiento funciona en los siguientes tipos de objeto usando el método expuesto para controlar el tamaño de la cara en el render.

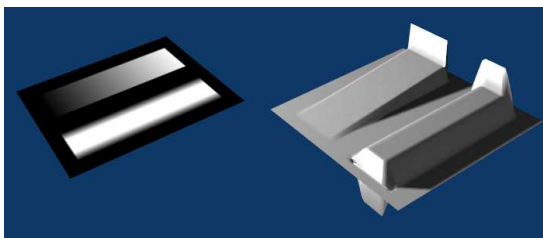
- Mallas Subsurf. El tamaño de la cara renderizada está controlado con el nivel subsurf de render. Al Desplazamiento le favorecen las normales suaves.
- Mallas Subsurf simples. Controla las caras renderizadas con el nivel subsurf de render. Hay sin embargo un escollo en las aristas agudas si la textura no es gris neutro.
- Mallas subdivididas manualmente (modo edición). Controle las caras de render con el número de subdivisiones (esto puede ser combinado con los métodos anteriores). Desplaza exactamente la misma Subsurf Simple, pero ralentiza la edición por el trabajo de OpenGL para dibujar las caras extra (de este modo no se puede bajar el nivel de subdivisión de edición).
- Metabolos. Controle las caras de render con wiresize (tamaño de alambre) de render. Alambre pequeño = más caras.

Los siguientes están disponibles, pero actualmente no funcionan bien. Es aconsejable convertir lo siguiente a mallas antes de aplicar el desplazamiento.

- Superficies Nurbs abiertas. Controle las caras de render con U/V DefResolu. Números más altos dan más caras (note los errores en las normales).
- Superficies Nurbs cerradas. Controle con DefResolu (note los errores en las normales, y cuán implícita se muestra la costura).
- Curvas y Texto. Controle con DefResolu. Más alto da más caras de render (note que las superficies planas grandes tienen pocas caras de render que desplazar).

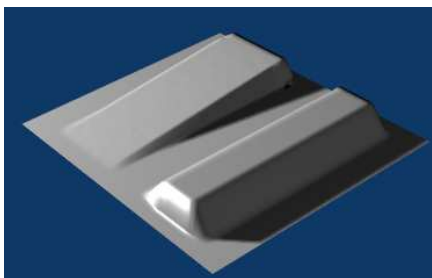
Ejemplos

Primero, un ejemplo que no funciona tan bien (*Textura y Mapa de Desplazamiento*):



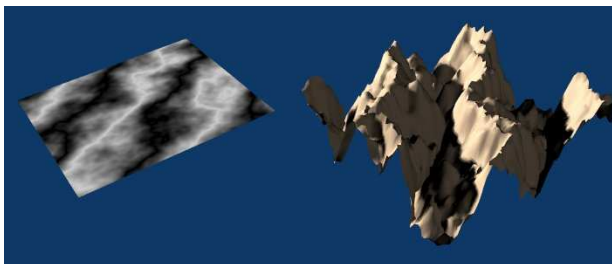
Textura y Mapa de Desplazamiento. El resultado muestra algunos errores.

Las transiciones de pronunciado contraste de negro a blanco producen problemas. Para corregirlo, use un poco de desenfoque (blur) gaussiano en la textura (*Una Textura desenfocada*).



Una Textura desenfocada da el resultado correcto.

Si usa una Textura (como *Marble*) sin transiciones pronunciadas, el desplazamiento funciona bastante bien (*Un Mapa de Desplazamiento para crear un paisaje*).



Un Mapa de Desplazamiento para crear un paisaje.

Los Materiales avanzados usan Mapas de Desplazamiento frecuentemente. Aquí una Textura *Marble* se aplicó a varios Valores *Map To*, incluyendo *Disp*. De otro modo, el borde del cometa sería liso (*Un Mapa de Desplazamiento para materiales avanzados*). La esfera tiene 1024 caras.

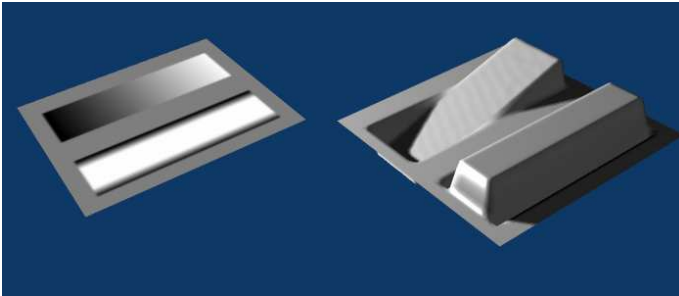


Un Mapa de Desplazamiento para materiales avanzados.

Cómo crear un Mapa de Desplazamiento

Al hacer mapas de desplazamiento a medida, empiece con un fondo de 50% gris. Este color no produce ningún desplazamiento. Puede realizarse algún ajuste usando el Panel *Colors* y los deslizadores *Bright* (brillo) y *Contr*, pero es mejor empezar de forma correcta.

Las líneas finas pueden causar problemas en los Mapas de Desplazamiento, puesto que una cara de render puede ser requerida a mover sólo uno de sus vértices una gran distancia en relación a los otros 2 o 3. Los resultados tienden a ser mejores si se realiza un pequeño desenfoque gaussiano sobre la imagen antes.



Es fácil crear un Mapa de Desplazamiento.

OSA: El OSA de la Textura no funciona actualmente para imágenes mapeadas a desplazamiento.

Plugins de Textura

Como nota final sobre texturas, echemos un vistazo al cuarto botón de tipo de textura, *Plugin*. Blender permite, al momento de la ejecución, el enlace dinámico de objetos compartidos, tanto de plugins de textura como de secuencia. En ambos casos, estos objetos son trozos de código C escrito de acuerdo a un estándar determinado (*chapter_plugin_reference*). En el caso de los plugins de textura, estos trozos de código definen funciones que aceptan coordenadas como entrada y proveen Color, Normal e Intensidad como salida, exactamente como lo hacen las texturas procedurales. Para usar un plugin de Textura, seleccione esta opción, y luego haga click en el botón *Load Plugin* que aparece en Texture Buttons. Una ventana vecina se convertirá en una ventana de selección de ficheros donde podrá seleccionar un plugin. Estos plugins son archivos *.dll* en Windows y archivos *.so* en sistemas Unix. Una vez que un plugin es cargado, convierte la ventana Texture Buttons en su propio set de botones, como se describe en las referencias individuales de plugins.

PARTE V - ILUMINACIÓN

La iluminación es un tema muy importante para el renderizado, tanto como el modelado, los materiales y las texturas. La escena mejor modelada y texturizada con total precisión podría estropearse si no tiene un buen esquema de iluminación, mientras que un modelo simple puede llegar a ser muy realista si esta hábilmente iluminado. La iluminación, por tristemente, se suele menospreciar por los artistas con menos experiencia, que normalmente creen que, en el mundo real las escenas que suelen estar iluminadas por una sola luz (una lámpara, el sol, etc.) producirán el mismo efecto con una sola luz en gráficos por ordenador. Esto es falso, debido a que en el mundo real, hay una sola luz presente, ésta se ve modificada por saltos y se vuelve a irradiar en toda la escena por los objetos en los que rebota, haciendo sombras ligeras y zonas no completamente oscuras, pero iluminadas parcialmente. Las físicas de las ondas lumínicas son simuladas por los renders de trazado de rayos (Ray Tracing) y pueden ser simuladas con Blender activando el motor "Radiosity" (*chapter_radiosity*). El trazado de rayos y la radiosidad son procesos lentos. Blender puede ofrecer un render mucho mas rápido con su renderizados interno de escaneo de líneas (Scanline). Un buen renderizador de escaneo de líneas, sin embargo. Este tipo de motor de render es mucho más rápido debido a que no intenta simular el comportamiento real de la luz, asumiendo muchas hipótesis que lo simplifican. En este capítulo analizaremos los diferentes tipos de luz en Blender y su comportamiento, analizaremos sus puntos fuertes y débiles, y terminaremos describiendo un esquema básico "realista" de iluminación, conocido como el método de los tres puntos, como uno de los más avanzados, realistas, pero por supuesto, de intenso trabajo de CPU, esquema de iluminación.

Capítulos

- Tipos de Luces
- Sombras con el Trazador de Rayos
- Sombras con el Buffer

Tipos de luces

Blender proporciona cuatro tipos de luces:

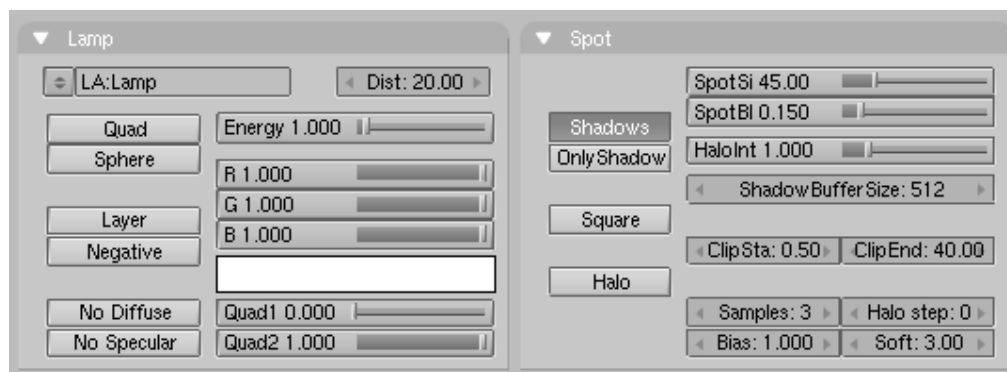
- Sun Light
- Hemi Light
- Lamp Light
- Spot Light

Cualquiera de estas luces pueden añadirse a una escena pulsando **SPACE** y seleccionando la opción *Lamp* del menú. Esta acción añade una luz de tipo *Lamp Light*. Para seleccionar otro tipo diferente, o para modificar sus parámetros, necesitas cambiar a la ventana Shading Context (contexto de sombreado) *Botones de luces*. (F5) y al sub-contexto Lamp (☀️). Una columna de botones, en el panel *Preview*, te permitirá escoger el tipo de luz.



Botones de luces.

Los botones de luces puede dividirse en dos categorías: los que afectan directamente a la luz, agrupadas en los paneles *Lamp* y *Spot*, y los que definen texturas para la luz, que están en el panel *Texture*, situado a la derecha, y que tiene dos pestañas. Las pestañas son muy similares a las relativas a los materiales. En las subsecciones siguientes nos centraremos en los primeros dos paneles (*Botones generales de iluminación.*), dejando una breve discusión sobre texturas a la sección `<command>Afinando la Iluminación (Tweaking light) </command>`.



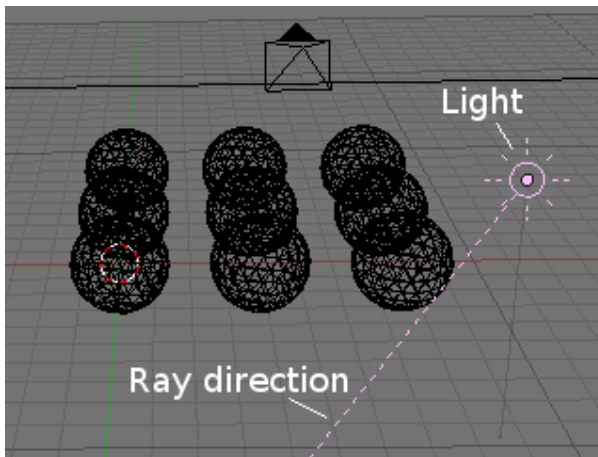
Botones generales de iluminación.

El panel *Lamp* contiene botones que son en su mayoría comunes a todos los tipos de luces, y merecen por tanto ser explicadas. *Negative* - hace que la fuente emita luz 'negativa', es decir, la luz arrojada por el foco es sustraída, en lugar de añadida, a la proyectada por otras luces en la escena. *Layer* - hace que la luz proyectada por el foco afecte únicamente a los objetos que están en la misma capa que éste. *No Diffuse* - hace que la fuente emita luz que no afecta al

modificador de material 'Diffuse', proporcionando únicamente reflejos 'especulares'. *No Specular* - hace que la fuente emita una luz que no afecta and modificador de mtaterial 'Specular', proporcionando únicamente sombreado 'Diffuse'. *Energy* - la energía irradiada por la luz. deslizadores *R*, *G*, *B* - las compoentes roja, verde y azul de la luz proyectada por la fuente.

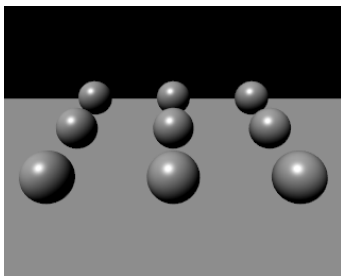
Luz Solar (Sun Light)

El tipo de luz más sencillo es probablemente la luz solar (Sun light)(*Luz solar.*). Una luz solar tiene una intensidad constante que viene de una dirección dada. En la vista 3D la luz solar se representa por un punto amarillo metido en un círculo, el cual se pone de color púrpura cuando se selecciona, además de una línea de puntos. Esta línea indica la dirección de los rayos del sol. Por defecto es normal a la vista en la que se añadió la luz solar a la escena y puede rotarse seleccionando el sol y pulsando **R**.



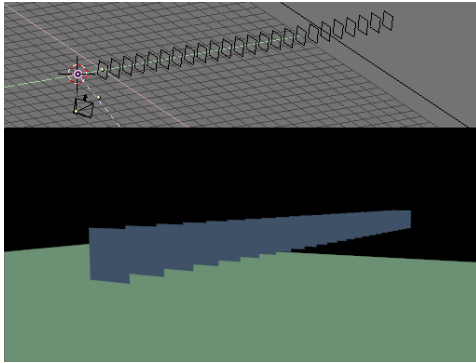
Luz solar.

Los botones de luces que se usan con el sol son los descritos en la sección 'general'. Un ejemplo de iluminación solar se muestra en *Ejemplo de luz solar.*. Como es evidente, la luz viene de una dirección constante, tiene una intensidad uniforme y *no arroja* sombras. Esto último es un punto muy importante de comprender en Blender: ninguna luz, excepto las del tipo "Spot", arrojan sombras. La razón para esto reside en la implementación de la luz en un renderizador por líneas y se discutirá brevemente en las subsecciones 'Puntual' (Spot) y 'Sombras' (shadows). Por último, es importante darse cuenta que en tanto que la luz solar se define por su energía, color y *dirección*, la *ubicación* real en sí de la luz solar no es importante.



Ejemplo de luz solar.

Ejemplo de luz solar. muestra una segunda configuración, compuesta por una serie de planos que distan 1 unidad de blender entre sí, iluminados con una luz solar. La uniformidad de la iluminación es aún más evidente. Esta imagen se usará como referencia para las comparaciones con otros tipos de luces.

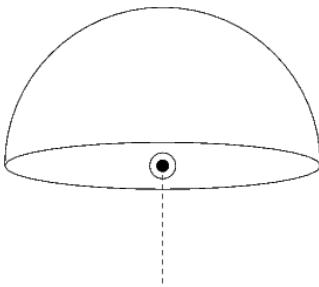


Ejemplo de luz solar.

Trucos solares: Una luz solar puede ser muy útil para la iluminación de un espacio abierto a plena luz del día. El hecho de no arrojar sombras puede solucionarse añadiendo luces puntuales (spot) de tipo 'solo sombra' (shadow only). ¡Mira la sección Afinando la iluminación (Tweaking Light)!

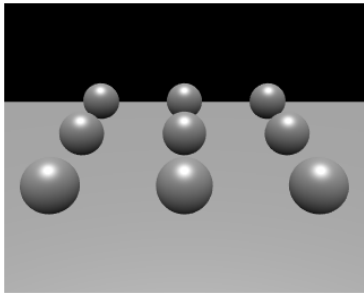
Luz Hemi

La luz Hemi es un tipo muy particular, diseñada para simular la luz proveniente de un cielo muy nublado o de un cielo uniforme. En otras palabras, es una luz que se proyecta, uniformemente, por un hemisferio que rodea la escena (*Esquema conceptual de la luz Hemi.*). Es probablemente la menos usada, pero merece ser tratada antes de las otras dos iluminaciones de Blender debido a su simplicidad. La configuración de esta luz básicamente recuerda a la de la luz solar. Su localización no es importante, mientras que su orientación sí que lo es. La línea de puntos representa la dirección en la que se radia la máxima energía, que es la normal al plano que define el corte del hemisferio, apuntando hacia el lado oscuro.



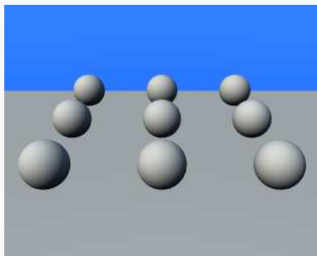
Esquema conceptual de la luz Hemi.

Los resultados de una luz Hemi para la configuración de 9 esferas se muestra es *Ejemplo de luz hemi.* Es evidente la superior suavidad de la luz Hemi en comparación con la solar.



Ejemplo de luz hemi.

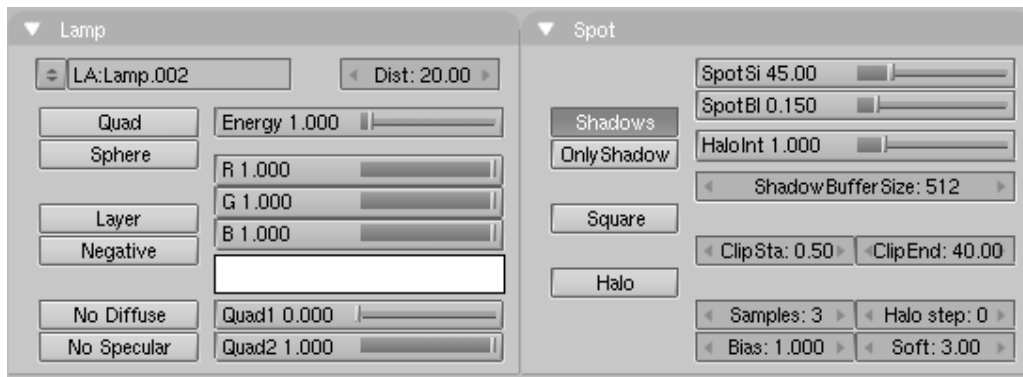
Truco de luz Hemi: Para conseguir una iluminación exterior más realista, sin el problema de la ausencia de sombras, puedes usar una luz solar, de energía 1.0 y color amarillo(naranja cálido, y una luz Hemi más débil simulando la luz que viene de un cielo despejado. *Ejemplo de iluminación exterior. Energía de la luz solar=1, RGB=(1,0.95,0.8). La dirección del sol en un sistema de referencia polar es (135°,135°). Energía de la luz Hemi=0.5, RGB=(0.64,0.78,1.) dirigida hacia abajo* muestra un ejemplo con parámetros relativos. La figura también usa un Mundo. Revisa el capítulo correspondiente.



Ejemplo de iluminación exterior. Energía de la luz solar=1, RGB=(1,0.95,0.8) La dirección del sol en un sistema de referencia polar es (135°,135°). Energía de la luz Hemi=0.5, RGB=(0.64,0.78,1.) dirigida hacia abajo.

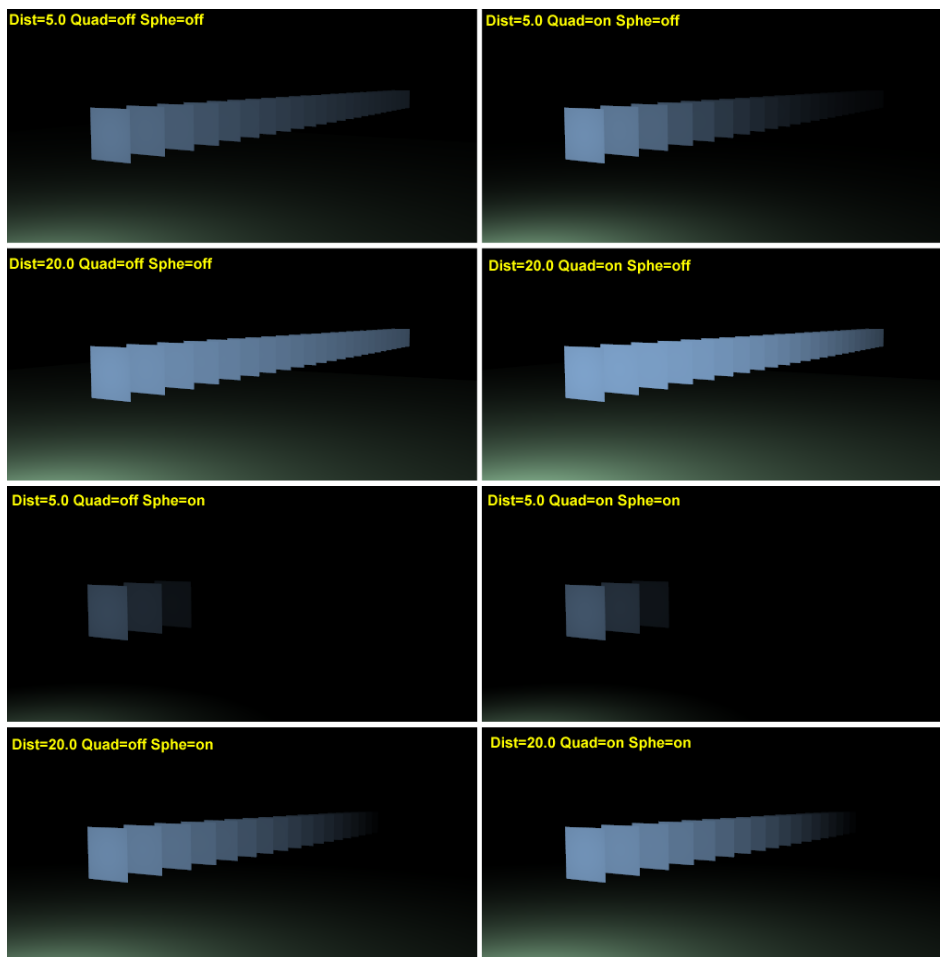
Luz lámpara

La luz de tipo lámpara es una luz de tipo omni-direccional, que es un punto sin dimensiones que irradia la misma cantidad de luz en todas las direcciones. En blender se representa por un punto amarillo encerrado en un círculo. La dirección de los rayos de luz en la superficie de un objeto es dada por la línea que unes el punto de la fuente de luz y el punto en la superficie del objeto. Es más, la intensidad de la luz decrece acorde a un radio dado de la distancia del foco. Además de los mencionados antes, se usan otros tres botones y dos deslizadores en el panel *Lamp* en una luz lámpara(*Botones de luz lámpara.*): *Distance* - da la distancia a la que la intensidad de la luz es la mitad de la Energía. Los objetos que están más cerca que ese punto, reciben más luz, mientras que los que están más lejos reciben menos luz. *Quad* - Si este botón está desactivado, se usa un ratio de atenuación lineal en distancia- en lugar del carente de física. Si está activado, se usa una atenuación más compleja, la cual puede ser definida con más precisión por el usuario desde una atenuación lineal, como la usada por defecto en Blender, a un ratio de atenuación cuadrático en distancia -físicamente correcto. Esto último es un poco más difícil de aprender. Se gobierna por los dos botones numéricos *Quad1* y *Quad2* y se explicará más adelante. *Sphere* - Si se presiona este botón, la luz arrojada por la fuente resulta confinada en la esfera de radio Distance en lugar de dispersarse hacia el infinito con su radio de atenuación.



Botones de luz lámpara.

A continuación *Ejemplo de luz lámpara*. En los ejemplos *Quad Quad1=0, Quad2=1*. se muestra la misma configuración que en el último ejemplo de luz solar, pero con una luz lámpara con diferentes valores Distance y con la atenuación Quadratic activada y desactivada.



Ejemplo de luz lámpara. En los ejemplos *Quad Quad1=0, Quad2=1*.

El efecto del parámetro Distance es evidente, mientras que el del botón Quad es más sutil. En cualquier caso la ausencia de sombras sigue siendo un problema considerable. En realidad, únicamente debería iluminarse el primer plano, porque todos los demás deberían caer bajo la sombra del primero. Para los entusiastas de las matemáticas, y para los que deseen

profundizar un poco más, las leyes que gobiernan la atenuación son las siguientes. Haz que D sea el valor del botón numérico Distance, E el valor del deslizador Energy y r la distancia de la lámpara al punto donde la intensidad de la luz I va a ser calculada. si los botones Quad y Sphere están desactivados:

$$I = E \frac{D}{D+r}$$

Es evidente lo que se afirmó antes: que la intensidad de la luz es igual a la mitad de la energía por $r=D$.

si el botón Quad está activado:
$$I = E \frac{D}{D + Q_1 r} \frac{D^2}{D^2 + Q_2 r^2}$$

Esta es un poco más compleja y depende de los valores de los deslizadores *Quad1* (Q_1) y *Quad2* (Q_2).

No obstante se nota que la atenuación es lineal para Nevertheless it is apparent how the decay is fully linear for

$$Q_1=1, Q_2=0$$

y totalmente cuadrática para

$$Q_1=0, Q_2=1$$

siendo ésta última la usada por defecto. Es suficientemente interesante si

$$Q_1=Q_2=0$$

entonces la luz no se atenúa Si el botón Sphere está por encima de la intensidad de luz calculada

$$I$$

se modifica posteriormente multiplicándolo por el término que tiene una progresión lineal para

$$r$$

de 0 a D

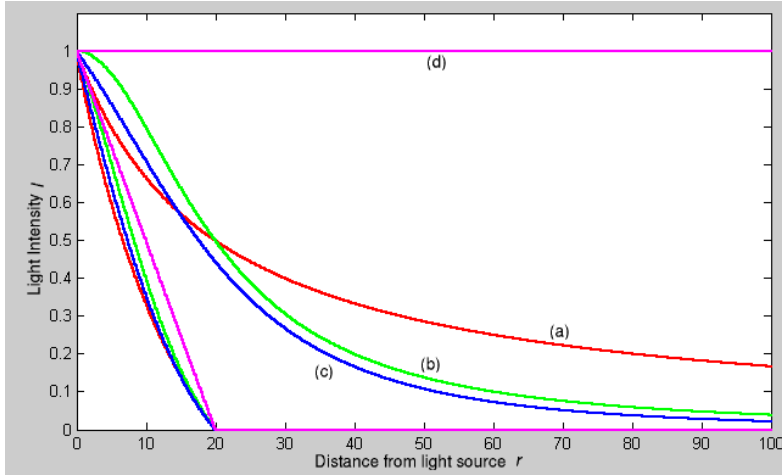
y es exactamente 0 en cualquier otro caso. Si el botón Quad está desactivado y el Sphere está activado:

$$I_s = E \frac{D}{D+r} \frac{D-r}{D} \text{ ; } \text{if } r < D; \text{ ; } 0 \text{ de lo contrario}$$

Si los botones Quad y Sphe están activados:

$$I_s = E \frac{D}{D + Q_1 r} \frac{D^2}{D^2 + Q_2 r^2} \frac{D-r}{D} \text{ ; } \text{if } r < D; \text{ ; } 0 \text{ en caso contrario}$$

Atenuación de la luz: a) Lineal, por defecto en Blender; b) Cuadrática, por defecto en Blender con Quad1=0, Quad2=1; c) Cuadrática, con Quad1=Quad2=0.5; d) cuadrática, con Quad1=Quad2=0. También se muestran en el gráfico las mismas curvas, en los mismos colores, pero con el botón esfera activado. *puede resultar útil a la hora de comprender estos comportamientos de manera gráfica.*

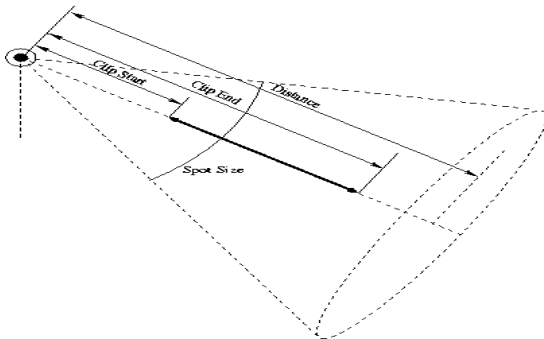


Atenuación de la luz: a) Lineal, por defecto en Blender; b) Cuadrática, por defecto en Blender con Quad1=0, Quad2=1; c) Cuadrática, con Quad1=Quad2=0.5; d) cuadrática, con Quad1=Quad2=0. También se muestran en el gráfico las mismas curvas, en los mismos colores, pero con el botón esfera activado.

Truco de luz lámpara: Como la luz lámpara no proyecta sombras, brilla a través de los muros y objetos similares. Si quieres lograr efectos como fuego, o una habitación iluminada por una vela vista desde fuera de una ventana, la opción a considerar es el uso de una Esfera. Trabajando con cuidado en el valor de Distance (distancia), puedes conseguir una tenue luz que sólo ilumine dentro de la habitación, mientras que ilumines el exterior con una suave luz de luna, obteniendo ésta última con una luz de tipo solar (sun), hemi o ambas.

Luz focal (Spot)

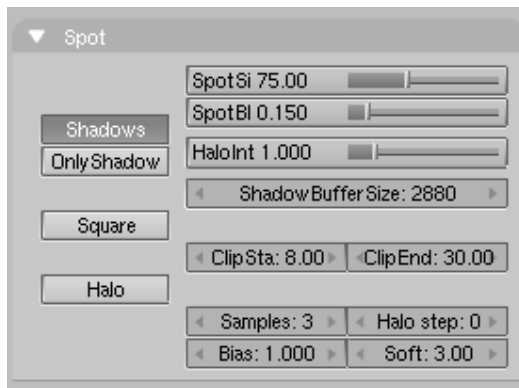
La luz focal es la más compleja de las luces de Blender y es de las más usadas gracias al hecho de ser la única que puede arrojar sombras. Una luz focal es un rayo con forma de cono que parte de la ubicación de la fuente luminosa, la cual es el vértice del cono, en una dirección determinada. *Esquema de luz focal.* debería aclarar este punto.



Esquema de luz focal.

La luz focal usa todos los botones de una luz de tipo lámpara, y con el mismo significado, pero es tan complejo que necesita un segundo panel de botones (*Los botones de opciones de lámpara*): *Spot*.

Opciones de luz focal



Los botones de opciones de lámpara

Shadows - activa/desactiva la proyección de sombras para este foco. *Only Shadow* - hace que el foco sólo proyecte la sombra y no la luz. Esta opción se analizará posteriormente en la sección 'Afinando la iluminación'. *Square* - Las luces focales habitualmente proyectan un cono de luz de sección circular. Hay casos en los que sería útil una sección rectangular, y tener una pirámide de luz en lugar de un cono. Este botón cambia esta opción. *Halo* - permite al foco proyectar un halo como si los rayos de luz atravesaran un medio nublado. Esta opción se explica posteriormente en la sección 'Luz volumétrica'.

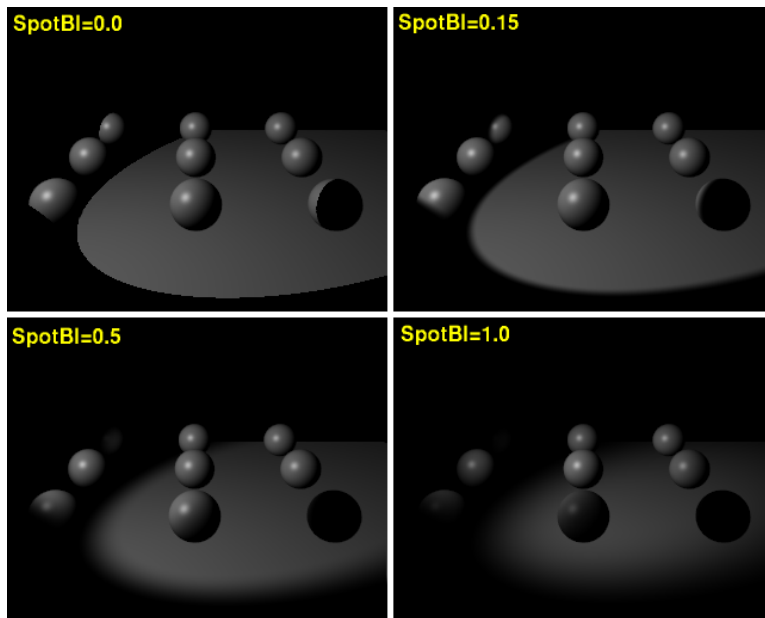
Botones de luz focal



Botones de luz focal.

La columna de botones situada más a la derecha en el Panel *Spot* controla la geometría del foco y las sombras (*Botones de luz focal*): *SpotSi* - el ángulo en el vértice del cono, o la abertura del foco. *SpotBl* - la unión entre el cono de luz y el área en penumbra que lo rodea. Cuanto más bajo sea el valor, más contrastado es el cambio, y cuanto más elevado, más suave. Fíjate que esto sólo se aplica a los bordes de los focos, no a la suavidad de los bordes de las sombras proyectadas por el foco. Estos últimos son gobernados por otro conjunto de botones descritos en la subsección 'Sombras'. *HaloInt* - Si el botón *Halo* está activado, este deslizador define la intensidad del halo del foco. De nuevo, te remitimos a la sección 'Luz volumétrica'. El grupo de botones inferior de la luz focal gobierna las sombras y es un tema lo

suficientemente amplio como para merecerse una sección. Antes de cambiar a las sombras, *Ejemplos de luz focal para SpotSi=45°* muestra el resultado de usar una luz focal para iluminar nuestro primer caso de prueba en diferentes configuraciones.



Ejemplos de luz focal para SpotSi=45°

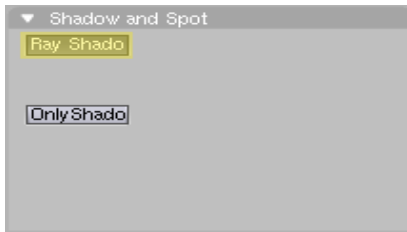
Nota: En *Ejemplos de luz focal para SpotSi=45°* las sombras se desactivan! Las sombras se tratarán en la siguiente sección.

Sombras con el Trazador de Rayos

A diferencia de las sombras producidas con Buffer, las sombras producidas por raytracing se obtienen proyectando rayos desde una fuente lumínica uniformemente y en todas direcciones. Se registra cuales pixeles de la imagen final fue tocado por un rayo de luz y cuales no. Estos últimos son entonces oscurecidos por una sombra.

Los tipos de luces que Blender ofrece proyectan rayos de manera distinta. Por ejemplo, una luz Spot genera los rayos en las direcciones dentro de un cono, mientras que una luz tipo Sol (Sun) emite los rayos como si viniesen desde un punto en el infinito por lo cual todos esos rayos son paralelos y tienen la dirección de la luz en cuestión.

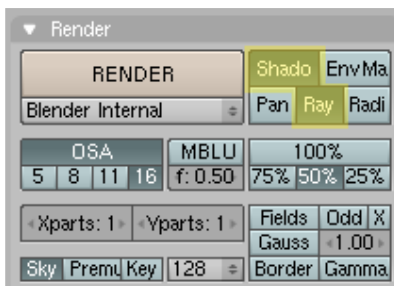
Debido a que cada luz emite una gran cantidad de rayos en la escena, las sombras por raytracing tienden a ser mucho mas lentas que las calculadas usando un Buffer, pero los límites de las areas iluminadas/oscuras son mucho más cortantes y definidos. Para poder utilizar sombras por raytracing para una determinada luz, vaya a los botones *Lamp Buttons* y en la pestaña *Shadow and Spot*, active el botón *Ray Shadow* como se ve en *La pestaña Shadow and Spot*.



La pestaña Shadow and Spot.

Todos los tipos de luces excepto la luz Hemi Light pueden producir sombras por raytracing. Por el momento solamente las Luces de Área, proveen parámetros de configuración adicionales para controlar las sombras por raytracing. Para más información sobre Luces de Área, consulte el capítulo ZZ y la figura *Area Light Options*, para ver las opciones de Luces de Área.

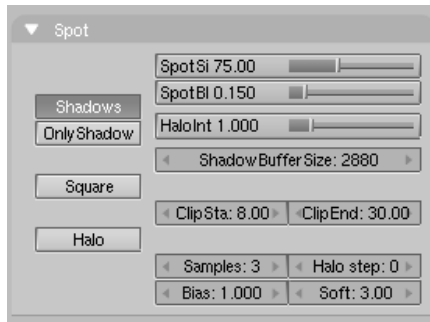
Por supuesto, al igual que en otras características que utilizan raytracing en Blender, debe activar además el botón *Ray* en la pestaña *Render* de los botones de Escena (F10), como se ve en *La pestaña Render.*, y también el botón *Shadow* al igual que para el método de Sombras con Buffer.



La pestaña Render.

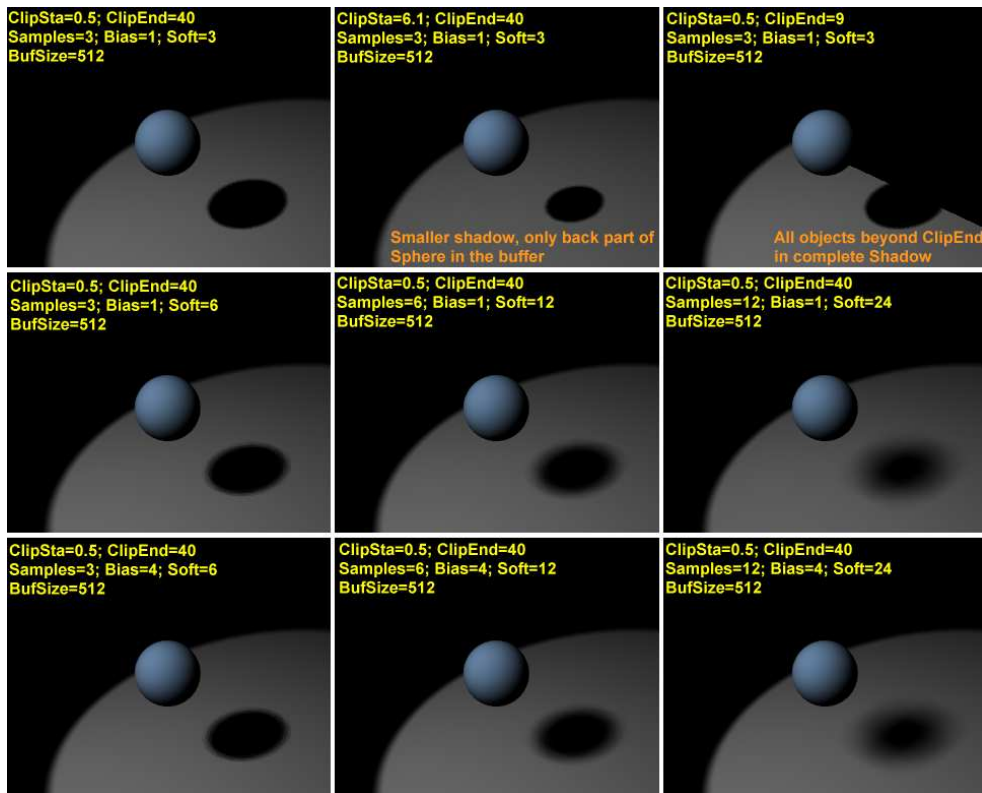
Sombras con el buffer

Los esquemas de iluminación analizados hasta ahora producen en los objetos solo áreas que son más o menos luminosas, pero no proyectan sombras, y una escena sin las sombras apropiadas pierde profundidad y realismo. Por otro lado, un cálculo apropiado de sombras requiere un trazado de rayos completo (y lento). En un renderizador por líneas como Blender, las sombras son calculadas usando el *shadow buffer en el caso de sombras producidas por las luces*. Esto implica que una 'imagen', es vista desde la misma la luz Spot y renderizada y se almacena la distancia de cada punto al foco. Cualquier punto de la imagen renderizada que este mas lejos que cualquiera de estos puntos es entonces considerado como sombra. El shadow buffer almacena estos datos. Para mantener el algoritmo compacto, eficiente y rápido el shadow buffer tiene un tamaño que es fijado desde el comienzo y que en en Blender puede ser desde 512x512 hasta 5120x5120. El valor mayor es el mas preciso. El usuario puede controlar el algoritmo a traves de los botones inferiores en el panel *Spot (Botones de sombra de la luz Spot.)*.



Botones de sombra de la luz Spot.

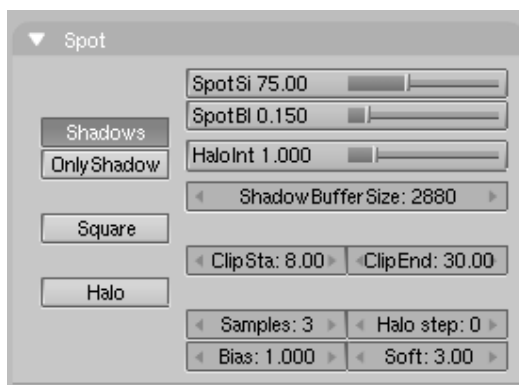
ShadowBufferSize - Botones Numericos, desde 512 hasta 5120, definen el tamaño del buffer de sombras. *ClipSta*, *ClipEnd* - Para aumentar la eficiencia del cómputo de sombras, las mismas funcionan en realidad solo en un rango predefinido de distancias desde la posición del foco. Este rango va desde *ClipSta*, cercano a la luz Spot, hasta *ClipEnd*, mas lejos (BSG.LIG.F.S68.112). Los objetos mas cercanos al *ClipSta* desde la luz Spot nunca son considerados para las sombras, y estan siempre iluminados. Los objetos mas lejanos al *ClipEnd* nunca son considerados para iluminarlos y estan siempre en la sombra. Para conseguir una sombra realista *ClipSta* debe estar mas cerca que la menor distancia entre cualquier objeto relevante de la escena y el foco, y *ClipEnd* mas grande que la mayor distancia. Para el mejor uso de la memoria asignada y una mejor calidad de sombras, *ClipSta* debe ser lo mas grande posible y *ClipEnd* lo menor posible. Esto minimiza el volumen donde las sombras deben ser calculadas. *Samples* - Para obtener sombras suaves el shadow buffer, una vez calculado, es renderizado por medio de su propio algoritmo de anti-aliasing que trabaja por promedio de valores de sombra sobre un cuadrado de un lado de un numero dado de pixeles. *Samples* es el numero de pixeles. Este es 3 por defecto, o sea un cuadrado de 3X3. Valores mayores dan un mejor anti-aliasing y un mayor tiempo de calculo. *Bias* - El bias es usado en el cálculo de las sombras, de nuevo mientras mayor es, mejor el resultado, y más lento. *Soft* - Controla la suavidad del limite de la sombra. A mayor valor, más suave y extendida es el borde. Comunmente este debe tener asignado un valor en el rango entre el valor del botón numérico *Sample* y el doble de este valor. *Halo step* - Activa el paso del muestreo de halo para sombras volumetricas. Esto será explicado en *lighting_volumetric_light*.



Ejemplos de luz Spot con sombra.

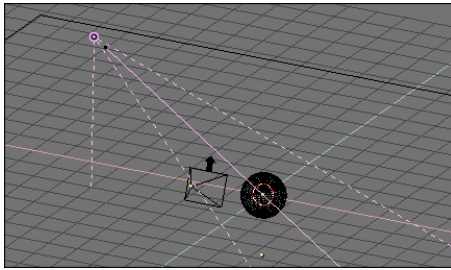
Luz Volumétrica

La luz volumétrica es el efecto que se puede ver en el aire brumoso, cuando los rayos de luz se hacen visibles debido a la dispersión que ocurre debido a niebla, bruma, partículas de polvo etc. Si se usa cuidadosamente puede añadir mucho realismo a una escena... o echarla a perder. En Blender la luz volumétrica sólo puede ser generada para lámparas tipo Spot, una vez que el botón *Halo* en el panel *Spot* es presionado (*Botón halo de la lámpara tipo Spot.*).

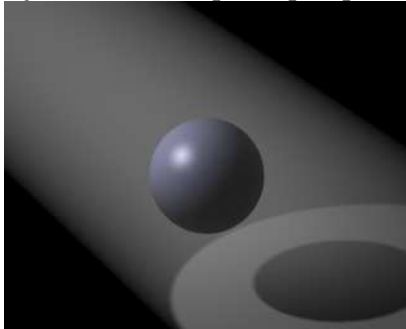


Botón halo de la lámpara tipo Spot.

Si la escena de prueba mostrada en la *Ajustes de la lámpara tipo Spot* es creada y el botón halo presionado, la imagen renderizada será como la *Renderizado con halo.*.



Ajustes de la lámpara tipo Spot



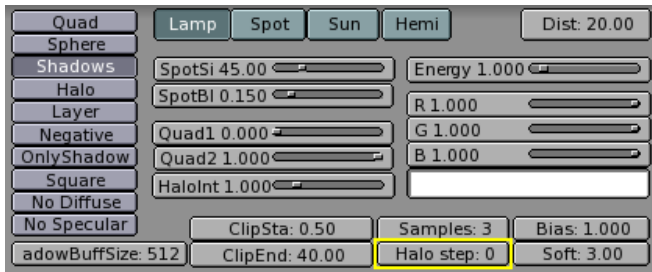
Renderizado con halo.

El efecto de luz volumétrica es más bien fuerte. La intensidad del Halo puede ser ajustada con el deslizador *HaloInt* (*Deslizador de intensidad de halo.*). Valores bajos producen halos débiles.

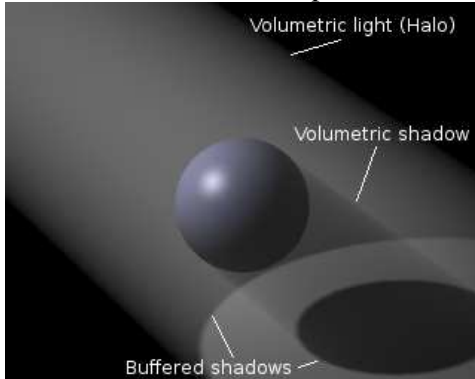


Deslizador de intensidad de halo.

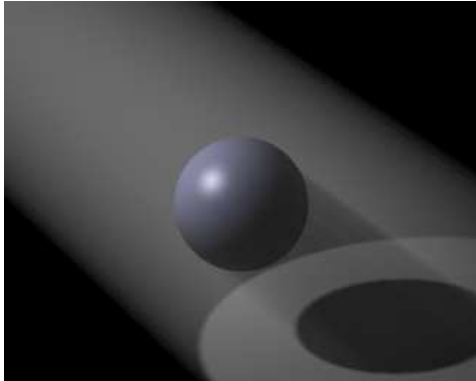
El resultado es interesante. Se obtiene luz volumétrica, pero carecemos de sombra volumétrica! El halo pasa a través de la esfera y, sin embargo, se proyecta sombra. Esto se debe a que el halo tiene lugar en la totalidad del cono de la lámpara Spot a menos que se le indique a Blender que haga otra cosa. El cono debe ser muestreado para obtener sombra volumétrica, y el muestreo ocurre con una medida definida por el botón numérico *HaloStep* (*Botón numérico Halo Step.*). El valor por defecto de 0 significa que no se hará muestreo alguno, por lo cual no hay sombra volumétrica. Un valor de 1 da un muestreo más detallado y por tanto mejores resultados, con el costo de hacer más lento el renderizado (*Halo con sombra volumétrica, Halo Step = 1*), aunque un valor mayor da resultados pobres con un renderizado más rápido (*Halo con sombra volumétrica, Halo Step = 12*).



Botón numérico Halo Step.



Halo con sombra volumétrica, Halo Step = 1

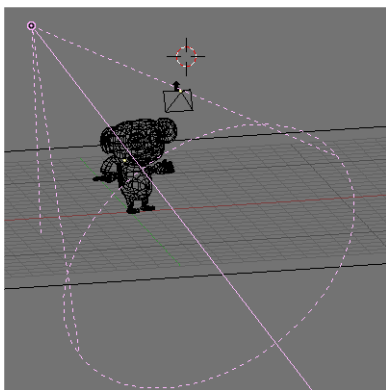


Halo con sombra volumétrica, Halo Step = 12

Valores de HaloStep: Un valor de 8 corresponde, usualmente, a un buen balance entre rapidez y precisión.

Ajustando las sombras con buffer

Ok, ahora que conoces lo básico, realmente podemos hablar de iluminación. Trabajaremos con un solo ejemplo, más complejo que una simple configuración 'esfera sobre plano', para ver lo que podemos conseguir en iluminación realista con Blender. Recurriremos a la configuración en la *Ajustando la iluminación de Cornelius*. El simio de la figura es Cornelius, el hermanito de Suzanne. Tiene un material ligeramente pulido de color marrón. ($R=0.8$, $G=0.704$, $B=0.584$, $Ref=0.7$, $Spec=0.444$, $Hard=10$ - Sí, no muy mono, pero estamos hablando de luces, no de materiales!) y se apoya sobre un plano azul ($R=0.275$, $G=0.5$, $B=1.0$, $Ref=0.8$, $Spec=0.5$, $Hard=50$). Por ahora están iluminados por un simple Spot ($Energy=1.0$, $R=G=B=1.0$, $SpotSi=45.0$, $SpotBl=0.15$, $ClipSta=0.1$, $ClipEnd=100$, $Samples=3$, $Soft=3$, $Bias=1.0$, $BufSize=512$).



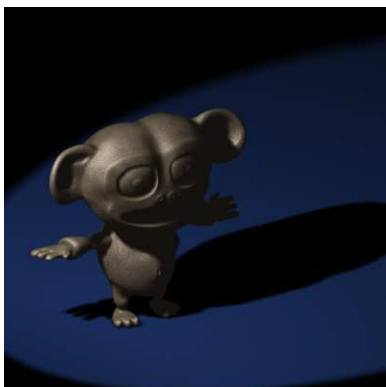
Ajustando la iluminación de Cornelius.

Un render de Cornelius con esta configuración, con $OSA=8$ y sombras activadas, da el resultado en *Configuración de un simple spot*.. El resultado es malo. Obtienes una sombra muy negra y poco realista de Cornelius, y la sombra que crea Cornelius sobre si mismo es inaceptable.



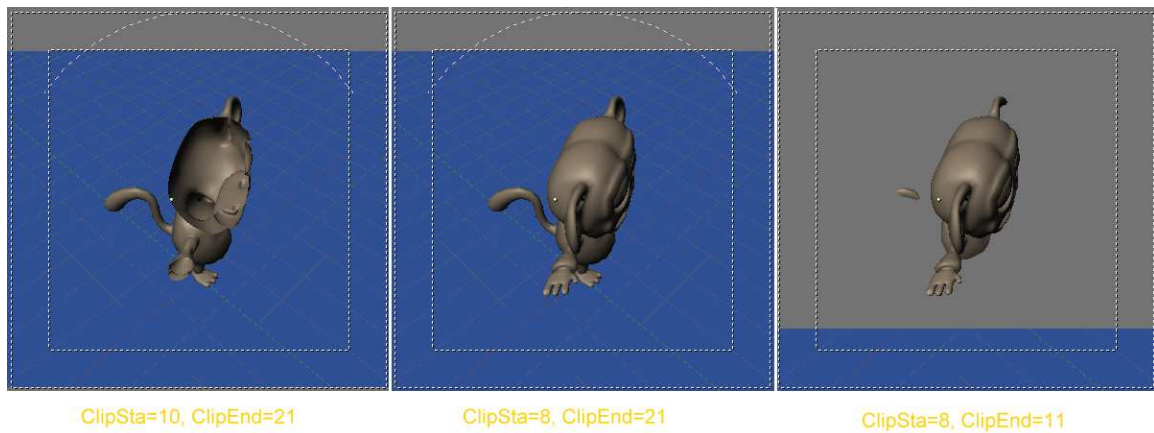
Configuración de un simple spot.

El primer ajuste es sobre *ClipSta* y *ClipEnd*, Si se ajustan para abarcar la escena lo más justo posible ($ClipSta=5$, $ClipEnd=21$) el resultado mejora, al menos para la sombra proyectada. La sombra de Cornelius sigue demasiado oscura (*Configuración de un simple Spot con truncamiento apropiado*).



Configuración de un simple Spot con truncamiento apropiado.

Para poner buenos valores de truncamiento (límites), este es un buen truco: Cualquier objeto en Blender puede actuar como una cámara en la vista 3D. Por lo tanto puedes seleccionar la luz Spot y cambiar a una vista desde ella pulsando **CTRL-NUM0**. Lo que deberías ver en modo sombreado se muestra en *Ajustes de truncamiento de la luz Spot*. Izquierda: *ClipSta demasiado alto*; Centro: *Bien*; Derecha: *ClipEnd demasiado bajo*.. Todo lo que haya más cerca de la Spot que *ClipSta* y mas lejos de la Spot que *ClipEnd* no es visible. Por lo tanto puedes ajustar estos valores con precisión verificando que todos los objetos que arrojan sombras sean visibles.

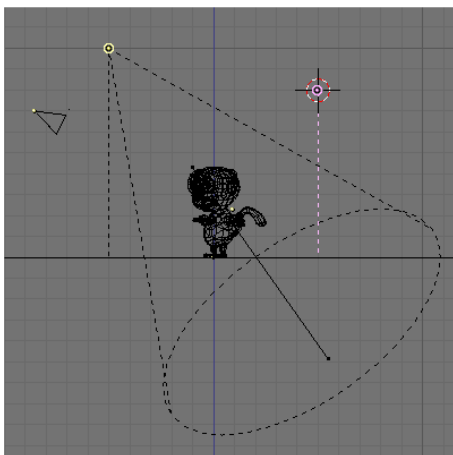


Ajustes de truncamiento de la luz Spot. Izquierda: ClipSta demasiado alto; Centro: Bien; Derecha: ClipEnd demasiado bajo.

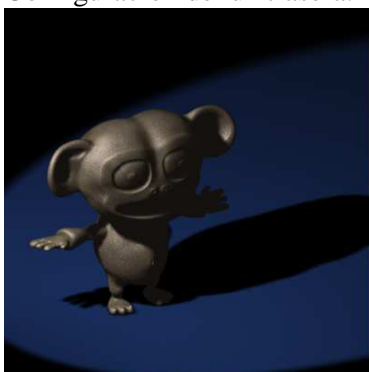
De lo que todavía carece, es del fenomeno fisico de la difusión. Un cuerpo iluminado emite luz (reflejada), por lo tanto las sombras no son totalmente negras, porque algo de luz escapa de las zonas cercanas iluminadas. Esta difusión de la luz es tomada en cuenta correctamente por un trazador de rayos (Ray Tracer), y también por Blender, Atraves del motor de radiosidad. Pero hay configuraciones que pueden simular este fenomeno de forma aceptable. Los analizaremos desde el más simple, hasta el más complejo.

Iluminación de tres puntos

La configuración de iluminación de tres puntos es un clásico y un modo muy simple de obtener una escena con iluminación suave. Nuestra luz Spot es la principal, o luz *Clave*, de la escena, la que arroja sombras. Añadiremos dos luces más para simular la difusión. La siguiente luz que necesitamos se llama *Luz trasera*. está colocada detrás de Cornelius (*Configuración de luz trasera*). Esta ilumina el lado oculto de nuestro personaje, y nos permite separar el primer plano de nuestra imagen, de el fondo, añadiendo en general una sensación de profundidad. Normalmente la luz trasera es igual de fuerte que la luz *Clave*, si no lo es más. Aquí hemos usado una Lamp con energia 1 (*Solo luz clave (Izquierda)*, *Solo luz trasera (centro)*, y *ambas (derecha)*).



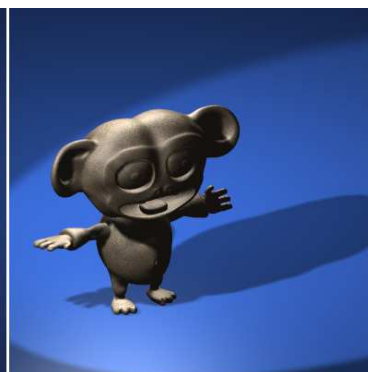
Configuración de luz trasera.



Key light



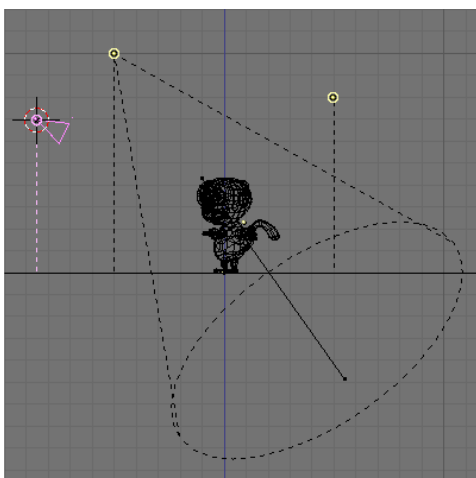
Back Light



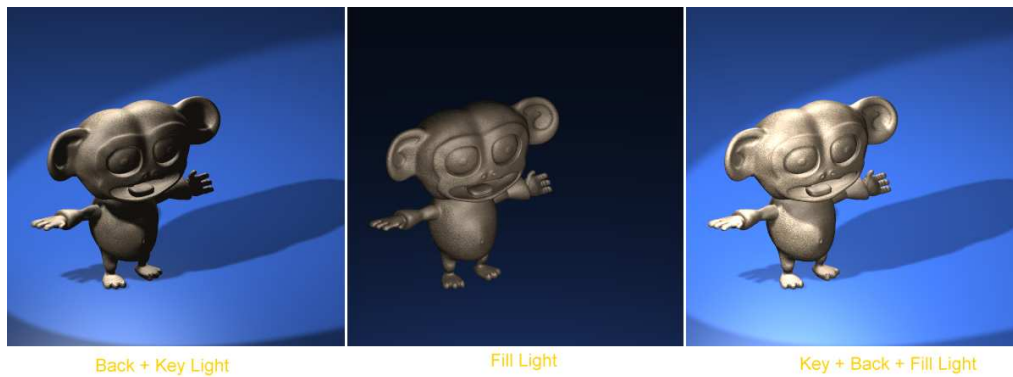
Back + Key Light

Solo luz clave (Izquierda), Solo luz trasera (centro), y ambas (derecha).

El resultado es ya mucho mejor. Finalmente, la tercera luz es la de *Relleno*. La finalidad de la luz de relleno es iluminar las sombras en frente de Cornelius. Colocaremos la luz de relleno exactamente donde está la cámara, con una energía más baja que la más baja de las otras dos luces. (*Configuración de la luz de relleno*). Para este ejemplo se eligió $Energy=0.75$ (*Solo luz Clave y Trasera (izquierda), Solo luz de Relleno (centro), y las tres (derecha)*).



Configuración de la luz de relleno.

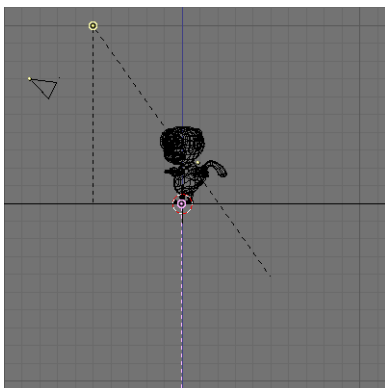


Solo luz Clave y Trasera (izquierda), Solo luz de Relleno (centro), y las tres (derecha).

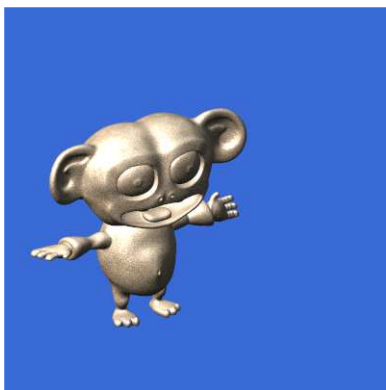
La luz de Relleno hace visibles partes del modelo que estaban completamente en la oscuridad con la luz Clave y Trasera unicamente.

Iluminación de tres puntos - Exteriores

Al usar una luz Spot como luz Clave, el metodo anterior está tristemente relegado a escenas de interior o, como mucho, escenas exteriores nocturnas. Esto se debe a que la luz clave tiene una distancia finita, sus rayos se degradan y el suelo no es iluminado uniformemente. Si estuviéramos al aire libre en un día soleado y despejado, todo el suelo estaría uniformemente iluminado y los objetos arrojarían sombras. Para obtener una iluminación uniforme, sobre todo el suelo, una luz de tipo Sun es lo mejor. Y si añadimos una luz tipo Hemi para simular la luz que viene desde todo el cielo (como en *BSG.LIG.F.S68.111*) podemos obtener una bonita iluminación exterior... pero no tendremos sombras! La configuración de la luz Clave (el sol, $R=1.0$, $G=0.95$, $B=0.9$, $Energy=1.0$) y las luces Relleno/Trasera (ambas representadas por la Hemi, $R=0.8$, $G=0.9$, $B=1.0$, $Energy=0.4$) se muestran en *Configuración de luces Sun y Hemi para exteriores.* y el correspondiente render en *Render de luces Sun y Hemi para exteriores.*

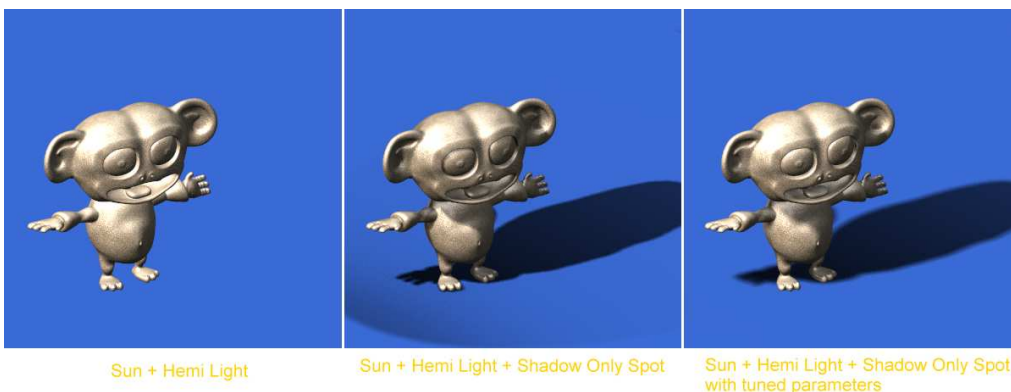


Configuración de luces Sun y Hemi para exteriores.



Render de luces Sun y Hemi para exteriores.

La falta de sombras hace que Cornelius parezca flotar en el espacio. Para obtener sombra, colocar una Spot en el mismo lugar que que el sol y con la misma orientación. Haciendo que el Spot sea de solo sombra (Shadow Only Spot) con el botón apropiado. Si la energía es disminuida a 0.9 y mantenemos el resto de parametros del anterior ejemplo (*BufSize=512, Samples=3 Soft=3 Bias=1 ClipSta=5, ClipEnd=21*) el resultado es el de *Render de exteriores*. (centro).



Render de exteriores.

La sombra es un poco cuadriculada porque Cornelius tiene muchos detalles finos el *BufSize* es demasiado pequeño, y el valor de Sample es demasiado bajo para tenerlo en cuenta correctamente. Si *BufSize* es elevado a 2560, *Samples* a 6 y *Bias* a 3.0 el resultado es *Render de exteriores*. (derecha). Mucho más liso.

Seudo Luz de Area

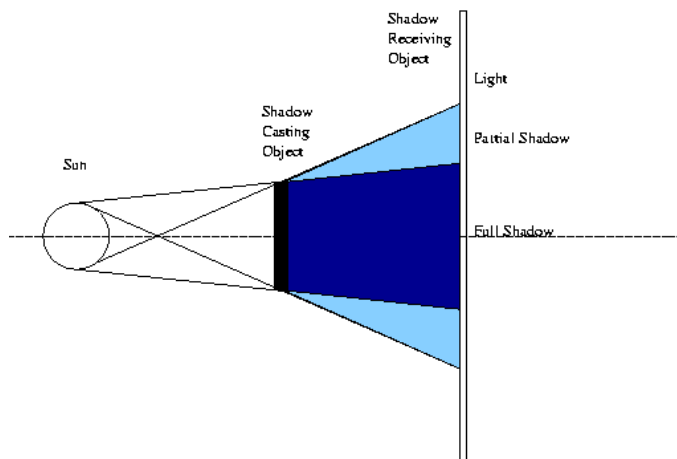
El concepto de luz proveniente de un punto es una aproximación. ninguna luz en el mundo real está carente de dimensión. Toda luz proviene de superficies, no de puntos. Esto posee un par de implicaciones interesantes, principalmente sobre las sombras: <itemizedlist mark='opencircle'>

- Las sombras nítidas no existen: tienen

bordes borrosos.

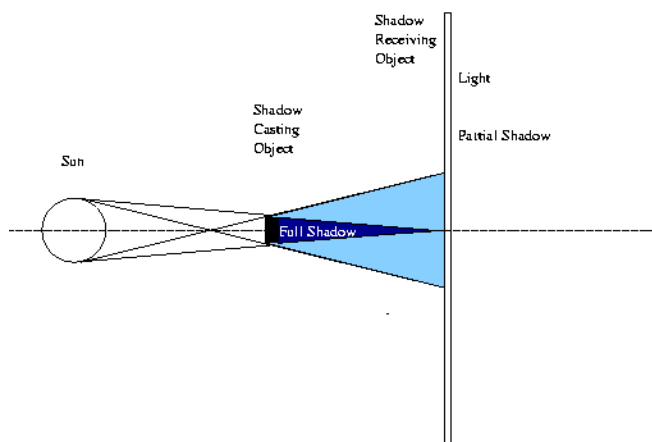
- lo difuminado del borde de la sombra depende de la posición

y tamaño relativo de la luz, del objeto que produce las sombras y del objeto que las recibe. La primera de estas situaciones es aproximada con el ajuste 'Soft' de la luz Spot, pero la segunda no. Para entender claramente este punto imagina un poste fino y alto en el centro de un plano llano iluminado por el sol. El sol no es un punto, tiene una dimensión y, para nosotros los terrestres, esta es medio grado de ancho. Si miras la sombra, te darás cuenta de que es muy nítida en la base del poste y se va haciendo más difusa conforme vas hacia la sombra de la punta. Si el poste es suficientemente alto y delgado, la sombra se desvanecerá. Para entender mejor este concepto, echale un ojo a *La luz de Area y su sombra.* El sol emite luz, en el medio el objeto obstruye completamente los rayos del sol solo en la zona azul oscuro. para cada punto en la zona azul claro el sol es parcialmente visible, por lo tanto cada una de estas zonas es parcialmente iluminada.



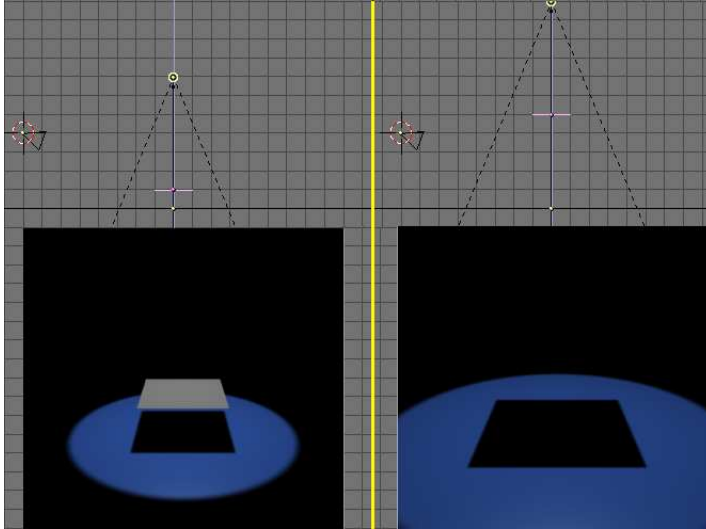
La luz de Area y su sombra.

La zona azul claro es una zona parcialmente sombreada, donde la iluminación va desde luz total hasta sombra total. Es además evidente, en *La luz de Area y su sombra.* que esta zona de transición es más pequeña cerca del objeto que produce la sombra y se hace más grande conforme se aleja de este. Además, si el objeto que produce la sombra es más pequeño que el objeto que emite la luz (y si el objeto que emite la luz es el sol como es el caso) hay una distancia más allá de la cual solamente permanece la sombra parcial *La luz de Area y su sombra 2.*



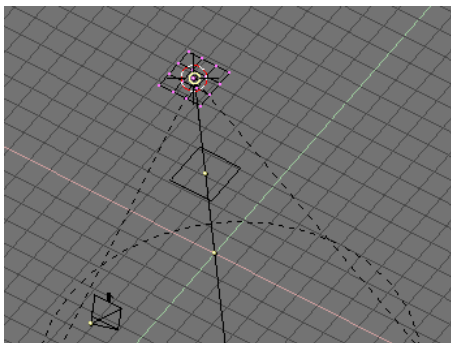
La luz de Area y su sombra 2

En Blender, si colocamos un solo Spot a una distancia fija desde un primer plano, y miramos la sombra producida en un segundo plano, nos daremos cuenta de que la sombra se hace más grande conforme se aleja, pero no más difusa (*La luz Spot y su sombra*)

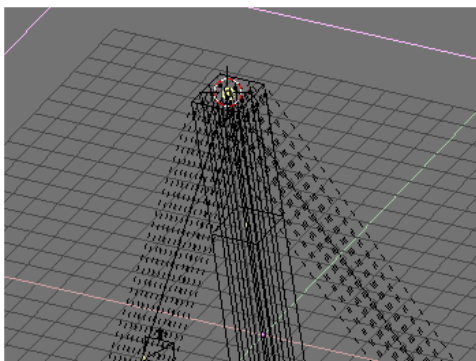


La luz Spot y su sombra

Para simular una luz de tipo area en Blender, podemos usar varios Spots, como si muestreáramos el area luminosa con un numero discreto de puntos de luz. Esto se puede conseguir colocando varios Spots a mano, o usando la función DupliVert de Blender (*special_modelling_dupliverts*), la cual es más eficiente. Añade una malla tipo rejilla (Grid) 4x4, donde está el Spot, asegurate de que las normales estén apuntando hacia abajo haciendo que Blender las muestre e invirtiéndolas (si fuese necesario), como se explica en *mesh_modelling_editmode_basic* (*Configuracion de la rejilla*). Emparenta el Spot a la rejilla, selecciona la rejilla y en el panel *Anim Settings* (F7) pulsa *DupliVert* y *Rot*. *Rot* no es estrictamente necesario pero te ayudará a colocar la luz de area más tarde. Tendrás un bloque de Spots como en *Luces Spot y sus DupliVerts*.

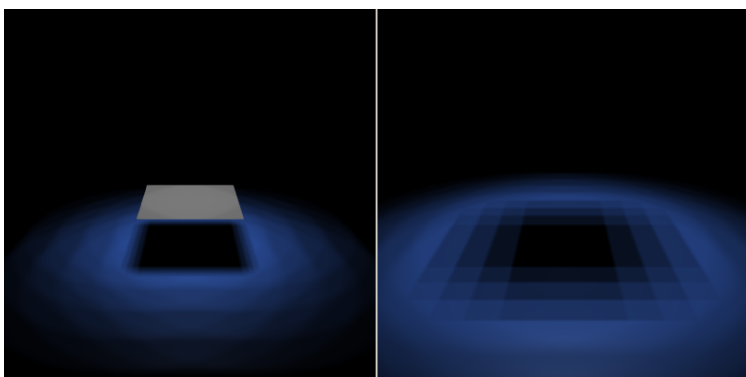


Configuracion de la rejilla



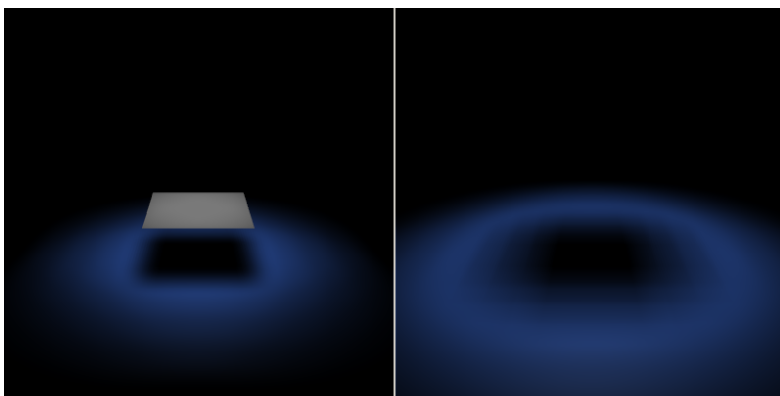
Luces Spot y sus DupliVerts

Entonces disminuye la energía del Spot. Si para un solo Spot usabas una cierta cantidad de energía, ahora debes dividir esa energía entre todos los duplicados. Aquí tenemos 16 Spots, así que a cada uno se le debe asignar $1/16$ de esa energía (esto es $Energy=0.0625$). Los dos mismos renders de arriba, con esta nueva luz de área trucada dará este resultado *Luz de Área trucada con multiples Spots..* El resultado está lejos de lo que se esperaba, porque el muestreo de las luces Spot para ese área es demasiado grueso. Por otro lado, un mejor muestreo implicaría un mayor número de Spots y por ello unos tiempos de render inaceptablemente altos.



Luz de Área trucada con multiples Spots.

Pueden obtenerse resultados mucho mejores, suavizando los Spots, con valores como estos $SpotBl=0.45$, $Sample=12$, $Soft=24$ y $Bias=1.5$ (*Luz de Área trucada con multiples Spots suavizados.*)



Luz de Área trucada con multiples Spots suavizados.

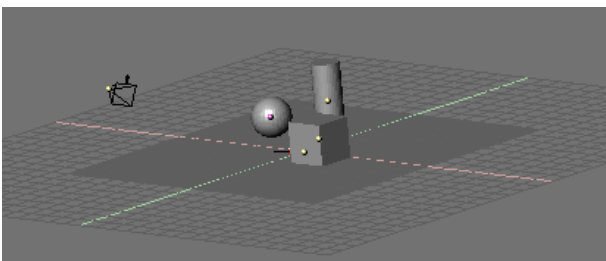
Finalmente, *Cornelius bajo una luz de Area*. muestra que ha pasado con Cornelius una vez la luz Clave ha sido sustituida con 65 Spots duplicados de Energía=0.0154 en un patrón circular. Fíjate como la sombra va suavemente desde nítida en el pie, hasta suave y difusa conforme se aleja de él. Este es el comportamiento físico correcto.



Cornelius bajo una luz de Area.

Seudo Iluminación Global

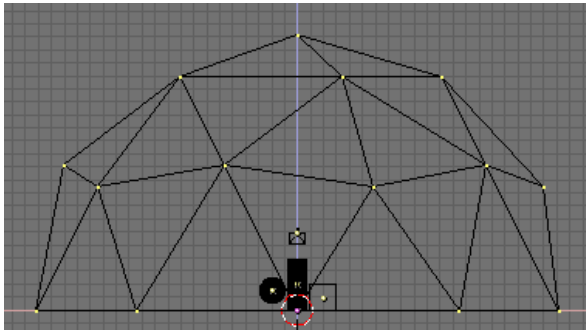
Las técnicas antedichas trabajan bien cuando hay una sola, o por lo menos un número finito de luces, arrojando sombras distintas. Las únicas excepciones son: la configuración para exteriores donde la luz Hemi simula la luz emitida por el cielo, y la luz de Area donde multiples Spots simulan una fuente de luz de dimensiones finitas. la primera de estas dos está muy cerca de una buena iluminación exterior, excepto porque la luz Hemi no arroja sombras y por lo tanto no obtiene resultados realistas. Para obtener una configuración de exteriores realmente buena, especialmente para días nublados, necesitas que la luz provenga de todo el cielo y que cree sombras! Esto se puede obtener aplicando una técnica muy similar a la usada en la configuración de luz tipo Area, pero usando media esfera como malla padre. A esto se le llama normalmente "Iluminación Global". Puedes usar una UVsphere o una IcoSphere, la última tiene los vértices uniformemente distribuidos, mientras la primera tiene una gran concentración de vértices en los polos. Por lo tanto una IcoSphere consigue una iluminación más 'uniforme', todos los puntos del cielo irradian a la misma intensidad; en cambio, una UVsphere irradia mucha más luz desde su(s) polo(s). Personalmente recomiendo la IcoSphere. Preparemos una configuración, compuesta de un plano y algunos solidos, como en *Escena de Iluminación Global*.. Usaremos figuras simples para apreciar mejor los resultados.



Escena de Iluminación Global.

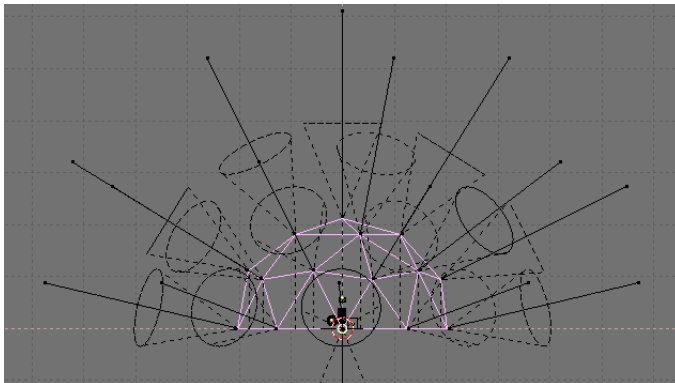
Cambia a la vista superior y añade una IcoSphere, con un nivel 2 de subdivisión es suficiente normalmente, un nivel 3 consigue resultados aún más suaves. Escala la IcoSphere de modo que contenga la escena entera. Cambia a vista frontal y, en modo edición, borra la mitad

inferior de la IcoSphere (*Bóveda celeste*). Esta será nuestra "bóveda celeste" donde los spots serán emparentados y DupliVerteados.



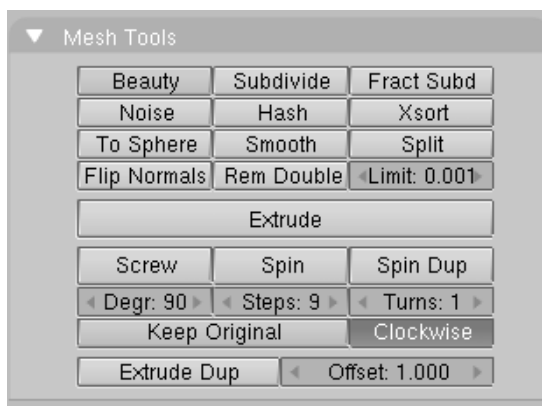
Bóveda celeste.

De nuevo en la vista superior añade una luz Spot, emparentala a la media IcoSphere (CTRL-P) y pulsa los botones *DupliVert* y *Rot* exactamente como en el ejemplo anterior. El resultado, en la vista frontal, es el de *Bóveda celeste con Spots duplicados*.



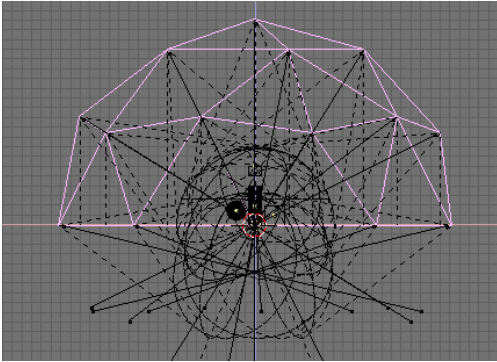
Bóveda celeste con Spots duplicados.

Esto no es lo que queremos, ya que todos los Spots apuntan hacia afuera y la escena no es iluminada. Esto ocurre porque las normales de la IcoSphere apuntan hacia afuera. Es posible invertir la dirección seleccionando todos los vértices en modo edición y pulsando el botón *Flip Normals* en el panel *Mesh Tools* que esta en la ventana de edición (F9) (*Invertiendo las normales*).



Invertiendo las normales.

Esto conduce a la nueva configuración en *Bóveda celeste correcta y luces Spot DupliVerteadas.*

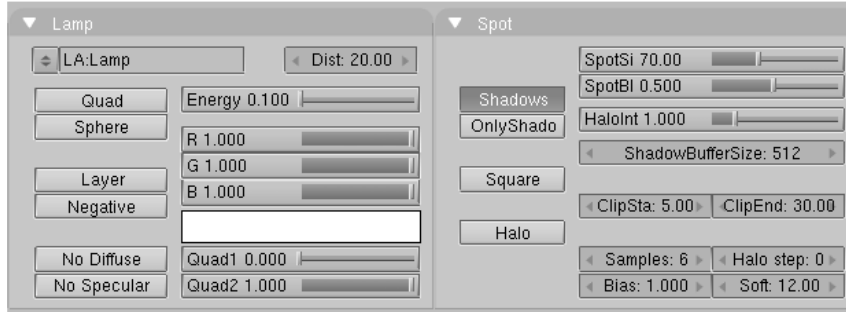


Bóveda celeste correcta y luces Spot DupliVerteadas.

Para obtener buenos resultados selecciona la luz Spot original y cambia sus parametros a un ángulo ancho con contorno difuso (*SpotSi=70.0; SpotBl=0.5*); con los valores convenientes de *ClipSta* y *ClipEnd*

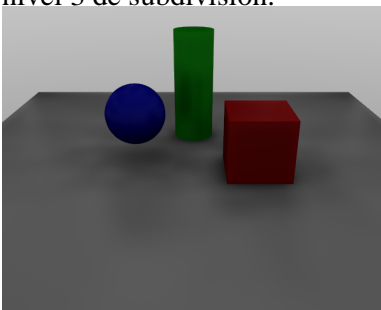
en este caso 5 y 30, respectivamente,

en cualquier caso usa los valores necesarios para envolver toda la escena; incrementa Samples a 6 y Softness a 12. Decrementa la energía a 0.1; recuerda que estás usando muchos spots, así que cada uno de ellos debe ser débil. (*Configuración de luz Spot.*).



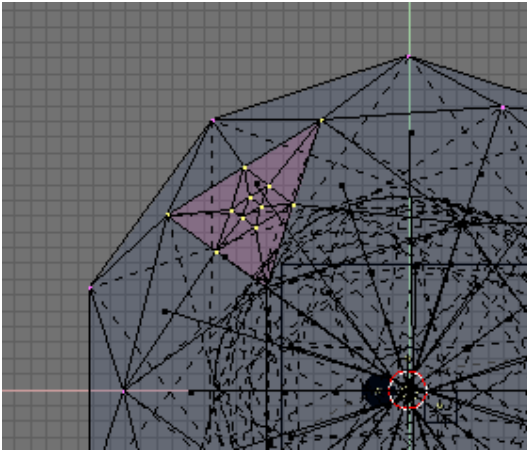
Configuración de luz Spot.

Ahora puedes renderizar. Si se le dan algunos materiales y un entorno, el resultado debe ser algo como *Configuración de luz Spot.* fijate en las sombras suaves y en la iluminación 'omnidireccional'. Todavía se pueden conseguir mejores resultados con una IcoSphere con nivel 3 de subdivisión.



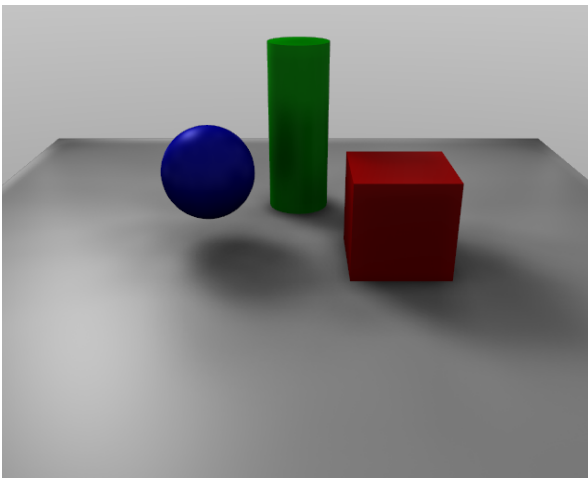
Configuración de luz Spot.

Esta técnica de iluminación global sustituye efectivamente, a muy alto costo de computación, la Hemi de la anterior configuración para exteriores. Es posible añadir un componente de luz direccional simulando el sol mediante un solo Spot o una luz de Area. Una posible alternativa es hacer la IcoSphere menos uniforme subdividiendo una de sus caras un número de veces, como se aprecia en una de las caras posteriores en *Haciendo más densos los Spots en un area..* Esto se hace seleccionando una cara y pulsando el botón *Subdivide*, de nuevo en el panel *Mesh Tools* de la ventana de Edición (F9). Entonces deseleccionamos todo re-seleccionamos únicamente la pequeña cara interior y subdividimos de nuevo, etc...




Haciendo más densos los Spots en un area.

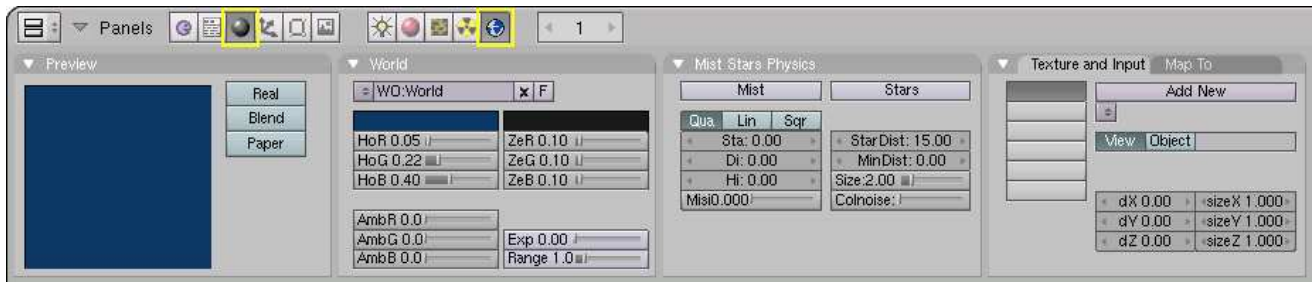
El resultado es una luz direccional muy suave junto con una bóveda celeste de iluminación global o, más sencillo, una bóveda celeste anisotrópica (*Render de la bóveda celeste Anisotrópica.*). Es muy bueno para condiciones nubladas, pero no tan bueno para días soleados y despejados. Para días realmente despejados, es mejor dejar la bóveda celeste separada de la luz del sol, permitiendonos usar diferentes colores para cada una.



Render de la bóveda celeste Anisotrópica.

PARTE VI - EL MUNDO Y EL UNIVERSO

Blender proporciona una gran cantidad de opciones muy interesantes para completar las generaciones de imágenes (renderings) añadiendo un bello fondo y unos cuantos efectos 'de profundidad' interesantes. Todo esto está disponible mediante el Shading Context (Contexto de sombreado)(F5) y el subcontexto World Buttons (Botones de Mundo) () mostrados en *Botones del mundo*. Por defecto se presenta un mundo uniforme y muy sencillo, aunque se puede editar este Mundo o añadir uno nuevo.



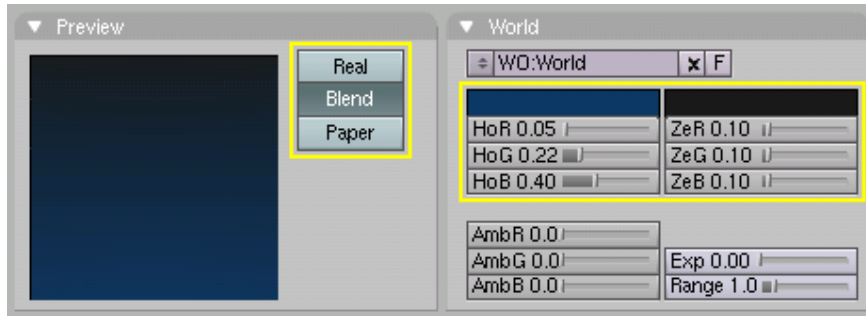
Botones del mundo

Capítulos

- El Fondo del Mundo
- Oclusión Ambiental
- Exposición
- Niebla
- Estrellas

El Fondo del Mundo

La forma más fácil para usar los "World Buttons" es dar a las imágenes un fondo bonito hecho con gradientes para imágenes. Los botones en el *World Panel (Background colors)* permiten definir un color en el horizonte (botones *HoR*, *HoG*, *HoB*) y en el cenit (botones *ZeR*, *ZeG*, *ZeB*).



Background colors

Estos colores se interpretan de forma distinta, según se usen los Botones en el *Preview Panel (Background colors)*:

- *Blend* - El color de fondo

se degrada desde el horizonte hacia el cenit. Si sólo está pulsado botón, el gradiente va desde la parte inferior a la superior de la imagen generada independientemente de la orientación de la cámara

- *Real* - Si este botón también está pulsado

el degradado depende de la orientación de la cámara. El color del horizonte será exactamente ése en el horizonte (en el plano x-y), y el color del cenit se usará para los puntos que estén justo en la vertical de la cámara, tanto arriba como abajo.

- *Paper* - Si se pulsa este botón

el gradiente se produce en los colores cenit-horizonte-cenit. De esta forma, habrá dos transiciones en la imagen, lo que refleja la rotación de la cámara pero mantiene el color del horizonte hacia el centro y el color del cenit en los extremos. El menú *World Buttons* también tiene un Panel de *Textura* con dos Pestañas. Se usan prácticamente igual que las texturas de *Material*, excepto por un par de diferencias (*Botones de textura*):

- Sólo hay seis canales de textura.
- *Texture mapping* - Tiene sólo las opciones

Object y *View*, siendo *View* (Vista) la orientación por defecto.

- *Affect* - La Textura afecta sólo al

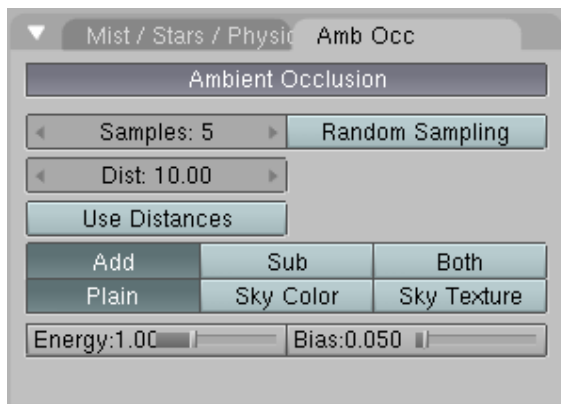
color, pero de cuatro maneras diferentes: Puede afectar al canal de degradado *Blend*, haciendo que el color del Horizonte aparezca donde la textura es distinta de cero; el color del horizonte *Horizonta*; y el color del cenit, arriba o abajo (*Zen Up*, *Zen Down*)



Botones de textura

Oclusión Ambiental

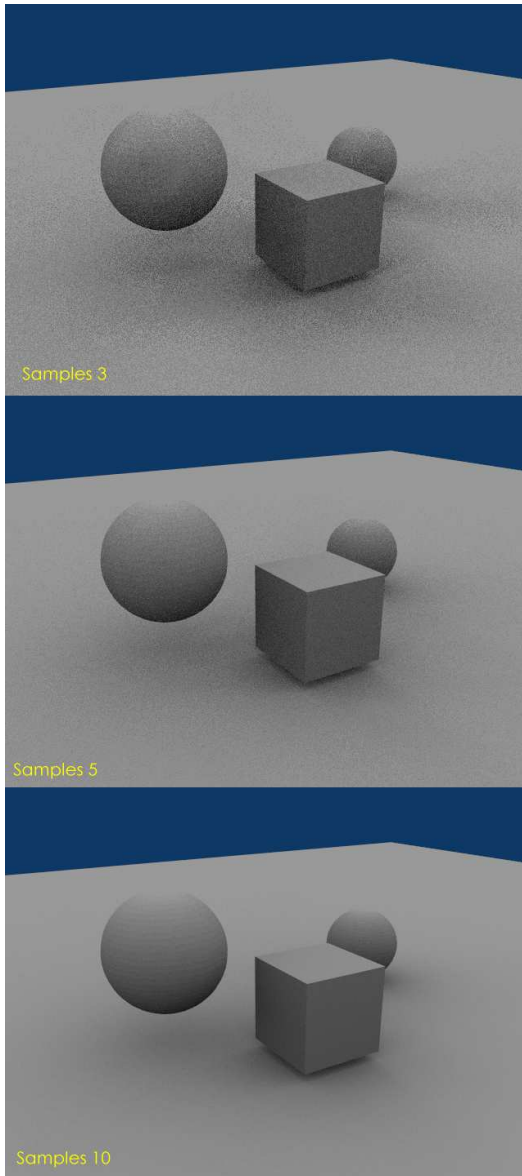
La Oclusión Ambiental, es un sofisticado truco ambiental que simula la iluminación global "blanda", teniendo en cuenta la cantidad de cielo (que se asume como fuente de luz) visible desde un punto dado. Esto se logra de hecho, emitiendo rayos desde cada punto visible, y contando cuantos de ellos logran alcanzar el cielo, y cuántos por el contrario, son bloqueados por objetos. La cantidad de luz sobre el punto es entonces proporcional al número de rayos que han "escapado" y han alcanzado el cielo. Se dispara una hemi-esfera de rayos-sombra. Si un rayo pega contra una cara (es ocluido) entonces el rayo es considerado "sombra", de otro modo se lo considera "luz". La relación entre la cantidad de rayos "sombra" y "luz", define cuán brillante será el pixel. Los parámetros de Oclusión Ambiental ("Ambient Occlusion" - AO) están en la pestaña *Amb Occ* del Sub-contexto Botones de World, en el Contexto de Sombreado. Por omisión, la AO está desactivada, si se la activa la pestaña se llena de botones (*Panel de Oclusión Ambiental*.).



Panel de Oclusión Ambiental.

Los rayos se disparan a la hemiesfera siguiendo un patrón aleatorio, esto causa diferencias apreciables en el patrón de oclusión de pixeles vecinos a menos que el número de rayos disparados sea lo suficientemente grande como para producir buenos datos estadísticos. Esta es la razón por la que la AO genera un patrón granuloso que parece suciedad, si no hubiera rayos suficientes. La cantidad de rayos disparados se controlan por medio del botón numérico *Samples*. El valor por omisión de 5 es en general bueno para previsualizaciones. La cantidad

efectiva de rayos disparados es el cuadrado de este número. (por ejemplo *Samples* (*Muestreos*)=5 significa 25 rayos). La *Efecto de distintas cantidades de muestreos*. muestra una escena simple con número creciente de muestreos. Por supuesto, el tiempo de render ; aumenta con el número de muestreos !



Efecto de distintas cantidades de muestreos.

Los botones *Dist* y *Use Distances* permiten un control sutil del sombreado estableciendo una conducta dependiente de la distancia y amortiguamiento de la oclusión. La fila de botones *Add*, *Sub* y *Both* controlan como se comporta la oclusión:

- *Add* - El pixel recibe luz de acuerdo al número

de rayos no-obstruidos. La escena es clareada.

- *Sub* - El pixel recibe sombra

(luz negativa) de acuerdo al número de rayos obstruidos. La escena se oscurece.

- *Both* - Ambos efectos tienen lugar,

La escena tiene más o menos el mismo brillo.

La fila de botones *Plain*, *Sky Color* y *Sky Texture* controla el color de la luz:

- *Plain* - El pixel recibe luz blanca pura

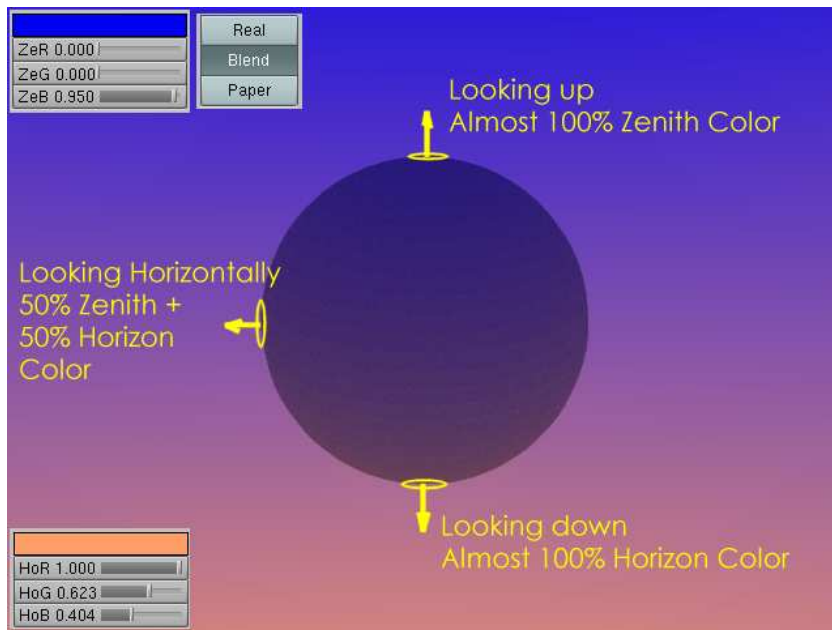
de acuerdo al número de rayos no-obstruidos.

- *Sky Color* - El pixel recibe luz coloreada, el color se calcula

basandose en la porción del cielo tocada por los rayos no-obstruidos (*Oclusión Ambiental con Color Cielo. Zenith es azul, Horizon es anaranjado, y el tipo es Blend para que el cielo sea un anaranjado total en el Nadir.*).

- *Sky Texture* - Debe haber una textura de Imagen para el Cielo, preferentemente un

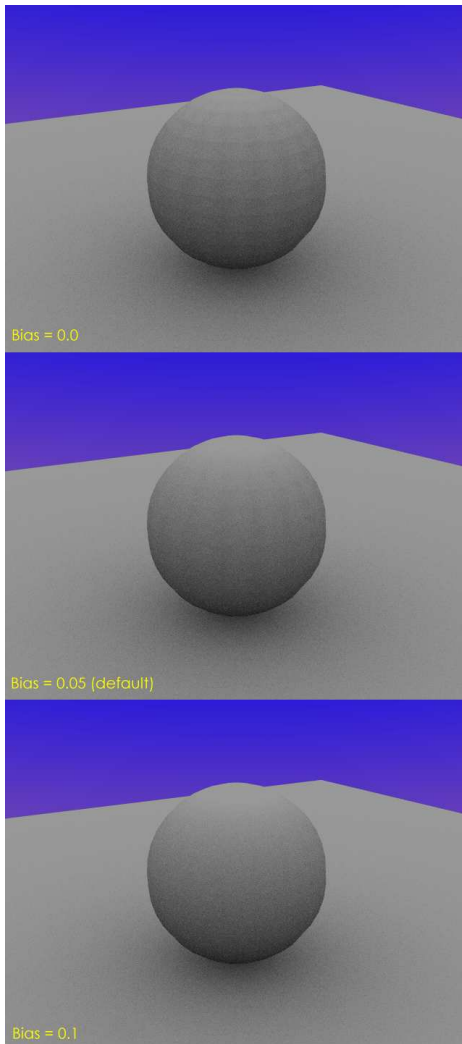
Mapa Angular o un *Mapa Esférico*. Funciona como *Sky Color* pero el color del rayo depende del color del pixel de la textura Cielo alcanzado por el mismo.



Oclusión Ambiental con Color Cielo. Zenith es azul, Horizon es anaranjado, y el tipo es *Blend* para que el cielo sea un anaranjado total en el Nadir.

El deslizador *Energy* controla la cantidad real de luz/sombras que crea el procedimiento AO. Debido a que la AO tiene lugar sobre la malla facetada original, es posible que la luz AO haga visibles las caras sobre objetos con 'smooth' activado. Esto se debe a la manera en que se disparan los rayos AO, y puede controlarse con el deslizador *Bias*. El valor *Bias* permite controlar la suavidad con la que las caras "suavizadas" aparecerán en el renderizado con AO.

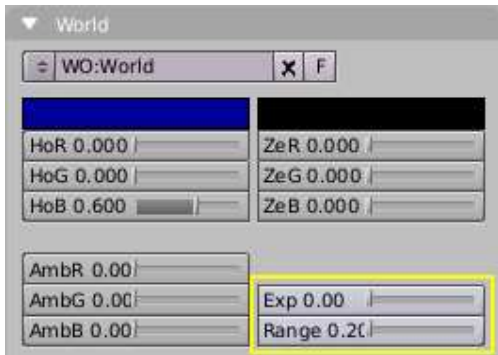
El Bias ("desviación") representa el ángulo (en radianes) que se estrechará la hemiesfera. Típicamente valores de 0.05 a 0.1 funcionan bien (*Valores de Bias de Oclusión Ambiental*).



Valores de Bias de Oclusión Ambiental.

Por favor, tenga presente que esto es sólo trazado de rayos, por lo que tiende a ser lento. Más aún, la performance depende del tamaño del Octree, ver el Capítulo de Renderizado para detalles adicionales.

Exposición y Rangoe



Exposure and Range Buttons

Los botones numéricos *Exposure* y *Range* en el Panel *World* de los botones *World* (*Botones Exposure* y *Range*) funcionan de manera análoga a la herramienta "Colour Curves" en Gimp o Photoshop.

Antiguamente Blender recortaba el color directamente en '1.0' (o 255) cuando excedía el espacio RGB posible. Esto produce feas bandas de color cuando la luz es excesiva (*Una tetera sobre-expuesta*).

Utilizando una fórmula de corrección exponencial, este defecto puede corregirse perfectamente.

- *Exp* es la curvatura exponencial, donde 0.0 es lineal, y 1.0 es curvada.
- *Range* define el rango de colores de entrada que son mapeados como colores visibles (0.0-1.0).

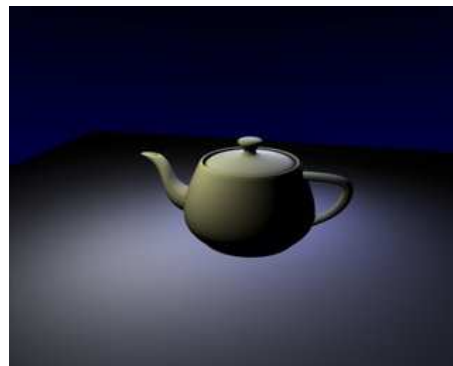
Por lo tanto, sin *Exposure* obtendremos una corrección lineal de todos los valores de color.

1. *Range* > 1.0: la imagen se torna más oscura (*Range 2.0*).
2. *Range* < 1.0: la imagen se torna más brillante (*Range 0.5*).

Con una corrección lineal, cada valor de color se cambiará, lo cual probablemente no sea lo deseado. *Exposure* aclara los píxeles más oscuros, de manera que las partes más oscuras de la imagen no se cambiarán para nada (*Range 2.0, Exposure 0.3*).



Una tetera sobre-expuesta.



Range 2.0



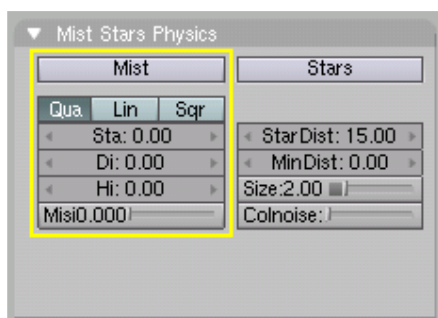
Range 0.5



Range 2.0, Exposure 0.3

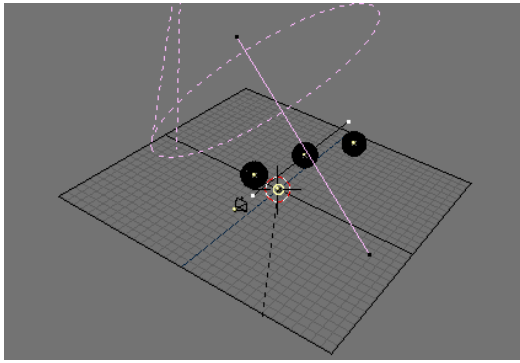
Niebla (Mist)

La niebla puede mejorar enormemente la sensación de profundidad en la imagen generada. Para crear niebla, Blender mezcla simplemente el color de fondo con el objeto del color y aumenta la intensidad del primero conforme el objeto esté más lejos de la cámara. Las opciones de Niebla están en "Mist settings" del Panel *Mist Stars Physics* y se muestran en *Botones de niebla*.



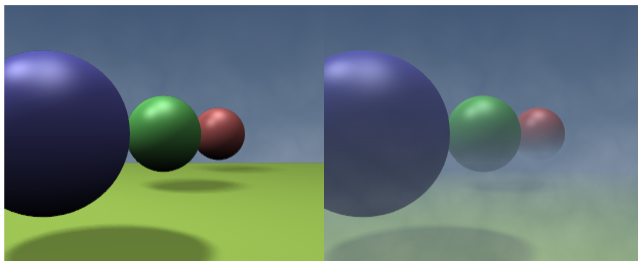
Botones de niebla

El botón *Mist* activa o desactiva la niebla. La fila de tres botones "Toggle" (conmutar) debajo de este botón fija el grado de degradación de la niebla según las funciones cuadráticas, lineal, y raíz cuadrada (*Quadratic*, *Linear*, and *Square Root*). Estas opciones controlan la regla que gobierna la intensidad de la niebla a medida que el objeto se aleja de la cámara. La niebla comienza a una distancia de la cámara definida por el botón *Sta*., y se computa según la distancia definida por el botón *Di*:. Los objetos que estén alejados de la cámara más allá de *Sta+Di* estarán totalmente ocultos por la niebla. Por defecto, la niebla cubre toda la imagen uniformemente. Para producir un efecto más realista se podría hacer que la niebla disminuyera con la altura (eje z) usando el NumButton *Hi*:. Si el valor de este botón es distinto de cero, ajusta, en unidades de Blender, un intervalo alrededor de $z=0$ en el cual la niebla cambia de intensidad máxima (abajo) a cero (arriba) Finalmente, el Numbutton *Misi*:. define la intensidad o densidad de niebla *Configuración de la niebla* muestra un ejemplo de prueba.

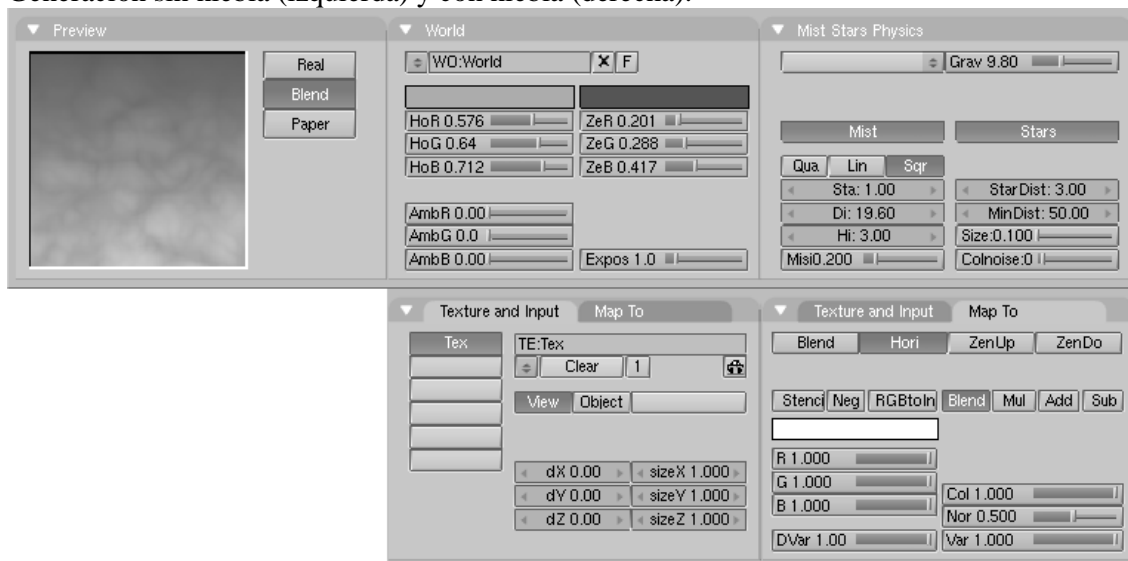


Configuración de la niebla

Generación sin niebla (izquierda) y con niebla (derecha). muestra los resultados con y sin niebla. Los valores de control se muestran en *Configuración de Mundo.*; la textura es la de una nube sencilla procedural con nivel de ruido *Hard*.



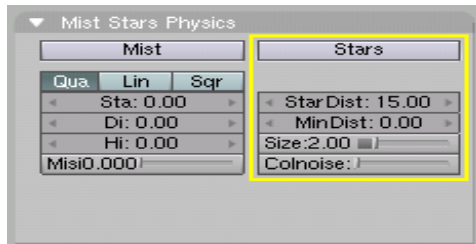
Generación sin niebla (izquierda) y con niebla (derecha).



Configuración de Mundo.

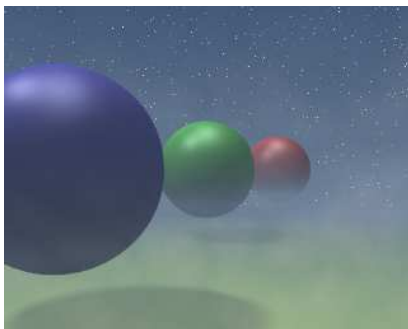
Estrellas

Las estrellas se colocan aleatoriamente como objetos con halo que aparecen en el fondo. Las opciones para estrellas también se encuentran en el Panel *Mist Stars Physics* (*Botones de Estrella*).

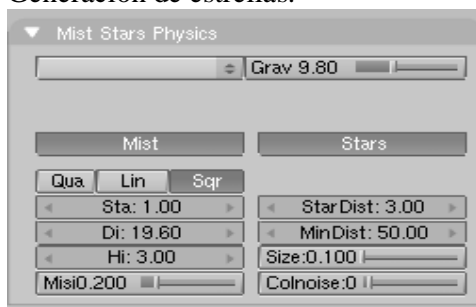


Botones de Estrella

Cuando se quieren crear estrellas, deben entenderse previamente algunos conceptos importantes: *StarDist*: es la distancia *media* entre estrellas. Las estrellas son características intrínsecamente 3D ubicadas en el espacio, no en una imagen. *Min Dist*: es la distancia *mínima* desde la cámara a la que se colocan las estrellas. Ésta debería ser mayor que la distancia desde la cámara al objeto *más alejado* de la escena, a menos que se quiera correr el riesgo de que las estrellas aparezcan *delante* de los objetos. El botón *Size*: define el tamaño real del halo de la estrella. Debe mantenerse este valor mucho más pequeño que el propuesto por defecto para mantener el material más pequeño que el tamaño del pixel y crear así estrellas hechas de un sólo punto, que son mucho más realistas. El botón *Colnoise*: añade un tono de color aleatorio a las estrellas que de otro modo son blancas. Normalmente es una buena idea añadir un poco de *ColNoise*. *Generación de estrellas.* muestra la misma imagen neblinosa que capítulo Niebla, pero con estrellas añadidas. Se muestra la configuración para las estrellas en *Configuración de las estrellas.*



Generación de estrellas.



Configuración de las estrellas.

PARTE VII - CONCEPTOS BÁSICOS DE ANIMACIÓN

Los objetos se pueden animar de varias formas. Pueden animarse como Objetos, cambiando su posición, orientación o tamaño en el tiempo; pueden animarse deformándolos; esto es, animando sus vértices o puntos de control; o pueden animarse mediante una compleja y flexible interacción con una clase especial de objeto: el Esqueleto.

En este capítulo cubriremos el primer caso, pero los conceptos básicos dados aquí son igualmente vitales para entender también los siguientes capítulos.

Hay tres métodos usados normalmente en programas de animación para hacer moverse a un objeto en 3D:

- *Key frames (fotogramas clave)* La posición completa se guarda por unidad de tiempo (frames o fotogramas). Una animación se crea interpolando fluidamente el objeto a lo largo de los fotogramas. La ventaja del método es que te permite trabajar con unidades claramente visibles. El animador puede trabajar desde una posición a la siguiente, puede cambiar posiciones creadas anteriormente, o moverlas en el tiempo.
- *Motion Curves (curvas de movimiento)* Pueden dibujarse curvas para cada componente XYZ de la posición, rotación y tamaño. Esto forma el gráfico del movimiento, con el tiempo dispuesto horizontalmente y el valor en vertical. La ventaja de este método es que te da control preciso sobre el resultado del movimiento.
- *Path (Camino)* Se dibuja una curva en un espacio 3D y se le asigna al objeto para que la recorra de acuerdo a una función de tiempo dada para la posición a lo largo del camino.

Los primeros dos sistemas en Blender se integran completamente en uno único, el *IPO* (sistema de Interpolación). Fundamentalmente, el sistema IPO consiste en curvas de movimiento estándar. Una simple pulsación de un botón cambia la IPO a un sistema de claves, sin conversión ni cambios en los resultados. El usuario puede trabajar como quiera con las claves, cambiar a curvas de movimiento y volver de nuevo, de la forma que mejores resultados le produzca o satisfaga sus preferencias personales.

El sistema IPO también tiene mucha importancia en las animaciones por Caminos.

Capítulos

- El Bloque IPO
- Los Cuadros Clave
- Curvas y Claves IPO
- Siguiendo un Camino

El Bloque IPO

El bloque IPO en Blender es universal. No diferencia si se controla el movimiento de un objeto o la configuración del material. Una vez has aprendido a trabajar con IPOs para objetos, será obvio cómo trabajar con otros IPOs. De todas formas, Blender hace distinción entre diferentes *tipos* de IPOs y la interfaz lo tiene en cuenta automáticamente.

Cada tipo de bloque IPO tiene un número fijo de *canales* disponibles. Cada uno tiene un nombre (*LocX*, *SizeZ*, etc.) que indica cómo se aplican. Cuando añades una curva IPO a un canal, la animación empieza inmediatamente. A tu albedrío (y hay canales separados para ello), una curva puede ser enlazada directamente a un valor (*LocX...*), o puede aplicarse a un incremento del mismo (*dLocX...*). Éste último te permite mover un objeto tal y como se haría normalmente arrastrándolo con la tecla G, sin interrumpir la IPO. La posición actual se determina entonces con curvas IPO *relativas* a esa posición.

El interfaz de Blender ofrece varias opciones para copiar IPOs, enlazar IPOs a más de un objeto (una IPO puede animar a varios objetos), o borrar enlaces IPO. En la sección "Ventana de Referencia de IPO" se da una descripción de tallada de esto. Este capítulo se limita a las principales opciones para su aplicación.

Fotogramas clave



Menú Insertar Clave.

La forma más fácil para crear un objeto IPO es con el comando {Literal|Insertar clave}; (TECLA-I) en la ventana 3D, con un Objeto seleccionado. Un menú emergente ofrece una amplia selección de opciones (*Menú Insertar Clave.*). Seleccionaremos la primera opción: *Loc*. Ahora se ha guardado la posición actual X-Y-Z y todo se coloca automáticamente:

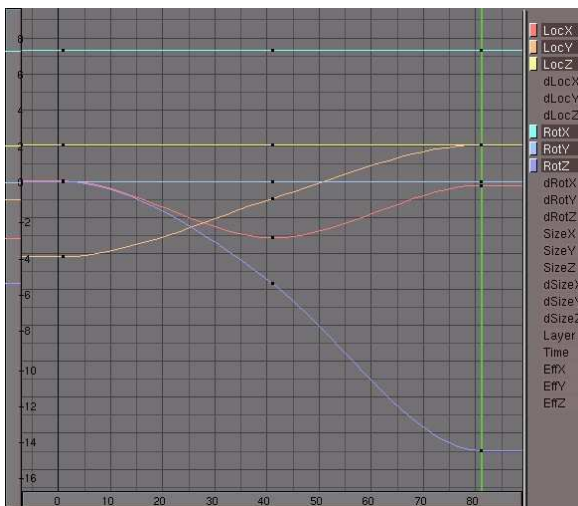
- Si no hay un bloque IPO, se crea uno nuevo y se asigna al objeto.
- Si no hay Curvas IPO en los canales *LocX*, *LocY* y *LocZ*, se crearán.
- Entonces se añadirán los vértices a las Curvas IPO con los valores exactos de la posición del objeto.

Nos movemos 30 fotogramas más lejos (pulsando tres veces la **ARRIBA**) y movemos el objeto. De nuevo usamos **TECLA-I**. Ahora podemos pulsar inmediatamente **ENTER** para que Blender recuerde nuestra última opción y la seleccione. La nueva posición se inserta en la Curva IPO. Podemos verlo lentamente volviendo hacia atrás a través de los cuadros (**DERECHA**). El objeto se mueve entre las dos posiciones. En este caso, puedes crear la

animación moviéndote a lo largo de los cuadros, posición por posición. Observa que la posición del objeto es *directamente* enlazada a las curvas. Cuando cambias fotogramas, las IPOs son siempre reevaluadas y reaplicadas. Puedes mover libremente el objeto en el mismo fotograma, pero tan pronto como cambies de fotograma, el objeto 'salta' a la posición que había sido determinada por el IPO. La rotación y el tamaño del objeto son completamente independientes en este ejemplo. Pueden ser cambiadas o animadas con el procedimiento de Insertar Clave seleccionando del menú que aparece al pulsar la **TECLA-I** las otras opciones como Rotación (Rotation), Tamaño (Size) y cualquiera de sus combinaciones.

Las Curvas y las Claves IPO

Las Curvas IPO



La ventana IPO.

Ahora queremos ver exactamente qué ha pasado. La primera ventana que aparece al cargar el archivo estándar de inicio de Blender es estupenda para esto. Actívala con **CTRL-LEFTARROW**. A la derecha podemos ver la Ventana IPO representada (*La ventana IPO.*). Desde luego, puedes convertir cualquier ventana en una ventana IPO eligiendo este tipo en el menú de Tipo de Ventana, pero es más manejable tener la vista 3D y la ventana IPO a la vez. Ésta muestra todas las Curvas IPO, los canales usados y los disponibles. Puedes hacer zoom en la Ventana IPO y moverla tal y como se haría en cualquier otra ventana de Blender.

Además de los canales estándar, que pueden ser inicializados pulsando la tecla **I**, tienes las opciones *delta*, como la *dLocX*. Estos canales te permiten asignar un cambio *relativo*. Esta opción se usa principalmente para controlar múltiples objetos con la misma IPO. Además, es posible trabajar con "capas" de animación. Puedes conseguir efectos sutiles de esta forma sin tener que dibujar curvas complicadas.

Cada curva puede seleccionarse individualmente con el botón derecho del ratón (**RMB**). Además, los modos de Mover y Escalar funcionan exactamente como en una vista 3D. Puedes seleccionar IPOs también seleccionando el botón coloreado en la columna de nombres de canales de la derecha. Seleccionando el nombre de canal IPO muestras u ocultas la curva

relativa. Seleccionando todas las curvas (con la tecla "a", A) y moviéndolas a la derecha (con la tecla "g" G), puedes mover y completar la animación a lo largo del tiempo.

Cada curva puede ser posicionada individualmente en modo edición, o puede hacerse colectivamente. Selecciona las curvas y pulsa TAB. Ahora se muestran los vértices individuales y los *manejos* de la curva. Los manejos de Bézier están programados tal y como aparecen en el Objeto Curva:

- Free Handle (Manejo Libre, en negro). Puede usarse para lo que quieras. Atajo de teclado: tecla H (HKEY) (cambia entre Libre y Alineado).
- Aligned Handle (Manejo Alineado, en rosa). Esto coloca todos los manejos en línea recta. Atajo: tecla H (HKEY) (cambia entre Libre y Alineado).
- Vector Handle (Manejo por Vector, en verde). Las dos partes del manejo siempre apuntan hacia el manejo anterior o siguiente. Atajo: Tecla V (VKEY).
- Auto Handle (Auto Manejo, en amarillo). Este manejo tiene una dirección y longitud totalmente automática. Atajo: Mayúsculas+Tecla H (SHIFT-HKEY).

Los manejos pueden moverse seleccionando primero el vértice central con el botón derecho del ratón (RMB). Esto selecciona también los otros dos vértices. Inmediatamente después puedes moverlos manteniendo pulsado el RMB y moviendo el ratón. Los manejos pueden *rotarse* seleccionando primero el final de uno de los vértices y después moviéndolo manteniendo pulsado el RMB y moviendo el ratón.

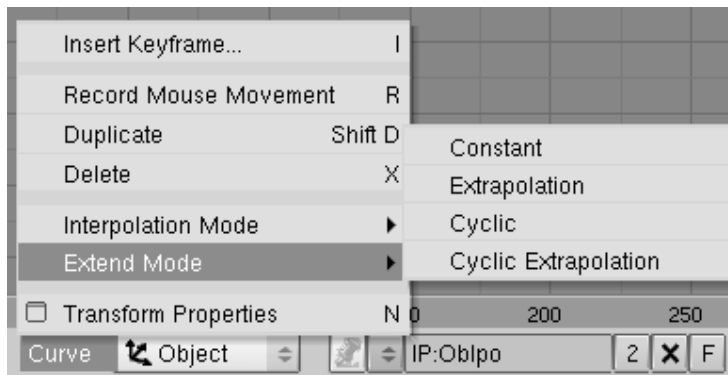
En cuanto los manejos son rotados, su tipo cambia automáticamente:

- Auto Handle se vuelven Aligned.
- Vector Handle se vuelven Free.

Los "Auto" manejos se colocan por defecto en la curva. El primer y último Auto manejo siempre se mueve horizontalmente, lo que crea una interpolación fluida.

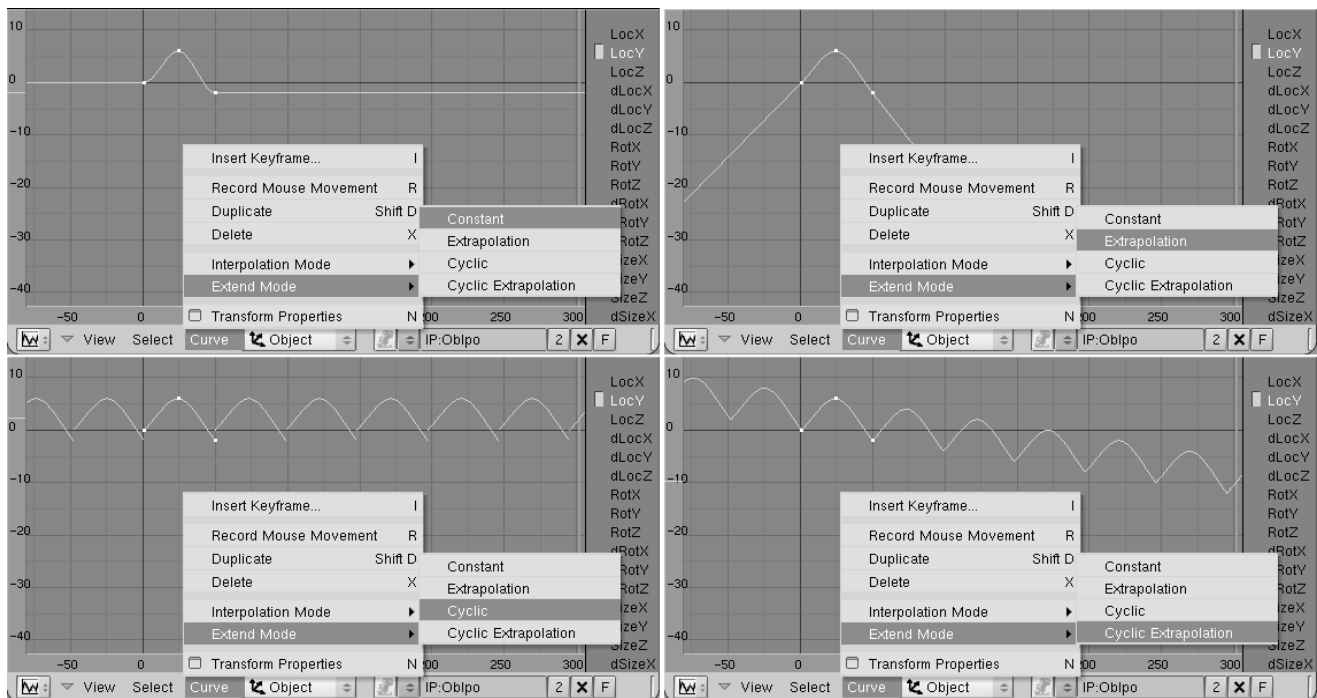
Las Curvas IPO tienen una característica importante que las distingue de las curvas normales: es imposible poner más de un segmento de curva horizontalmente. Ciclos y círculos en una IPO son ambiguos y no tienen sentido. Una IPO sólo puede tener un valor en un determinado momento. Esto se detecta automáticamente en la Ventana IPO. Al mover parte de la Curva IPO horizontalmente, puedes ver que los vértices seleccionados se mueven "a lo largo" de la curva. Esto te permite duplicar partes de una curva (SHIFT-D) y moverla hasta otro cuadro de tiempo.

También es importante especificar cómo una Curva IPO debe ser interpretada *desde fuera* de la misma curva. Hay cuatro opciones para esto en el Submenú *Curve>>Extend Mode* de la cabecera de la ventana IPO (*IPO extension options.*).



Modo Extendido.

El efecto de cada una puede apreciarse en (*IPOs extendidos.*).



IPOs extendidos.

De izquierda a derecha:

- Extend mode Constante: Los extremos de la Curva IPO seleccionada son extrapolados de forma continua (horizontalmente). Es el comportamiento por defecto.
- Extend mode Extrapolación: Los extremos de la Curva IPO seleccionada continúan en la dirección en la que fueron terminados.
- Extend mode Cíclica: La anchura completa de la Curva IPO se repite cíclicamente.
- Extend Mode Extrapolación Cíclica: La anchura completa de la Curva IPO se extrapola cíclicamente.

Además de las Béziérs, hay otros dos posibles tipos de Curvas IPO. Usa el comando de la tecla "T" (TKEY), y el diálogo emergente que surge, o la entrada del submenú *Curve>>Interpolation Mode* para elegirla. La interpolación de las Curvas IPO seleccionadas pueden ser:

- *Constante* - tras cada vértice de la curva, su valor permanece constante. No hay interpolación..
- *Lineal* - se interpola linealmente entre los vértices.
- *Bezier* - la interpolación suave estándar.

No hay por qué crear la Curva IPO sólo mediante Cuadros Clave. También se pueden dibujar "a mano". Usa el comando CTRL-LMB. Las reglas son las siguientes:

- No existe aún un bloque IPO (en esta ventana) y un *canal* está seleccionado: se crea un nuevo Bloque IPO a lo largo de la primera Curva IPO, cuyo primer vértice se coloca donde se pulsó el ratón.
- Ya existe un bloque IPO, y se ha seleccionado un *canal* sin una curva IPO: se añade una nueva Curva IPO con un vértice.
- Ya hay un Bloque IPO, y se ha seleccionado un *canal* que tiene una curva IPO: Se añade un nuevo punto a la Curva IPO seleccionada. Esto *no* es posible si hay varias Curvas IPO seleccionadas o en Modo Edición.

{{Note|Haz rotar un objeto| Éste es el mejor método para especificar ejes de rotación rápidamente: Selecciona el objeto; en la Ventana IPO, pulsa uno de los canales "Rot" y utiliza CTRL-LMB para insertar dos puntos. Si el eje de rotación ha de ser continuo, debes usar la entrada del menú *Curve>>Extend Mode>> Extrapolation*.

Una desventaja de trabajar con curvas de movimiento es que la *libertad* de las transformaciones está limitada. Puedes trabajar bastante intuitivamente con curvas de movimiento, pero sólo si ésta puede ser procesada en una base XYZ. Para una posición esto es trivial, pero para un tamaño o rotación hay disponibles mejores descriptores: matrices (3x3 números) para el tamaño y cuaterniones (4 números) para rotaciones. Éstas también podrían procesarse en los canales, pero conllevaría fácilmente a situaciones confusas y matemáticamente complicadas.

Limitar el *tamaño* a los tres números XYZ es obvio, pero esto limita a una distorsión rectangular. Un escalado diagonal como un "esquilado" (shearing) es imposible. Simplemente trabajando con jerarquías puede resolverse esto. Un objeto padre escalado *no*-uniformemente influenciaría la rotación de un objeto hijo como en un "esquilado".

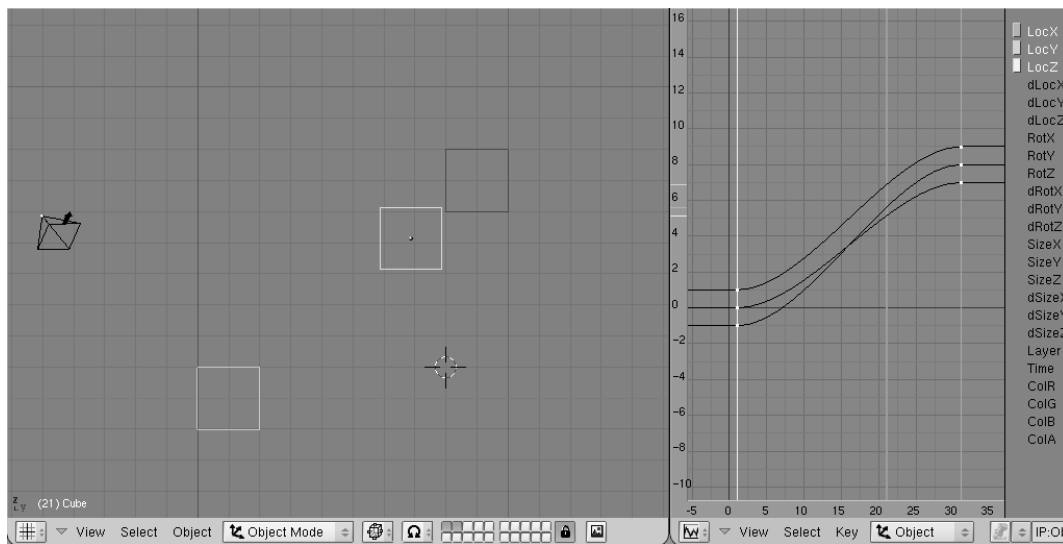
La limitación a las rotaciones de tres números XYZ es menos intuitiva. La llamada "rotación euleriana" no es uniforme (la misma rotación puede expresarse con diferentes números) y tiene el fastidioso efecto de que *no* es posible rotar desde cualquier posición a cualquier otra por el famoso "*gimbal lock*". Mientras está trabajando con distintas claves de rotación, el usuario puede enfrentarse de repente con interpolaciones inesperadas, o puede volvérselo imposible efectuar una particular rotación sobre un eje cuando hace cambios manuales. Aquí,

de nuevo, una solución mejor es trabajar con jerarquías. Un objeto padre *siempre* asignará el eje de rotación especificado al hijo. (Es conveniente saber que las rotaciones en X, Y y Z se calculan una *detrás* de la otra. La curva que afecta al canal *RotX* *siempre* determina la rotación en el eje X).

Afortunadamente, Blender calcula todo internamente con matrices y cuaterniones. Las jerarquías trabajan así normalmente, y el modo Rotación hace lo que tendría que hacer. Sólo las IPOs son una limitación aquí, pero en este caso la facilidad de uso prevalece sobre una no demasiado intuitiva pureza matemática.

Las Claves IPO

La forma más fácil de trabajar con curvas de movimiento es convertirlas a claves IPO. Volvemos a la situación del ejemplo anterior: hemos especificado dos posiciones en un objeto IPO en el cuadro 1 y el 31 con la tecla "i" (IKEY). A la derecha de la pantalla, puedes ver una ventana IPO. Vamos a situarnos en el frame 21 (*El modo Clave IPO*).



El modo Clave IPO.

Pulsa la tecla "k" (K) mientras el cursor del ratón está en la ventana 3D. Pasarán dos cosas:

- La Ventana IPO cambia al modo clave de IPO.
- Al objeto seleccionado se le asigna la opción de "dibujado de clave" (DrawKey).

Las dos acciones tienen significados distintos.

- La ventana IPO ahora dibuja líneas verticales a lo largo de todos los vértices de todas las curvas IPO visibles (las IPOs ahora son negras). Los vértices con el mismo valor de "cuadro" están relacionadas a las líneas verticales. Las líneas verticales (las "claves IPO") pueden seleccionarse, moverse o duplicarse, al igual que los vértices en Modo de Edición. Puedes mover las claves IPO únicamente en horizontal.

- El objeto no sólo se muestra en su posición actual, sino que también se muestran objetos "fantasma" en todas las posiciones Clave. Además de poder ver las posiciones clave del objeto, también puedes modificarlas *en* la ventana 3D. En este ejemplo, usa el modo de movimiento (Grab) en el objeto para cambiar la clave IPO *seleccionada*.

Abajo hay unas cuantas instrucciones para utilizar mejor la potencia del sistema:

- Sólo puedes usar el **RMB** para seleccionar claves IPOKeys en la ventana de IPO. La selección por borde y la selección *extendida* también funcionan aquí. Selecciona todas las claves IPO para transformar el sistema de animación completa en la ventana 3D.
- La "Inserción de Clave" (Insert Key) siempre afecta a *todos* los objetos seleccionados. La Clave IPO para múltiples objetos también puede transformarse simultáneamente en la ventana 3D. Usa el comando **SHIFT-K: Muestra y selecciona todas las claves** para transformar animaciones completas de un grupo de objetos todos a la vez.
- Usa los comandos **PAGEUP** y **PAGEDOWN** para seleccionar claves subsiguientes en la ventana 3D.
- Puedes crear claves IPO en cada conjunto de canales. *Excluyendo* a consciencia ciertos canales, puedes forzar una situación en la cual los cambios en las posiciones clave de la ventana 3D sólo afectarán a los valores especificados por los canales visibles. Por ejemplo, con sólo el canal *LocX* seleccionado, las claves sólo se moverán en la dirección X.
- Cada Clave Ipo está formada por los vértices que tienen *exactamente* el mismo valor de cuadro. Si los vértices se mueven manualmente, puede pasar que tengamos un enorme número de claves y cada una teniendo sólo una curva. En este caso, usa el comando "Unir" (Join) pulsando la tecla "j" (JKEY) para combinar las Claves IPO seleccionadas. También es posible asignar vértices seleccionados de Claves Ipo a *todas* las curvas visibles: usa la tecla "i" (I) en la ventana de IPO y elige "Selected keys".
- La opción de Dibujado de Claves (DrawKey) y el modo Clave IPO pueden ser activados y desactivados independientemente. Usa los botones EditButtons->DrawKey para desactivar esta opción u objeto. Puedes activar y desactivar el modo Clave IPO tú mismo con la tecla "k" (K) en la ventana de IPO. Sólo pulsando **K** en la ventana 3D activa o desactiva a la vez los modos Dibujado de Clave y Clave IPO.

Otras aplicaciones de las Curvas IPO

Relevante para Blender v2.31 Hay muchas otras aplicaciones para las IPOs aparte de la simple animación del movimiento de objetos. Los botones del menú "IPO Type" de la cabecera *La ventana IPO*. permiten la selección del tipo del Bloque IPO, la activa ahora es el Objeto IPO descrito hasta ahora, pero hay Material IPO, Mundo (World) IPO, Claves de Vértices (Vertex Keys) IPO, Restricciones (Constraints) IPO y Secuencia (Sequence) IPO. No todas las opciones están siempre presentes, dependiendo del contexto. El bloque Curva IPO aparece si el objeto seleccionado es una curva y no una Malla; la Luz (Lamp) IPO aparece sólo si el objeto seleccionado es una luz.

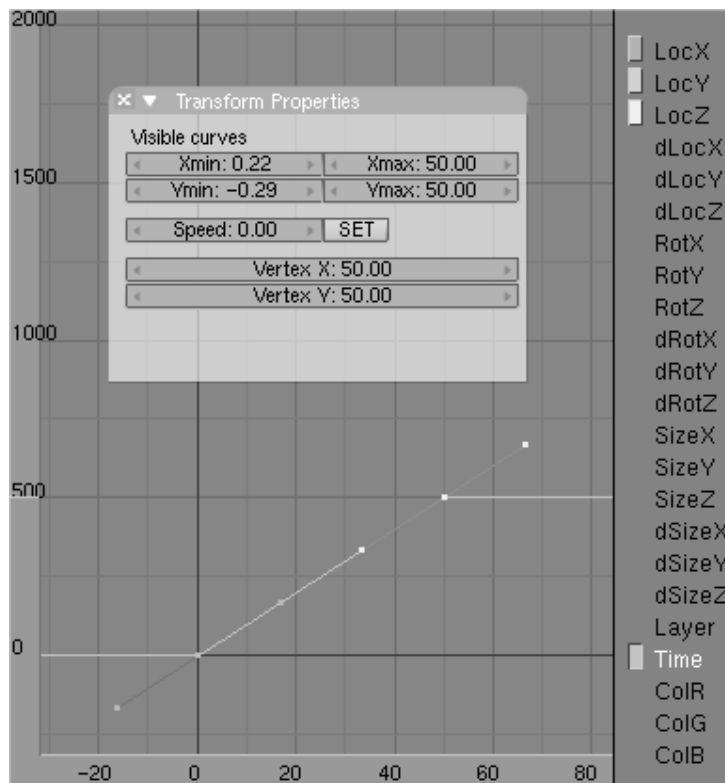


La ventana IPO.

El Material IPO es una forma de animar un Material. Como con los objetos, las Curvas IPO pueden usarse para especificar 'posiciones clave' para Materiales. Con el ratón en la ventana de botones (ButtonsWindow), pulsando la tecla "i" (IKEY) abre un menú emergente con opciones para las distintas variables del Material. Si estás en el bloque IPO de Material, Luz (Lamp) o Mundo (World) entonces un pequeño botón numérico (Num Button) aparece al lado del Menú de tipo de IPO en la barra de herramientas de la ventana IPO. Esto indica qué canal de textura está activo. ¡El mapeado para los 8 canales puede ser controlado con Curvas IPO! Estrictamente hablando, se puede animar de otras dos formas usando texturas. Dado que hay objetos que pueden asignar las coordenadas de texturas a otros objetos (cada objeto en Blender puede ser usado como una fuente de coordenadas de textura. Para hacer esto, debe seleccionarse la opción objeto (Object) en el botón verde titulado Origen de Coordenadas (Coordinates input) y escribir el nombre del objeto en él. Se realizará una transformación inversa en las coordenadas de renderización globales para obtener las coordenadas *locales* del objeto), es posible animar la textura simplemente animando la posición, tamaño y rotación del objeto. Además, en cada cuadro (frame), Blender puede cargar otra imagen (numerada) como mapa de texturas en lugar de tener una sola imagen fija. También es posible usar archivos de películas en formato SGI o AVI para hacer esto.

La Ipo del tiempo

Con la curva Ipo del tiempo (llamada en adelante Ipo-tiempo) puede manipular el tiempo de los objetos en la animación sin variar la animación u otras curvas Ipos. De hecho, dicha curva cambia el mapeado del tiempo de la animación a tiempo global (*Curva Ipo-tiempo lineal*).



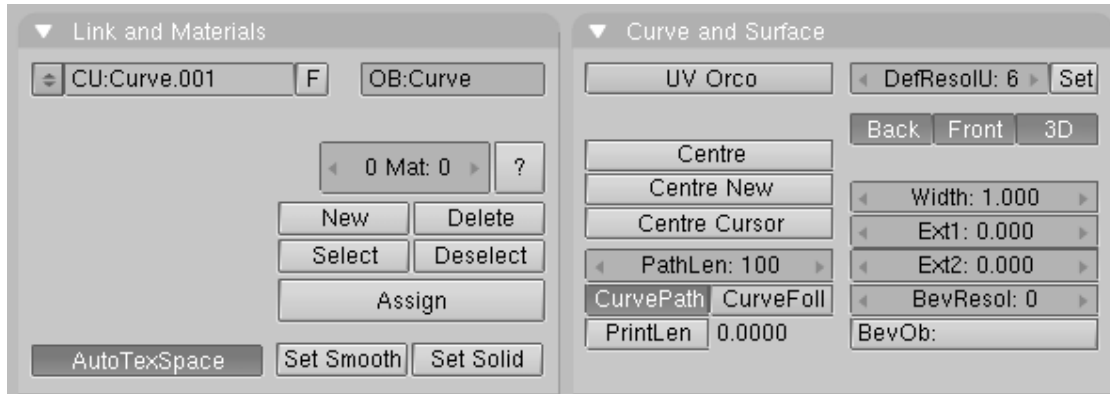
Curva Ipo-tiempo lineal

Para asimilar este concepto, haga una sencilla animación consistente en mover un objeto, desde una posición a otra en, digamos, 50 frames (imagenes). Entonces seleccione el canal *Time* (Tiempo) y cree una Ipo-Tiempo en la Ventana Ipo que vaya desde el punto (1,1) al punto (50,50). Es sencillo configurar el comienzo y el final de una IPO usando la Tecla **N** e introduciendo los valores numericamente. En los frames donde la pendiente de la Ipo-Tiempo sea positiva, su objeto avanzará en la animación. La velocidad depende del valor de la pendiente. Una pendiente más grande que 1 provocará una animación más rápida que la animación base. Una pendiente inferior a 1 provocará que la animación sea más lenta. Una pendiente igual a 1 no producirá cambios en la velocidad de la animación, y una pendiente negativa hará que la animación vaya hacia atrás. La Ipo-Tiempo es especialmente interesante para sistemas de partículas, permitiendo "congelar" las partículas o animar partículas para que sean absorbidas en lugar de emitidas. Otras posibilidades son crear un lapso de tiempo o una animación a cámara lenta.

Multiples curvas IPO-tiempo: Necesita copiar la curva Ipo-Tiempo para todo el sistema de animación para obtener un movimiento a cámara lenta. Pero deteniendo unicamente algunas animaciones, y continuando la animación, por ejemplo, de la cámara puede conseguir algunos efectos muy interesantes (como esos usados para simular que el tiempo va más lento en la película "The Matrix")

Trayectoria de la animación

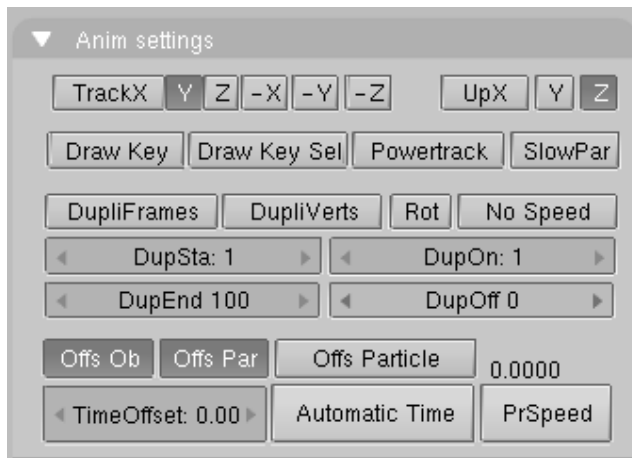
Una manera diferente para tener objetos moviéndose en el espacio consiste en restringir su movimiento para que siga una trayectoria dada. Cuando los objetos necesitan seguir una trayectoria, o es demasiado difícil realizar alguna clase de movimiento especial con el método del keyframe (Piense en un planeta siguiendo su movimiento alrededor del Sol. Animar esto con keyframes es virtualmente imposible) los objetos curva pueden usarse para indicar la trayectoria de la animación. Si el objeto Curva contiene más de una curva solo la primera de ellas, en el objeto, será usada.



La ventana de acciones con los botones de trayectorias.

Hay dos métodos de animación con trayectorias, el antiguo, anterior a Blender 2.30 que es descrito en esta sección y el método nuevo, el cual actualmente define una restricción, y será descrito *character_constraints*. Cuando emparente un Objeto a una Curva tendrá la opción de elegir entre *Normal Parent* (*emparentado normal*) o *Follow Path* (*seguir una trayectoria*). El método antiguo es el que necesita para realizar una tradicional animación que sigue una trayectoria, pero otras acciones necesitarán llevarse a cabo después. La segunda opción crea una restricción del tipo "sigue la trayectoria", y es todo lo que necesita hacer. Cualquier clase de curva puede convertirse en una trayectoria activando o poniendo a "ON" la opción *CurvePath* en la ventana de botones de animación (F7) *La ventana de acciones con los botones de trayectorias..* Cuando una curva tiene hijos, esta puede convertirse a una trayectoria seleccionandola, yendo al contexto de Edición (F9) y activando el botón *CurvePath* en el panel "Curve and Surface". Ahora los objetos hijos de la curva se moverán a lo largo de la trayectoria especificada. Es una buena idea convertir la curva en tridimensional a través del botón *3D* de los botones de edición de curvas para que de esta forma la curva pueda ser libremente modelada. De otra manera, en el menú 'ADD' bajo Curve->Path, existe ya una primitiva con la configuración correcta. Es una curva NURBS de quinto orden, que puede ser usada para crear un muy fluido y continuo movimiento. Normalmente una trayectoria tiene una longitud de 100 frames y es seguida por el hijo en esos 100 frames. Puede hacerla más larga o más corta variando el botón *PathLength*:. La *velocidad* a lo largo de una trayectoria es determinada con una curva en la ventana IPO. Para visualizarla, en la ventana IPO debe seleccionar el tipo *Curve* (*curva*) para el bloque IPO. Un único canal, es la *velocidad*. La trayectoria completa recorre en la ventana IPO los valores verticales de cero a uno. Trazar una curva entre esos valores le permitirá enlazar el tiempo a la posición de la trayectoria. Para la mayoría de las trayectorias, una curva IPO debe moverse *exactamente* entre los valores de 0.0 a 1.0 en Y. Para realizar esto, use el menú numérico (Tecla N) en la ventana IPO. Si la curva IPO es eliminada, el valor de *PathLen* determina la duración de la

trayectoria. En este caso se definirá un movimiento lineal. La velocidad de la IPO es una manera delicada de controlar la longitud de la trayectoria. La trayectoria tiene una longitud de 1 para la IPO-tiempo, y si la IPO-tiempo va de 0 a 1 en 200 frames entonces la trayectoria tiene 200 frames de longitud. Usando la opción *CurveFollow (Seguir curva)*, en el panel *Curve and Surface*, se proporcionará una rotación a los objetos hijos de la trayectoria, de tal manera que apunten permanentemente en la dirección de la trayectoria. Use los botones "tracking" en el panel *Anim settings* del contexto Objeto (F7) para especificar el efecto de la rotación (*Botones de Tracking*) que vaya hacer para el Tracking:

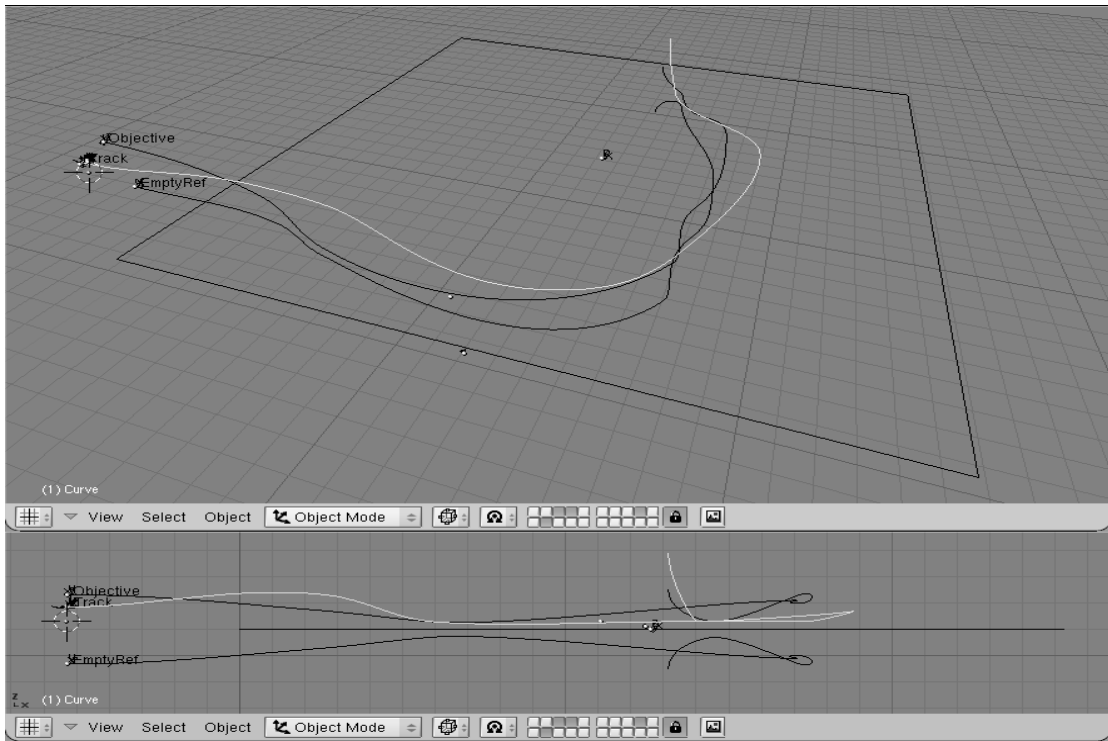


Botones de Tracking

TrackX, Y, Z, -X, -Y, -Z Esto especifica la dirección del eje, en el ejemplo el eje está situado en la trayectoria. *UpX, UpY, UpZ* Especifica que eje debe apuntar "hacia arriba", en la dirección del eje local positivo Z. Si los ejes *Track (pista)* y *Up (hacia arriba)* coinciden se desactiva.

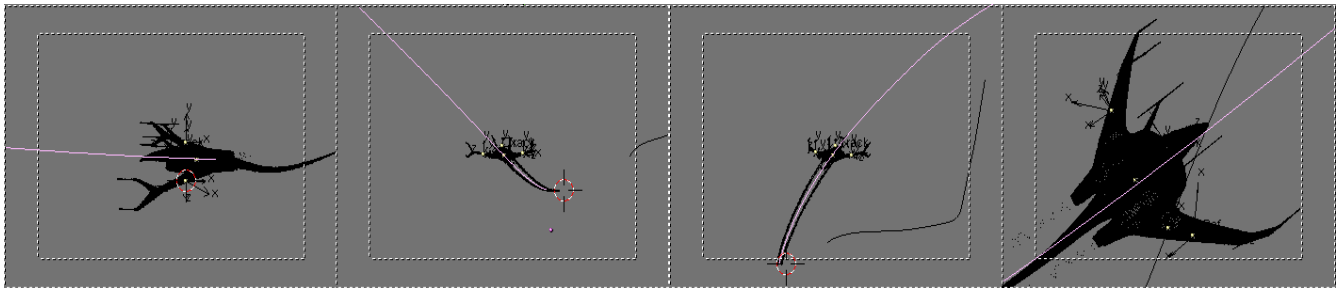
Note: Las trayectorias tienen el mismo problema que las curvas 'Bevel' en lo concerniente a la definición de la dirección 'hacia arriba'.

Para visualizar precisamente esas rotaciones, debemos posibilitar que un hijo tenga sus propias rotaciones. Elimine la rotación del hijo con **ALT-R**. También elimine la "Parent Inverse": **ALT-P**. El mejor método es emparentar un hijo sin rotación a la trayectoria con el comando **SHIFT-CTRL-Tecla P**: "Make parent without inverse (Hacer padre sin inversa)". Ahora el hijo salta directamente a la trayectoria y los puntos del hijo están en la dirección correcta. Las trayectorias tridimensionales tienen un valor extra por cada vértice: el 'tilt (inclinación)'. Esto puede ser usado específicamente en la rotación de un eje. Use la Tecla **T** en modo edición para cambiar el 'tilt' de los vértices seleccionados en modo edición, por ejemplo para tener un hijo moviéndose alrededor como si fuera una peonza. La *Trayectoria compleja de animación* muestra una aplicación compleja. Queremos hacer que un avión vaya a través de un cañón, volando próximo al agua y entonces levante el vuelo, todo esto siguiéndolo con nuestra cámara y, posiblemente, teniendo reflejos en el agua! Para conseguir esto necesitaremos tres trayectorias. La trayectoria 1 tiene un avión emparentada a ella, es decir, el avión volará siguiéndola.



Trayectoria compleja de animación

El avión tiene un Empty llamado 'Track (pista)' emparentado a él en una posición estratégica. Una cámara es entonces emparentada a otra curva, la trayectoria 2, y la sigue, siguiendo el Empty 'Track'. El avión tiene la *velocidad IPO* constante, mientras que la cámara no. Primero va rápido, luego despacio y siempre siguiendo el empty, y por lo tanto el avión, así nosotros tendremos unos movimientos muy fluidos de la cámara desde el lateral del avión, al frontal del avión, al otro lado, desde atrás, etcétera (*Algunas imágenes de la cámara siguiendo al avión.*).



Algunas imágenes de la cámara siguiendo al avión.

Desde que nosotros queremos que nuestro avión sobrevuele un río, necesitaremos configurar un mapa entorno para obtener reflejos de la superficie del agua. Pero el Empty usado para los cálculos siempre tiene que estar en una posición especular con respecto a la cámara... Y la cámara se está moviendo a lo largo de una trayectoria!. La trayectoria 3 es por lo tanto creada como un espejo de la trayectoria 2 con respecto al plano el agua, duplicándola, y usando la Tecla M, en modo edición con respecto a cursor, una vez que cursor esté en el plano. El Empty para el cálculo del mapa de entorno es entonces emparentado a su nueva trayectoria, y la IPO-tiempo de la trayectoria 2 es copiada a la trayectoria 3. La *Una imagen de la animación final.* nos muestra una imagen renderizada. Algunos sistemas de partículas se

usaron para crear la estela. La escena presenta muchos trucos pequeños, como las partículas para los motores del avión, la niebla, un cielo añadido a una esfera rodeando la escena y así.



Una imagen de la animación final.

VIII - DEFORMACIONES BÁSICAS

Animar un Objeto/Material no es lo único que puede hacerse en Blender. ¡Pueden cambiarse, remodelarse o deformarse los objetos en el tiempo!

De hecho, hay muchas maneras de conseguir esto, y una de estas técnicas es tan poderosa y general, que hay un capítulo entero para ella: Animación de caracteres. El resto de técnicas son cubiertas aquí.

Capítulos

- Garfios
- Claves de Forma
- Claves de Vértices Absolutas
- Claves de Vértices Relativas
- Animación de Enrejados

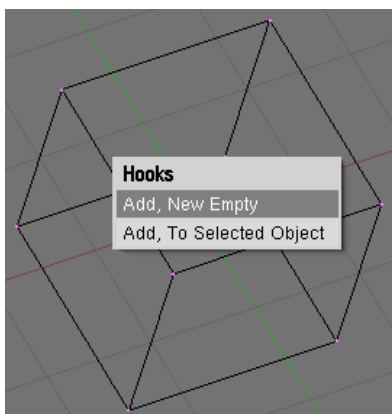
Garfios

Los Garfios (Hooks) proveen acceso a nivel de objeto, a la geometría subyacente de mallas, curvas, superficies o retículas. Un Garfio, es una característica de objeto, y es como el padre de un objeto, pero para vertices. En un objeto puede crear tantos garfios como quiera, y asignar para cada uno distintos vertices de influencia. También es posible superponer garfios, en cuyo caso se dispone de un factor de ponderación de peso para cada uno, que determina el nivel de influencia de cada garfio sobre los vertices compartidos.

Nota: Cuando remodela completamente algún objeto, muy probablemente también deba reasignar los garfios existentes.

Agregando garfios

Como los garfios están relacionados con vértices o puntos de control, la mayoría de las opciones de edición se encuentran disponibles en el ModoEdición para mallas, curvas, superficies y retículas. Seleccione un número cualquiera de vertices y presione **CTRL-H** para acceder al menú Garfios.



Menú de garfios

- *Add, New Empty* Crea un nuevo garfio y un nuevo objeto Empty, que será el padre de la selección y será ubicado en el centro de la misma.
- *Add, To Selected Object* Cuando hay seleccionado otro objeto (puede hacerlo en ModoEdición con **CTRL-RMB**) se crea el nuevo grafio y se lo emparenta a ese objeto.

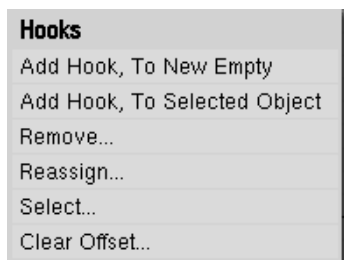
Usando garfios

Dentro de Modo Edición los garfios se deshabilitan para permitir un modelado cómodo. Los garfios pueden usarse únicamente estando en Modo Objeto. Todas las opciones y transformaciones a nivel objeto son posibles, incluyendo el uso de jerarquías, restricciones, y animaciones de recorrido y de curvas IPO.

También puede hacer que el padre garfio sea un hijo del objeto original si no desea que las transformaciones del objeto deformen los garfios.

Opciones de Modo Edición

Una vez que en un objeto hay garfios disponibles, el menú **CTRL-H** nos ofrecerá opciones adicionales:

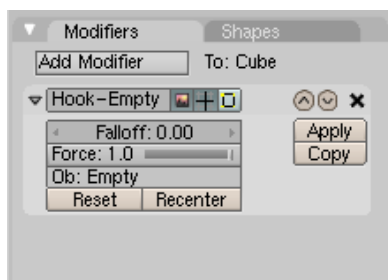


Menú extendido de garfios

- *Remove...* (Eliminar...) Nos ofrecerá un nuevo menú con una lista de garfios para eliminar.
- *Reassign...* (Reasignar...) Usela si desea reasignar nuevos vertices al garfio.
- *Select...* (Seleccionar...) Para seleccionar los vertices de un garfio determinado.
- *Clear Offset...* (Borrar Desplazamiento...) Neutralizará las trnasformaciones actuales del padre garfio.

Panel Hooks

Puede encontrar botones para Garfios en la pestaña *Hooks* del contexto de objeto (F7). Aquí se le puede dar un nombre nuevo al garfio, el nombre por omisión es el nombre del padre; se le puede asignar un nuevo padre escribiendo el nombre del nuevo padre, o se le puede modificar el factor de ponderación de peso *Force*.



Panel de Garfios

- *Force* (Fuerza) Como varios garfios pueden actuar sobre los mismos vertices, con este control puede ponderar la influencia de cada garfio. Las reglas de ponderación son:
 - Si la suma total de todas las fuerzas es menor que 1.0, el resto, 1.0 - Fuerzas, será el factor que la posición orginal tiene como fuerza.
 - Si la suma total de todas las "fuerzas" es mayor que 1.0, solo se utilizan las transformaciones producidas por garfios, promediando sus pesos.
- *Falloff* Si es distinto de cero, el falloff es la distancia a la que la influencia de un garfio vale cero. Actualmente emplea un ainterpolación suave, parecida a la *Herramienta de Edición Proporcional*. (Ver *Herramienta de Edición Proporcional*)
- *Delete* Borra el garfio del objeto.
- *Clear offset* Neutraliza la transformación actual de un garfio.

Claves de Forma



Un ejemplo de Claves de Forma combinadas

Las Claves de Forma permiten el almacenamiento de formas relativas a partir de una malla base. Otras aplicaciones 3D pueden llamar a esto 'objetivos de morph'. Estas formas pueden ser mezcladas en base a un porcentaje con otras claves de forma para conseguir el efecto deseado. Tomemos por ejemplo una cara humana. El usuario puede modelar una cara con una expresión neutral y tener claves de forma para la sonrisa, el ceño fruncido, el ojo izquierdo cerrado, el ojo derecho cerrado, etc. Entonces, las claves de forma tendrían que permitir la combinación de ellas. Así, la malla podría estar sonriendo con su ojo izquierdo cerrado completamente y su ojo derecho cerrado al 50%.

Uso

Creando Claves de Forma

Insertando una Malla Base

(nótese que a lo largo de este documento nos referiremos a las Claves de Forma como Claves o Formas, invariablemente)

Empezando con una malla básica, el usuario habilita las claves de forma insertando una clave base. Ello se consigue seleccionando el objeto malla y presionando **I**; luego, seleccionando *Mesh* (malla). Alternativamente, este paso puede hacerse seleccionando el objeto malla, y luego haciendo clic en el botón *Add Shape Key* (añadir clave de forma) del panel *Shapes* (formas) en los Botones de Edición (**F9**).

Insertando Claves de Forma adicionales

Una vez la clave de forma base ha sido creada, el usuario puede crear claves de forma adicionales del mismo modo. Con la malla seleccionada, bien presionando **I** y luego seleccionando *Mesh*, o bien haciendo clic en el botón *Add Shape Key*. En el panel *Shapes*, se creará una nueva Clave. Si es la primera forma creada después de la forma base, se denominará *Key 1* (clave 1).

Una cosa a tener en cuenta aquí: al crear una nueva forma, ésta estará basada en la forma seleccionada actualmente. Al crear formas a partir de la forma base, la nueva forma será una copia de ésta; si el usuario crea una forma nueva mientras otra forma alterada está seleccionada, la nueva forma será una copia de esa forma alterada.

Las claves adicionales pueden ser renombradas fuera de sus nombres automáticos haciendo clic en el nombre de la Forma en el panel *Shapes*.

Alterando las Claves de Forma

Una vez la clave adicional ha sido creada, puede ser editada seleccionándola desde el panel *Shapes*. Esto puede realizarse de 2 maneras. Puede escogerse desde el menú emergente, o bien ir pasando por las diferentes claves usando los botones de flecha.



Situación de las herramientas de selección de Claves de Forma en el panel *Shapes*

Cuando la clave deseada es seleccionada, ésta puede ser editada entrando en modo Edición TAB y moviendo los vértices.

Nota: Añadir o borrar un vértice una vez las claves de forma han sido añadidas puede ser problemático. Los cambios se propagan a las otras formas basándose en su posición en la forma actual; ello puede tener un efecto drástico en otras formas.

Al seleccionarse las formas en el panel *Shapes*, las formas alteradas son mostradas en el objeto malla.

Enclavando

Cuando un objeto malla no tiene una forma enclavada, puede mostrar múltiples formas al mismo tiempo. Para evitarlo, haga clic en el icono enclavar.



Situación del Icono Enclavar en el panel *Shapes*

Esto encadenará el objeto malla actual a esa clave de forma. Una vez enclavado, el objeto malla sólo mostrará esa clave. Esta característica es útil cuando múltiples claves están afectando al objeto malla y el usuario desea ver el efecto de una clave en particular.

Otro uso del enclavamiento es para crear una galería de formas. Puesto que Blender permite la creación de Duplicados Enlazados que comparten datos subyacentes (como datos de malla y de forma), pueden crearse múltiples copias de una malla (**Alt D**). Una vez se ha creado el duplicado, puede ser enclavado a una clave de forma. Si se realiza lo mismo para cada clave, tendremos una especie de galería para ver y editar claves fácilmente.

Combinando Claves de Forma

Las Claves de Forma pueden ser combinadas de varias maneras. Estos métodos tienen el mismo efecto neto que creando datos IPO, pero se diferencian en la interfaz.

Panel *Shapes* (formas)



Panel *Shapes* y Claves de Forma

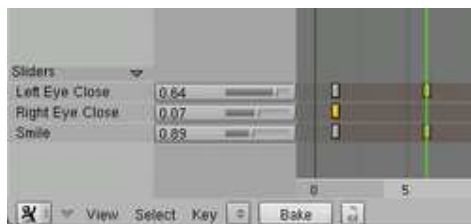
Cuando una forma no está enclavada, el panel *Shapes* tiene una fila de botones adicional: Valor, *Min* y *Max*.

El primer deslizador es el valor actual de esa clave en el fotograma actual. Ajustando el valor, se insertará o ajustará un punto de control en la curva de control de la clave de forma en el fotograma actual. Seleccionando cada forma y ajustando su valor, puede crearse una forma combinada para el fotograma actual.

Los botones *Min* y *Max* ajustan los valores extremos para el deslizador Valor, tanto en el panel *Shapes* como en la Ventana *Action Editor* (editor de acciones).

Lo mejor es configurar las formas y los ajustes *Min/Max* en el panel *Shapes*, y combinarlas en el *Action Editor*.

Ventana *Editor de Acciones*



Editor de Acciones y Claves de Forma

En la Ventana del *Editor de Acciones*, expanda el botón *Sliders* (deslizadores) y use los deslizadores para ajustar los valores de las claves en el fotograma actual. Los valores *Min / Max* definidos en el panel *Shapes* se reflejan en los deslizadores de la ventana del *Editor de Acciones* para esa forma.

Si no hay ya un fotograma clave en el fotograma indicado por la línea verde, se añadirá un nuevo fotograma clave.

Ventana *IPO*

La Ventana *IPO* representa los datos subyacentes que están siendo controlados por los otros métodos de combinación. Es en esta ventana donde los valores son mapeados a fotogramas para cada clave de forma. Para acceder a las IPOs de la clave de Forma, seleccione *Shape* de entre las entradas del menú emergente 'tipo de IPO', en la cabecera de la Ventana *IPO*. Cada clave de forma se mostrará en la lista de IPOs en la parte derecha, y puede ser alterada del mismo modo que cualquier IPO en Blender. Vea la documentación sobre IPOs para más información.

Controladores IPO

Vea la documentación sobre Controladores IPO para información acerca de cómo crear o ajustar un Controlador IPO. Los Controladores IPO funcionan del mismo modo para IPOs de Claves de Forma que para cualquier otro tipo de IPO.

Claves de Vértices Absolutas

Además de las Claves de Objeto (Objects Keys) en las que se especifican las posiciones de los objetos también se pueden crear en Blender las Claves de Vértice (Vertex Keys) en las que se especifican las posiciones de los vértices *dentro* de un Objeto. Ya que esto podría involucrar miles de vértices, se usa el sistema tradicional de la Clave de Posición. Una Curva IPO sencilla se usa para determinar como se realiza la interpolación y los momentos en los que se puede ver una Clave de Vértice.

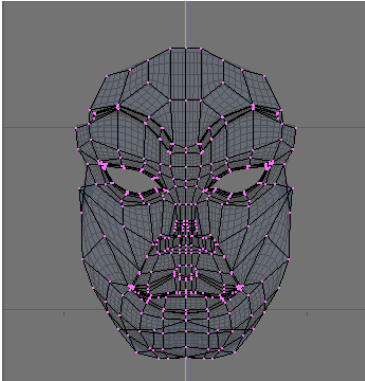
Las Claves de Vértice son parte de los datos del objeto no del objeto. Cuando se duplican los datos del objeto, el bloque asociado de Claves de Vértices también se copia. En Blender no se permite que múltiples objetos compartan la misma Clave de Vértices, ya que esto no sería muy práctico.

El bloque de Clave de Vértice es universal y comprende la distinción entre una malla, una curva, una superficie o una retícula. La interfaz y uso está por lo tanto unificada. En esta sección se explica con detalle como trabajar con Claves de Vértice de una malla, que contiene cierto número de comentarios breves sobre los otros Datos del Objeto.

La primera posición de clave de vértice que se crea es siempre la clave de *referencia*. Esta clave define las coordenadas de la textura. Solo si esta Clave está *activa* se pueden cambiar las caras y las curvas, o el *numero* de vértices. Se permite asignar otras claves a un diferente número de vértices. El sistema de Claves las interpola automáticamente.

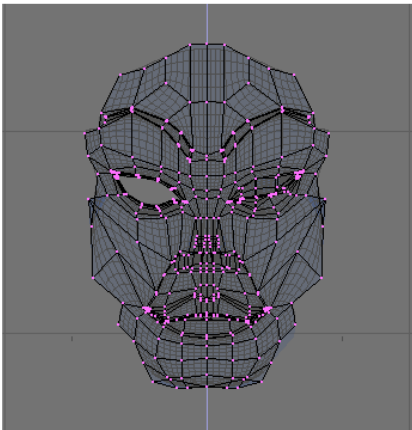
A continuación se da un ejemplo práctico. Cuando se trabaja con Claves de Vértices, es muy cómodo tener una ventana IPO abierta. Usa la primera Pantalla del fichero estándar de Blender, por ejemplo. En la ventana IPO, tenemos que especificar que queremos ver las Claves de Vértices. Para hacer esto usa el Botón de Menú IPO y selecciona *Vertex*. Ve a la Ventana3D con el cursor del ratón y aprieta **IKEY**. Con un objeto Mesh seleccionado y activo. El menú de la "Tecla Insert" tiene varias opciones, y la última es *Mesh*. Tan pronto

como esto haya sido seleccionado, aparece un nuevo cuadro de diálogo (*Menú de Insertar Clave*) preguntando por Claves de Vértices Absolutas o Relativas.



Menú de Insertar Clave.

Elegiremos *Absolute Keys*; se dibujará una línea horizontal amarilla en la ventana IPO. Esta es la primera clave y por lo tanto la Clave de *referencia*. También se crea una curva IPO para la "Velocidad" (Speed) (*Clave de Referencia y Velocidad IPO*).



Clave de Referencia y Velocidad IPO.

Creación de Claves de Vértices: Crear Claves de Vértices en Blender es muy simple, pero el hecho de que el sistema sea muy sensible en términos de su configuración puede hacer que ocurran un número de cosas "no visibles". Por lo tanto, la siguiente regla se debe tener en cuenta.

Tan pronto como se inserta una posición de Clave de Vértice, se activa *inmediatamente*. Todos los cambios siguientes en la Malla se enganchan a *esta* posición Clave. Por lo tanto, es importante que la posición de la clave se añada *antes* de comience la edición.

Adelanta unos pocos cuadros y vuelve a seleccionar: **IKEY**, *Mesh* (en la Ventana 3D). La segunda Clave se dibuja como una línea azul claro. Esta es una clave Normal; esta clave y todas las siguientes Claves afectan sólo a la información de Vértices. Aprieta **TAB** para EditMode y traslada uno de los vértices de la malla. Entonces vuelve atrás unos cuadro: ¡no pasa nada! tanto en cuanto estamos en EditMode, *no* se aplican otras Claves de Vértices. Lo que se ve en el ModoEdición es *siempre* la Clave de Vértice *activa*.

Sal del ModoEdición y navega por los fotogramas de nuevo. Ahora se puede ver el efecto del sistema de Claves de Vértice. Las Claves de Vértice sólo se pueden seleccionar en la Ventana IPO. Siempre se debe *salir* del ModoEdición: el 'contenido' de las claves de Vértices se muestran de modo temporal en la malla. Se puede editar la clave especificada entrando en el ModoEdición.

Hay tres métodos para trabajar con Claves de Vértices:

- **El método 'animación cronológica'.** Este método trabaja enteramente en Modo Edición, cronológicamente de posición en posición:
 - Inserta Clave. Se especifica la referencia.
 - Unos fotogramas después: Inserta Clave. Edita la Malla para la segunda posición.
 - Otros pocos fotogramas después: Inserta Clave. Edita la Malla para la tercera posición.
 - Continúa con el proceso anterior...
- **El método 'edición'.**
 - Primero insertamos todas las claves que se necesiten, a menos que ya hayamos creado las claves usando el método descrito anteriormente.
 - Blender *no* está en Modo Edición.
 - Selecciona una Clave. Ahora entra en Modo Edición, cambia la Malla y sal del ModoEdición
 - Selecciona una Clave. Entra en Modo Edición, cambia la Malla y sal del Modo Edición.
 - Continúa el proceso anterior...
- **El método 'inserción'.** En este método no importa si ya se han creado Claves o si estamos en Modo Edición.
 - Se va hasta el fotograma en el que se quiere insertar la nueva Clave.
 - Se inserta la Clave.
 - Se va a un nuevo fotograma, se inserta la Clave.
 - Continúa con el proceso anterior...

Mientras se está en el ModoEdición, las Claves no se pueden cambiar . Si el usuario intenta hacerlo, aparecerá un aviso.

Cada Clave se representa por una línea que se dibuja a cierta altura. La altura será de modo que la clave intersecte la "Velocidad" IPO en el fotograma en el que se encuentra la Clave.

Tanto la Curva IPO y la Clave de Vértice pueden seleccionarse separadas con **RMB**. Ya que sería muy difícil de otro modo trabajar con ellas, la selección de las líneas de las Claves está desactivada cuando la curva está en Modo Edición. El botón *channel* se puede usar para ocultar temporalmente la curva (**SHIFT-LMB** on "Speed") para hacer más fácil la selección de Claves.

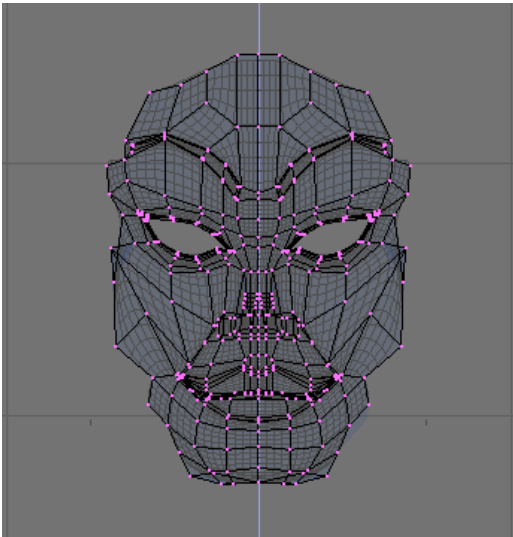
Las líneas de Claves en la Ventana IPO, una vez se toman, pueden situarse en cualquier posición vertical. Selecciona la línea y usa el Modo Grab para hacerlo. La Curva IPO también puede procesarse aquí del mismo modo que se describe en el capítulo anterior. En vez de un 'valor', la curva determina la interpolación entre las Claves, e.g. una curva sinodal se puede usar para crear una animación cíclica.

Durante la animación el contador de fotogramas muestra un cierto valor de la velocidad IPO, que se usará para elegir la(s) Clave(s) que se va(n) a usar, posiblemente con interpolación, para producir la malla deformada.

La Velocidad IPO tiene el comportamiento estándar de un IPO, también para interpolar. La línea de Clave tiene tres tipos distintos de interpolación. Presiona **TKEY** con una línea de Clave seleccionada para abrir un menú con las opciones:

- *Lineal*: La interpolación entre las Claves es lineal. La línea de Clave se muestra como una línea punteada.
- *Cardinal*: La interpolación entre las Claves es fluida, la asignación estándar.
- *BSpline*: La interpolación entre las Claves es extrafluida e incluye cuatro Claves en el cálculo de la interpolación. Las posiciones no se muestran de modo preciso. La línea de Clave se dibuja como una línea de guiones.

Claves Absolutas muestra una animación simple de Vértices de un cilindro. Cuando se ejecuta el cilindro se convierte en una gran estrella, entonces se deforma a una pequeña estrella, entonces ya que la Velocidad IPO retrocede a 0 la deformación se repite en orden inverso.



Claves Absolutas.

Algunas pistas útiles:

- Las posiciones de las Claves se añaden *siempre* con **IKEY**, incluso si no se encontrasen en la misma posición. Esto se usa para copiar posiciones mientras se inserta. Se pueden usar dos líneas en la misma posición para cambiar el efecto de la interpolación.
- Si *no* se seleccionan Claves, el ModoEdición se puede invocar como es normal. En cualquier caso, cuando se deja el ModoEdición, se deshacen todos los cambios. Inserta la Clave *en* el ModoEdición en este caso.
- Para las Claves, *no* hay diferencia entre seleccionada y *activa*. Por lo tanto, no es posible seleccionar múltiples Claves.
- Cuando se trabaja con Claves con diferente número de vértices, las caras se pueden desordenar. No hay herramientas para especificar de un modo *preciso* una secuencia

de vértices. Actualmente esta opción está sólo asequible para Mallas que tienen vértices como Halos.

Claves de Curva y Superficie

Como se mencionó anteriormente, las Claves de Curva y Superficie trabajan exactamente igual que las Claves de Malla. Para las Curvas, es particularmente interesante localizar las Claves de Curva en el objeto bevel. Aunque esta animación *no* se mostrará en tiempo real en la ventana 3D, si será renderizada.

Claves de Retícula

Tan pronto como una Clave esté presente en una Retícula, los botones que se usan para determinar la resolución se bloquean.

Claves de Vértices Relativas

Las Claves de Vértices Relativas (RVK) trabajan de modo diferente tanto en cuanto que solo se almacena la diferencia entre la malla de referencia y la malla deformada. Esto permite doblar varias claves juntas para conseguir animaciones complejas.

Pasaremos por RVK por medio de un ejemplo.

Crearemos una animación facial con RVK. Mientras que las Claves de Vértice absolutas se controlan con sólo *una* curva IPO, las Claves Relativas de Vértices se controlan con una curva de interpolación por cada posición de clave, lo que pone de manifiesto 'cuanta' de esa deformación relativa se usa para producir la malla deformada. Es por esto que las claves relativas se pueden mezclar (sumar, restar, etc.).

Para la animación facial, la base de partida podría ser una posición relajada con una boca ligeramente abierta y los párpados medio abiertos. Entonces las claves podrían definirse para los ojos izquierdo/derecho cerrado, felicidad, tristeza, sonriendo, frunciendo el ceño, etc.

El truco con las claves de vértice relativas es que sólo los vértices que han cambiado entre la base y la clave afectan la salida final de la deformación. Esto significa que es posible tener varias claves que afecten al objeto en diferentes sitios todos al mismo tiempo.

Por ejemplo, una cara con tres claves: sonreír, el ojo derecho cerrado y el ojo izquierdo cerrado podría ser animado para sonreír, entonces cerrar el ojo izquierdo, después cerrar el derecho, entonces abrir los dos ojos y finalmente dejar de sonreír - todo deformando tres claves. Sin las claves relativas de vértice se hubiesen necesitado seis claves de vértice, una para cada posición intermedia.

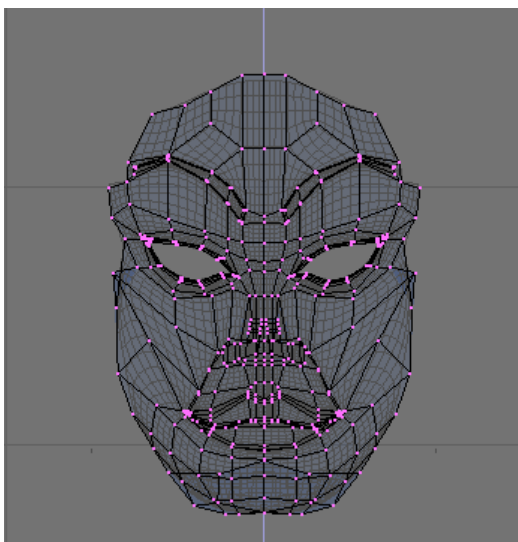
Consideremos la cabeza femenina de *La cabeza femenina que queremos animar*.



La cabeza femenina que queremos animar.

Para añadir RVK se aprieta **IKEY** y se selecciona *Mesh(Malla)* igual que para las claves absolutas, pero, desde el menú pop up selecciona *Relative Vertex Keys*. Esto almacena la clave de referencia y aparecerá como una raya horizontal amarilla en la ventana IPO.

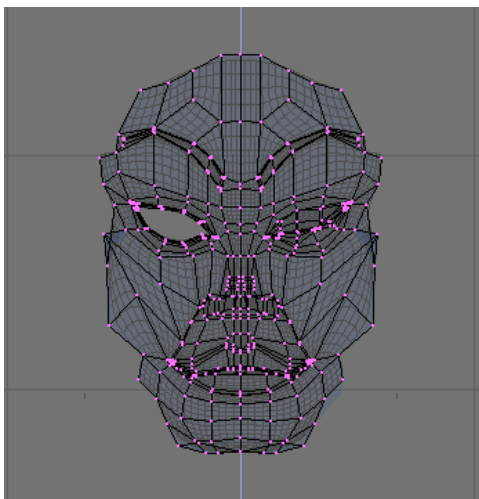
La Claves Relativas se definen insertando otras claves de vértice. Cada vez que se apriete **IKEY** y que se seleccione *Mesh* aparecerá una nueva línea horizontal en la ventana IPO. Si el número de fotograma se incrementa cada vez la línea horizontal se sitúa por encima de la anterior. Para simplificar el modelado vamos a ocultar todos los vértices excepto los de la cara *Todos los vértices ocultos salvo los de la cara.*



Todos los vértices ocultos salvo los de la cara.

Ahora nos movemos a otro fotograma, digamos que el número 5, y añadimos una nueva Clave. Una línea cian aparecerá por encima de la amarilla, que ahora estará naranjada. Cambiamos a modo Edición y cerramos el párpado izquierdo.

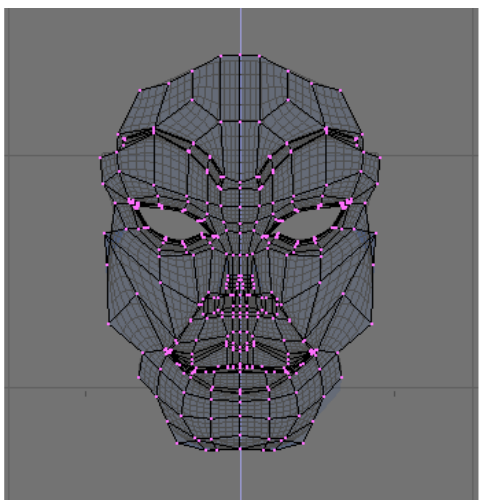
Cuando salgamos del modo Edición. Si se selecciona la Clave de Referencia se verá la malla original. Si se selecciona la primera Clave de Vértice Relativa se verá la malla deformada (*Ojo izquierdo cerrado*).



Ojo izquierdo cerrado.

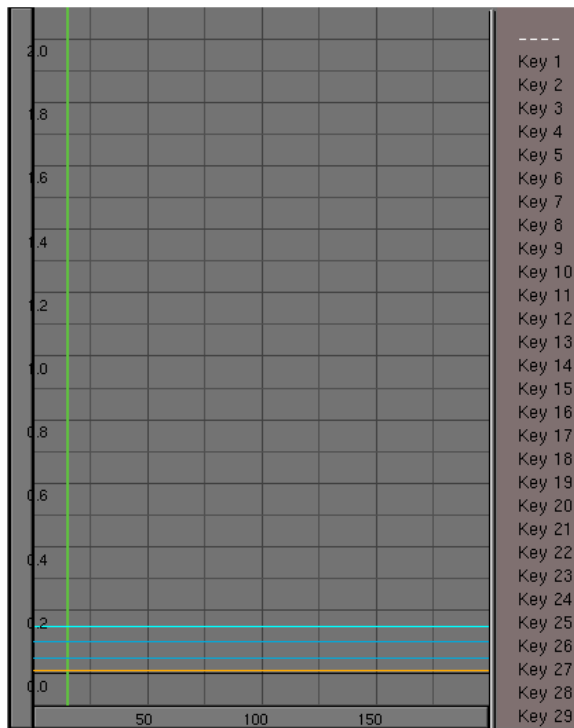
Repitamos el paso anterior para el ojo derecho. Hay que tener cuidado de que la clave nueva que se inserta se basa en la malla de la clave que está actualmente como activa, así que en general es buena idea seleccionar la clave de referencia antes de apretar **IKEY**.

Ahora añadimos una sonrisa (*Sonriendo*).



Sonriendo.

La ventana IPO debe ser algo como *Claves en la Ventana IPO*.

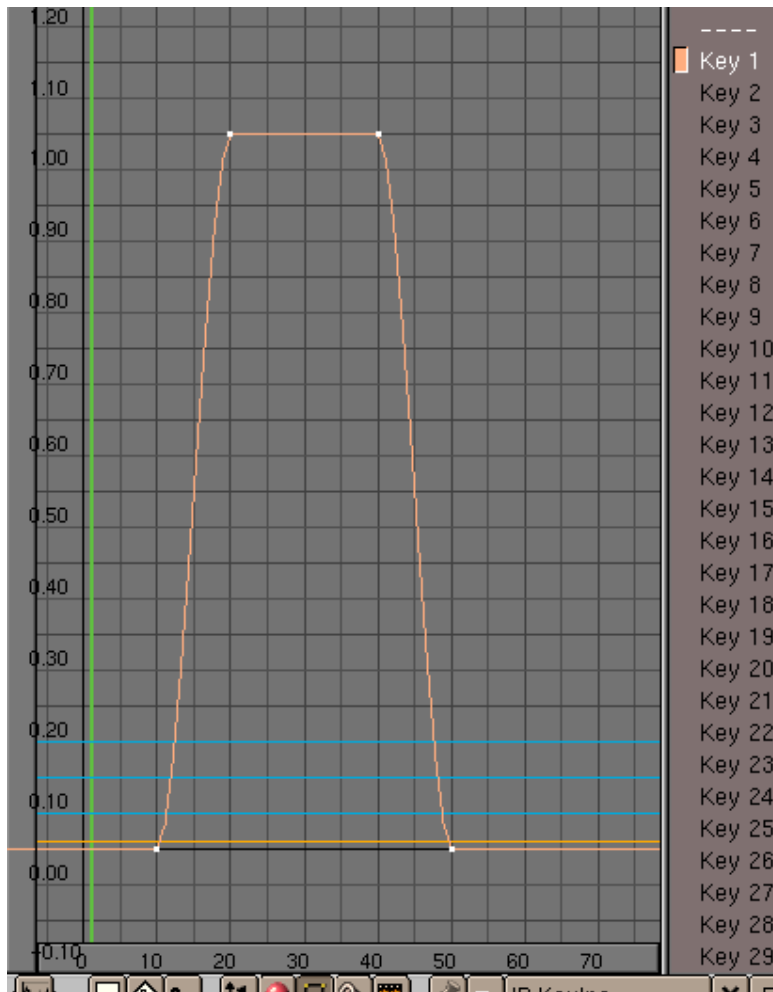


Claves en la Ventana IPO.

El orden vertical de las Claves de Vértice (Las líneas azules) de abajo a arriba determina su curva IPO correspondiente, i.e. la línea de clave azul más baja será controlada por la curva *Key1*, la segunda más baja será controlada por la curva *Key2*, y así sucesivamente.

No hay IPO para la malla de referencia ya que es la malla que se utiliza cuando cualquier otra clave tiene un IPO con valor cero en determinado fotograma.

Selecciona *Key1* y añade un IPO con tu método favorito. Haz que se parezca algo a *La curva IPO de la clave 1*.

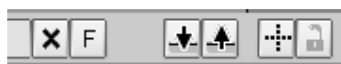


La curva IPO de la clave 1.

Esto hará que nuestra malla no se deforme hasta el fotograma 10, entonces desde el 10 hasta el 20 la Clave 1 (Key 1) empieza a afectar a la deformación. Del fotograma 20 al fotograma 40 la Clave 1 sustituye completamente a la malla de referencia (el valor de IPO es 1), y el ojo estará completamente cerrado. El efecto irá desapareciendo del fotograma 40 al 50.

Se puede comprobar con ALT-A, o asignado los números de fotograma a mano. La segunda opción es mejor, a menos que el ordenador se muy potente.

Copia este IPO usando el botón de la flecha hacia abajo en la barra de herramientas de la Ventana IPO (*Botones copiar/pegar*). Selecciona la Key 2 (Clave 2) y pega la curva con la flecha hacia arriba. Ahora ambas claves tendrán la misma influencia en la cara y ambos ojos se cerrarán al mismo tiempo.

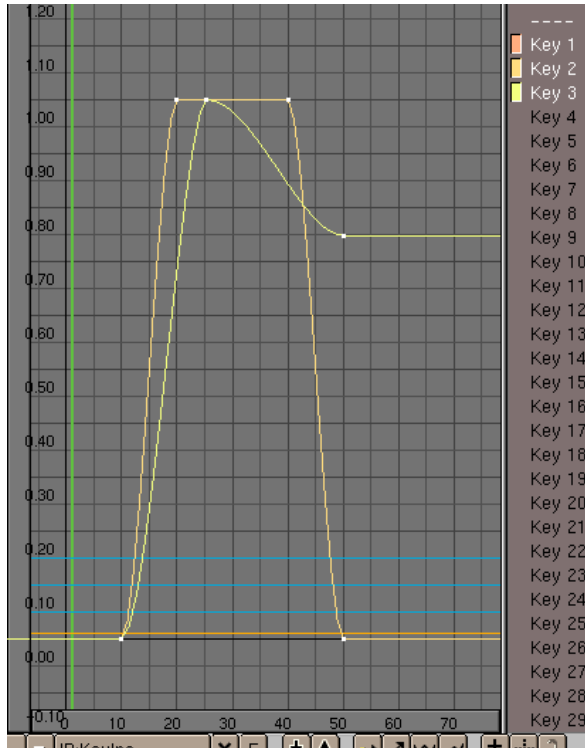


Botones copiar/pegar.

Desplazando la Barra de Herramientas: Puede pasar que la barra de herramientas sea más larga que la ventana y algunos de los botones no se vean. Se puede desplazar cualquier barra de herramientas horizontalmente It may happen that the toolbar is longer than the window

and some buttons are not shown. You can pan horizontally pulsando el botón del medio y arrastrándola con el ratón.

Añade también un IPO para la *Key 3*. Pero hagamos esta diferencia (*Todos los IPOs*).



Todos los IPOs.

De esta forma los ojos se cierran y ella comienza a sonreír, la sonrisa está al máximo con los ojos cerrados, entonces ella sonríe 'menos' mientras los ojos se vuelve a abrir y se queda sonriendo (*Secuencia*).



Secuencia.

La curva IPO de cada Clave controla la deformación entre las claves relativas. Estas curvas deben crearse del modo típico. La posición final se determina añadiendo todos los efectos de cada curva IPO.

Valores fuera del rango [0,1]: Una parte importante de las Claves Relativas es el uso de posiciones aditivas o extrapoladas. Por ejemplo, si la posición base para una cara es con la boca recta, y una clave se define para sonreír, entonces es posible que la aplicación negativa de la clave resulte en curvar la boca al contrario de una sonrisa. En el caso de extendamos la Curva IPO por encima de 1.0 se "extrapolará" esa clave, construyendo una sonrisa extrema.

Animación de Enrejados

Emparentar una malla a un enrejado es una buena forma de aplicar deformaciones a la malla mientras se modela, pero ¡también es la forma de hacer las deformaciones en el tiempo!.

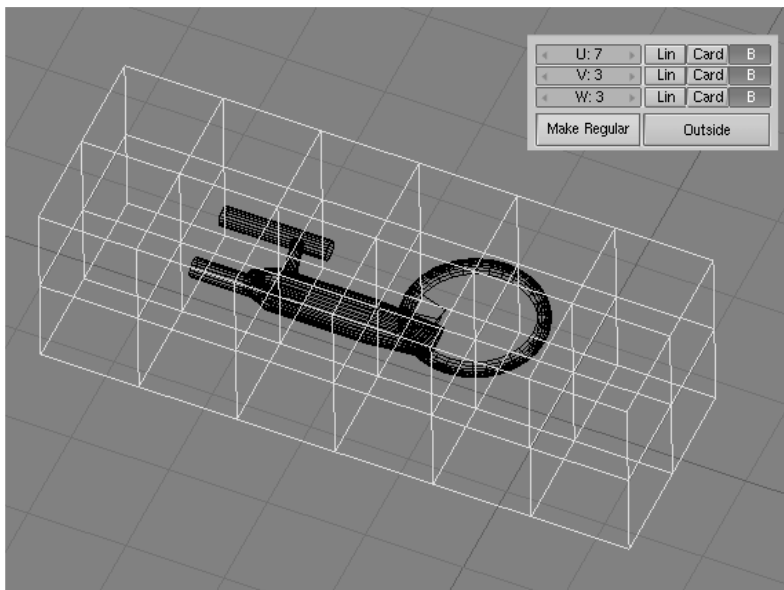
Se pueden usar Enrejados en animaciones de dos maneras:

- Animar los vértices con claves de vértice (o claves de vértice relativas);
- Mover el enrejado o el objeto hijo del enrejado.

La primera técnica es básicamente lo contenido en las dos secciones previas pero aplicado al enrejado que tiene un objeto emparentado.

Con la segunda clase se pueden crear animaciones que aplaste cosas entre rodillos, o para obtener el bien conocido efecto de la aceleración de una nave espacial a la velocidad de doblado.

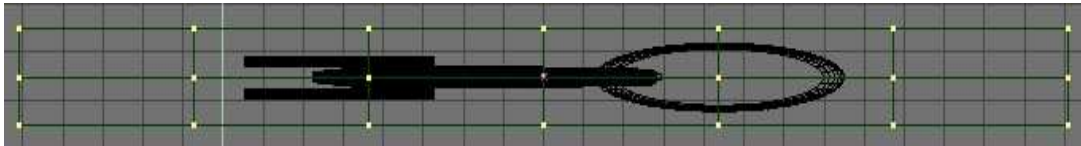
Hacemos una nave espacial y añadimos un enrejado alrededor de la nave. Hacemos el enrejado con los parámetros de *Configuración del Enrejado*.



Configuración del Enrejado

Selecciona la nave, extiende la selección al enrejado (manteniendo apretada la tecla **SHIFT** mientras seleccionamos), y presionando **CTRL-P** para hacer el enrejado el objeto padre de la nave. No se debería ver ninguna deformación de la nave porque el enrejado todavía es regular.

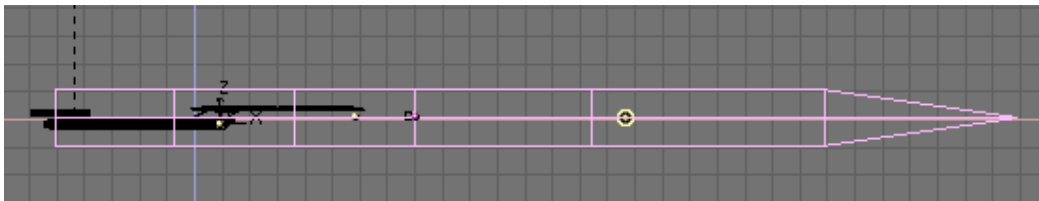
Para los próximos pasos es importante que los hagamos en Modo Edición. Así que ahora se selecciona el enrejado, se entra en Modo Edición, se seleccionan todos los vértices (**AKEY**), y se escala el enrejado a lo largo de su eje x (presiona el botón del medio **MMB** mientras se inicia el escalado) para conseguir el estrechamiento que se quiere. La malla de la nave muestra inmediatamente la deformación causada por el enrejado (*Estrechando*).



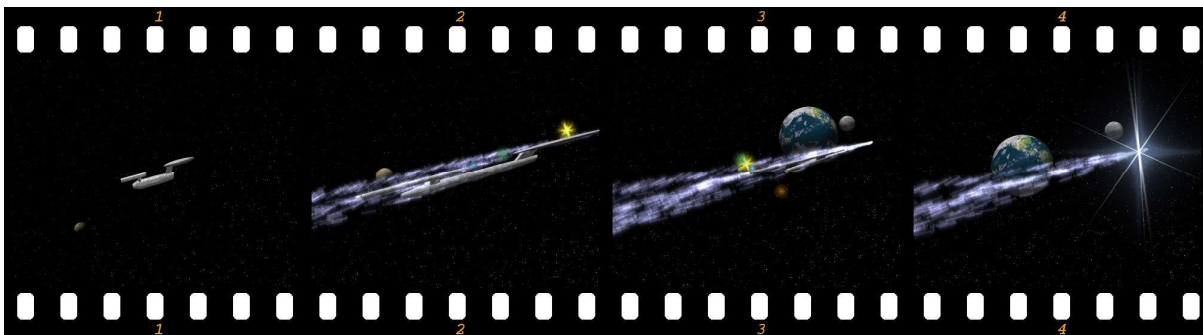
Estrechando

Ahora edita el enrejado en Modo Edición de forma que los vértices de la derecha tengan una distancia mayor entre ellos. Esto incrementará el estrechamiento así como la nave va dentro del enrejado. Los vértices del extremo de la derecha se han escalado al mínimo de manera que prácticamente están en un punto; esto causará el desvanecido de la nave al final (*Deformación Final del Enrejado*).

Selecciona la nave de nuevo y muévela a través del enrejado para ver un previo de la animación. Ahora ya se puede hacer una animación normal con claves de fotograma para conseguir que la nave vuele a través del enrejado.



Deformación Final del Enrejado



Algunos fotogramas de la animación resultante.

PARTE IX - EL SISTEMA DE ARMATURES

Como hemos visto en su primera animación en 30 Minutes II Blender utiliza *Armatures* para la animación de personajes. Un *armature* es solo un esqueleto el cual una vez que hemos emparentado nuestra malla de persona al mismo, nos permite definir una serie de *poses* a lo largo de la línea de tiempo de nuestra animación.

Un *armature* está compuesto de un número arbitrario de *huesos*. El tamaño, la posición y orientación de cada hueso en el *armature* depende de ti, y a lo largo de este capítulo encontrarás distintas situaciones que requerirán un arreglo particular de huesos para que tu personaje funcione correctamente.

A medida que animes tu *armature* verás que es mucho mejor organizar las distintas poses en algo llamado una *action*, lo que es más o menos lo mismo que sucede en el mundo real. Cuando caminamos podemos imaginarnos atravesando distintas poses instantáneas como si fueran los cuadros de una película, el proceso entero de la caminata es una acción a fin de cuentas. Pero hay acciones y acciones. Como animador necesitarás desarrollar la habilidad de reconocer cómo dividir un movimiento natural o acción en acciones más simples con las que sea más sencillo trabajar. Usar acciones más simples normalmente ahorra tiempo y trabajo (y por qué no: dinero !) ya que es común que estas acciones sean reutilizables en varias animaciones.

Una vez que hayas creado tus primeras acciones, podrás combinarlas utilizando el poderoso editor de *Animación No Lineal* (o *NLA*), de Blender, otorgándole a tu personaje un aspecto vivo y natural.

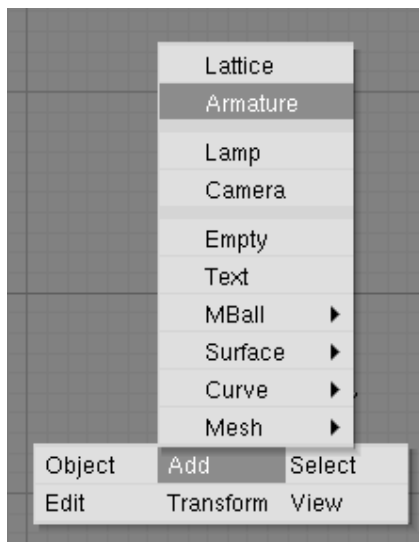
En este capítulo cubriremos cada detalle de las funcionalidades de Blender relativas a *Armatures*, *Acciones* y el Editor *NLA*. Incluso veremos varias configuraciones de *Armatures* que te darán un punto de partida para aplicar a tus propias creaciones. Relajate y disfruta.

Capítulos

- El Objeto *Armature*
- FK, IK and other Constraints
- Skinning
- Modo Pose
- La Ventana de Acción
- Animación No Lineal
- Rigging examples

El Objeto Armature

El objeto Armature es el objeto básico para animación de personajes. Es un objeto que consiste en una serie de "huesos" que pueden o no estar interconectados. Una serie de huesos interconectados puede ser una "Cadena de Cinemática Inversa (IK)" o simplemente una "Cadena" de huesos. Una Cadena IK es algo más complejo que una relación Parent estándar en el sentido que no solo los movimientos del hueso "Padre" son transmitidos a los huesos "hijos", sino que los movimientos del último "hijo" en la cadena son transmitidos a lo largo de la cadena hasta el hueso "padre" si se utilizara una Solución de Cinemática Inversa. Los huesos pueden moverse como si fuesen un grupo de Objetos rígidos indomables con uniones articuladas perfectas. Piensa en un armature como si se tratara de un esqueleto de una criatura viva. Los brazos, piernas, columna y cabeza son todas partes del mismo objeto esqueleto.



Agregando un Armature

Para crear un nuevo armature, selecciona **SPACE>>Add>>Armature** de la barra de herramientas (*Agregando un Armature*). Aparecerá un nuevo hueso con su raíz en la posición del cursor 3D. A medida que muevas el ratón, el hueso cambiará de tamaño correspondientemente. **LMB** finalizará el hueso y comenzará otro nuevo como hijo del anterior. De esta manera puedes crear una cadena completa. Pulsando **ESC** cancelarás la creación de un nuevo hueso. Puedes agregar otro hueso al armature mientras estés en Modo Edición con **SPACE>>Add>>Armature** otra vez en la caja de herramientas. Esto empezará otra vez el modo de agregar huesos y los nuevos huesos serán parte de la armature actual pero formarán una cadena separada. También puedes "extrudar" huesos de los huesos existentes seleccionando el extremo de un hueso y presionando **E**. El hueso recién creado será hijo del hueso de donde fue extrudado pero *no* de su cadena IK. Estando en Modo Edición puedes realizar las siguientes operaciones sobre los huesos de un armature.

- *Ajuste* - Selecciona uno o mas extremos de huesos y utiliza cualquiera de las operaciones de transformación estándar para ajustar la posición y orientación de cualquier hueso en el armature. Notar que las cadenas IK no pueden tener ninguna separación entre los huesos que la componen por lo que mover el extremo final de un hueso moverá también el extremo inicial, o *raíz* de sus hijos. Se puede seleccionar una

cadena IK completa en un solo paso moviendo el puntero del ratón sobre un extremo de hueso en la cadena y pulsando **L**. También se puede usar la herramienta de selección por límites (**B**).

- *Borrado* - Se pueden borrar uno o más huesos seleccionando el comienzo y final de los mismos. Al hacer esto observarás que el hueso entero se resaltará con un color. Pulsando **X** se borrarán los huesos resaltados. Observar que seleccionando solo uno de los extremos no es suficiente para borrar el hueso..
- *Ajuste a punto* - Es posible ajustar los extremos de un hueso a la cuadrícula o al cursor usando el menú Snap accesible con **SHIFT-S**.
- *Modo Numérico* - Para una edición más precisa, pulsando **N** aparecerá la caja de diálogo numérica. Aquí se puede ajustar la posición de los extremos iniciales y finales así como la rotación del hueso sobre su propio eje. Una forma sencilla de orientar automáticamente el eje Z de todos los huesos seleccionados (necesario para un correcto funcionamiento de la opción "pose invertida") es pulsar **CTRL-N**. No olvides hacer esto antes de empezar cualquier animación del armature.
- *Deshacer* - Estando en Modo Edición puedes cancelar los cambios realizados en la sesión de edición actual presionando **U**. El armature se revertirá al estado anterior al comienzo de la edición.

También se pueden unir dos armatures juntos en un único Objeto. Para hacerlo, asegurate de estar en Modo Objeto, selecciona ambos armatures y pulsa **CTRL-J**.

Nombrando Huesos

Es importante asignar nombres significativos a los huesos por varias razones. Primero que nada, hará las cosas más fáciles cuando edites acciones en el Editor de Acciones. Segundo, los nombres de los huesos se utilizan para asociar canales de Acción a los huesos cuando intentes reutilizar Acciones, y tercero los nombres se usan para aprovechar la ventaja de la opción de pose inversa automática.. Fíjate que los nombres de los huesos solo necesitan ser únicos dentro de una unidad armature. Puedes tener varios huesos llamados "Cabeza" siempre y cuando se encuentren en distintas armatures. Para cambiar los nombres de uno o más huesos, selecciona los huesos en Modo Edición y cambia a los botones de Contexto Edición con **F9**. Una lista de todos los huesos seleccionados debería aparecer en el Panel *Armature Bones* (*Botones de Edición para un Armature*). Cambia el nombre de un hueso haciendo **SHIFT-LMB** en la caja con el nombre del hueso y escribe un nuevo nombre.



Botones de Edición para un Armature

Es más sencillo nombrar los huesos editandolos de a uno a la vez o asegurándose de activar la opción *DrawNames* en los botones de edición **F9** (*Opciones de Visualización para Armatures*).

Convenciones de Pose Invertida: Típicamente, las armatures de personajes son axialmente simétricas. Esto significa que muchos elementos se encuentran de a pares, unos sobre la izquierda y otros sobre la derecha. Si nombras correctamente Blender puede invertir una pose dada alrededor del eje de simetría facilitando la tarea de animar ciclos de andar, por ejemplo. Para cada hueso que tenga par, añádele un sufijo a los nombres para distinguir el par izquierdo y el derecho. Puede ser tanto ".L" y ".R" como ".Left" y ".Right". Nota: No funcionará escribirlo en español, y dará lo mismo si ".L" es realmente la izquierda o la derecha. Los huesos que estén sobre el eje de simetría o que no tengan un homólogo no lo necesitan. La parte del nombre que precede al sufijo debe ser idéntica para ambos lados . Así por ejemplo si tenemos dos manos, deberían llamarse "Mano.R" y "Mano.L".

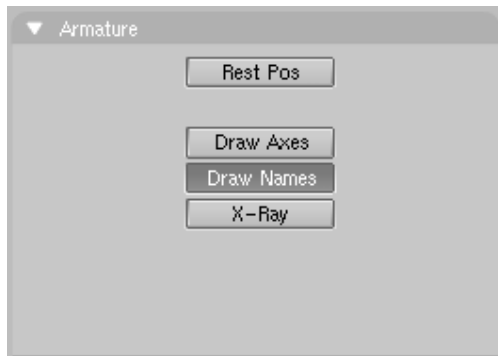
Emparentado y cadenas IK

Para modificar relaciones de emparentado dentro del armature, selecciona el hueso que debería ser *hijo* y entra al Panel *Armature Bones* de la ventana de Botones de Edición. Al lado del nombre del hueso verás una lista desplegable con la etiqueta *Child Of*. Para convertir este hueso en hijo de otro, elige de la lista el hueso que será padre. Observa que esto es mucho más sencillo si los huesos han sido nombrados correctamente. Para eliminar una relación de emparentado, elige la opción en blanco de la lista. Observa que el menú de emparentado solo contiene los nombres de padres válidos. Aquellos huesos que no pueden ser padres (por ejemplo los hijos del hueso actual) no serán mostrados. El botón IK junto a cada hueso que tiene padre, se usa para determinar si la solución IK deberá propagar sus efectos a través de

esta articulación. Si el botón IK está activo, el extremo final del padre se moverá para coincidir con el extremo inicial del hijo. Esto ocurre para cumplir con el requerimiento de que no debe haber brechas o espacios en una cadena IK. Desactivando el botón IK no se restaurará la posición del extremo inicial del hijo a su posición original, pero mover esa articulación no afectará ya al extremo final del padre.

La orientación de los *roll* handles de los huesos es importante para obtener buenos resultados del sistema de animación. Puedes ajustar el ángulo de revolución de la articulación del hueso, seleccionado la articulación y pulsando **NKEY**. El número exacto que debe ser ingresado aquí depende de la orientación del hueso. El eje Z de cada hueso de orientarse en una dirección coherente en los huesos homólogos. Una buena solución es hacer que los ejes Z apunten hacia arriba (o hacia adelante si el hueso es vertical). Esta operación se facilita mucho si se activa la opción "Draw Axes" en el Panel *Armature* en la Ventana de Botones de Edición.

El panel Armature



Opciones de Visualización para Armatures.

Este panel contiene solo unos pocos botones. Cuando se activa el botón *Rest Pos* (*Opciones de Visualización para Armatures*), el armature se mostrará en su posición de "descanso". Esto es muy útil si se necesita editar la malla asociada con el armature después de haber animado o cambiado la pose del armature. Nota que las Acciones y poses aún están allí pero han sido temporalmente desactivadas mientras el botón esté activado. Los botones *Draw Axes* y *Draw Names* permiten visualizar los ejes locales y los nombres de cada hueso en la Vista 3D. El botón *X-Ray* evita que los huesos del armature sean ocultados por el modelo cuando nos encontramos en Modo Sólido o Sombreado.

Skinning

Una vez que el Armature - el 'esqueleto del personaje' - está listo, es necesario emparentarle la "piel" del personaje. El "Skinning" es una técnica para crear deformaciones de la malla suaves usando un armature. Esencialmente el skinning es la relación entre los vértices de una malla y los huesos de un armature, y cómo afectarán las transformaciones de cada hueso a la posición de los vértices de la malla. Cuando se emparenta una malla a un armature se presentarán varias opciones:

- Parent to Bone

En este caso, aparecerá un menú que te permitirá elegir a qué hueso se debe emparentar el o los objetos hijos. Esto funciona muy bien para robots, que tienen cuerpos compuestos de mallas *separadas* que no se supone que deban doblarse o deformarse cuando se mueven.

- Parent to Armature

Eligiendo esta opción se deformarán la o las mallas hijas de acuerdo con sus grupos de vértices. Si las mallas hijas no tienen grupos de vértices definidos, serán sometidas a skinning automático. En efecto, un segundo menú aparecerá con las opciones:

- *Don't create groups* - No hace nada más, se usa skinning automatico;
- *Name Groups* - crea grupos de vértices vacíos cuyos nombres concuerdan con los nombres de los huesos, pero sin asignarles vértices;
- *Create from closest bone* - en el caso que quieras crear y llenar automáticamente los grupos de vértices.

- Parent to Armature Object

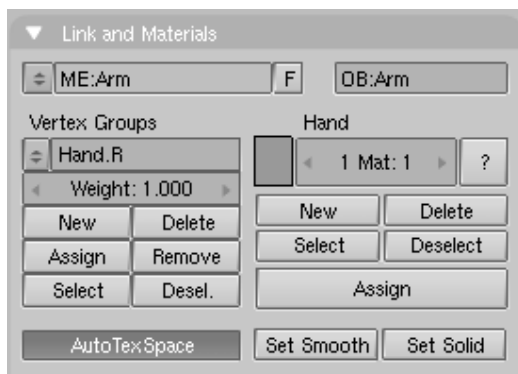
Eligiendo esta opción hará que los hijos vean al armature como un Empty para todo propósito.

Si estás preparando un animación de personajes, la mayoría de las veces emparentarás tu personaje al armature usando la opción "Armature". Se aconseja enérgicamente usar la opción *Name Groups*. Esto te proveerá de los grupos de vértices, ahorrándote el tedioso trabajo de crearlos y nombrarlos, evitándote errores de tipo. La opción *Create from closest bone* se encuentra actualmente en desarrollo. Utilizará los "Tipos de huesos" que se pueden definir por medio del menú a la derecha de los botones *IK* (*Botones de Edición para un Armature*) para resultados óptimos. Actualmente solo las opciones *Skinnable* y *Unskinnable* son funcionales. La primera opción hace que los Grupos de Vértices sean creados (y asignados si así lo has indicado) para el hueso dado, la segunda opción hace que el hueso sea ignorado en el proceso de skinning.

Note: El algoritmo de asignación de vértices actual, crea grupos de vértices no óptimos, por ello se recomienda verificarlos uno a uno.

Si una malla no posee ningún grupo de vértices y se convierte en hija de un armature, Blender intentará calcular la información de deformación al vuelo. Esto es muy lento y no se recomienda. Es aconsejable crear y usar grupos de vértices en su lugar.

Grupos de Vértices

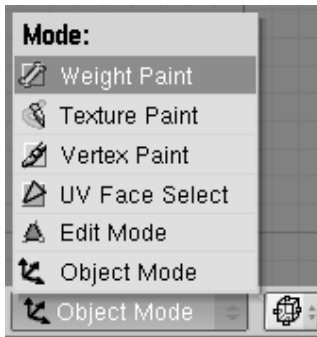


Grupos de Vértices

Para definir qué huesos deforman a qué vértices, se necesitan los Grupos de Vértices. Un vértice puede ser miembro de varios grupos, en cuyo caso la deformación será un promedio ponderado de las deformaciones producidas por los huesos a que está asignado. De esta manera es posible crear articulaciones suaves. Para agregar un nuevo grupo de vértices a una malla debes estar en Modo Edición. Crea un grupo nuevo de vértices haciendo click en el botón *New* en los botones de edición de la malla en el Panel *Mesh Tools 1 (Grupos de Vértices)*. Un grupo de vértices puede ser eliminado pulsando sobre el botón *Delete*. Cambia el grupo activo eligiendo uno de la lista desplegable de grupos. Los grupos de vértices tienen que tener el *mismo* nombre que los huesos por los que serán manipulados. Tanto la ortografía como las mayúsculas importan aquí. ¡ Por ello es que es tan cómoda la creación de nombres automática !. Renombra un grupo de vértices haciendo **SHIFT-LMB** sobre el botón con el nombre y escribiendo un nuevo nombre. Observa que los grupos de vértices deben ser únicos en una malla dada. Se pueden asignar los vértices al grupo activo, seleccionándolos y pulsando en el botón *Assign*. Dependiendo del valor indicado en el botón *Weight*, los vértices recibirán más o menos influencia de ese hueso. Este peso, es importante solamente en los vértices que son miembros de más de un grupo. El valor de *Weight* no es un valor absoluto sino uno relativo. Para cada vértice, el sistema calcula la suma de los pesos de todos los huesos que afectan al vértice. Las transformaciones de cada hueso se dividen por esta cantidad o sea que cada vértice siempre recibe una deformación del 100%. Asignando peso 0 a un vértice lo eliminará del grupo activo en la práctica. Para eliminar vértices del grupo actual, selecciónalo y pulsa en el botón *Remove*. Presionando el botón *Select* añadirá a la selección actual, los vértices del grupo elegido. Pulsando el botón *Deselect* deseleccionará los vértices del grupo actual. Esto es práctico para determinar qué vértices están en un grupo dado.

Pintura de Pesos (Weight Painting)

La pintura de pesos es una técnica alternativa para asignar pesos a los vértices en los grupos de vértices. El usuario puede "pintar" pesos sobre el modelo y ver los resultados en tiempo real. Esto permite lograr articulaciones suaves en forma más fácil. Para activar el modo de pintura de pesos, selecciona una malla con grupos de vértices y haz click sobre el icono de *Weight Paint (El botón Pintura de Pesos (Weight Paint))*.



El botón Pintura de Pesos (Weight Paint).

La malla activa será dibujada en modo Color-de-Pesos. En este modo, el azul oscuro representa áreas con peso nulo para el grupo de vértices actual, mientras que el rojo representa áreas con peso lleno. Solo puede visualizarse un grupo a la vez. Cambiando el grupo de vértices activo en los botones de edición, cambiarás la representación de la pintura de pesos. Los pesos se pintan sobre la malla usando una técnica similar a la utilizada en la Pintura de Vértices (vertex-painting), con algunas diferencias. El "color" es el valor de peso especificado en los botones de edición. El desplazador de *opacidad* (*opacity*) en los botones de vertex paint se usa para modular el peso. Para borrar el peso de los vértices, pon el peso a "0" y comienza a pintar.

Modo Pose

Para manipular los huesos en un armature debes entrar al Modo Pose. En el Modo Pose solo pueden seleccionarse huesos del armature activo. A diferencia del Modo Edición no se pueden ni agregar ni borrar huesos en este modo.

Para ingresar al Modo Pose, selecciona el armature y pulsa **CTRL-TAB**. Alternativamente puedes activar el Modo Pose seleccionando el armature y pulsando en la entrada *Pose Mode* en el menú *Modo* de la cabecera de la Ventana 3D (%%XREF:BSG.CHA.F.S68.108"/>). Puedes salir del Modo Pose con el mismo método o entrando a Modo Edición. <figure id="BSG.CHA.F.S68.108" float="1"> <title>La opción de menú Modo Pose.</title> <mediaobject> <imageobject> <imagedata fileref="PartA/character_animation/gfx/PoseModeIcon.png" format="PNG" Dentro del Modo Pose puedes manipular los huesos del armature seleccionandolos con **RMB** y usando las teclas de transformaciones estándar: **RKEY**, **SKEY** y **GKEY**. No puedes "mover" huesos que son hijos IK de otro hueso ya que la cadena IK debe mantenerse continua.

Para insertar Keyframes en los huesos seleccionados pulsa **IKEY**.

Si quieres borrar la pose de uno o más huesos, seleccionalos y pulsa **ALT-R** para borrar las rotaciones, **ALT-S** para borrar el escalado y **ALT-G** para borrar los desplazamientos. Ejecutando estos tres comandos sobre los huesos seleccionados volverán el armature a su posición inicial de reposo.

Frecuentemente resulta conveniente copiar poses de un armature a otro. o de una acción a un punto diferente de la misma acción. Aquí es donde intervienen las herramientas de copiado de poses en el menú *Armature*.

Para obtener mejores resultados asegurate de seleccionar todos los huesos en Modo Edición y pulsar **CTRL-N** para auto-orientar los huesos antes de comenzar a animar.

Image:Manual.es/PartIX/PoseFlipIcons.png
El botón Modo Pose.

Para copiar una pose, selecciona uno o más huesos en Modo Pose, luego selecciona la opción *Armature >> Copy Current Pose* del menú en la cabecera del a Ventana 3D (%%%%XREF:BSG.CHA.F.S68.109" />). Las transformaciones de los huesos seleccionados serán almacenadas en un buffer hasta que sea necesario o hasta que se efectúe otra operación de copia.

Para "pegar" una pose, simplemente selecciona la opción *Armature >> Paste Pose* del menú (<xref linkend="BSG.CHA.F.S68.109" />). Si se encuentra activado el autoinsertado de keyframes para *Action*, los KeyFrames serán generados automáticamente. Para pegar la "pose invertida" (si el personaje estaba inclinado hacia la izquierda en la pose copiada, se pegará inclinado hacia la derecha en la pose invertida), usa la opción *Armature >> Paste Flipped Pose* del menú (<xref linkend="BSG.CHA.F.S68.109" />). Observar que si el armature no fue configurado correctamente, la técnica de pegar la pose invertida no funcionará según lo esperado.

Ventana de Acciones

Una Acción está hecha de uno o más canales de Acción. Cada canal corresponde a uno de los huesos del armature, y además, cada canal tiene una curva IPO de Acción asociada al mismo. La Ventana de Acción provee una forma de visualizar y editar juntas todas las curvas IPO asociadas con la Acción. {{Tip|Ventana de Acción| Para cada grupo de keys en una IPO de Acción, en la Ventana de Acción se mostrará un marcador en el frame que corresponda. Esto es parecido al modo "Key" en la Ventana IPO. Para los canales con IPOs de Restricciones, habrá uno o más canales de restricción adicionales debajo de cada canal de Acción. Estos canales se pueden seleccionar independientemente de los canales de sus dueños. (*Ventana de Acción con una Restricción*).

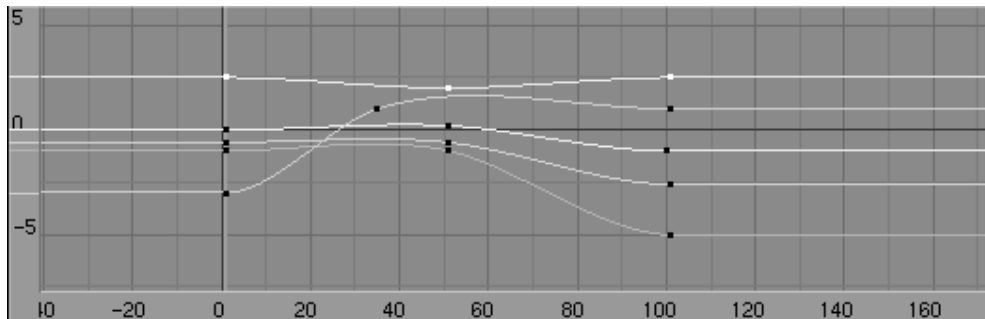


Ventana de Acción con una Restricción

Un bloque de keys de Acción se puede seleccionar tanto con **RMB** sobre él como utilizando la herramienta de selección por áreas (**B**). Los keys son resaltados en amarillo. Una vez seleccionados, los keys pueden moverse pulsando **G** y moviendo el ratón. Manteniendo presionada **CTRL** el movimiento se hará en intervalos de un cuadro. Haciendo **LMB** se terminará el desplazamiento de keys, mientras que con **ESC** se cancela y se vuelve al estado inicial. Un bloque de keys de Acción también puede escalarse horizontalmente (lo que acelerará o ralentizará la Acción), para ello basta con seleccionar un número de keys y presionar **S**. Moviendo el ratón horizontalmente se escalará el bloque. Con **LMB** se terminará la operación. Una o más keys de Acción pueden borrarse pulsando **X** cuando el puntero del ratón se halle sobre el área de KeyFrames en la Ventana de Acción. Un bloque de Acción puede duplicarse y moverse dentro de la misma Acción, seleccionando los keys que se deseen

y pulsando **SHIFT-D**. Esto entrarás inmediatamente en modo Mover con lo que se podrán trasladar el nuevo grupo de keys. El movimiento lo finalizamos con **LMB**. Con **ESC** se abortará el desplazamiento pero no eliminará los duplicados generados. También puedes borrar una o más Acciones o canales de restricción completos (con todas las keys asociadas) seleccionando los canales en la parte más a la izquierda de la Ventana de Acción (los canales seleccionados se resaltarán en azul). Con el puntero del ratón sobre esta zona izquierda de la ventana presiona **X** y confirma el borrado. Observa que no hay deshacer así que realiza esta operación cuidadosamente. Asimismo observa que si borras un canal de Acción que contiene canales de restricción, borrarás esos canales de restricción también.

Puedes ver la IPO de Acción asociada a un hueso en la Ventana IPO en lugar de la Ventana de Acción simplemente de hecho cambiando a una ventana IPO (*IPO de Acción*). La IPO de Acción es un tipo de IPO especial que solo es aplicable a huesos. En lugar de ángulos de Euler para codificar la rotación, las IPO de Acción utilizan quaterniones, los que dan una mejor interpolación entre Posturas.

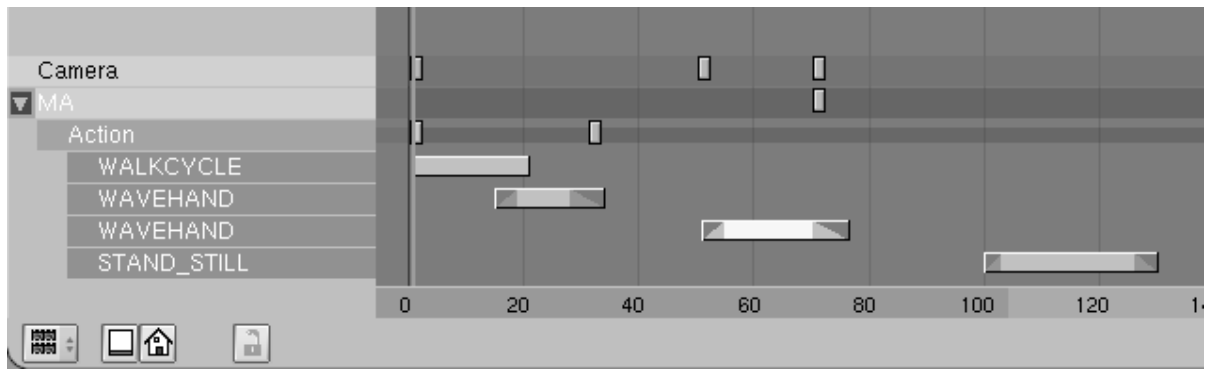


IPO de Acción

Los Quaterniones usan vectores de cuatro componentes. En general es difícil y poco intuitivo describir las relaciones entre estos quaterniones y la orientación resultante pero a menudo esto no es necesario. Es mucho mejor generar los Keyframes de quaterniones manipulando directamente los huesos y editando estas curvas solo para ajustar transiciones de ataque y decaimiento.

Animación No Lineal

La Animación No Lineal (NLA) es una técnica de alguna manera similar a RVK, utilizada para mezclar Acciones simples individuales para generar Acciones complejas y fluidas. La ventana NLA ofrece un panorama general de toda la animación que transcurre en la escena. Desde aquí puedes editar el timing de *todas* las IPOs, como si estuvieran en la Ventana de Acción. La mayor parte de la funcionalidad es idéntica a la de la Ventana de Acción. La Ventana NLA puede verse con **CTRL-SHIFT-F12** (*Ventana NLA*).



Ventana NLA

También puedes usar esta ventana para realizar mezclas de Acciones y otras tareas de Animación No Lineal. Puedes agregar y mover Bandas de Acción en una manera semejante a la del Editor de Secuencias y generar transiciones de mezcla en las mismas. En la ventana NLA, las Acciones se muestran como bandas individuales debajo de la banda del objeto; todos los Keyframes de la Acción (incluyendo canales de restricción) se representan en una línea (*Acción Expandida en la Ventana NLA*). Para ver una vista expandida de la Acción, utiliza la Ventana de Acción.

Image:Manual-Part-IX-Action in nla.png
Acción Expandida en la Ventana NLA

Los Objetos con canales de restricción mostrarán una o más bandas adicionales debajo de la banda del objeto. La banda de restricción se puede seleccionar independientemente del objeto al que pertenece. (*Restricción Expandida en la Ventana NLA*).



Restricción Expandida en la Ventana NLA

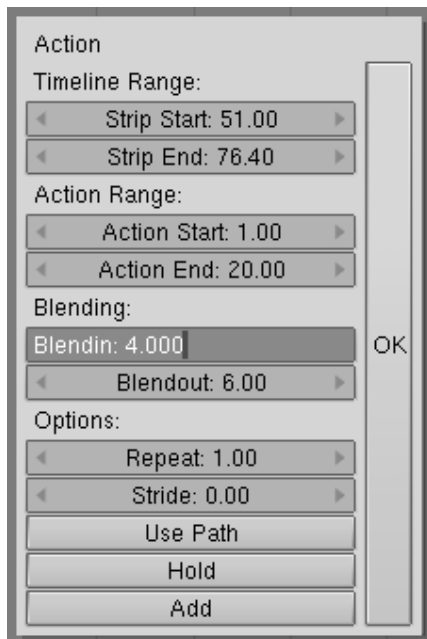
Haciendo click con **RMB** sobre los nombres de los objetos en la Ventana NLA seleccionará los objetos correspondientes en la Ventana 3D. Las bandas seleccionadas se representan en azul y las deseleccionadas en rojo. Puedes eliminar los canales de restricción de los objetos haciendo click con **RMB** sobre el nombre del canal de restricción y luego presionando **X**.

Nota: Observa que solo armatures u objetos con IPOs aparecerán en la ventana NLA.

Trabajando con Bandas de Acción

Se pueden agregar Bandas de Acción solamente a objetos Armature. Los objetos no tienen que tener necesariamente una Acción asociada a ellos inicialmente. Para agregar una Banda de acción a un objeto, mueve el puntero del ratón sobre el nombre del objeto en la Ventana NLA, pulsa **SHIFT-A** y elige la Acción deseada del menú emergente. Observa que solo puedes tener una banda de Acción por línea. En la Ventana NLA puedes seleccionar, mover y

borrar bandas de Acción junto con otros Keyframes. Las bandas se evalúan de arriba hacia abajo. Los canales especificados en bandas sucesivas reemplazarán a los canales especificados en las bandas precedentes. Puedes también crear una animación sobre el armature propiamente dicho. Los canales en la Acción local del armature, reemplazarán a los canales de las bandas. Observa que una vez creado un canal en la Ventana de Acción, siempre reemplazará todas las acciones. Si deseas reemplazar solo una parte de la línea de tiempo, puedes convertir la Acción local en una Banda de Acción presionando **C** con el ratón sobre el nombre del armature en la ventana NLA. Esto quita la Acción del armature y la pone al final de la lista de bandas de acción. Cada banda posee varias opciones a las que puedes acceder seleccionando la banda y pulsando **N** (*Opciones de la Banda de Acción NLA*). Las opciones disponibles son las siguientes:



Opciones de la Banda de Acción NLA

- *StripStart/StripEnd* - El primero y último cuadros de la banda de Acción en la línea de tiempo.
- *ActionStart/ActionEnd* - El rango de keys de la Acción que se leerán. El final puede ser menor que el comienzo lo que provocará que la Acción sea reproducida en sentido inverso (de atrás hacia adelante).
- *Blendin/Blendout* - La cantidad de cuadros de transición que se generarán entre esta Acción y la anterior en la lista de bandas de Acción.
- *Repeat* - El número de veces que el rango de la Acción debe repetirse. No es compatible con la opción *Use Path*.
- *Stride* - La distancia (en unidades de Blender) que el personaje se desplaza en un único ciclo de Acción (normalmente un ciclo de andar). Este campo solo es necesario si se activa la opción *Use Path*.
- *Use Path* - Si un armature es hijo de un path o curva y tiene un valor de *Stride*, este botón elige qué cuadro de la animación se mostrará de acuerdo a la posición del objeto sobre el path. Muy útil para ciclos de andar.
- *Hold* - Si esta está activada, el último cuadro de la Acción se mostrará eternamente a menos que sea reemplazada por otra Acción. De otra forma el armature volverá a su posición de descanso.

- *Add* - Especifica que las transformaciones de esta banda deberán *sumarse* a los datos de animación existente en lugar de reemplazarlos.

Rigging de bípedos.

Veremos cómo crear y configurar un armature para animar un personaje bípedo. La primer parte muestra cómo crear el armature, y la segunda cómo configurarlo agregándole constraints para hacerlo completamente funcional. En la parte 3 ajustaremos algunas propiedades de los huesos para obtener resultados naturales, y en la Parte 4 mostraremos y resolveremos típicos problemas de rigging en Blender.

Rigging Parte 1: Creando el armature

Rigging Parte 2: Añadiendo Restricciones

Rigging Parte 3: Ajustando límites de rotación

Rigging Parte 4: Problemas y Soluciones

Rigging de bípedos.

Introducción:

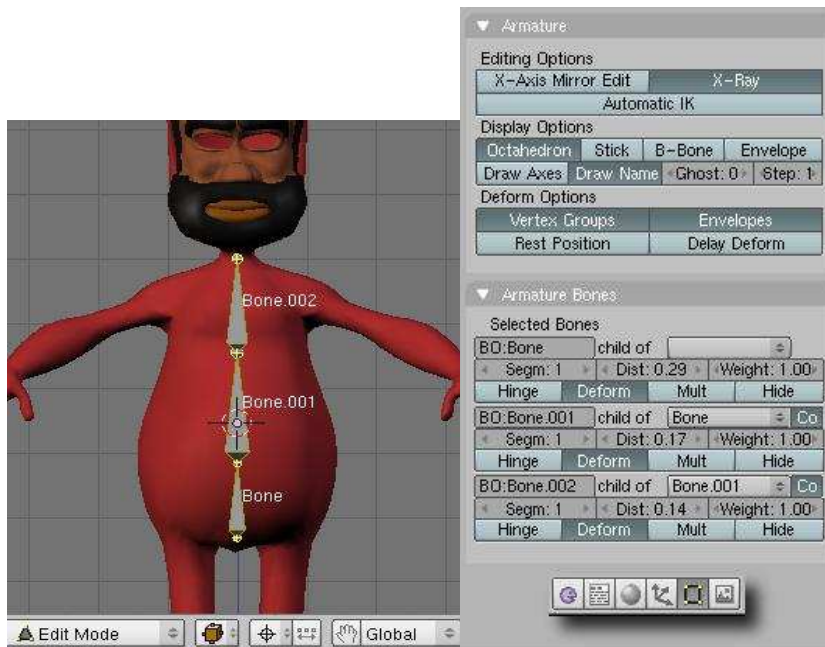
Hemos visto las propiedades básicas de un armature, ahora utilizaremos los conceptos aprendidos para crear un personaje bípedo tradicional que son la mayoría de los personajes.

Durante este capítulo utilizaremos el modelo "malefico", disponible para descarga.

Comenzando.

Cargaremos el modelo de nuestro personaje en la vista frontal. Si bien podríamos crear el armature completamente sin la ayuda de ningún personaje modelado, el tenerlo allí como referencia será útil a los fines didácticos.

Colocaremos el personaje en el centro de coordenadas, y allí ubicaremos el cursor. Vamos a crear un armature de tres huesos verticales como hemos visto en el capítulo anterior. Los 3 huesos harán las veces de cadera, abdomen y pecho por lo que les daremos dimensiones apropiadas. Crearemos los tres huesos haciendo click en la pantalla con CTRL pulsado. Una vez terminados, los ajustaremos para que queden perfectamente verticales. Les daremos nombres apropiados a cada uno.

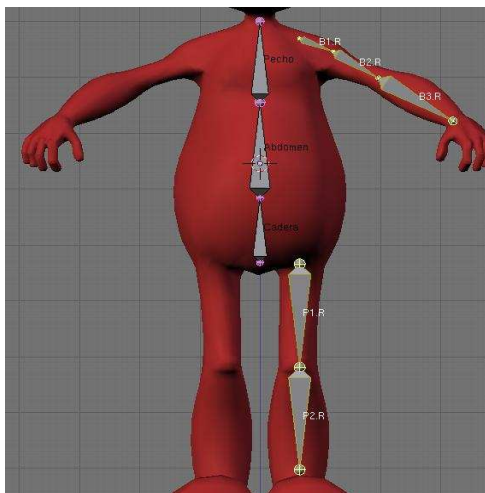


Ahora duplicaremos esta cadena y la usaremos para crear un brazo, también en la vista frontal. Solo haremos el brazo derecho. Nombraremos los 3 huesos de la siguiente manera: "B1.R", "B2.R" y "B3.R".

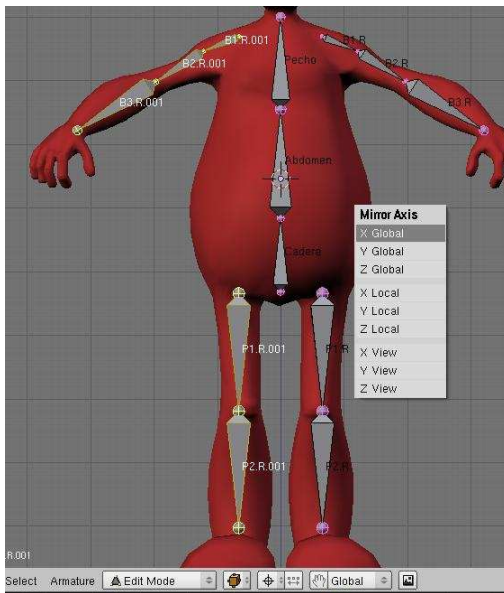
Ya hemos visto el significado del ".R" final.

Crearemos ahora la pierna derecha del personaje, moviendo el cursor a la zona derecha de la cadera, y usando el toolbox -> Add Bone. Crearemos una cadena de dos huesos solamente, con la articulación en la "rodilla" del personaje. Nombraremos los huesos como "P1.R" y "P2.L".

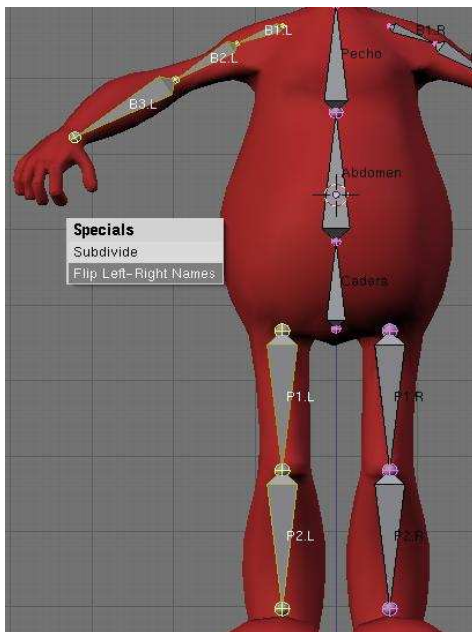
Saldremos momentaneamente del modo Edición y ubicaremos el cursor en el centro del armature.



Sin cambiar el cursor de lugar, entraremos otra vez a Modo Edición y seleccionaremos las cadenas del brazo y pierna (pulsar "L" sobre un hueso de cada cadena para seleccionarla entera). Las duplicaremos (ambas a la vez) usando SHIFT+D y presionando la barra espaciadora SIN MOVER los huesos nuevos. Ahora activaremos la REFERENCIA CURSOR, y pulsaremos "M" para entrar al menú "Mirror Axis". Elegiremos la opción "X Global" y pulsaremos la barra espaciadora para confirmar el comando. Quizás notemos algún tipo de rotación axial de los huesos al hacer el "mirror", si esto ocurre, lo corregiremos con CTRL+N (Recalc Roll Bones Angle) las veces que haga falta.



Tenemos ahora las dos cadenas en la posición opuesta, pero con nombres inútiles para nuestros fines. Seleccionaremos únicamente estas dos cadenas, y pulsaremos "W" y la opción "Flip Left-Right Names". Ya está ! tenemos ahora un armature básico listo !



Modelado simétrico.

Activaremos la opción "X-Axis Mirror Edit" en los botones de edición. Con esta opción activada y habiendo nombrado los huesos siguiendo la convención ".R/.L", al editar una mitad del armature se actualizarán los cambios en la otra mitad.

Hasta ahora habíamos trabajado exclusivamente en la vista frontal, no tenemos idea del aspecto de nuestro armature en la vista lateral, pero les aseguro que no está nada bien.

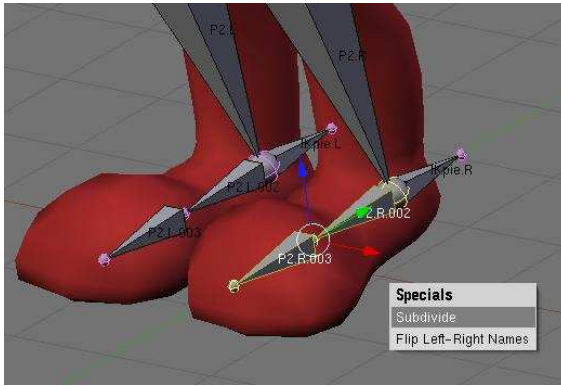
Teniendo activado el "X-Axis Mirror Edit" solo deberemos preocuparnos de editar una mitad del armature. Haremos entonces los ajustes necesarios en la vista lateral. Será de suma utilidad emplear los controles de traslación (widgets) accesibles a través del icono de la palma en la cabecera de la ventana 3D.

La idea es que los huesos queden totalmente cubiertos por la malla del personaje, como lo estarían los huesos cubiertos por carne en una persona real.



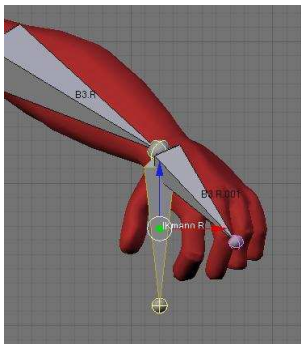
Seleccionaremos la cola del hueso P2.R y extrudaremos una vez "hacia atrás" en la vista lateral (fácilmente pulsando la tecla "Y" luego de la "E" de extrudar). Veremos que al mismo tiempo un hueso homólogo se crea en la otra mitad del armature.

Volveremos a seleccionar la cola del hueso P2.R y otra vez extrudaremos, pero esta vez "hacia adelante" hasta la punta del pie (otra vez usaremos la tecla "Y" para restringir a esa coordenada). Seleccionaremos este hueso entero, y los subdividiremos una vez (W -> Subdivide).



Los huesos que van "hacia atrás" los llamaremos "IKpie.R" e "IKpie.L" respectivamente.

Haremos algo similar en los brazos. Seleccionaremos la cola del hueso B3.R y lo extrudaremos "hacia abajo", luego volveremos a extrudarlo "hacia adelante" para crear un hueso que abarque totalmente el largo de la mano.



Los huesos que hicimos "hacia abajo" los llamaremos "IKman.R" e "IKman.L" respectivamente.

Por ahora no crearemos más huesos en la mano, eso lo veremos en un capítulo más adelante.

Jerarquías y emparentados

Seleccionaremos el hueso P1.R y sin deseleccionarlo, seleccionaremos también el hueso Cadera (en ese orden). Emparentaremos ambos huesos pulsando CTRL+P. Elegiremos la opción "Keep Offset". Una línea punteada se dibujará uniendo ambos huesos. Emparentaremos a la cadera el hueso P1.L también (de la misma manera).

Al emparentar estamos creando una jerarquía de huesos, significa que si movemos el hueso cadera (en modo Pose) también se moverán los huesos P1.L y P1.R, que a su vez moverán al resto de los huesos de las piernas del personaje.

Podemos distinguir entonces dos tipos de emparentado, aquel que une el hueso P1.R y al hueso P2.R por ejemplo, que sería un emparentado "Conectado" ya que no media entre ambos huesos ninguna distancia o separación; y aquel que une al hueso Cadera al hueso P1.R, o un emparentado "manteniendo la distancia", que se indica con una línea punteada.

Emparentaremos ahora los brazos (huesos B1.R y B1.L) al hueso Pecho.

Básicamente hemos terminado, tenemos un armature humanoide.

Como último paso de la creación de este armature nos aseguraremos que la orientación de los ejes de cada hueso sea coherente, de modo que cuando hablemos del "eje Z del hueso X", estemos seguros que ese eje apunta en la misma dirección que en los otros huesos.

Para ello nos pondremos en la Vista Frontal, seleccionaremos todos los huesos en Modo Edición (pulsar tecla "A") y luego pulsaremos CTRL+"N", lo que nos preguntará "Recalc Bone Roll Angle ?". Haremos clic con el ratón para afirmar.

Si somos observadores, veremos que algunos huesos se realinean a la vista de manera que ahora todo se ve más regular.

Ahora es el momento de empezar a añadir Constraints para que el armature se porte como es debido.

Rigging de bípedos.

Parte 2: Constraints

Introducción:

En la parte 1 hemos creado una serie de huesos, los hemos nombrado y aplicado jerarquías. En esta parte terminaremos el rigging básico aplicando "restricciones" o "Constraints" que automatizarán el comportamiento del armature completo dándole funcionalidades muy útiles para animar.

Las constraints son entidades que restringen el movimiento de los huesos y nos permiten controlar un armature y animarlo de manera más realista y natural. Las aplicaremos siempre en Modo Pose, y muchas veces será importante el orden de selección de los huesos por lo que se recomienda seguir las instrucciones "al pie de la letra".

Cinemática Inversa o "IK"

Cuando creamos los huesos de manos y pies, usamos nombres que a priori pueden parecer arbitrarios, en particular llamamos a algunos huesos "IKmano" o "IKpie". El prefijo "IK" que usamos está relacionado con "Inverse Kinematics" o Cinemática Inversa que es un mecanismo muy útil que pasaremos a explicar.

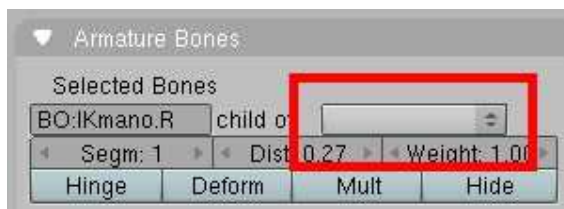
Cuando tenemos una cadena de huesos como se vió en "Jerarquías y emparentados", al mover el hueso raíz de la cadena, toda la cadena se mueve. Esto se llama "Cinemática Directa" o "Forward Kinematics" ("FK") en inglés. Imaginemos el caso de un brazo. Esto significaría que estando la raíz de la cadena en el hombro, si yo quisiera mover la mano (por ejemplo llevarla al pecho del personaje), debería primero rotar el hombro, luego rotar el biceps, y por último rotar la mano hasta la posición deseada.

Con el uso de IK en cambio, lo único que deberíamos hacer es tomar la mano y llevarla hasta el pecho del personaje. El hombro y el biceps rotarían automáticamente haciendo pivote en la raíz de la cadena.

Está claro que este tipo de sistemas son muchísimo más cómodos y prácticos para animar extremidades, que el sistema FK.

Vamos a ver como implementamos esta solución a nuestro armature en Blender.

Primero vamos a hacer unos ajustes a nuestro armature. El sistema de IK funciona con lo que se llama un "target" o blanco. Es decir, un hueso que al moverse indica la posición que deberá resolverse mediante rotaciones de los componentes de la cadena. Estos "targets" serán los huesos que hemos llamado "IKmano" o "IKpie". Para que puedan funcionar como targets, no deben estar emparentados a ningún hueso de la cadena que se quiere resolver por IK.



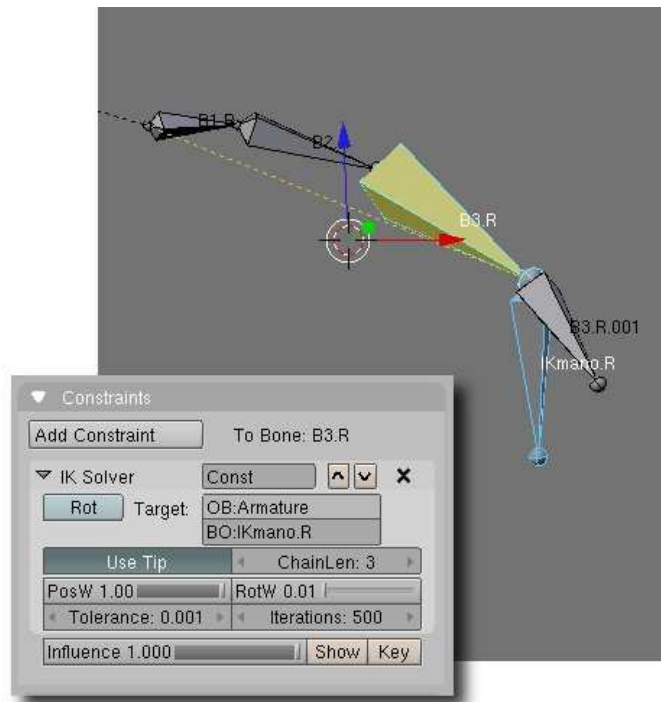
Por ello entraremos en Modo Edición, seleccionaremos el hueso "IKmano.R" y en los botones de edición, dejaremos en blanco el campo "Child of" ("hijo de"). Haremos lo mismo para todos los huesos "IK". Listo, ya podemos proseguir.

Restringiendo

Seleccionaremos el armature y entraremos en modo Pose. Vamos a crear una restricción llamada "IKSolver" en las extremidades de nuestro personaje que es la manera de aplicar en Blender la teoría explicada antes.

Seleccionaremos el hueso IKmano.R. y LUEGO el hueso B3.R. Ahora pulsaremos CTRL+I y elegiremos la opción "To Selected Bone". La lógica de selección sería, primero el target, luego el hueso que alojará la restricción.

Si bien hemos decidido usar huesos como blanco de las restricciones IK, también es posible utilizar empties. Usar uno u otro dependerá de la situación, a veces es mas sencillo utilizar empties, otras veces convendrá utilizar huesos.



El hueso B3.R se pondrá de color amarillo indicando que posee ahora una restricción de tipo IK Solver. Al mismo tiempo, aparecerá una línea punteada amarilla que va desde el "pivote" de la cadena, hasta el final de la misma. Los parámetros de esta restricción los podremos ver en los botones de Objeto, en el panel "Constraints", siempre que tengamos el hueso "amarillo" seleccionado en Modo Pose. Vemos que tiene activado el botón "Use Tip". Esto simplifica la configuración, ya veremos otros casos donde no se usará así.

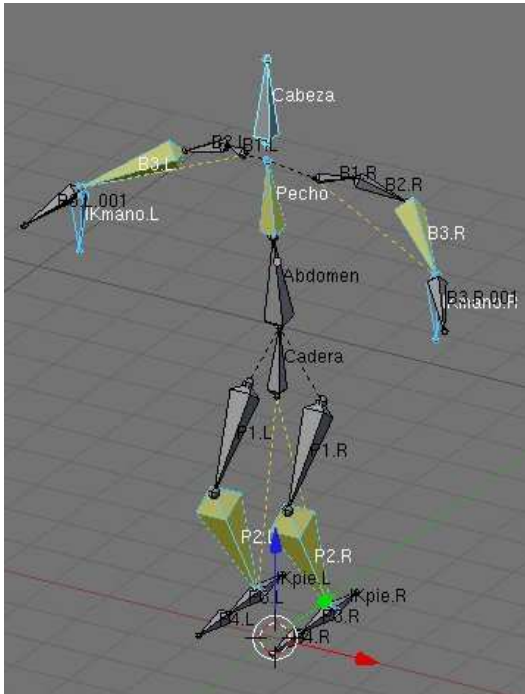
Si ahora (siempre en modo Pose) tomamos el hueso IKmano.R y lo movemos (no rotarlo) veremos como la cadena del brazo se acomoda para seguir a este hueso. No solo eso, otros huesos rotarán también, por ejemplo los huesos de la espina del personaje.

Esto se debe a que por defecto, la cadena IK se propaga siguiendo todas las relaciones jerárquicas que existan.

Para limitar el efecto, deberemos indicar un valor distinto de cero en la casilla ChainLen (Longitud de cadena) de la restricción IK, en este caso usaremos un valor igual a 3, puesto que la cadena se compone de 3 huesos.

Seguiremos poniendo restricciones IK para los huesos B3.L (apuntando a IKmano.L), el hueso P2.R (usando como target a IKpie.R), y P2.L (usando como target a IKpie.L).

Podemos probar la configuración moviendo estos huesos en Modo Pose. Los brazos y piernas deberían moverse en consonancia con los "blancos".



También podemos agregarle una restricción IK, a la columna. Probemos seleccionando el hueso "Cabeza" y luego el hueso "Pecho" y pulsando CTRL+I. Ahora al mover el hueso Cabeza, la columna se moverá haciendo pivote en el hueso "Cadera".

Algunas Bisagras

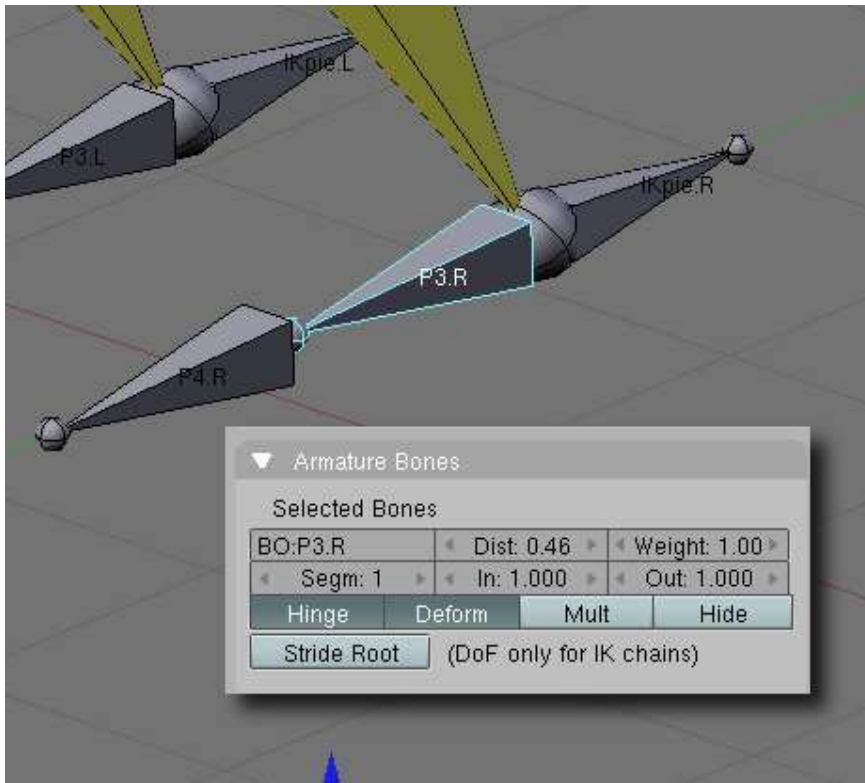
Observemos por un momento el pie del personaje. De la manera que hemos configurado los IK de pies y manos, si tomamos el hueso Cadera en Modo Pose y lo movemos "hacia abajo", veremos que ocurre algo curioso.

En efecto, los pies del personaje rotan haciendo pivote en el "talón". Esto por supuesto es algo extraño y complicará la animación pues deberemos rotar los pies para mantenerlos horizontales (es decir, "apoyados en el suelo").

Hay una manera muy sencilla de evitar este comportamiento y es mediante el uso de "Bisagras" ("Hinges" en inglés).

Cualquier hueso puede ser configurado como una bisagra con lo cual, la rotación del hueso precedente, no lo afectará de ninguna manera. En este caso lo que deseamos es que la rotación del hueso P2 no afecte a la orientación de los huesos P3. O lo que es lo mismo, queremos que P3 sea un hueso bisagra.

Para ello seleccionaremos el hueso P3 en Modo Edición y pulsaremos el botón "Hinge" en los botones de edición. Eso es todo.



Si ahora volvemos a mover el hueso Cadera en Modo Pose, veremos como el pie se mantiene horizontal, independientemente de lo que ocurra con el resto de la pierna.

Parte 3: Límites de Rotación

Introducción:

En la parte 2 hemos empezado a configurar nuestro armature añadiendole algunas restricciones IKSolver. Si bien todo parece funcionar maravillosamente, hay aún algunos detalles que deberemos ajustar.

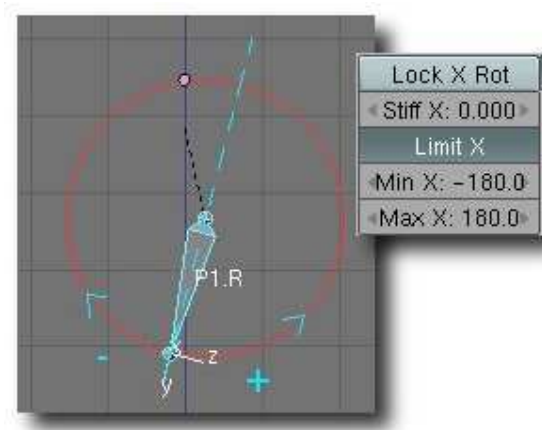
Cuando movemos el blanco de una cadena IK, por ejemplo para animar el brazo de un personaje, puede ocurrir que la cadena adopte rotaciones o ángulos anti naturales. Rotaciones que normalmente serían imposibles en un personaje real.

Las rotaciones posibles de un hueso se conocen como sus "grados de libertad" o "Degree of Freedom" (abreviado DoF, no confundir con Depth of Field, que también se abrevia DoF y que se refiere al bien conocido efecto de "Profundidad de Campo"). Así si un hueso tiene DoF=1 significa que solo puede rotar en un eje, si fuera DoF=2, podría rotar alrededor de 2 de sus ejes, etc.

Para evitar estos resultados desagradables, disponemos de una manera efectiva de limitar o acotar las rotaciones de los huesos que forman una cadena IK, es decir limitar sus grados de libertad.

Límites para las piernas

Para cada hueso que forma una cadena IK podemos optar entre bloquear completamente la rotación sobre cualquier eje local, o bien restringir esta rotación entre un par de valores desde -180 a +180 grados. Estos ángulos deben entenderse como tomados desde el cero en el eje local del hueso en reposo, y teniendo valores positivos en sentido antihorario, de la misma manera que se hace normalmente.

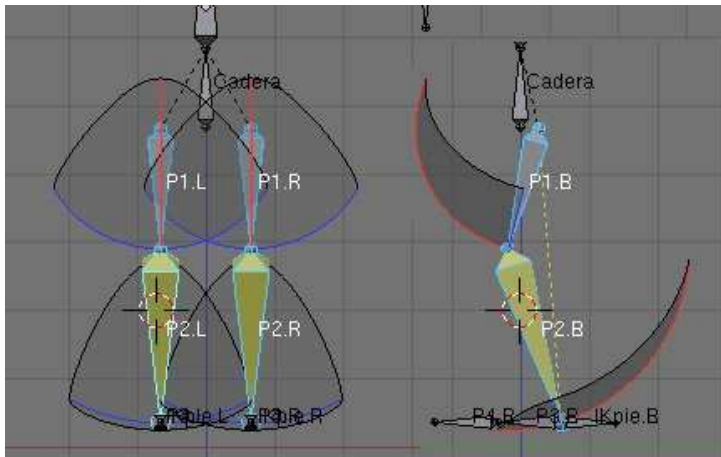


El rango -180 a +180 aparece como un círculo rojo

Ni bien activamos los botones "Limit X", "Limit Y" o "Limit Z", aparecerán graficados en Modo Pose fragmentos de arco indicando el rango permitido de rotación. Si limitamos más de un eje, aparecerá también graficada el área elipsoidal cubierta por estos rangos.

Si seleccionamos el hueso P1.R de la pierna derecha de nuestro armature en Modo Pose y vamos a los botones de edición, veremos en el panel Armature Bones, los botones de límites de rotación.

Para el muslo del personaje, podemos pensar que lo natural sería limitar las rotaciones en X (hacia adelante y atrás) y en Z (hacia izquierda y derecha). La rotación alrededor del eje Y (longitudinal) podemos anularla completamente pulsando el botón "Lock Y". Le daremos a Limit X un valor límite de modo que al levantar el muslo, no supere el rango entre 0 y -100 grados, de esta manera la rodilla del personaje se articulará siempre de manera natural. Para Limit Z usaremos un rango que limite las rotaciones laterales del hueso. En este caso opté por un rango -60 a +60 grados.



Los límites de rotación aparecen como áreas de esfera

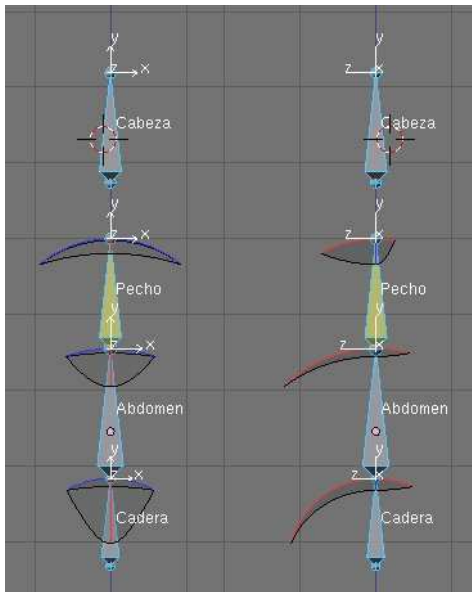
De la misma manera, limitaremos la rotación del hueso P2.R para que la rodilla nunca se doble "hacia adelante" (rotación alrededor del eje X) superando el ángulo de la rodilla (-20 grados), y que nunca se doble "hacia atrás" tanto como para que el personaje se pegue con los talones en las nalgas (70 grados). Los valores se resumen en la figura que sigue. Para las rotaciones "laterales" del hueso P2.R he optado por un valor conservativo de -30 a +30 que es más o menos lo que puedo mover yo mismo sin quebrarme la pierna y le he añadido un valor de Stiffness (Rigidez) a esa articulación para hacerla menos flexible.



Valores optativos para las piernas, he añadido algo de "Stiff" que actúa como "Rigidez" de la articulación.

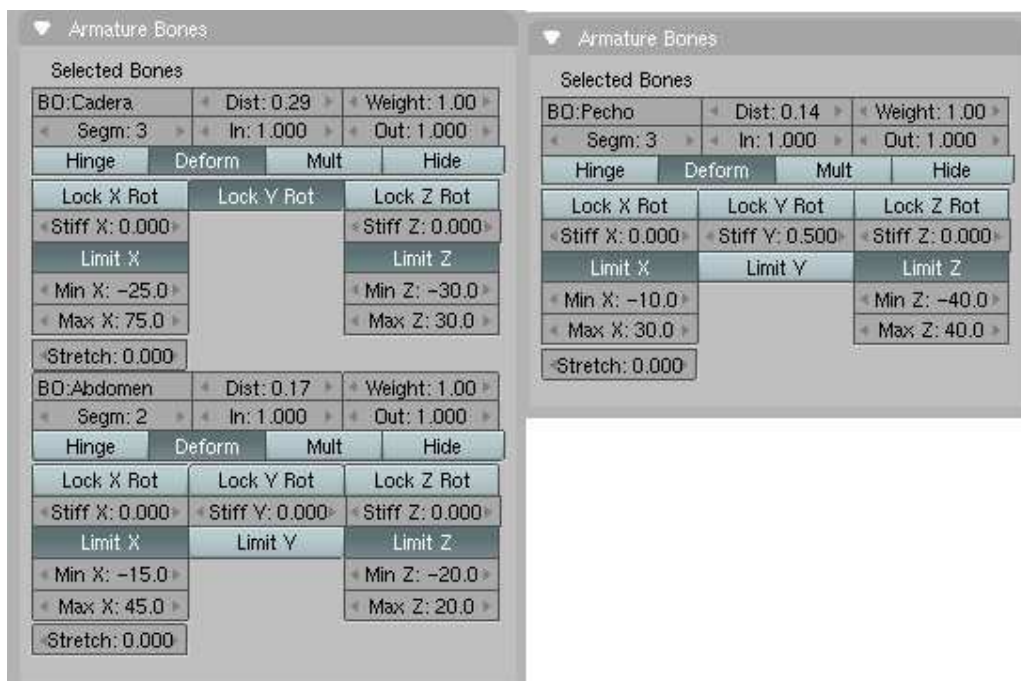
Límites para la columna

Procederemos a aplicar límites a los huesos de la columna, de forma que la cadera no pueda rotar libremente sobre el eje longitudinal Y (se vería realmente extraño si uno pudiera retorcerse 360 grados) y acotaremos las posibilidades de flexión de la cintura.



Vista frontal y lateral de la columna y sus límites de rotación

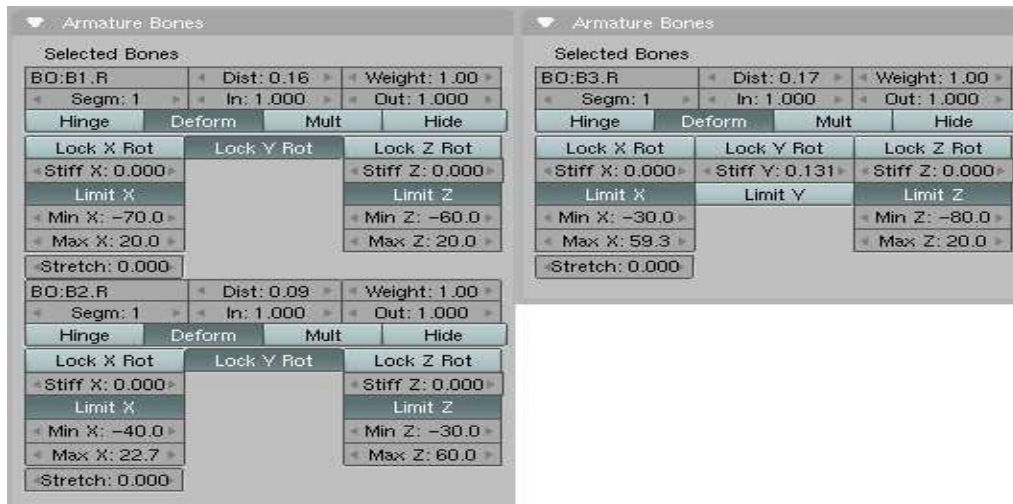
Observar que una vez activados los botones Limit, siempre se visualizarán en Modo Pose, los rangos de rotación permitidos de cada hueso.



Valores de limite de rotación para los huesos de la columna

Límites para los brazos.

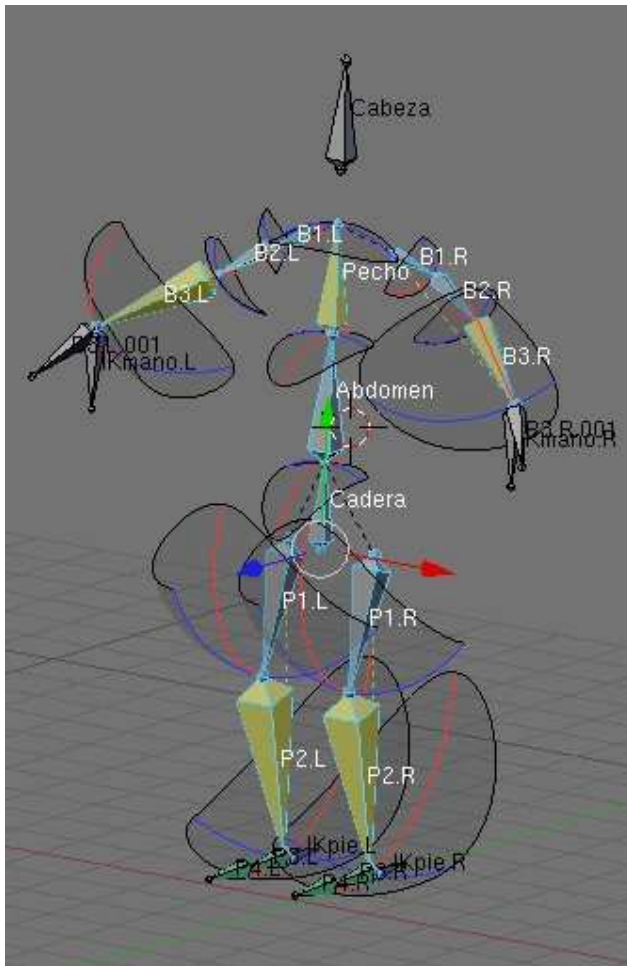
Para el caso de los huesos del brazo, adoptaremos límites que correspondan con los giros normales de hombro, biceps y antebrazo. Observar que si bien el biceps no puede "retorcerse", es decir, girar sobre su eje longitudinal, el antebrazo sí puede hacerlo en cierto grado, al igual que la muñeca. El hombro en cambio está más limitado.



Valores de limites de rotación para los huesos del brazo.

Conclusión

Ahora que hemos aplicado algunos límites de rotación, nuestras cadenas IK se comportan de manera más realista.



El armature completo con sus límites de rotación

Por supuesto, en cada personaje serán convenientes distintos valores de límites, podemos querer que nuestros personajes se porten más o menos rigidamente según la animación, así que estos valores que se muestran deben considerarse simplemente como ejemplos y no como leyes universales de la animación.

Parte 4: Problemas y Soluciones

Introducción:

Se han explicado hasta ahora varias situaciones de uso común como crear armatures para extremidades, añadirle restricciones, configurar límites de rotación, etc. Todo parece claro y funcionar bien, sin embargo hay situaciones en las que las cosas no funcionarán como esperamos.

En este capítulo se intentará mostrar y aclarar las típicas situaciones problemáticas al configurar piernas y brazos. Sobre el caso particular de manos y dedos, se verá un capítulo aparte más adelante.

El clásico problema de la rodilla invertida

Es bastante común que modelemos un personaje con las piernas totalmente extendidas para facilitar el mapeado de texturas. Cuando esto ocurre, en general creamos los huesos de la pierna totalmente extendidos y alineados perfectamente.

Esta configuración de pierna, normalmente provoca que al asignarle una restricción IK al extremo inferior (el pie) y flexionar la pierna, se produzca una articulación invertida en la rodilla. Es decir, en lugar de flexionarse "hacia atrás", la pierna se flexiona "hacia adelante" como si el personaje se hubiera fracturado la rodilla.

Este es un viejo problema, y en mi experiencia, los límites de rotación no lo han solucionado. La solución para este problema, es bastante simple. Lo único que deberemos hacer es en Modo Edición modificar nuestro armature en la vista lateral para que la "rodilla", es decir la cola del hueso "femur" y la raíz del hueso "tibia", se adelante, produciendo un ángulo menor de 180 grados entre ambos huesos. Asimismo, se moverá el "talón" "hacia atrás", para acentuar ese ángulo (hacerlo todavía menor).

No es necesario modificar la malla para esto. La deformación ocurrirá normalmente aunque los huesos no estén alineados dentro de la pierna o que incluso sobresalgan un poco los huesos por encima de la malla.

Alineación de Codos y Rodillas

A veces, no importa lo bien que configuremos nuestro armature, las rodillas y los codos rotarán de la manera equivocada. Intentemos animar nuestro personaje llevando su mano a la boca. Al usar cadenas IK, el codo flexionará pero no necesariamente como deseamos para que se vea natural. Esta rotación puede controlarse mediante un leve "torque" del hueso anterior al que posee la restricción IK, sin embargo este enfoque es bastante incómodo de implementar.

Para tener mayor control sobre estos giros de los huesos, se suele utilizar un hueso adicional, no conectado a ningún hueso de la pierna o el brazo, que sirve de "puntero" para la rodilla o el codo.

Veamos un ejemplo concreto. A nuestro armature favorito (ejem), le añadiremos un hueso a la altura del codo, pero alejado del cuerpo. Le pondremos un nombre adecuado a la convención para nombres, como "PunteroCodo.L". Este hueso lo emparentaremos al torso o el abdomen del personaje.

Ahora le añadiremos una restricción IK al BICEPS (hueso B2.L), y la apuntaremos a este "PunteroCodo.L". Lo más sencillo es seleccionar el hueso PunteroCodo.L, y luego el hueso B2.L, (en ESE ORDEN) y pulsaremos CTRL+I, para crear una restricción IK correcta.

Una vez hecho esto podemos probar cómo funciona. Si movemos el brazo usando el hueso ManoIK.L, el codo quedará en una posición que podría no ser la deseada. Si ahora movemos el hueso PunteroCodo.L, el brazo rotará siguiendo este "Puntero" y nos permitirá acomodar el codo en forma más natural.

De igual manera que los codos, para resolver la orientación de las rodillas, usaremos un hueso "PunteroRodilla.L" que lo configuraremos de manera análoga al codo. Esta vez la restricción IK la añadiremos al hueso P1.L y por supuesto usaremos como "Target" o "Blanco" al hueso "PunteroRodilla.L". Esta vez emparentaremos los huesos "PunteroRodilla" al hueso "Cadera".

PARTE X - CUERPOS BLANDOS Y FLUIDOS

El sistema SoftBody (cuerpos blandos) de Blender permite que los vértices se muevan basándose en las leyes de la física. Esto significa que pueden ser configurados para reaccionar a la gravedad y al viento. Los Objetos de Blender pueden ser configurados como cuerpos blandos. En la versión 2.37 únicamente están implementados los objetos *Mesh* y *Lattice*.

El sistema SoftBody está diseñado principalmente para mejorar los sistemas de animación, incluyendo la animación de personajes. Efectos como la piel flexible u ondulante son ahora muy fáciles de lograr.

En los archivos demo de la versión 2.37 (4 MB) pueden encontrarse dos ejemplos de cuerpos blandos, `softbody_basics.blend` y `wind_soft.blend` (<http://download.blender.org/demo/test/test237a.zip>).

nota: añadir algo sobre fluidos en estra intro

Capítulos

- Cuerpos Blandos
- Simulación de Fluidos

Cuerpos Blandos

Las bases

Hay dos métodos principales para controlar el efecto de cuerpo blando:

Goal (objetivo) - *Goal* actúa como una etiqueta en un conjunto de vértices seleccionados, controlando cuanto efecto tiene el cuerpo blando en ellos.

Con *Goal* activo al máximo (1.0), el objeto actuará como cualquier objeto animado normal (sin efecto de cuerpo blando). Ajustando *Goal* a 0.0, el objeto es influido únicamente por las leyes de la física. Ajustando los valores de *Goal* entre 0.0 y 1.0, se puede combinar entre tener un objeto al que sólo afecta el sistema de animación, y tenerlo únicamente afectado por el efecto de cuerpo blando.

Goal sirve también como memoria, para asegurarnos de que los objetos blandos no se deforman demasiado, acabando en la forma animada no blanda.

Utilizando el sistema de pesos de *Vertex Group* (grupo de vértices), puede definirse un peso *Goal* por vértice. Para hacer que esto parezca más natural, pueden definirse fuerzas elásticas para controlar qué distancia pueden alejarse los vértices de su posición original.

Springs (fuerza de retorno a la posición original) - *Edge Spring Stiffness* (rigidez de la arista) define cuánto intentan las aristas mantener su tamaño original. Por ejemplo, añadiendo aristas diagonales dentro de un cubo, lo volveremos más rígido (menos "gelatinoso"). Ajustando el parámetro *E Stiff*, los objetos intentarán, en mayor o menor medida, mantener su forma original, pero continuarán ajustándose a las dinámicas.

Note: Cuando activamos el efecto de Cuerpo Blando en un objeto, será siempre simulado con antelación en el tiempo. Al movernos hacia atrás en el tiempo o saltando en pasos superiores a 9 fotogramas reiniciará el cuerpo blando a su posición original. Use el play back de la ventana TimeLine (línea de tiempo) para ajustar interactivamente los efectos de Cuerpo Blando.

Una vez esté satisfecho con la simulación, puede hornear (*Bake*) la simulación en un sistema de animación estático. Un cuerpo blando horneado se anima mucho más rápidamente en pantalla, y deja de depender de la simulación.

Note: Se recomienda hornear los cuerpos blandos al renderizar animaciones, puesto que la simulación no funciona correctamente para renderizado con Motion Blur (borrosidad de movimiento), o para renderizado por fragmentos a través de un sistema de renderizado por red.

El uso de *SoftBody* para simulación de tela, especialmente con detección de colisión, está todavía en fase de pruebas. El código de colisión actual necesita mejoras, planeadas para una futura versión. Para un control más preciso sobre la simulación de tela, se añadirán opciones de arista especiales en una próxima versión. Actualmente, todas las aristas tienen un efecto idéntico en el cuerpo blando.

Puesto que los vértices en los Cuerpos Blandos son tratados como partículas, las opciones *Force fields* y *Deflectors* se aplican igualmente. Tenga en cuenta que deflection (colisiones) funciona únicamente en Mallas no deformadas (sin usar hooks, armaduras, lattices, etc). Véanse las notas de la versión para más información.

Interfaz

Se puede acceder a la interfaz de SoftBody mediante el panel *Object* (F7), bajo la pestaña *Softbody*. Véase *Ajustes de Softbody*.

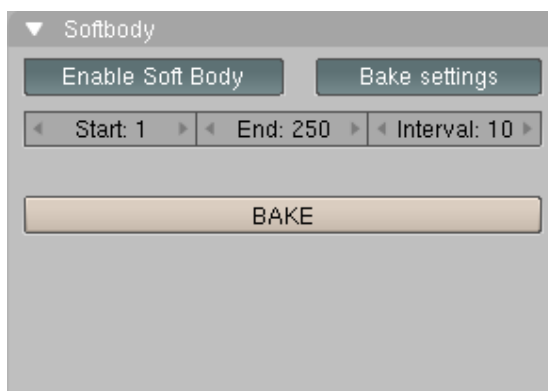


Ajustes de Softbody.

Ajustes de Softbody

- **Enable Soft Body** (habilita Cuerpo Blando) - Habilita el sistema softbody en el objeto seleccionado.
- **Bake settings** (opciones de horneado) - Abre las opciones de horneado. Véase *Ajustes de Bake*.
- **Friction** (fricción) - Controla la cantidad de fricción que tiene el objeto. Un valor alto significa que las fuerzas, por ejemplo *Gravity* (gravedad) o *Wind* (viento), tienen más facilidad para tomar el objeto y moverlo.
- **Grav** - Gravedad, cantidad de fuerza en la dirección del eje Z negativo. El valor de la gravedad terrestre es aproximadamente 9.8.
- **Error Limit** (límite de error) - Límite *Runge-Kutta*. Define el tamaño del paso durante la simulación. Valores pequeños dan una precisión mayor, pero incrementan el tiempo de cálculo.
- **Mass** (masa) - Valores de masa para los vértices. Mayor masa frena el movimiento, excepto para la gravedad, donde el movimiento es fijo independientemente de la masa.
- **Speed** (velocidad) - Con este valor puede controlarse la temporización interna del sistema softbody.
- **Apply Deform First** (aplicar la deformación primero) - Aplica los cálculos de softbody después de otras deformaciones, tales como las causadas por Lattices o Armaduras.
- **Use Goal** (usar objetivo) - Usa el movimiento de animaciones para la simulación (Ipo, Deform, Parents, etc). "Goal" es la posición final deseada para los vértices basada en esta animación. Cómo tratará el softbody de adquirir este objetivo puede ser definido usando fuerzas de flexibilidad y atenuación de la oscilación.
- **Goal** (objetivo) - El peso objetivo por defecto para todos los vértices cuando no hay grupo de vértices asignado (Vertex Group). Si hay grupo de vértices asignado, este botón muestra el nombre del grupo de vértices objetivo.

- **G Stiff** - Rigidez del objetivo. Un valor pequeño crea una fuerza de retorno muy débil ("agarre" más flexible al objetivo), un valor alto crea una fuerza de retorno fuerte ("agarre" más rígido al objetivo).
- **G Damp** - La atenuación de oscilación del objetivo. Valores altos atenúan el efecto elástico del objetivo en el cuerpo blando.
- **GMin/GMax** - Cuando se pintan los valores en grupos de vértices (usando el modo WeightPaint - pintado de pesos), pueden usarse GMin y Gmax para afinar (fijar) los valores de peso. El menor peso de vértice (azul) será GMin, el valor más alto (rojo) será GMax.
- **Use Edges** (usar aristas) - Las aristas de un objeto Malla (si las hay, comprobar en Editing->Mesh Panel) pueden usarse también como elementos flexibles con fuerza de retorno.
- **Stiff Quads** (rectángulos rígidos) - Para caras rectangulares, las aristas diagonales se usan con fuerza de retorno. Esto evita que las caras se aplasten por completo.
- **E Stiff** - Rigidez de las aristas (cuánto se esturan las aristas). Un valor bajo significa fuerzas de retorno muy débiles (material muy elástico), un valor alto es una gran fuerza de retorno (material más rígido).
- **E Damp** - Atenuación de la oscilación de las aristas. Valores altos atenúan el efecto E Stiff.



Ajustes de Bake.

Ajustes de Bake

- **Start/End** (principio/fin) - Configura el rango del cuerpo blando a ser horneado.
- **Interval** (intervalo) - Indica el número de fotogramas entre cada "paso" de horneado (La "resolución" del resultado horneado). Las posiciones intermedias serán calculadas usando los pasos como fotogramas clave, con interpolación B Spline.
- **Bake** (hornear) - Inicia el proceso de horneado. Dependiendo de la complejidad, esto puede demorar un poco. Puede pulsarse **ESC** para detener el horneado. Una vez horneado, este botón cambia a un botón "Free Bake" (liberar el horneado). Debe liberarse el resultado horneado para modificar los ajustes de cuerpo blando.

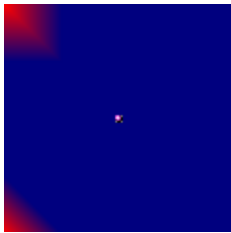
Ejemplo

Nuestro ejemplo mostrará un modo de hacer una simple bandera ondeando al viento.

Cree un plano en vista frontal y subdivídalo 3 veces. Vaya a los botones de Editing **F9** y active **Subsurf**. Coloque subsurf level a **2** para mejores resultados. Presione **Set Smooth**.

Crearemos ahora dos puntos de sujeción en nuestra bandera en la esquina superior e inferior de nuestro plano.

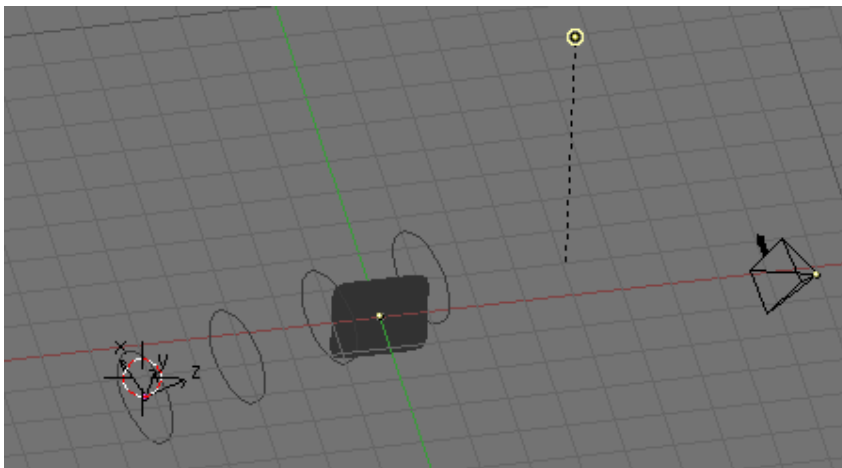
- Cree un nuevo Vertex Group, y ajuste *Weight* a 0. Seleccione todos los vértices, y presione **Assign**.
- Ahora, seleccione la esquina superior e inferior de un lado de la bandera. Coloque *Weight* a 1.0, y presione **Assign** nuevamente. Esto hará que estos vértices se mantengan donde están durante la simulación de softbody. En modo *Weight Paint* debería verse algo como en *Ejemplo de configuración de pesos*.
- Ahora, salga de modo Edit, y vaya al panel Softbody en los botones Object F7. Haga clic en **Enable Soft Body**. Incremente *Grav* a 9.8. Active el botón *Use Goal*. Haga clic en el pequeño botón junto a *Use Goal* y elija el nombre del *Vertex Group* a usar como objetivo; en este caso, la única opción debería ser el nombre por defecto *Group*. Ahora ajuste la rigidez de arista *E Stiff* a 0.9, coloque *Mass* a 0.5, *Friction* a 0.14 y *Speed* a 2.
- Ahora, puede presionar **ALT-A** para ver la bandera reaccionando a la gravedad.



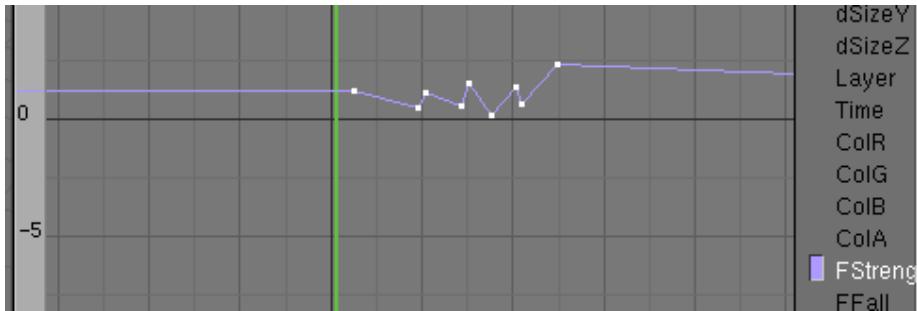
Ejemplo de configuración de pesos.

Ahora añadiremos algo de viento a la simulación.

- Añada un objeto empty a la escena donde estará situada la fuente de viento. Seleccione la pestaña *Particle Interaction* y active el botón *Wind*. Ajuste *Strength* a 1.
- Ahora rote y mueva el empty de tal modo que el eje Z apunte hacia la bandera. Véase *Ejemplo de IPO de configuración de viento*.
- Ahora puede presionar **ALT-A** para ver el la bandera reaccionando al viento.
- Añadir una curva IPO para simular la fuerza cambiante del viento añadirá más realismo a la animación. Véase *Ejemplo de IPO de fuerza del viento*.



Ejemplo de configuración de viento.



Ejemplo de IPO de fuerza del viento.

Simulación de Fluidos

El siguiente capítulo describe cómo animar líquidos usando el simulador de fluidos integrado en Blender a partir de la versión 2.40.



Ejemplo de animación de una presa rompiéndose.

Visión general del flujo de trabajo

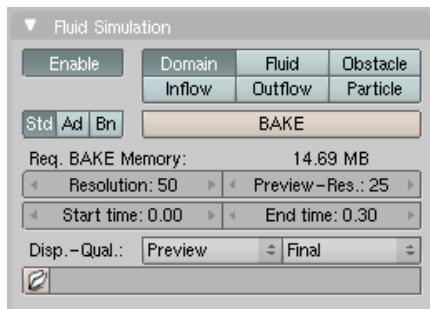
- Al modelar una escena con Blender, ciertos objetos pueden ser marcados para participar en la simulación de fluidos, por ejemplo como fluido, o como obstáculo. La caja envolvente de otro objeto será usada para definir una región con forma de caja para simular el contenedor del fluido.
- Los parámetros globales de la simulación tales como la viscosidad y gravedad pueden ser establecidos para este objeto dominio.
- Utilizando el botón *Bake* (hornear), la geometría y los ajustes son exportados al simulador y la simulación de fluidos se lleva a cabo, generando una malla de superficie junto con una vista previa para cada fotograma de la animación, y guardándolos en disco.
- La superficie del fluido apropiada para cada fotograma es cargada desde disco y mostrada o renderizada en Blender.

Las dos imágenes encima y debajo de este párrafo son un ejemplo de animaciones de fluidos creadas con el simulador El'Beem en Blender (y renderizadas usando Yafray).

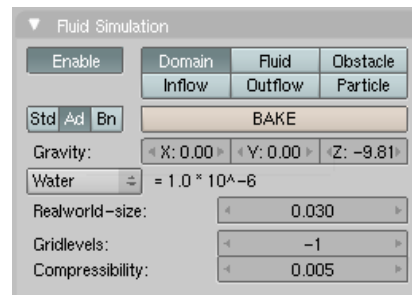


Otra animación de ejemplo de una gota cayendo.

Explicación de los Ajustes de GUI



Las opciones básicas (y necesarias frecuentemente) de simulación de fluidos.



Opciones avanzadas necesarias con menos frecuencia.

Volcado de pantalla de la GUI actual para un objeto dominio. Después de habilitar la simulación de fluidos para un objeto, puede escogerse el tipo (dominio, fluido, obstáculo,...), dando más ajustes para cada tipo.

Después de habilitar la simulación de fluidos para un objeto malla, están disponibles los siguientes tipos de objeto:

- *Domain* (dominio): la caja envolvente de este objeto será el receptáculo de la simulación. Note que la forma del objeto no importa (por lo que normalmente no habrá motivo para usar otra forma que una caja). Si necesita obstáculos u otros receptáculos que no sean una caja, debe insertar objetos obstáculo adicionales. Actualmente debería haber un único objeto dominio de simulación de fluidos. Las longitudes de los lados de la caja envolvente pueden ser diferentes.
 - *Resolution* (resolución): La resolución a la que se realiza la simulación de fluidos. Es probablemente el parámetro más importante de la simulación puesto que determina la cantidad de detalle en el fluido, el uso de disco y memoria, así como el tiempo de cálculo. Note que la cantidad de memoria necesaria se incrementa rápidamente: una resolución de 32 requiere 5 MB, 64 requiere 40 MB, mientras 128 ya necesita más de 250 MB. Actualmente la resolución está limitada a 200 (850 MB), para evitar el colapso de Blender debido al uso masivo de memoria. Si el dominio no es cúbico, la resolución se tomará para el lado más largo. Las resoluciones a lo largo de los otros lados serán reducidas de acuerdo a sus longitudes.
 - *Preview-Res.*: Es la resolución a la que se generarán las mallas de superficie de previsualización. Por lo tanto no influencia la simulación actual, e incluso si no

se visualiza nada en la previsualización, podría haber una fina superficie de fluido que no puede ser resuelta en la previsualización.

- *Start time* (tiempo de inicio): Momento en la simulación (en segundos) del primer fotograma de blender. Así, esta opción hace que la animación de Blender se inicie más tarde en la simulación.
 - *End time* (tiempo final): Momento en la simulación del último fotograma de blender.
 - *Disp.-Qual.*: Cómo mostrar una simulación horneada en la GUI Blender (primer menú desplegable) y en el render (segundo): geometría original, malla de previsualización o malla final. Cuando no se encuentran datos horneados, se mostrará la malla original por defecto.
 - Directorio *Bake* (hornear): Directorio y prefijo de archivo con los que almacenar las mallas de superficie horneadas. Es similar a los ajustes de salida de render, sólo que la selección de archivo es un poco especial: al seleccionar cualquiera de las superficies de malla previamente generadas (como `untitled_OBcube_fluidsurface_final_0132.bobj.gz`), el prefijo será establecido automáticamente (`untitled_OBcube_` en este ejemplo). Así la simulación puede realizarse varias veces con ajustes distintos, y permite cambios rápidos entre los diferentes ajustes de los datos de superficie.
 - Botón *Bake*: Lleva a cabo la simulación de fluidos actual. La GUI de Blender se congelará y mostrará el fotograma simulado actualmente. Presionando **ESCAPE** se abortará la simulación. Seguidamente dos `.bobj.gz` aparecerán en el directorio seleccionado para cada fotograma.
 - Botón *Advanced* (avanzado): Haciendo clic en este botón se mostrarán más opciones avanzadas ,que normalmente no deben modificarse con mucha frecuencia.
 - *Gravity vector* (vector de gravedad): Fuerza y dirección de la aceleración de gravedad. Actualmente, la componente principal debería ir a lo largo del eje Z negativo [m/s^2]. Actualmente no debería ser cero (por lo menos un número pequeño).
 - *Viscosity* (viscosidad): Viscosidad, es decir "grosor" del fluido. Puede bien introducirse un valor directamente o usar uno de los preajustes. Para entrada manual, el valor se especifica en punto flotante, que debería ser alrededor de 1.0 con exponente negativo. Ello simplifica la entrada de números muy pequeños, puesto que la viscosidad de por ejemplo el agua, es alrededor de 10 a la menos seis.
 - *Real-World size* (tamaño mundo real): Tamaño del objeto dominio en el mundo real en metros. Si desea crear un vaso de agua, ello podría ser 0.2 metros, mientras que para una simple gota sería más adecuado un centímetro (así 0.01m). El tamaño establecido aquí es para el lado más largo de la caja dominio envolvente.
 - *Gridlevel* (nivel de rejilla): Cuántos niveles de rejilla adaptativa a ser usados durante la simulación. Estableciéndolo a -1 se realizará la selección automática.
 - *Compressibility* (compresibilidad): Si tiene problemas con grandes regiones de fluido estático a alta resolución, reducir este número puede ayudar (note que ello aumentará el tiempo de cómputo).
- *Fluid* (fluido): Todas las regiones de este objeto que están dentro de la caja dominio envolvente se usarán como fluido actual en la simulación. Si coloca más de un objeto

Fluido dentro del Dominio, no deberían interseccionar. Asegúrese también de que las normales de superficie están apuntando hacia fuera. A diferencia de los objetos dominio, la geometría de la malla actual es usada para los objetos fluido.

- *Initial velocity* (velocidad inicial): Velocidad del fluido al inicio de la simulación en metros por segundo.
- *Obstacle* (obstáculo): Este objeto se usará como obstáculo en la simulación. Como con los objetos fluido, la geometría de malla actual es utilizada para los obstáculos.
 - Sin ajustes actualmente...
- *Inflow* (flujo de entrada): Este objeto añadirá fluido a la simulación (piense en un grifo de agua).
 - *Initial velocity* (velocidad inicial): Velocidad del fluido que se crea dentro del objeto.
- *Outflow* (flujo de salida): Cualquier fluido que entra en la región de este objeto será borrado (piense en un desagüe). Puede ser útil en combinación con el flujo de entrada para evitar el desbordamiento del objeto dominio.
 - Sin ajustes aquí...

Información general de fondo

La animación de fluidos puede tomar mucho tiempo. Cuanto mejor comprendamos cómo funciona, más fácil resultará estimar cómo serán los resultados. El algoritmo utilizado en este proyecto es el *Lattice Boltzmann Method* (LBM); hay otras aproximaciones, como los solucionadores *Navier-Stokes* (NS) y los métodos *Smoothed Particle Hydrodynamics* (SPH). Puede asumirse el LBM como a medio camino entre los otros dos. En general es muy duro para los ordenadores simular correctamente incluso un tanque de agua de 1 metro. Para simular una ola rompiendo contra una ciudad, se necesitaría probablemente uno de los superordenadores más caros que pudiéramos conseguir, y aún y así podría no funcionar adecuadamente (independientemente de cuál de los 3 algoritmos anteriores usáramos). Pero de forma similar a lo que los cineastas han estado haciendo en películas "analógicas" durante años, podemos pretender tener una ola en una ciudad construyendo un modelo menor, poner una ola pequeña en el modelo, y esperar que nadie se dé cuenta de la diferencia entre una ola de 100m y una de 1m.

Para el solucionador LBM, las siguientes cosas harán que la simulación sea más dura de calcular:

- dominios largos
- larga duración
- viscosidades bajas
- altas velocidades

La viscosidad del agua es ya realmente baja, por lo que especialmente para resoluciones bajas, la turbulencia del agua no puede ser capturada correctamente. Si nos fijamos atentamente, la mayoría de las simulaciones de fluidos en gráficos por ordenador no parecen agua real todavía hasta ahora.

En general, no confíe demasiado en los ajustes físicos (tal como tamaño físico del dominio, o duración de la animación en segundos). Más bien intente conseguir el movimiento general correcto con una resolución baja, y luego incremente la resolución tanto como sea posible o deseable.

Notas/FAQ/Problemas conocidos

- No se sorprenda, pero se encontrará con un amplio abanico de archivos de malla (.bobj.gz) después de la simulación; básicamente 2 por fotograma. Contienen el resultado de la simulación, así que los necesitará. Actualmente estos archivos no serán borrados automáticamente, por lo que es una buena idea crear una carpeta dedicada a almacenar los resultados de la simulación. Hacer una simulación de fluidos es similar a hacer clic en el botón *ANIM*: debe preocuparse de organizar las mallas de superficie de fluido usted mismo. Si quiere dejar de utilizar la simulación de fluidos puede simplemente eliminar todos los archivos *fluid*.bobj.gz.
- Si la configuración parece no funcionar asegúrese de que todas las normales son correctas (en modo edición, seleccione todo, y recalcula las normales a un tiempo).
- Antes de ejecutar una simulación a alta resolución que puede tomar horas, repase la animación entera mediante ejecuciones a resoluciones más bajas.
- En los objetos dominio sólo se utiliza la caja envolvente, pero los objetos fluido y obstáculo pueden ser mallas con geometrías complejas. Si son muy finos podrían no aparecer, sin embargo, si la resolución elegida es demasiado ruda para resolverlos (incrementarla podría solucionar el problema).
- Actualmente hay un problema con la simulación con gravedad cero. Simplemente seleccione una unidad de gravedad muy pequeña hasta que se solucione.
- Note que todos los parámetros animados u objetos no son tomados en consideración durante la simulación. Sólo se utiliza la configuración inicial en el fotograma 1, pero será posible en futuras versiones.
- Blender se congela después de pulsar *BAKE*: presionando **ESC** vuelve a hacerlo funcionar, después de un lapso de tiempo (ello puede




Otro ejemplo de simulación de fluidos creado con Blender y Yafray

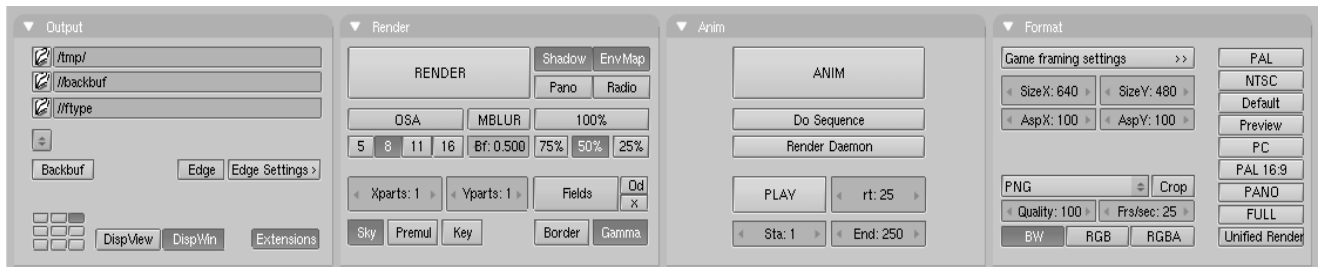
suceder si la resolución es muy grande y la memoria es intercambiada en disco, haciéndolo todo horriblemente lento. Reducir la resolución debería ayudar en este caso.

- Blender se cuelga al presionar *BAKE*: esto puede pasar si la resolución es realmente alta y se usan más de 2GB, causando el colapso de Blender. Reducir la resolución debería ayudar también...
- Las mallas deberían ser cerradas, así que si alguna parte de por ejemplo un fluido no son inicializadas como fluido en la simulación, asegúrese de que todas las partes de vértices conectados son mallas cerradas. Por desgracia, la malla Suzanne (mono) de Blender no es una malla cerrada (los ojos están separados).
- Si experimenta problemas, o algo no funciona como usted piensa que debería funcionar, envíe el archivo .blend y una descripción del problema a *nils arroba thurey punto de*.

PARTE XI - RENDERIZADO

El render es el proceso final del CG (antes de la postproducción, por supuesto) y es la fase en la que la imagen que se corresponde con la escena 3D se crea finalmente.

La ventana de botones de render es accesible a través del panel Escena (Scene Context) y el subpanel render. (F10 o el botón ). Los paneles y botones de render se pueden ver en *Botones de render*.



Botones de render.

El render de la escena actual se realiza presionando el botón grande *RENDER* en el panel *Render*, o presionando F12. El resultado del render es almacenado en un buffer y se ve en la misma ventana. Puede ser guardado presionando F3 o a través del menú *File>>Save Image*.

La imagen es renderizada en función de las dimensiones definidas en el panel *Format* (*Tipos de imágenes y dimensiones*).



Tipos de imágenes y dimensiones.

Por defecto las dimensiones *SizeX* y *SizeY* son 320x256 y pueden ser modificadas con cualquier botón numérico. Los dos botones más abajo definen el ratio de proporción de los pixels. Éste es un ratio entre las dimensiones X e Y de los pixels de la imagen. Por defecto está en 1:1 dado que los pixels de la pantalla son cuadrados, pero puede ser modificado si es para televisión, ya que sus pixels no son cuadrados. Para simplificar, el bloque de botones trae algunas configuraciones predeterminadas (*Dimensiones de imagen predeterminadas*)



Dimensiones de imagen predeterminadas.

- *PAL* 720x576 pixels ratio 54:51.
- *NTSC* 720x480 pixels ratio 10:11.
- *Default* Igual que PAL, pero con todas las opciones para TV, como se explicará en las siguientes secciones.
- *Preview* 640x512 con ratio de 1:1. Automáticamente reduce la imagen al 50% para producir una imagen real de 320x256.
- *PC* 640x480 con ratio de 1:1.
- *PAL 16:9* 720x576 con ratio 64:45, para renders a TV en widescreen 16:9.
- *PANO* Configuración panorámica estándar 576x176 con ratio de 115:100. Más info sobre el render 'panorámico' en la sección pertinente.
- *FULL* 1280x1024 con ratio de 1:1.

Capítulos

- Antialiasing
- Animaciones
- Formatos de Salida
- Opciones de R nder
- El Renderizador Unificado
- YafRay
- Preparando su Trabajo para V deo

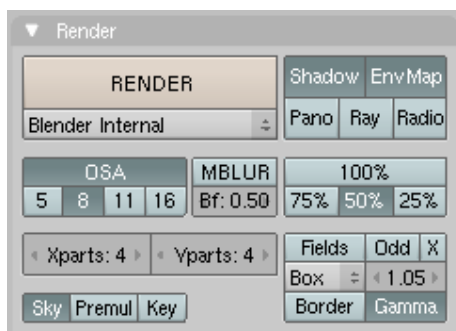
Antialiasing

Una imagen generada por ordenador está compuesta de píxeles. Estos píxeles pueden ser por supuesto sólo de un color. En el proceso de renderizado el motor de render debe asignar un color a cada pixel basándose en qué objeto se muestra en ese pixel.

A menudo esto conduce a unos resultados pobres, especialmente en los límites muy contrastados, o donde hay líneas delgadas, y es particularmente evidente en las líneas oblicuas.

Para solucionar este problema, que se conoce como *Aliasing*, es posible recurrir a una técnica Anti-Aliasing. Básicamente, cada pixel se 'sobremuestra', renderizándolo como si fuera 5 píxeles o más, y asignando un color 'medio' al pixel renderizado.

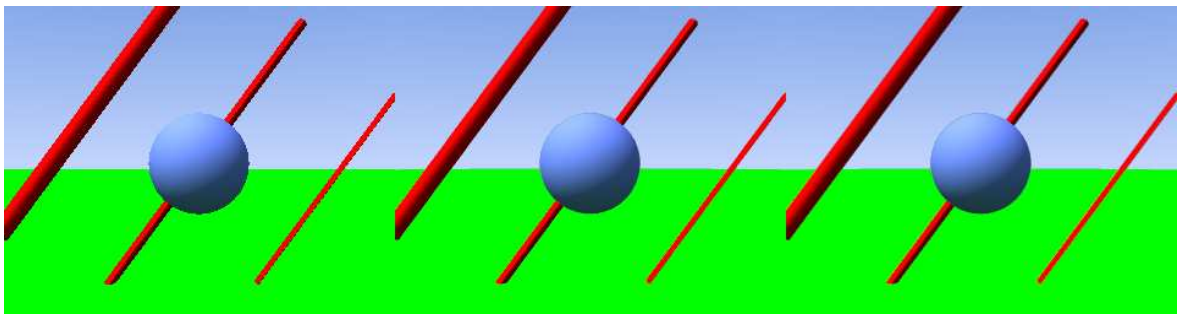
Los botones para controlar el Anti-Aliasing, o el SobreMuestreo (OSA, OverSampling), se encuentran debajo del botón de render en el panel *Render*. Presionando el botón *OSA* se activa el antialiasing. Seleccionando uno de los cuatro botones numéricos bajo el, se determina el nivel de sobremuestreo (de 5 a 16).



Panel Render.

Blender usa un sistema de renderizado de Acumulación Delta con muestreo jittered. Los valores de *OSA* (5, 8, 11, 16) son números predefinidos que especifican el número de muestras; un valor más alto produce mejores bordes, pero ralentiza el proceso de renderizado.

La figura *Render sin OSA (izquierda)* con *OSA=5 (centro)* y *OSA=8 (derecha)* muestra un render con OSA desactivado y con 5 u 8 muestras OSA.



Render sin OSA (izquierda) con OSA=5 (centro) y OSA=8 (derecha).

Animaciones

El renderizado de una animación se controla a través del panel *Anim* (*Botones para renderizar una animación*).



Botones para renderizar una animación.

El botón *ANIM* comienza el renderizado. El primero y el último frame de la animación se configura con los dos botones numéricos del fondo (*Sta:* y *End:*), que por defecto están a 1 y a 250. Por defecto la animación de la escena 3D se renderiza, entonces para hacer uso del editor de secuencia el botón *Do Sequence* debe estar seleccionado.

Por defecto la animación se renderiza en el directorio especificado en el Panel *Output* (*Localización de la animación y las extensiones*). Si se ha seleccionado el formato AVI, entonces el nombre será #####_#####.avi donde '#####' indica el frame inicial y el final de la animación con 4 numeros enteros añadiendo deros a la izquierda si fuese necesario.



Localización de la animación y las extensiones.

Si se elige, por otro lado, un formato de imagen, una serie de imagenes llamadas #####, (siendo '#####' el numero de frame correspondiente) se crea en el directorio. Si fuese necesaria la extension del fichero, esta puede obtenerse presionando el botón *Extensions* (*Localización de la animación y las extensiones*).

Animaciones complejas: A menos que su animación sea muy sencilla, y espere renderizarla en media hora o menos, siempre es una buena idea, desde el principio, renderizar la animación como frames separados en formato Targa más que en un formato AVI.

Esto permite una recuperación sencilla si algo falla en la computadora y tiene que comenzar

de nuevo a renderizar, ya que los frames que se han renderizado siempre estarán ahí.

Este método también le vendrá bien, en el caso de que un error ocurra en unos pocos frames, ya que puede hacer las correcciones oportunas y renderizar de nuevo solo los frames afectados.

Puede entonces crear el AVI con los frames afectados con el editor de secuencia de Blender o con otro programa que se lo posibilite.

Formatos de Salida

El fichero se salva en el formato que se haya seleccionado en el Panel *Format*. Aquí puede seleccionar varios formatos de imagen o animación (*Formatos de imagen y animación*).



Formatos de imagen y animación.

El tipo de imagen por defecto es *Targa*, pero desde que la imagen se guarda en un buffer y luego se guarda, es posible cambiar el tipo de formato de imagen después de que se haya renderizado usando este menú.

Por defecto Blender renderiza imágenes en color (*RGB*) pero también es posible renderizar en blanco y negro (*BW*) y en color con canal alpha (*RGBA*).

Blender *no* añade automáticamente la extensión a los ficheros, con lo que cualquier extensión *.tga* o *.png* se tiene que escribir en la ventana donde se guardan los ficheros.

Excepto el formato Jpeg, que es de compresión con pérdidas, todos los demás formatos son más o menos equivalentes. Generalmente es una mala idea utilizar el formato Jpeg ya que un formato con pérdidas. Es mejor usar el formato Targa y entonces convertirlo a Jpeg, ya que de esta manera mantenemos el original.

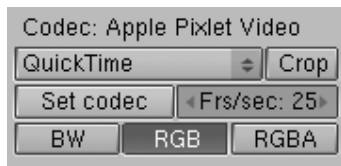
Para lo que concierne a los otros formatos: *TARGA raw* es el formato Targa sin compresión y usa un montón de espacio de disco. *PNG* es Portable Network Graphics, y es un estándar que se creó para intentar reemplazar al formato GIF y soporta imágenes con color verdadero. *HamX* es un formato de 8 bits con compresión RLE (Run Length Encoded bitmap); crea

ficheros extremadamente compactos que se pueden mostrar rapidamente. Para usarse unicamente con la opción "Play". *Iris* es el formato estandar de SGI e *Iris + Zbuffer* es el mismo formato añadiendole la información del Zbuffer (buffer de profundidad).

Finalmente *Ftype* usa un fichero "Ftype", para indicarle que este fichero sirve como ejemplo del tipo de formato gráfico en el cual Blender debe guardar las imagenes. Este método permite procesar el 'mapa de color' (colormap). El mapa de color se lee del fichero y se usa para convertir a 24 o 32 bit. Si la opción "RGBA" se especifica, el número de color estandar '0' se usará como color transparente. Blender lee y escribe mapas de colores en los siguientes formatos: (Amiga) IFF, Targa, (SGI) Iris y CDinteractive (CDi) RLE.

Por lo que concierne a las animaciones:

- *AVI Raw* - guarda un AVI sin compresión, sin pérdidas pero con un enorme tamaño.
- *AVI Jpeg* - guarda una AVI como una serie de imagenes Jpeg. Tiene perdidas, tamaño pequeño pero no tan pequeño como puede obtener con un algoritmo de compresión mejor. Además el formato AVI Jpeg no lo lee la mayoría de los reproductores.
- *AVI Codec* - guarda un AVI comprimido con un codec. Blender automaticamente le proporciona la lista de los codecs disponibles en su sistema y le permite seleccionar los distintos parámetros. También es posible cambiarlo o cambiar sus parametros, una vez seleccionado, a través del botón *Set Codec* que aparece en *Configuración de los codec del AVI*.
- *QuickTime* - saves a QuickTime animation.



Configuración de los codec del AVI.

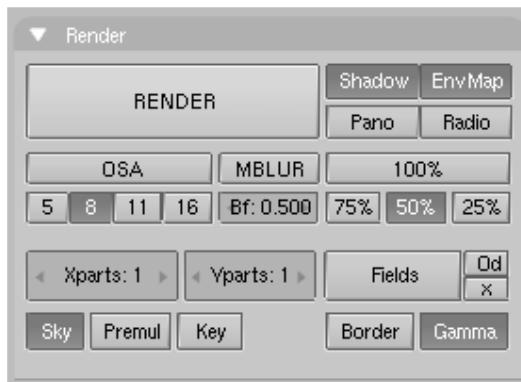
Para una animación en formato AVI también es posible seleccionar el ratio de frames (framerate) (*Configuración de los codec del AVI*) que, por defecto, es de 25 frames por segundo.

Opciones de Rénder

Renderizando por Partes

Es posible renderizar una imagen por partes, una después de la otra, en lugar de toda de una vez. Esto puede ser útil para escenas muy complejas, donde renderizar pequeñas secciones, una a continuación de otra, solo necesita el procesamiento de una pequeña parte de la escena, lo cual utiliza menos memoria.

Al asignar valores diferentes de 1 en *Xpart* y en *Ypart* en el Panel *Render Botones de Renderizado por partes* se obliga a Blender a dividir la imagen en una grilla de *Xpart* por *Ypart* sub-imágenes, que son renderizadas una a continuación de la otra y finalmente ensambladas como una sola imagen.



Botones de Renderizado por partes.

Nota:

Blender no puede manejar más de 64 partes.

Renders panorámicos

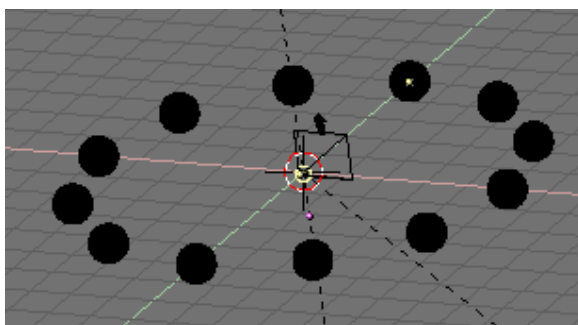
Para conseguir bonitos renders panorámicos, de hasta 360 grados; con vista al horizonte, Blender nos provee de un procedimiento automático.

Si *Xparts* es mayor que 1 y el botón *Pano* del panel *Render* está presionado (*Botón panorama*), entonces la imagen renderizada se crea *Xparts* veces *SizeX* de ancho y *SizeY* de altura, renderizando cada parte rotando esa parte la cámara tan lejos como sea necesario para conseguir la imagen con el menor número de uniones posible.



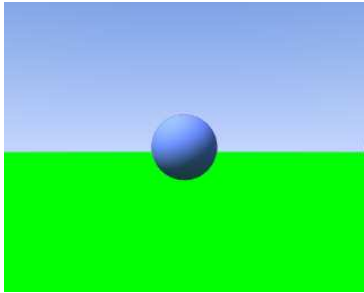
Botón panorama.

Configuración test panorámico muestra una configuración test con 12 esferas alrededor de una cámara. Dejando la cámara como está obendrá el render mostrado en *Renders no panorámicos*. Si pone *Xparts* en 3 y seleccionando *Panorama* el resultado será una imagen tres veces más ancha, mostrando un plano de cámara a mayores a la derecha y otro a la izquierda (*Renderizado panorámico*).

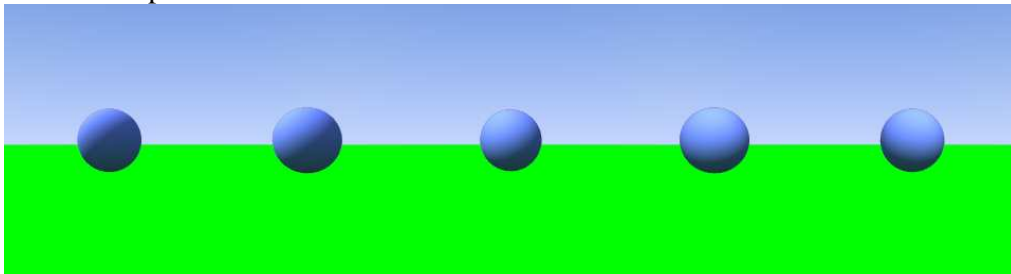


Configuración test panorámico.

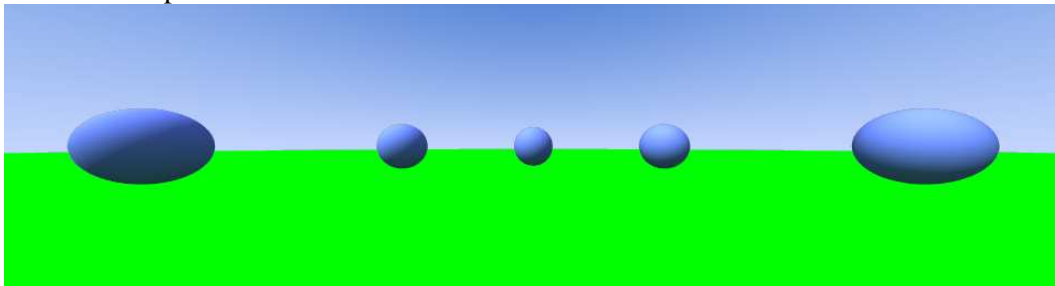
Para conseguir algo similar sin la opción Panorámica, el único camino es reducir la longitud focal de la cámara. Por ejemplo *Renderizado "ojo de pez"* muestra una vista comparable, a una obtenida con una longitud focal de 7.0, equivalente a un ángulo muy ancho, o lentes de "ojo de pez". La distorsión es evidente.



Renders no panorámicos.

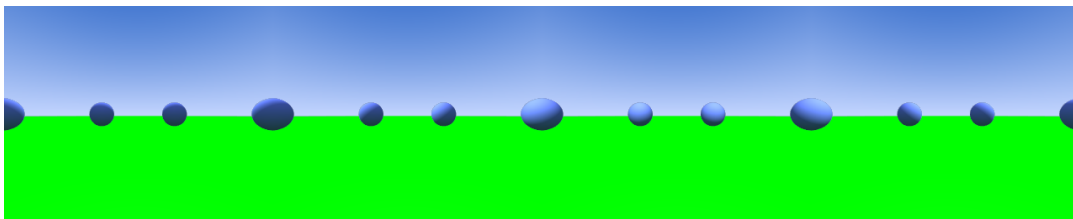


Renderizado panorámico.



Renderizado "ojo de pez".

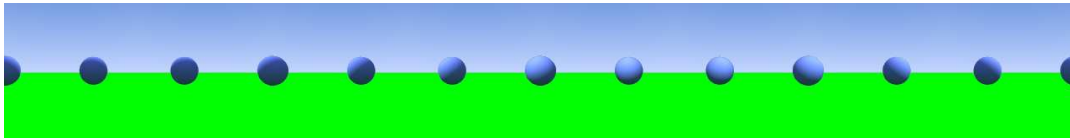
Para conseguir una vista completa de 360 grados; es necesario hacer algunos trucos. Es sabido que una longitud focal de 16.0 corresponde con un ángulo de visión de 90 grados. Entonces un render panorámico con 4 Xparts y una cámara con una lente de 16.0 supone una vista completa de 360 grados, como se ve en *Panorámica 360 grados con lentes 16.0*. Queda sumamente distorsionado, ya que una lente de 16.0 es muy ancha, y distorciona en los lados.



Panorámica 360 grados con lentes 16.0.

Para conseguir vistas sin distorsionar, la distancia focal debería ser de alrededor de 35.0. *Panorámica de 360 grados con lente de 38.5* muestra el resultado de una panorámica con 8

Xparts y una cámara con una lente de 38.5, que corresponde a un ángulo de visión de 45 grados.



Panorámica de 360 grados con lente de 38.5.

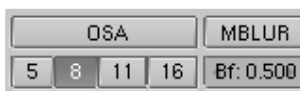
La imagen está mucho menos distorsionada, pero se debe prestar especial atención a la proporción. La imagen original era de 320x256 pixels. La panorámica en *Panorámica 360 grados con lentes 16.0* es de 4x320 de ancho. Para mantener a esta nueva panorámica con el mismo ancho, el valor *SizeX* de la imagen debe ponerse a 160 de forma que $8 \times 160 = 4 \times 320$. Pero el ancho del ángulo de visión de cámara está para la dimensión mayor, de forma que, si *SizeX* se mantiene en 256 la imagen se prolonga 45 grados verticalmente pero menos que horizontalmente, así el resultado final no es una panorámica de 360 grados. O bien *SizeX* es mayor que *SizeY* o tendrá que estar dispuesto a hacer unos cuantos test.

Desenfoque de Movimiento (Motion Blur)

Las animaciones de Blender son, por defecto, renderizadas como una secuencia de imágenes *perfectamente quietas*.

Esto no es realista, ya que los objetos que se mueven a gran velocidad, realmente parecen estar "en movimiento", es decir, desenfocados por su propio movimiento, tanto en un fotograma de una película como en una fotografía obtenida con una "cámara del mundo real".

Para obtener ese tipo de efecto de Desenfoque de Movimiento (Motion Blur), se le puede decir a Blender que haga el render del cuadro actual y de algunos cuadros más entre los cuadros reales, y combinarlos para obtener una imagen donde los detalles de los objetos que se mueven rápidamente aparecen "desenfocados".

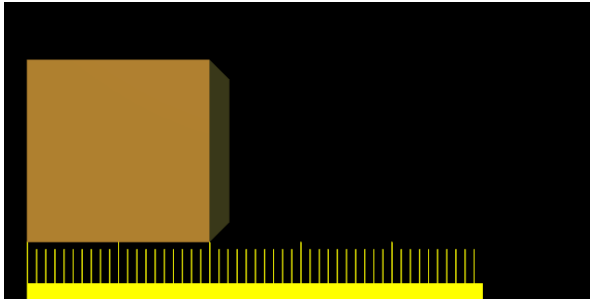


Botones de Motion Blur.

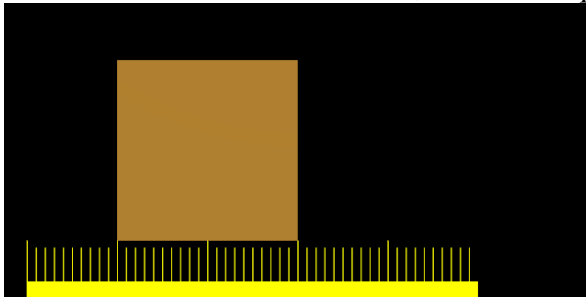
Para acceder a esta opción hay que seleccionar el botón *MBLUR* que se encuentra al lado del botón *OSA* en el Panel *Render (Botones de Motion Blur)*. Esto hace que Blender renderice tantos cuadros "intermedios" como indique el valor en oversampling (5, 8, 11 o 16) y los acumule, uno sobre otro, en un sólo cuadro. El botón-número *Bf*: o Factor de Desenfoque (Blur Factor) define la duración del tiempo de obturación, como se mostrará en el ejemplo de abajo. Activar los botones de *OSA* no es necesario, ya que el proceso de Desenfoque de Movimiento (Motion Blur) agrega algo de antialiasing de todas formas, pero para tener una imagen realmente suavizada, *OSA* puede ser también activado. Esto hace que cada imagen acumulada tenga antia-aliasing.

Para entender mejor este concepto, asumamos que tenemos un cubo moviéndose en uniformemente una (1) unidad de Blender hacia la derecha en cada cuadro. Esto es realmente rápido, especialmente porque el cubo tiene un lado de sólo 2 unidades de Blender.

Cuadro 1 del cubo en movimiento, sin Desenfoco de Movimiento (motion blur) muestra el render del primer cuadro, sin Desenfoco de Movimiento (Motion Blur), Cuadro 2 del cubo en movimiento, sin Desenfoco de Movimiento (motion blur) muestra el render del cuadro 2. La escala debajo del cubo ayuda a apreciar el movimiento de una (1) unidad de Blender.

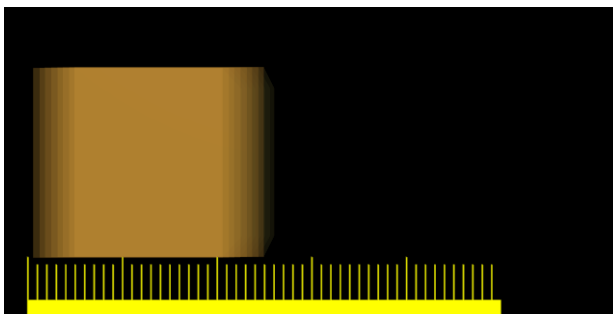


Cuadro 1 del cubo en movimiento, sin Desenfoco de Movimiento (motion blur).



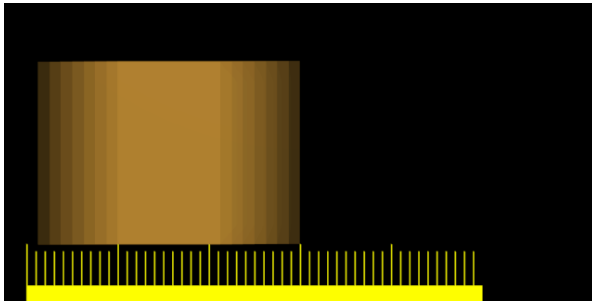
Cuadro 2 del cubo en movimiento, sin Desenfoco de Movimiento (motion blur).

Cuadro 1 del cubo en movimiento, con Desenfoco de Movimiento (motion blur), 8 muestras, $Bf=0,5$ por otra parte, muestra el render del cuadro 1 cuando Desenfoco de Movimiento (Motion Blur) está activado y 8 cuadros "intermedios" son computados con un periodo entre cuadros de 0,5, empezando desde el cuadro 1. Esto es bastante evidente ya que todo el "desenfoco" del cubo ocurre media unidad antes y media unidad después del cuerpo principal del cubo.

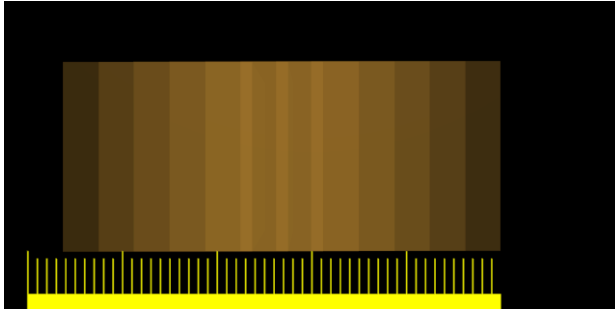


Cuadro 1 del cubo en movimiento, con Desenfoco de Movimiento (motion blur), 8 muestras, $Bf=0,5$.

Cuadro 1 del cubo en movimiento con Desenfoco de Movimiento (motion blur), 8 muestras, $Bf=1,0$ y Cuadro 1 del cubo en movimiento, con Desenfoco de Movimiento (motion blur), 8 muestras, $Bf=3,0$ muestran el efecto que produce aumentar los valores de Bf . Un valor más grande que 1 implica una obturación "lenta".



Cuadro 1 del cubo en movimiento con Desenfoque de Movimiento (motion blur), 8 muestras, Bf=1,0.



Cuadro 1 del cubo en movimiento, con Desenfoque de Movimiento (motion blur), 8 muestras, Bf=3,0.

Se pueden obtener mejores resultados que los mostrados anteriormente utilizando 11 o 16 muestras en lugar de 8, pero, por su puesto, ya que se necesitan tantos renders *separados* como muestras, un render con Desenfoque de Movimiento (Motion Blur) tarda bastante más tiempo que uno sin él.

Mejor Anti-Aliasing: Si Desenfoque de Movimiento (Motion Blur) está activo, aún si nada se mueve en la escena, Blender igualmente "agita" la cámara un poco entre un cuadro y el siguiente. Esto implica que, aun si OSA está desactivado, la imagen resultante tiene un buen Anti-Aliasing. Un Anti-Aliasing obtenido a través de Desenfoque de Movimiento (Motion Blur) es comparable a un Anti-Aliasing OSA del mismo nivel, pero generalmente menor.

Esto es interesante, ya que para escenas muy complejas donde un nivel de OSA de 16 no da resultados satisfactorios, se pueden obtener mejores resultados utilizando OSA y MBLUR. De esta forma se tienen tantas muestras por cuadro como cuadros "intermedios", dando de forma efectiva oversampling a niveles 25, 64, 121 y 256, si se eligen muestras de 5, 8, 11 y 16 respectivamente.

Profundidad de Campo

La Profundidad de Campo (Depth of Field) o DoF es un efecto interesante en Fotografía que realza mucho las imágenes generadas por computadora. También se conoce este efecto con el nombre de Borroneado Focal.

El fenómeno está vinculado al hecho de que una cámara fotográfica real puede enfocar un objeto a una distancia determinada, de manera que los objetos más cercanos y los muy lejanos

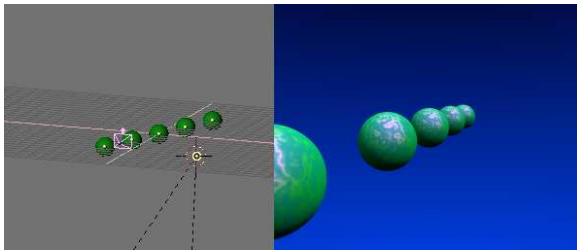
a la cámara quedarán fuera del plano focal, por lo que se los verá ligeramente borrosos en la fotografía resultante.

El nivel de borronado de los objetos más cercanos y los más lejanos varía muchísimo con las dimensiones de la longitud y la apertura focal de la lente, y si se la usa con destreza, puede dar efectos muy agradables.

El motor de render de Blender no ofrece un mecanismo automático para obtener el efecto DoF, pero existen dos caminos alternativos para lograrlo. Uno se basa solamente en los recursos internos de Blender y será descrito en estas líneas. El otro requiere un plugin de secuencia externo y será someramente descrito en el Capítulo Editor de Secuencias.

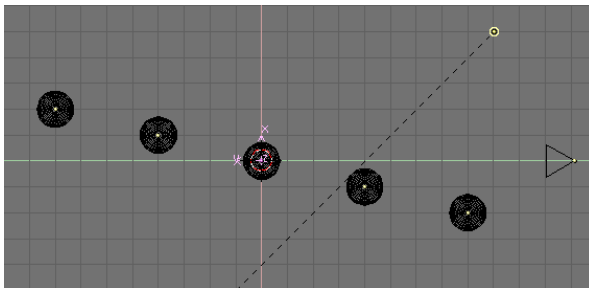
El truco para obtener el efecto DoF en Blender consiste en el uso hábil del efecto de Motion Blur o Borronado de Movimiento descrito anteriormente, haciendo que la cámara se mueva circularmente alrededor de los que sería la apertura de la lente en una "cámara en el mundo real", apuntando constantemente hacia un punto donde se desea tener el foco "perfecto".

Supongamos que tiene una escena con esferas alineadas como se ve en la parte izquierda de la *Escena de prueba de Profundidad de Campo*. Un render estándar de Blender dará como resultado la imagen en la parte derecha de la *Escena de prueba de Profundidad de Campo*, con todas las esferas perfectamente nítidas y en foco.



Escena de prueba de Profundidad de Campo.

El primer paso es ubicar un objeto Empty (**SPACE**>>**Add**>>**Empty**) donde se encontrará el foco. En nuestro caso en el centro de la esfera del medio (*Poniendo el Foco Empty*).



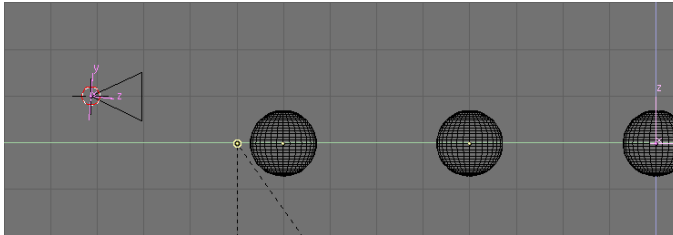
Poniendo el Foco Empty.

Luego, asumiendo que su cámara esté ya en la posición correcta, ponga el cursor sobre la cámara (seleccione la cámara, **SHIFT-S**>>**Curs->Sel**) y cree un círculo NURBS (**SPACE**>>**ADD**>>**Curve**>>**NURBS Circle**).

Salga de ModoEdición (**TAB**) y escale el círculo. Esto es muy arbitrario, y quizás desee re-escalarlo luego para lograr un resultado mejor. Básicamente, el tamaño del círculo está

vinculado a la diámetro físico de la apertura o diafragma de su "cámara real". Mientras mayor sea el círculo, más estrecha será la región de foco perfecto, y más borronados se verán tanto los objetos cercanos como los lejanos. Mientras más chico sea el círculo, menos notorio será el efecto DoF.

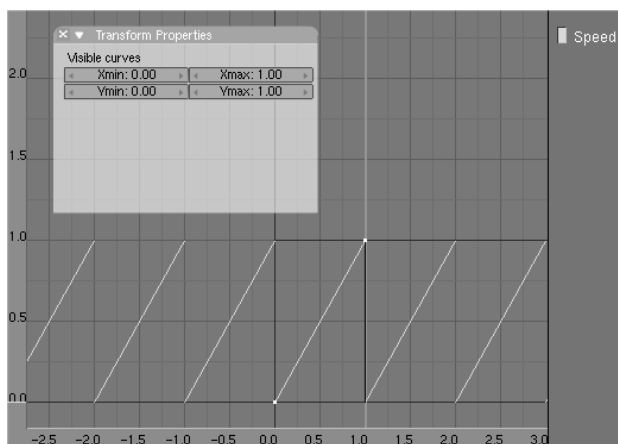
Ahora haga que el círculo siga al Empty con una restricción de seguimiento o bien con el viejo Seguimiento (Tracking) visto en *Círculo NURBS siguiendo al foco Empty*. Como la normal al plano que contiene al círculo está sobre el eje local Z, deberá configurar el seguimietno correctamente de modo que el eje local Z del círculo apunte al Empty y el círculo sea ortogonal con la línea que conecta su centro con el Empty.



Círculo NURBS siguiendo al foco Empty.

Seleccione la Cámara y luego el círculo y emparente la Cámara al círculo (CTRL+P). El círculo será el Recorrido de la cámara, por lo que puede utilizar una relación de parentesco normal y luego activar el botón *CurvePath* o emplear una relación de parentesco *Follow Path* (*Seguir Recorrido*).

Con el círculo aún seleccionado, abra una ventana IPO, seleccione el tipo de curva IPO Curve. La única IPO disponible es "Speed" (Velocidad). Con CTRL+LMB dos veces sobre un lugar al azar en la Ventana IPO agregue dos puntos cualquiera. Luego, usando NKEY, ponga numéricamente en estos dos puntos *Xmin* e *Ymin* a 0, *Xmax* e *Ymax* a 1. Para completar la edición de la IPO, haga acíclica usando la opción del menú *Curve>>Extend Mode>>Cyclic*. El resultado final debería verse como se muestra en *IPO Speed para el recorrido circular NURBS*.

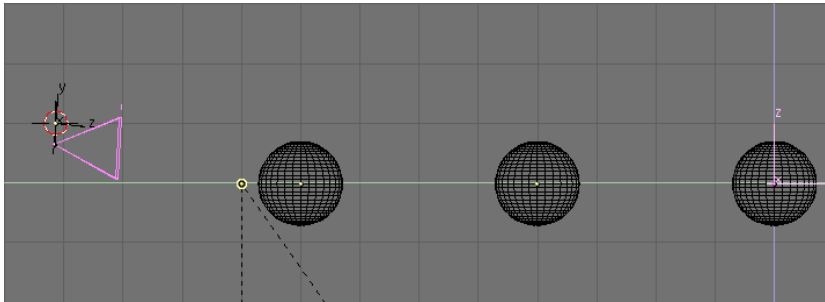


IPO Speed para el recorrido circular NURBS.

Con estos valores hemos hecho que efectivamente la cámara circule a lo largo del recorrido del círculo NURBS y alrededor de su posición original, en exactamente 1 cuadro. Esto hace

que la opción Motion Blur, vaya tomando distintas vistas de la escena ligeramente diferentes y cree al final un efecto similar al de Borroneado Focal.

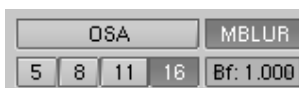
Falta realizar un ajuste más. Seleccione primero la cámara y luego el Empty Focal, y haga que la cámara siga al Empty de la manera que prefiera. La Cámara debería realizar un seguimiento del EMpty como se ve en *Cámara con seguimiento sobre el Empty Focal*.



Cámara con seguimiento sobre el Empty Focal.

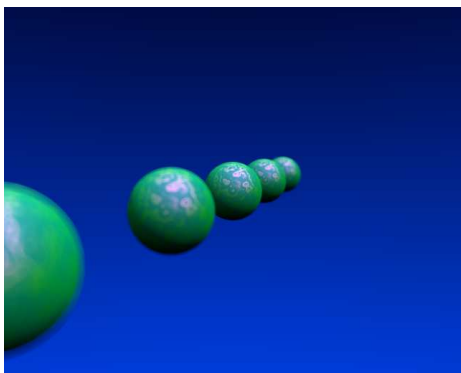
Si pulsa **ALT-A** no habrá ningún movimiento aparente ya que la cámara hace un giro en exactamente una vuelta completa al recorrido circular en un cuadro, por lo que parece estar quieta, sin embargo el motor de Motion BLur sí detectará estos movimientos.

El último retoque es luego ir a la ventana de botones de Renderizado (**F10**) y pulsar sobre el botón **MBLUR**. Muy probablemente no necesite activar la opción **OSA** ya que el Motion Blur realizará algún antialiasing implícitamente. Se recomienda enérgicamente que ponga el factor de Motion BLur a 1, ya que de esta manera recorrerá un frame completo para borronear, tomando el perímetro completo del círculo. Para resultados óptimos es necesario también poner el número de sobremuestras (oversamples) al nivel máximo (16) (*Valores de Motion blur*).



Valores de Motion blur.

Renderizando la escena (**F12**) obtendremos el resultado deseado. Esto puede tardar mucho más que un render sin DoF teniendo en cuenta que Blender renderizará 16 imágenes y luego las combinará en una sola. *Render final con Motion blur* muestra el resultado para comparación con la *Escena de prueba de Profundidad de Campo*. Se debe observar que el círculo ha sido escalado mucho menos para obtener esta imagen que lo que se ha mostrado en las capturas de pantalla de ejemplo. Estas últimas fueron hechas con un radio grande (igual a 0.5 unidades de Blender) para demostrar mejor la técnica. Por otro lado, *Render final con Motion blur* tiene un círculo cuyo radio es de 0.06 unidades de Blender.



Render final con Motion blur.

Esta técnica es interesante y con ella es bastante sencillo obtener pequeños grados de Profundidad de Campo. Para grandes Nieblas Focales está limitada por el hecho de que no es posible tener más de 16 sobremuestreos.

Bordes de Caricatura

Los nuevos sombreadores de materiales de Blender de la versión 2.28, incluyen sombreadores especulares y difusos de tipo caricatura (toon).

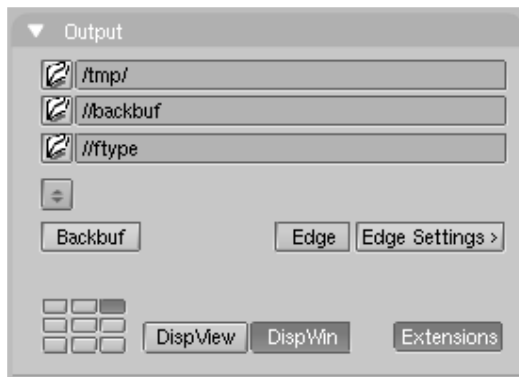
Utilizando estos sombreadores, le puede dar a sus renderizados un aspecto similar al de album de historieta o Manga, afectando los tonos de los colores como quizás pueda ver en *Una escena con materiales de Caricatura*.



Una escena con materiales de Caricatura.

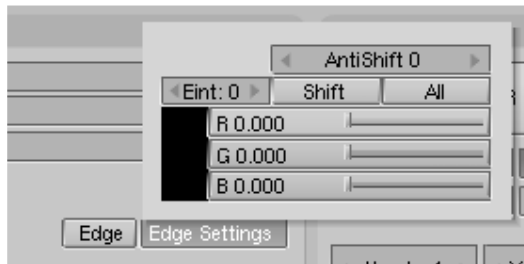
El efecto no es perfecto ya que las historietas y manga reales, también tienen normalmente contornos en tinta china. Blender puede añadir esta característica como una operación de post-procesado.

Para acceder a esta opción seleccione el botón *Edge* en el Panel *Output* de los Botones de Renderizado (F10) (*Botones de borde de caricatura*). Esto hace a Blender buscar los bordes en su renderizado y le añade una línea de contorno a los mismos.



Botones de borde de caricatura.

Antes de repetir el renderizado es necesario establecer algunos parametros. El botón *Edge Settings* abre una ventana para establecer estos valores (*Valores del borde de caricatura*).



Valores del borde de caricatura.

En esta ventana es posible indicar el color del contorno, que es negro por omisión, y su intensidad, *Eint* que es un número entero que va de 0 (más débil) a 255 (más intenso). Los otros botones son útiles si se usa el Renderizador Unificado (ver sección siguiente).

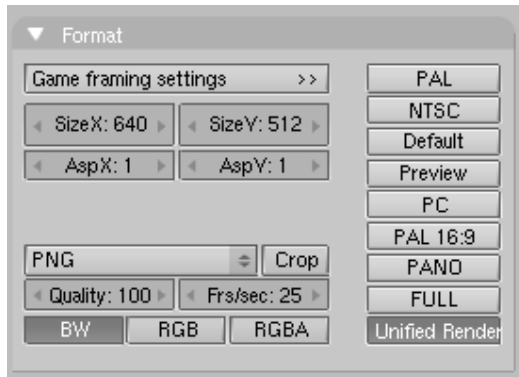
Escena re-renderizada con bordes de caricatura activados muestra la misma imagen que la *Una escena con materiales de Caricatura* pero con los contornos de caricatura habilitados en color negro y con máxima intensidad ($Eint=255$).



Escena re-renderizada con bordes de caricatura activados.

El Renderizador Unificado (Unified Renderer)

Una característica no muy conocida de Blender es el botón Renderizador Unificado (Unified Renderer) que se encuentra en la esquina derecha inferior del Panel *Format* de los Botones de Renderizado (*El botón Renderizador Unificado*).



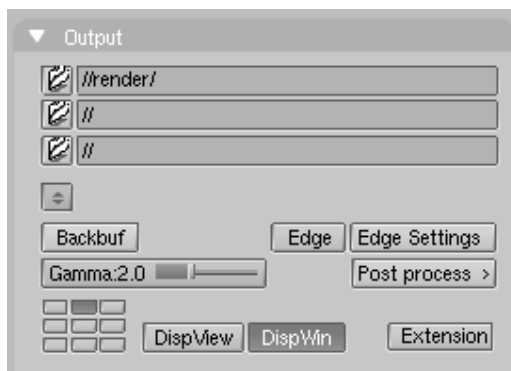
El botón Renderizador Unificado.

El motor de render de Blender *por omisión* está muy optimizado hacia la mejora en velocidad. Esto se ha conseguido subdividiendo el proceso de renderizado en varias "pasadas". Primero se procesan los materiales "normales", luego los materiales con transparencia (alfa); y, finalmente, se añaden los Halos y Flares.

Esto es rápido, pero puede llevar a resultados no óptimos, especialmente con Halos. El Renderizador Unificado, por el contrario, renderiza la imagen en una pasada única. Esto es más lento, pero da mejores resultados, particularmente en el caso de Halos.

Aún más, como los materiales transparentes se renderizan con los materiales convencionales, se le pueden aplicar también Bordes de Caricatura, pulsando el botón *All* en el diálogo de Edge Setting.

Si se selecciona el Renderizador Unificado, aparecerá un nuevo grupo de botones en el Panel *Output* (*Botones adicionales del Renderizador Unificado*).

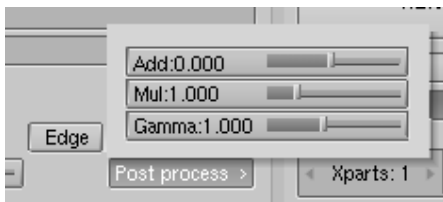


Botones adicionales del Renderizador Unificado.

El deslizador *Gamma* está relacionado con el procedimiento de OSA. Los sobremuestreos (oversampling) de píxeles se mezclan para generar el pixel renderizado final. El renderizador

convencional tiene un $\text{Gamma}=1$, pero en el Renderizador Unificado se puede variar este valor.

Los botones de *Post process* (*Post-procesamiento*) hacen aparecer una caja de diálogo (*El submenú de postprocesamiento del Renderizador Unificado*). Desde allí puede controlar tres tipos de postprocesamiento: el deslizador *Add* define una cantidad constante que será sumada al valor de color RGB de cada pixel. Los valores positivos hacen la imagen más brillante uniformemente, los valores negativos la hacen más oscura.



El submenú de postprocesamiento del Renderizador Unificado.

El deslizador *Mul* define un valor por el que se multiplican todos los valores RGB de todos los píxeles. Valores mayores que 1 hacen la imagen más brillante mientras que valores menores que 1 la hacen más oscura.

El deslizador *Gamma* realiza la corrección de contraste gamma estándar de cualquier programa de dibujo.

YafRay

YafRay (**Y**et **A**nother **F**ree **R**aytracer) es un raytracer libre, concebido bajo licencia LGPL, que usa un lenguaje de descripción de escenas en XML. Recientemente ha sido integrado al programa de modelado 3D Blender, y es usado frecuentemente para renderizar las escenas creadas con él. Su principales desarrolladores son Alejandro Conty Estévez y Alfredo de Greef.

Principales Características

- **Iluminación global completa:** YafRay puede iluminar escenas mediante este sistema, empleando aproximaciones Montecarlo y Quasimontecarlo.
- **Iluminación Skydome:** Luz originaria de un cielo emisor que causa sombras suaves en la escena.
- **Iluminación HDRI:** Este tipo de iluminación se basa en la información contenida en una imagen HDR. Puede usarse con los dos tipos de iluminación descritos anteriormente.
- **Cáusticas:** YafRay permite simular el comportamiento de los materiales reflectantes y transmisores de luz, como el cristal.

- **DOF Real:** Gracias a esto es posible reproducir en la escena el efecto de enfoque que tendría una lente real, produciendo desenfoque en los objetos mas lejanos y cercanos del punto de enfoque.
- **Reflexiones Borrosas:** Ciertos objetos producen una distorsión en la reflexión debida a una microscopia superficie rugosa, y este efecto puede ser simulado por el programa.
- **Otras características**
 - Estructura modular.
 - Integración en programas de modelación 3d mediante plug-in.
 - Multiplataforma.
 - Motor de render independiente.
 - Render distribuido y multihilo.

YafRay como motor de render integrado en Blender

Parte 1

La integración de YafRay es una de las mejores funcionalidades añadidas a Blender. La actual versión de Blender 2.35 tiene una sencilla y efectiva integración de Blender y YafRay, además de ser bastante estable. Desafortunadamente, el uso de YafRay está limitado para aquellos que ya manejan con soltura la interfaz de Blender. La primera parte de este tutorial corresponde a los pasos básicos necesarios para renderizar con YafRay, y las partes siguientes sirven como guía de uso para el resto de características disponibles en YafRay.

Interfaz

Blender tiene, por sí mismo, dos motores de render internos: su propio *raytracer* y el antiguo motor de render por *scan-line*. YafRay es, al contrario que los anteriores, un *raytracer* independiente. A través del interfaz de Blender se puede acceder a sus funcionalidades, exportando los parámetros de la escena a un formato entendible por YafRay. Esto se puede realizar de dos maneras. Para la primera de ellas Blender posee el soporte interno que permite usarlo como un plug-in, virtualmente como si YafRay fuese uno de los motores de render internos de Blender. La otra opción es que Blender exporte los datos de la escena a un fichero en el formato XML usado por YafRay, llamado *YBtest.xml*, y después YafRay lo renderice como programa independiente (de manera externa a Blender). Ambos métodos necesitan que YafRay se encuentre instalado en el sistema, de hecho este tutorial asume que tienes instalados tanto Blender 2.34 como YafRay 0.0.7.

La integración de YafRay en Blender se divide, en términos generales, en dos partes. Una relacionada con el interfaz para configurar la iluminación (luces / *lamps*). La otra se refiere al núcleo de las características de render de YafRay.

Paso-1

Hay que tener en cuenta que este paso podría no ser necesario en Windows. Antes de proceder, necesitamos configurar primero las preferencias de YafRay. Arrastra hacia abajo la barra de menu superior para mostrar las preferencias de usuario de Blender. Selecciona el botón *File Paths* para abrir el menú de los *Path* (rutas). En la parte superior izquierda hay una caja de texto, *YFExport*. Introduce el directorio donde quieres que se grabe el fichero XML

que contiene la exportación de la escena de Blender en formato YafRay. Esto se usa sólo cuando se renderiza utilizando la segunda opción. También es necesario si deseas salvar el fichero exportado para editarlo manualmente.

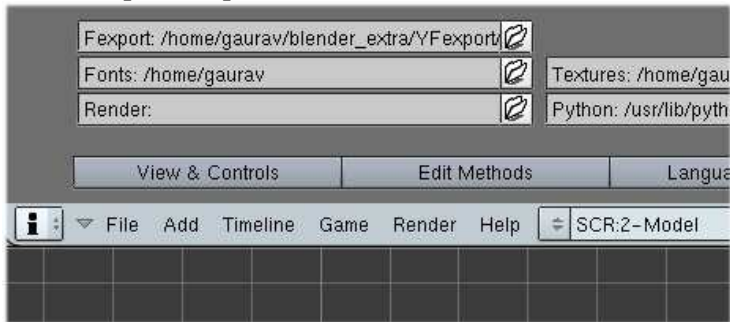


Figura 1. Selección del *Path* en el menú *User Preferences*.

Step-2

Como queda dicho, Blender permite elegir entre su propio *raytracer* interno y YafRay, por lo que para usarlo primero tendremos que indicárselo a Blender. Para ello, pulsa F10 para mostrar la ventana *Rendering Options*, y ve al panel *Render* (Figura 2). En dicho panel, selecciona YafRay en la lista desplegable (Figura 3), por defecto está seleccionado *Blender internal*.

⚠ Nota: No te confundas con el botón Ray. Activarlo no sirve para nada cuando se usa YafRay. No tiene ningún efecto sobre el resultado si YafRay está seleccionado como *raytracer* en la lista desplegable.



Figura 2. Las opciones por defecto en el panel *Render*.

Fíjate que al seleccionar YafRay aparecen dos nuevas solapas detrás de la de *Render* (Figura 3). Ambas solapas, *YafRay* y *YafRay GI*, se utilizan para controlar los parámetros de render y demás características de YafRay.



Figura 3. Selección de YafRay como raytracer y las dos solapas extra.

Ahora selecciona la solapa *YafRay* de entre las dos nuevas, aquí veremos algunas funciones nuevas. Por ahora sólo nos interesa el botón llamado *XML* (activado por defecto) (Figura 4). Púlsalo para tenerlo desactivado por ahora. Explicaremos por qué, como se ha dicho anteriormente, Blender tien dos opciones para renderizar con YafRay, o bien vía plug-in o bien llamando a YafRay de manera externa, como un programa de línea de comandos.

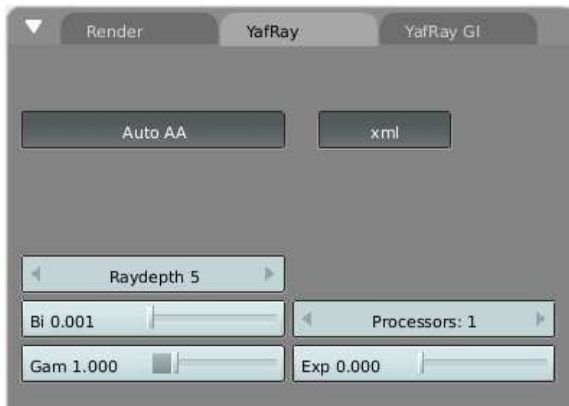


Figura 4. Solapa de YafRay por defecto con el fichero de exportación activado. (botón XML).

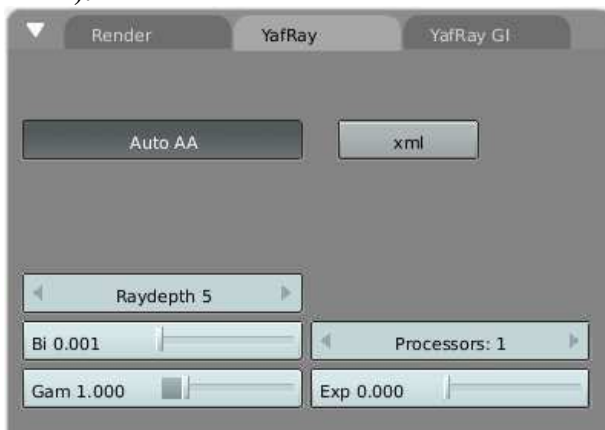


Figura 5. Exportación a fichero de YafRay desactivada.

Al escoger la primera opción de render, vía plug-in (botón XML desactivado), podremos ver el progreso del render en la ventana *Render* (Figura 6), prácticamente igual que con el propio motor de Blender, y esta es una de las razones por las que YafRay requiere y utiliza algo más de memoria que antes. Usando la segunda opción, la cual es activada por defecto (botón XML pulsado), primero ha de exportar la escena al directorio que definimos con YFExport en las preferencias, como vimos en (la sección Paso-1). Después se llama a YafRay como programa de línea de comandos, y empieza el proceso del fichero XML exportado. Sólo al final, cuando el *renderizado* ha acabado, se muestra el resultado en la ventana de *Render*. Por lo tanto, no se produce ninguna interacción, salvo por la información en modo texto mostrada en la ventana de terminal de Blender.

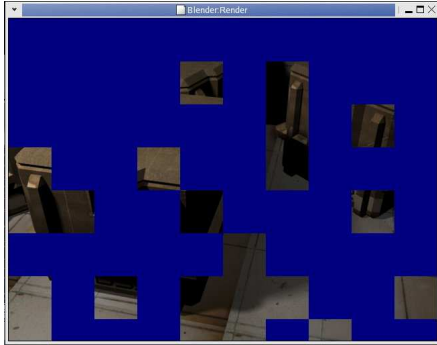


Figura 6. El render en progreso con XML desactivado.

Paso-3

Tras seguir los dos primeros pasos ya tienes todo lo que se necesita para realizar un render con YafRay. En este punto tienes tu escena con las luces que has elegido, lo mejor para seguir desde aquí será pulsar F12 para empezar el render. Y aquí es donde los problemas surgen para casi todo el mundo. Me explico... Dependiendo de la configuración de luces de tu escena, puedes ver una pantalla en negro o ver unos contornos débiles de los objetos de la escena. En otro caso si tus luces tienen una intensidad (valor) de luz muy alta verás la escena correctamente. Otra opción, en un caso extremo, es que YafRay y Blender den un error fatal.

En el caso de obtengas un render negro hay grandes posibilidades de que puedas renderizar con YafRay; todo lo que necesitas es ajustar los parámetros de las luces y tendrás tu propio render con YafRay en minutos.

Puntos para evaluar problemas

- El primer problema de los renders con YafRay es que prácticamente todas las veces la intensidad de luz es insuficiente para la escena. En realidad, esto no es un problema de YafRay sino que se debe a motivos de implementación de la integración con Blender. Por lo que, en este caso, necesitas incrementar la intensidad de luz, o su valor, de cada una de las fuentes de luz de la escena (más información en la sección llamada Parte 2).
- Si todo falla, necesitarás comprobar los valores de la distancia de las luces, ya que la atenuación de la luz es bastante más pronunciada en los gráficos por computador (para reducir cálculos). Los objetos en la escena podrían aparecer negros porque su distancia de *ray casting* (algo así como la distancia máxima de colisión del rayo con el objeto. *_N. del T.*), podría no alcanzar a ninguno de los objetos de la escena. Puedes solucionar esto ajustando el valor de la distancia en el panel *Lamps*.

- En caso de error fatal, por favor busca posibles soluciones en los foros de elysium (Normalmente los errores de este tipo (*crash*) están más relacionados con YafRay que con Blender, así que lo mejor es preguntar en el foco del problema: los foros de YafRay. *N. del T.*).

Parte 2

En lugar de enseñarte a modificar las configuraciones por aquí y por allá, creo que podría ser mejor explicar los tipos de luz de YafRay uno a uno. Esto te permitirá tomar decisiones con conocimiento de causa. Let there be light... And there was Yafray everywhere.

Hay cinco tipos de luz utilizadas en Blender y sólo cuatro de ellas están directamente soportadas por YafRay. Al activar YafRay como *raytracer* en el panel de render, inicialmente verás seis tipos de luz (en la ventanas de opciones de luz, F5), de las cuales YafRay utiliza *Lamp*, *Area*, *Spot*, y *Sun*. Volveré más tarde sobre las otras dos luces de la lista, *Hemi* y *Photons*



Figura 7. La escena.

Descripción de la escena: La escena en cuestion tiene dos fuentes de luz. Una es una luz de apoyo (*Sun*, sin valor de sombra, *shadow*, y con valor .200). Esta luz proporciona luz no direccional que es utilizada para potenciar la atmósfera de la escena. La otra fuente de luz es una plantilla que será cada una de los tipos de luz a explicar. Las posición de dicha fuente es estática y tiene activada el *Ray Shadow* (*shadow casting*) para todos los tipos de luz que utilizaremos (ver Figura 8).



Figura 8. La escena como se ve en el Viewport.

⚠ Nota: Los usos sugeridos para los tipos de luz definidos aquí son generales, como ilustración para su uso. No han de ser tomados como la verdad absoluta para la iluminación de tus escenas. La iluminación varía dependiendo de la visión del artista sobre la misma. Para obtener mejores resultados, experimenta con las luces de tu escena hasta encontrar el mejor resultado para tí.

💡 Importante = Los tiempos de render de las distintas imágenes se han redondeado, ya que son relativos y sólo los utilizamos para dar una idea de la variación de los tiempos de render para la misma escena, con diferentes tipos de luz y configuración.

Lamp

Esta es la fuente de luz más simple de entre todas las disponibles. En algunos casos, es como una luz solar, en el sentido de que es omnidireccional (es decir, emite luz en todas las direcciones desde el punto origen) y esférica por naturaleza. Aunque lanza rayos de luz en todas las direcciones, dichos rayos pueden perderse con lo que van a desperdiciar parte del proceso de cálculo. Aunque los *raytracers* utilizan una serie de trucos para eliminar cálculos indeseados, sigue siendo un poco más lenta que otras fuentes de luz direccionales, en el caso de que la mayor parte de la luz recaiga dentro de la parte visible de la escena.

El uso más frecuente de la fuente de luz *Lamp* es la iluminación interior, como en habitaciones y pasillos, pero no está limitada a este uso ya que también se utiliza como luz de ayuda o de relleno.

Valores: Cuando una simple *Lamp* se usa en una escena de YafRay (a partir de ahora y a lo largo de todo el tutorial al referirme a YafRay quiero decir Blender + YafRay), generalmente no proporcionará una iluminación suficiente para la escena, a no ser que la pongamos a un valor muy alto. En la escena de ejemplo, el valor de *Lamp* está puesto a 10 (máximo) pero aún así se aprecia la necesidad de iluminación adicional (pero lo dejaremos así con el propósito de entenderla). Ver Figura 9.



Figura 9. Renderizado con Lamp y valores por defecto.

El *Lamp* arroja sombras más afiladas (nítidas) con el radio por defecto (cero). El parámetro *radius* en *Lamp* se utiliza para aumentar el tamaño del area que emite la luz. Si el tamaño del *Lamp* es mayor que los objetos, entonces la luz emitida por algunas partes de la lámpara podrían alcanzar directamente a partes de la escena donde otra parte de la lámpara produce una sombra. La intersección desemboca en que la sombra se diluye, siendo así suavizada (a esto se le conoce también como también sombras parciales, *partial shadows*). Puedes observar este fenómeno en el mundo real con la luz solar.

Enlace: Para más información sobre sombras parciales visita Ditto head's Light tutorial en blenderman.org (Inglés).

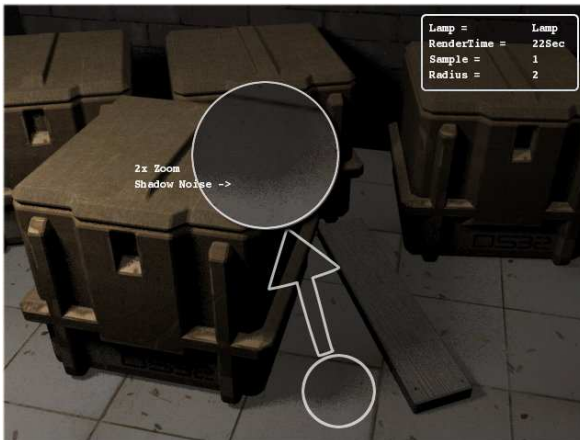


Figura 10. El ruido aparece cuando los valores de sampleado son bajos.

La escena de arriba(Figura 10) ha sido renderizada con *radius* 2 y un valor de sampleado (*sampling value*) 1. Como se puede ver, la escena tiene sombras parcialmente granuladas. El botón *samples* se utiliza simplemente para reducir estos granos en las sombras; los botones de sampleado controlas el número de muestras (*samples*) utilizados en los cálculos del sombreado. Incrementando el sampleado obtenemos sombras más suaves. Este botón está sólo disponible en luces del tipo *Lamp* y *Area*, funcionando de la misma manera para los dos.



Figura 11. Al incrementar el sampleado a 5 hemos eliminado el ruido.

Area

La luz de área (*Area Light*) es una fuente direccional. La forma del *Area Light* puede variar de cuadrada a rectangular utilizando la lista desplegable en el panel de *Lamp*. Debajo de la selección de forma encontramos también una opción para incrementar el tamaño del *Area Light*



Figura 12. Render básico con un área light de valor 4 y con configuración por defecto.

Valores: Para una escena con luz de área en YafRay, necesitarás unos valores de intensidad de luz más pequeños. La escena de ejemplo está iluminada por una luz con valor 4, y es más brillante que una renderizada con una *Lamp* con valor 10. Esto es debido a que la luz de área lanza toda la luz en una dirección desde un plano, mientras que una lámpara (*Lamp*, foco) lanza la luz en todas y cada una de las direcciones, distribuyendo energía donde no se necesita.



Figura 13. Aunque la escena parece similar a la Figura 11 de la sección *Lamp*, tiene más luz y mejores sombras.

⚠ Nota: Un punto importante a recordar es que en la versión oficial de Blender 2.34, usando dos o más luces de área, se obtiene un error en el render. Es un bug reconocido y ya ha sido arreglado en algunas versiones de desarrollo

⚠ Nota del Traductor: En el momento de realizar esta traducción Blender va por la versión 2.36 y este problema está ya solucionado.

Spot

La *Spot* (luz puntual) es también una fuente de luz direccional y el propio nombre sugiere que es un punto de luz circular, o dicho de manera más correcta, un cono de luz. El tamaño del cono puede ser controlado por `_SpotSi_` o el ángulo del rayo de la luz puntual. Con el ángulo más alto y el valor más cercano se comporta como una luz de área, pero con una gran diferencia. No puede producir sombras parciales (*partial shadows*). La luz *Spot* sólo tiene un parámetro para controlar el `_SpotBi_` o suavidad del límite del spot (del cono). Sólo se puede ver si la luz puntual está dentro de la vista de la escena, en otro caso no notarás diferencia entre *Spot* y otras luces.



Figura 14. Un render con luz *Spot* y valores por defecto.



Figura 15. Luz puntual (Spot) con atenuación de luz esférica.

Date cuenta de el incremento de brillo en la zona cercana a la fuente de luz en la [FiguraQuince][Figura 15]] al utilizar la atenuación de luz esférica (*quad* no está soportado por YafRay). Incrementar el valor de distancia en la atenuación esférica también incrementa el radio. Por ejemplo la Figura 15 tiene el valor por defecto de distancia, 20, mientras que la Figura 16 tiene el valor de distancia a 40. El efecto se parece al de las típicas lámparas caseras, en las cuales la atenuación es más acentuada.

⚠Nota: La atenuación de la luz en YafRay tiene sólo un decremento cuadrático, mientras que en Blender puede ser lineal, cúbica y una mezcla de ambas. Esto explica la relativamente acentuada caída de intensidad en los renders de YafRay



Figura 16. Luz Spot con atenuación esférica activada.

Sun

También es una fuente de luz direccional. Intenta emular la luz solar emitiendo luz de la misma intensidad en toda la escena sin atenuación (Figura 17). Esto produce que el entorno se ilumine con una atmósfera constante. Su valor debe mantenerse bajo. En la escena de ejemplo se usa una intensidad de luz de .100. Obviamente, esta luz es apropiada para escenas de exteriores, especialmente para emular un sol.


En escenas exteriores, si no quieres sombras duras y acentuadas, puedes desactivar la proyección de sombras (*shadow casting*) del *Sun* y utilizar otras fuentes de luz para la generación de sombras. Pero asegúrate de que tienen una intensidad de luz suficiente como para proyectar una sombra.



Figura 17. La *Sun Light* con valor 22. No es de extrañar que esta fuente de luz se llame Sol.

¿Más fuentes de luz?

- Hemi - YafRay no tiene soporte para la implementación de Hemi Light que posee Blender internamente; para ello simplemente usa la implementación de YafRay para *Sun Light*, por lo que puedes usar directamente una *Sun Light*. (N. del T.: Blender exporta las *hemilights* a YafRay como una *Sun Light* sin sombras).
- Photon - Finalmente, la fuente de luz *Photon Light* (luz de fotones) o el botón *Photons* no debe de ser confundido con un tipo de luz. No proyecta ninguna sombra o luz, solamente se utiliza para los cálculos de caústicas y ha de ser colocado donde quieras un cálculo de caústicas. Esta lámpara lanza fotones en un área. Los fotones son utilizados en YafRay como un rayo especializado solamente en calcular las caústicas pasando a través de los objetos como cristal o espejos, los cuales tienen la propiedad de distorsionar la luz que pasa a través de ellos; esto es conocido como reflexión interna total. Básicamente la colocación de una lámpara de fotones se realiza para permitir al usuario optimizar el renderizado. El cálculo de caústicas es uno de los métodos de *raytracing* más intensivos en términos de computación.

 **Nota:** Este tipo de lámpara y los fotones no han de ser confundidos con la opción *photon* en el método GI (*Global Illumination*, iluminación global). Dicho método será explicado en la Parte 3.

Parte 3

GI y otras características

YafRay soporta *Iluminación Global* (*Global Illumination*, GI). Vamos a volver al panel de Render para aprender más acerca de la implementación de GI llevada a cabo por YafRay. Echando un primer vistazo al panel de GI de YafRay puedes pensar, "¡Oh! ¡esto es muy

fácil!". Y tienes razón, las configuraciones están simplificadas para su uso. Las opciones disponibles son *Method* (Método) y *Quality* (Calidad). Veamoslas una a una. El parámetro *Quality* se utiliza siempre de la misma manera y lo explicaremos más adelante.



Figura 18. El panel de *GI* para YafRay.

Los métodos disponibles para *GI*

Skydome

Aunque que *Skydome* (Literalmente, *bóveda solar. N. del T.*) es un método simple para obtener iluminación de un cielo, o más correctamente, de una atmósfera, no proporciona iluminación global completa en sentido estricto, ya que no tiene en cuenta la luz indirecta proveniente de los rebotes de los rayos en la superficie de los objetos. Más bien afecta a la iluminación de la escena con el color de la atmósfera. También se controla a través de él la difusión de la luz en la atmósfera. Por ejemplo en la escena por defecto el color de la atmósfera está fijado a un tono rojizo (elegido precisamente para mostrar de forma clara su influencia). Observa en la Figura 19 cómo afecta la difusión a la escena.



Figura 19. La parte izquierda de la imagen tiene una potencia de 2 en el parámetro de difusión, mientras que la de la derecha tiene un valor de 4.

El color del fondo o de atmósfera puede cambiarse desde el panel *World* (Mundo) (F8). Para que el *Skydome* funcione completamente tienes que activar el *World*. *Skydome* no ofrece otras posibilidades. Es el método más rápido de iluminación global y, aunque no resulta muy correcto físicamente, sus resultados no son tan malos como para ignorar las ventajas que ofrece en cuanto a rapidez.

Full

En el otro extremo, la opción *Full* (Total) considera todos los rebotes de los rayos de luz reflejados o refractados en las superficies vecinas. Este método se asemeja a la simulación del comportamiento de la luz en el mundo real. Por esto la iluminación global produce imágenes con un grado de foto realismo mayor que cualquier otro método usado en gráficos por computador. También por ello requiere muchos más cálculos de computación y es más lento.



Figura 20. Al usar GI completa se incrementan tanto el tiempo como el realismo.

Fíjate en la diferencia de realismo entre las dos partes de la Figura 20. Los rebotes de la luz proporcionan una difusión suficiente en la segunda parte de la imagen, donde las las sombras se iluminan respecto a la distancia de las superficies en contacto. Por el contrario, en la primera parte la profundidad de las sombras arrojadas es prácticamente constante.

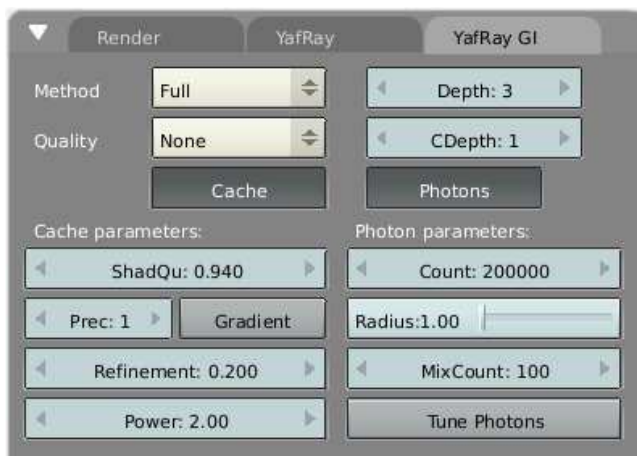


Figure 21. El panel GI de YafRay. Las opciones para el método *Full* son visibles

Parámetros para el método *Full* de GI

Depth:

(Profundidad). Está referido al número total de rebotes de la luz para un rayo refractado o difuso. En general un *Depth* 3 funciona bien para las escenas normales. Incrementar la profundidad repercute en una gran cantidad de cálculos por rayo.

Cdepth:

Este parámetro no es otra cosa que la profundidad de rebotes para caústicas, o para materiales transparentes como cristal, líquidos, piedras preciosas, etc. Para el cálculo de caústicas se requiere muchos cálculos de computación. Por esto la iluminación global en los gráficos por computador mantiene separado las luces normales de los fotones, a pesar de que en el mundo real los fotones son las partículas de luz. Para unos mejores resultados con caústicas, necesitamos por lo menos unos valores de 3 a 5 en *CDepth*, o más altos para mayor detalle.

Photon:

Este botón *Photon* no ha de ser confundido con la luz de fotones, *photon lamp*. Este botón se usa únicamente como apoyo en la iluminación global para YafRay, y no tiene relación con las luces de fotones ni con los fotones para caústicas

Count:

Se refiere al número total de fotones de la escena que estarán disponibles como ayuda para la iluminación global. El número de fotones variará según la escena, pero valores más altos proporcionan en general resultados más suavizados. De nuevo has de encontrar tú límite de tiempo/calidad adecuado.

Radius:

Radius (Radio) se refiere a la distancia dentro de la cual los fotones calculados tienen un mayor valor de precisión para efectuar la iluminación global, fuera de ella los fotones no influyen prácticamente. Entonces la idea es optimizar la longitud del radio para el tamaño del área que requiere el mejor cálculo de iluminación. Deja el radio a la mitad del tamaño de ese área.

MixCount:

Esto permite elegir el número de fotones que deberían ser tomados en cuenta dentro del radio (*Radius*). Estos dos parámetros, *Radius* y *MixCount* componen el mapa de fotones (*_Photon map*). El mapa de fotones no es más que el área optimizada para los fotones utilizados en los cálculos de iluminación global. YafRay procesa los fotones almacenados y el muestreo sobre el mapa de fotones, e ignora los fotones que caen fuera del mapa.

Tune Photons:

Poner a punto los fotones. Esto permite reutilizar los fotones guardados y las posiciones de la cache utilizados con éxito, los cuales ayudaron a los cálculos de iluminación global en la fase previa al render, y solamente recalcular el resto de los fotones para un resultado más rápido. No necesariamente es lo más rápido para todos los renders, pero generalmente proporciona una buena mejora de velocidad. Debido a esto, es bueno usarlo durante la fase de los renders de prueba para saber si estás cerca de los resultados deseados. Puedes desactivarlo cuando estés satisfecho con los resultados obtenidos en los render de prueba y quieres crear la versión final de la imagen haciendo los cálculos completos.

Quality:

Calidad. La lista desplegable *Quality* te permite elegir el nivel de calidad de la iluminación global tanto para el método *Skydome* como para el método *Full GI*. Los parámetros de calidad te permiten controlar el número de muestras (*samples*) utilizados en la iluminación global. Aunque su uso es automático por defecto, se puede probar a utilizar un valor máximo para MultiPasses y muestras por pasada (*samples per pass*), dependiendo del nivel de calidad. Si lo prefieres, puedes desactivar AutomaticAA y dar a YafRay las instrucciones pertinentes.

Otras funcionalidades de YafRay

Anti-Aliasing:

El panel de YafRay en la ventana de opciones de "*Rendering*" proporciona un listado de las posibilidades para el control manual sobre el anti-aliasing. Pulsa el botón AutoAA para mostrar las opciones manuales.

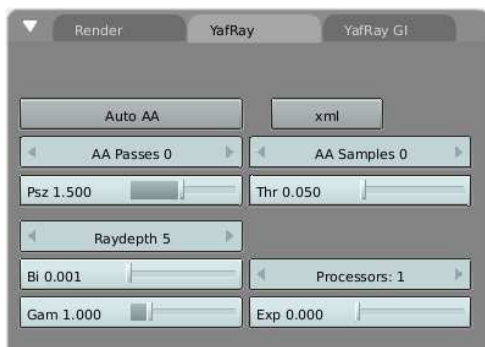


Figura 22. El panel YafRay.

- **AA Passes:** Se usa para el número de pasadas utilizadas en el Anti-Aliasing (AA). A mayor número de pasadas, mejores resultados y tiempos de render más altos. Si se realiza más de una pasada, cada paso muestrea el total de las muestras elegidas en **AA Samples**, de ese modo se logran mejor AA de los píxeles en las áreas oscuras. Si los resultados obtenidos con una sola pasada son sucios, contienen ruido, puedes incrementar el número de pasadas para un mayor control y también un mayor número de muestras por pasada. Una única pasada de AA con un valores de muestreo altos no es necesariamente mejor que menos muestras con más pasadas.
- **AA Samples:** El número de muestras por pasada. El muestreo total de AA puede ser calculando multiplicando **AA passes** por **AA samples**. En nuestro caso, para la imagen suavizada con el DoF utilizada más adelante, hemos utilizado $4 \times 4 = 16$ muestras de AA.
- **Psz:** Psz es el tamaño del filtro de pixel (*Pixel filter size*), que se utiliza durante los cálculos de AA para suavizar la superposición entre pixels vecinos.
- **Threshold: (Umbral).** Esto establece el umbral para la máxima diferencia de brillo entre píxeles vecinos. Cuando dicha diferencia está por encima del valor elegido, se toman muestras extras mientras el resultado esté por debajo del umbral o se alcance el máximo de muestras/pasada. Un umbral más bajo significa que más pixels serán

suavizados, por lo que con un valor de 0 todos los pixels en el render serán suavizados. Con un 1, no se realiza AA.

- Raydepth: (*Profundidad del Rayo*). Afecta al máximo número de rebotes de un rayo reflejado o refractado. Sólo es importante para cristales o espejos, un valor de profundidad más alto mejora la calidad de la reflexión/refracción en el cristal o espejo.

DoF: Profundidad de campo

YafRay soporta DoF ("*Depth of field*", profundidad de campo -visual-). La configuración del DoF está fácilmente accesible. Primero selecciona la cámara en la vista y después pulsa **F9**. Verás una solapa en el panel de la cámara (*Camera*) llamada *!YafRay DoF*. En la solapa "_Camera_" activa *Show limits* (mostrar límites); hacer esto te permitirá ver una cruz amarilla en la línea de *Show limits* de la cámara, dentro de la vista. La cruz amarilla es lo que usamos para establecer los objetos que queremos dejar enfocados. La distancia puede establecerse a través del botón numérico DoFDist. La configuración de la apertura (*Aperture*) es el valor que necesitas cambiar para conseguir el DoF que quieres obtener. Normalmente un valor entre .100 y .500 es suficiente para la mayor parte de las escenas.



Figura 23. Fíjate en el ruido presente en los tablones.

Como eliminar artefactos en las escenas renderizadas con DoF. El filtro DoF necesita más muestras por escena para obtener la cantidad suficiente de desenfoque (*blurring*). Para conseguirlo, has de deshabilitar el *Automatic AA* en la solapa YafRay del panel de render. Ahí intenta ajustar el valor de *AA Passes* y *AA Sample* adecuadamente. Si tienes valores grandes para la apertura, entonces necesitarás tener múltiples pasadas además de un mayor número de muestras por pasada. Incrementar el *anti-aliasing* también aumenta los tiempos de render.



Figura 24. Incrementar el muestreado manualmente y el tamaño de apertura soluciona el problema.

HDRI or High Dynamic Range Illumination

YafRay posee soporte para HDRI (Iluminación de rango dinámico alto). Para usarlo, ve a los botones *World* **F8**. Ve a la solapa *Texture and Input* y añade una nueva textura. Después ve a la selección de textura, **F6**, selecciona "imagen" en *Texture type*, pulsa el botón de cargar (*Load*) una imagen, localiza el fichero HDR, seleccionalo y pulsa Intro. Blender te mostrará la imagen HDRI en la solapa *World*, aunque se carga automáticamente durante el render. También puedes aumentar o reducir la exposición del HDRI desde el botón de brillo en la solapa de colores en el panel de texturas (**F6**). Los posibles valores de exposición son -1, 0 y 1 para los deslizadores (sliders) del brillo en 0, 1 y 2. Esto se debe a que la exposición en HDRI sólo se puede modificar en valores enteros. Como último paso, usa cualquiera de los métodos GI. Normalmente *Skydome* funciona correctamente.

Figure 25. HDRI render using Uffizi probe. It need more quality level.

Comentarios finales.

Quisiera dar las gracias a **eeshlo**, en gran medida él es el culpable de mi motivación para aprender cosas sobre YafRay, lo que me permitió escribir este tutorial. También quería agradecer su ayuda resolviendo mis dudas. También quiero agradecer a Dreamsgate por ayudarme en la edición y apoyo para este documento.

Este tutorial no está terminado, ya que hay varios temas que no he probado por mi mismo, como las caústicas, y técnicas de GI para cierto tipo de renders. Por ello llamo a este tutorial "versión 1". Las próximas mejoras no llegarán necesariamente en intervalos regulares. Si tienes cualquier duda, o hay algún punto por el que he pasado demasiado por encima o que no esté claro en todo lo escrito, estaré encantado de saber tu opinión. También son bienvenidas las sugerencias para posibles mejoras. E-mail: nawani_g AT yahoo.co.in (N. del T: Escribir en inglés ;-))

Preparando su trabajo para Vídeo

Una vez que ha dominado las técnicas de animación, seguramente comenzará a producir maravillosas animaciones codificadas con sus codecs favoritos, y probablemente los compartirá en Internet con el resto de la comunidad.

Pero tarde o temprano será tentado por el deseo de realizar una animación para Televisión o quizás para grabar sus propios DVDs.

Para ahorrarle alguna desilusión, aquí se presentan algunas pistas y secretos específicamente orientados a la preparación de vídeo. La primera y principal es recordar las dobles líneas blancas punteadas en la vista de cámara.

Si renderiza para PC, verá en su totalidad la imagen renderizada que es lo abarcado dentro del rectángulo punteado *exterior*. Para Televisión, algunas líneas y algunas partes de esas líneas se perderán debido a la naturaleza del barrido del haz de electrones en el tubo de rayos catódicos de su TV. Se garantiza que aquello que esté dentro del rectángulo punteado *interior* en la vista de cámara será visible en la pantalla. Todo aquello dentro del espacio entre los dos rectángulos, puede o no ser visible, dependiendo del equipo receptor de TV en el que esté viendo su vídeo.

Más aún, el tamaño del renderizado es dictado estrictamente por el estándar de TV. Blender ofrece tres configuraciones preestablecidas para su comodidad:

- *PAL* 720x576 píxeles con relación de aspecto de 54:51.
- *NTSC* 720x480 píxeles con relación de aspecto de 10:11.
- *PAL 16:9* 720x576 con relación de aspecto de 64:45 , para renderizado de TV en pantalla ancha de 16:9.

Recuerde lo relativo a "Relación de Aspecto". Las pantallas de TV *no* tienen píxeles cuadrados como tienen los monitores de computadora, sus píxeles son más bien rectangulares, por lo que es necesario generar imágenes *pre-distorsionadas* que se verán mal en una computadora, pero se verán perfectas en un equipo de TV.

Saturación de Color

La mayoría de las cintas de vídeo y señales de vídeo no se basan en modelos RGB sino en modelos YUV o YCrCb en el caso de Europa o YIQ en los Estados Unidos, siendo este último muy similar al primero. De ahí que también se necesite algún conocimiento al respecto de este tema.

El modelo YUV envía información como 'Luminancia', o intensidad (Y) y dos señales de 'Crominancia' , rojo y azul. En realidad, un equipo de TV blanco y negro sólo muestra luminancia; en cambio, los televisores a color, reconstruyen el mismo a partir de crominancias, siendo:

$$Y = 0.299R + 0.587G + 0.114B$$

$$U = Cr = R - Y$$

$$V = Cb = B-Y$$

Aunque una imagen estandar de 24 bits tiene 8 bits por canal, para mantener bajo el ancho de banda y, considerando que el ojo humano es más sensible a la luminancia que a la crominancia, la primera es enviada con más bits que las otras dos.

Esto lleva a un rango dinámico de colores menor, en Vídeo, que aquel al que está acostumbrado en los monitores. Debe, por lo tanto, tener en cuenta que no todos los colores se mostrarán correctamente. Una regla práctica es mantener los colores lo más "grisáceos" o "insaturados" que se pueda, esto se puede traducir básicamente en mantener el rango dinámico de sus colores por debajo de 0.8.

En otras palabras, la diferencia entre el valor de RGB más alto y el valor más bajo no debería sobrepasar 0.8(rango [0-1]) ó 200 (rango [0-255]).

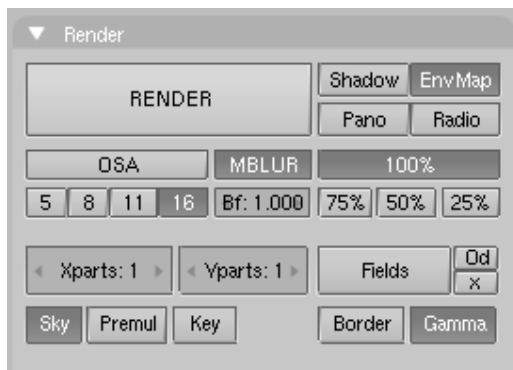
No es una regla estricta, valores algo mayores que 0.8 son aceptables, pero un material con RGB=(1.0,0,0) se verá muy mal.

Renderizando por Campos

El estándar de TV indica que debería haber 25 cuadros por segundo (PAL) ó 30 cuadros por segundo (NTSC). Como el fósforo de la pantalla no mantiene la luminosidad por mucho tiempo, esto puede provocar un parpadeo apreciable. Para minimizar este efecto, los televisores no representan cuadros como lo hace una computadora, sino que representan semi-cuadros, o *campos* (*fields*) a una velocidad de refresco doble, osea 50 semi-cuadros por segundo en PAL y 60 semi-cuadros por segundo en NTSC. Originalmente esto estaba relacionado con la frecuencia de las líneas de tensión en Europa (50Hz) y en los Estados Unidos (60Hz).

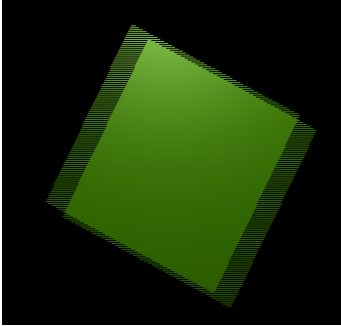
Los campos están "entrelazados" en el sentido que un campo presenta todas las líneas pares de un cuadro completo y el campo siguiente muestra todas las líneas impares.

Como no hay una diferencia de tiempo despreciable entre cada campo (1/50 ó 1/60 avas de segundo), normalmente no sirve renderizar un cuadro y separarlo en dos semi-cuadros. Se hará presente un temblor de los bordes en los objetos móviles claramente apreciable.



Configuración de Render por campos.

Para manejar de forma óptima este tema, Blender ofrece renderizado por campos. Cuando se presiona el botón *Fields* en el Panel *Render*(*Configuración de Render por campos*), Blender genera cada cuadro en dos pasadas. En la primera pasada, renderiza sólo las líneas pares, luego *avanza en el tiempo un medio paso* y renderiza todas las líneas impares.



Resultado de renderizar por campos.

Esto produce resultados extraños en una pantalla de PC (*Resultado de renderizar por campos*), pero se verá perfectamente en un televisor.

Uno de los dos botones al lado del botón *Fields* fuerza el renderizado del campo Impar primero (*Odd*) y el otro deshabilita el paso de medio cuadro entre campos (*x*).

Configurando el orden de campo correcto:

Por omisión, Blender producirá los campos Pares antes que los Impares, de acuerdo a los estándares PAL europeos. En cambio, en NTSC los campos impares se barren primero.

Por supuesto, si se equivoca al elegir las cosas se verán peor que si no hubiera renderizado por campos.

PARTE XII - RADIOSIDAD

Muchos modelos de render, incluyendo el trazado de rayos, asumen un modelo espacial simplificado, altamente optimizado para la luz que entra en el "ojo" para dibujar la imagen. Puede añadir reflexiones y sombras a este modelo para conseguir un resultado más realista. Mientras, hay un aspecto importante que falta. Cuando una superficie tiene un componente de luz reflexiva, no solamente muestra nuestra imagen, también brilla con la luz de las superficies en su alrededor. Y viceversa. De hecho, la luz rebota por todo el entorno hasta que toda la energía de la luz sea absorbida (o haya escapado!). La luz re-irradiada porta información sobre el objeto que la ha rebotado, normalmente color. Por lo tanto no solo las sombras son 'más oscuras' por la luz re-irradiada, sino que también tienden a mostrar el color de objeto más cercano y más iluminado. Un fenómeno comúnmente llamado como 'escape de color'. (*Ejemplo de radiosidad.*).



Ejemplo de radiosidad.

En entornos cerrados, la energía de la luz es generada por 'emisores' y es considerada como la reflexión o absorción de las superficies en el entorno. El parámetro el cual la energía queda en una superficie se llama la 'radiosidad' de la superficie. Al contrario de los métodos tradicionales de render, los métodos de radiosidad primero calculan todas las interacciones de la luz de un entorno de forma independiente de la vista. Así, pueden ser renderizadas varias vistas en tiempo real. En Blender, desde la versión 2.28, la Radiosidad es una herramienta tanto de modelado como de render. Esto quiere decir que puede activar Radiosidad sin un renderizado o usar la Radiosidad para pintar los colores de vértice y luces de vértice de sus mallas, para un uso posterior.

Capítulos

- Renderizado de Radiosidad
- Radiosidad como Herramienta de Modelado
- Un Jugoso Ejemplo de Radiosidad

Renderizado de Radiosidad

El método Radiosidad de Blender

Primero, ¡un poco de teoría! Puede saltar a la siguiente sección si lo desea, y volver aquí si surgen dudas. A finales de los ochenta y principios de los noventa, la Radiosidad era un tema candente en el mundo de los gráficos 3D por ordenador. Se desarrollaron métodos diversos; la más exitosa de esas soluciones estaba basada en el método del "refinamiento progresivo" con una esquema de "subdivisión adaptativa". Y esto es lo que Blender utiliza. Para poder sacar el máximo partido del método de Radiosidad de Blender, es importante comprender los siguientes principios:

- *Método de Elemento Finito*

Muchos métodos de simulación o de gráficos por ordenador asumen una simplificación de la realidad con "elementos finitos". Para una solución visualmente atractiva (e incluso científicamente válida), no es siempre necesario zambullirse hasta un nivel de detalle molecular. En su lugar, puede reducirse el problema a un número finito de elementos representativos y bien descritos. Es un hecho común el que tales sistemas convergen rápidamente en una solución estable y fiable. El método de Radiosidad es un ejemplo de método de elemento finito, puesto que cada cara está considerada un "elemento finito", y su emisión de luz está considerada como un todo.

- *Parches y Elementos*

En el universo de la Radiosidad, distinguimos entre dos tipos de caras 3D:

Parches.

Son triángulos o cuadrángulos capaces de emitir energía. Para una solución rápida es importante tener cuantos menos de estos Parches como sea posible. Pero para acelerar las cosas, la energía es modelada como si la energía se emitiera desde el centro del Parche; el tamaño de los Parches debería, por lo tanto, ser lo suficientemente pequeño para que hubiera una distribución realista de energía. Por ejemplo, cuando un pequeño objeto está situado encima del centro del Parche, toda la energía que el Parche envía es oscurecida por este objeto, ¡incluso si el Parche es grande! Este Parche debería ser subdividido en Parches más pequeños.

Elementos.

Estos son los triángulos o cuadrángulos que reciben energía. Cada Elemento está asociado a un Parche. En realidad, los Parches están subdivididos en pequeños Elementos. Cuando un Elemento recibe energía, absorbe parte de la misma (dependiendo de su color) y pasa el remanente al Parche, para su posterior radiación. Puesto que los Elementos son también las caras que se muestran, es importante mantenerlas cuanto más pequeñas posible, para producir bordes de sombras y gradientes de luz sutiles.

- *Refinamiento Progresivo*

Este método empieza examinando todos los Parches disponibles. El Parche con más cantidad de energía "no disparada" es seleccionado para disparar toda su energía al entorno. Los Elementos del entorno reciben esta energía, y la añaden a la energía "no

disparada" de los Parches que tienen asociados. Luego, el proceso se inicia una vez más para el Parche que tiene ahora más energía por disparar. Esto se sucede para todos los Parches hasta que no se recibe más energía, o hasta que la cantidad de energía "no disparada" converge por debajo de cierto valor.

- *El método del hemicubo*

El cálculo de cuánta energía da cada Parche a un Elemento se realiza mediante el uso de "hemicubos". Exactamente situado en el centro del Parche, un hemicubo (literalmente "medio cubo") consiste en 5 pequeñas imágenes del entorno. Para cada píxel en estas imágenes, cierto Elemento visible es codificado mediante color, y la cantidad de energía transmitida puede calcularse. Especialmente con el uso de hardware especializado, el método del hemicubo puede acelerarse significativamente. En Blender, sin embargo, los cálculos del hemicubo se realizan por software. Este método es, de hecho, una simplificación y optimización de la fórmula "real" de la Radiosidad (diferenciación del factor forma). Por esta razón, la resolución del hemicubo (el número de píxeles de sus imágenes) es aproximado, y su cuidadosa configuración es importante para prevenir defectos de aliasing.

- *Subdivisión adaptativa*

Puesto que el tamaño de los parches y los elementos en una malla define la calidad de la solución de la Radiosidad, se han desarrollado esquemas de subdivisión automática para definir el tamaño óptimo de Parches y Elementos. Blender posee dos métodos de subdivisión automática:

1. Parches Subdivisión-disparo.

Disparando energía al entorno, y comparando los valores del hemicubo con el "factor forma" matemático actual, pueden detectarse errores que indican la necesidad de una nueva subdivisión del Parche. El resultado son Parches menores y un mayor tiempo de resolución, pero un realismo de la solución más alto.

2. Elementos Subdivisión-disparo.

Disparando energía al entorno, y detectando cambios altos de energía (gradientes) dentro de un Parche, los Elementos de este Parche son subdivididos un nivel extra. El resultado son Elementos menores y un mayor tiempo de resolución, y quizá más aliasing, pero un nivel de detalle más alto.

- *Visualización y Post-procesado*

La subdivisión de Elementos en Blender está "equilibrada"; ello significa que cada elemento difiere un máximo de 1 nivel de subdivisión respecto a sus vecinos. Esto es importante para una visualización agradable y correcta de la solución de Radiosidad con caras sombreadas mediante Gouraud. Normalmente después de la resolución, la solución consiste en miles de pequeños Elementos. Filtrándolos y eliminando "dobles", el número de Elementos puede reducirse significativamente sin destruir la calidad de la solución de Radiosidad. Blender guarda los valores de energía en números de punto flotante. Ello hace posible la configuración de situaciones de iluminación dramática, cambiando los valores estándar de multiplicación y gamma.

- *Radiosidad para modelado*

El último paso puede ser sustituir las mallas de entrada con la solución de Radiosidad (botón *Replace Meshes* - Sustituir Mallas). En ese momento los colores de los vértices se convierten de un valor de punto flotante a un valor RGB de 24 bits. Los Objetos de Malla antiguos son eliminados y reemplazados por uno o más Objetos de Malla nuevos. Se pueden eliminar los datos de Radiosidad con *Free Data* (Liberar Datos). Los nuevos objetos reciben un material por defecto que permite un renderizado inmediato. Hay dos parámetros importantes en un Material para trabajar con colores de vértice:

VColPaint.

Esta opción trata los colores de los vértices como un reemplazo para el valor RGB normal en el Material. Deben añadirse Lamps para poder ver los colores de Radiosidad. De hecho, se puede usar la iluminación y sombreado de Blender como de costumbre, y conservar todavía una apariencia neta de Radiosidad en el render.

VColLight.

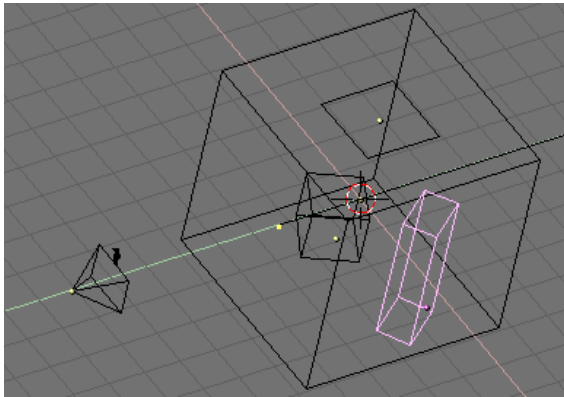
Los colores de los vértices se añaden a la luz al renderizar. Puede verse el resultado incluso sin Lamps. Con esta opción, los colores de los vértices son pre-multiplicados por el color RGB del Material. Esto permite un ajuste fino de la cantidad de "luz de Radiosidad" en el render final.

Como todo en Blender, la configuración de Radiosidad se almacena en un bloque de datos. Queda adjuntado a la escena, y cada escena de Blender puede tener un "bloque" de Radiosidad diferente. Puede usarse esta característica para dividir entornos complejos en escenas con solucionadores de Radiosidad independientes.

Renderizado de Radiosidad

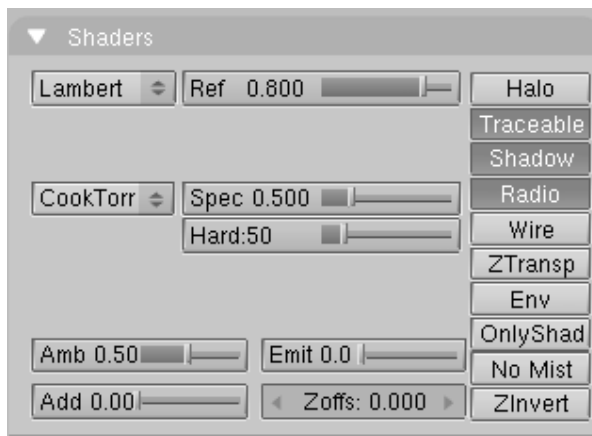
Asumamos que tenemos una escena lista, y que queremos renderizarla con Renderizado de Radiosidad. Lo primero que debe comprenderse al usar Radiosidad es que no es necesaria *ninguna Lamp*, aunque se requieren algunas mallas con la propiedad de material *Emit* mayor que cero, pues serán las fuentes luminosas. Puede construirse la escena de prueba mostrada en *Ejemplo de radiosidad.*; es bastante sencillo. Simplemente debe crearse un cubo grande para la habitación, darle materiales diferentes a las paredes laterales, añadir un cubo y un cubo estirado en su interior, y añadir un plano con un valor *Emit* diferente de cero junto al techo para simular la luz de área (*Decorado para la prueba de Radiosidad.*). Se asignan materiales como de costumbre a los modelos de entrada. El valor RGB del Material define el color del Parche. El valor '*Emit*' de un Material define si un Parche es cargado con energía al principio de la simulación de Radiosidad. El valor '*Emit*' es multiplicado por el área de un Parche para calcular la cantidad inicial de energía no disparada.

Caras emisoras: ¡Compruebe el número de 'emisores' en la consola de Blender! Si es cero, no puede suceder nada interesante. Es necesario como mínimo un Parche emisor para obtener luz, y en consecuencia, una solución.




Decorado para la prueba de Radiosidad.

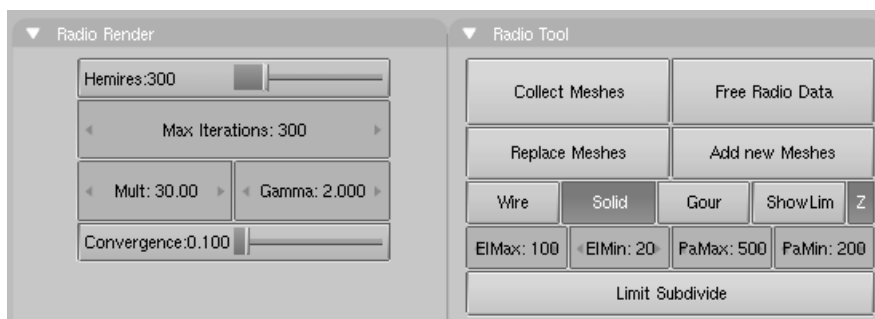
Al asignar materiales debemos asegurarnos de que todos tienen el conmutador *Radio* activado, para habilitar el Panel *Shaders* de los botones del subcontexto Material (*Material con la Radiosidad habilitada.*).



Material con la Radiosidad habilitada.

Nótese que la emisión de luz está gobernada por la dirección de las normales de una malla, por lo que el plano emisor de luz debería tener una normal *apuntando hacia abajo*, y el cubo exterior (la habitación) debería tener las normales apuntando hacia adentro (¡inviértalas!).

Seleccione el subcontexto Radiosidad  del Contexto Shading. Los Paneles, mostrados en Figure 18-4, son dos: *Radio Rendering* (renderizado de radiosidad) el cual gobierna la Radiosidad cuando es usado como herramienta de renderizado (caso presente), y *Radio Tool* (herramienta de radiosidad), el cual gobierna la Radiosidad como una herramienta de modelado (próxima sección).

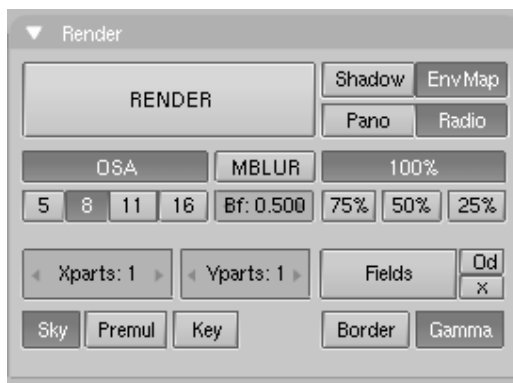


Botones Radiosidad para renderizado de radiosidad.

Los botones definen:

- *Hemires*: - La resolución del hemicubo, es decir, de las imágenes codificadas con color, utilizadas para encontrar los Elementos que son visibles desde un "Parche de disparo", y así recibir energía. Los Hemicubos no son almacenados, sino recalculados cada vez para cada Parche que dispara energía. El valor 'Hemires' determina la calidad de la Radiosidad e influye significativamente en el tiempo de resolución.
- *Max Iterations*: - El número máximo de iteraciones de Radiosidad. Si se pone a cero, la Radiosidad seguirá hasta que se cumpla el criterio de convergencia. Se recomienda fuertemente asignarle un número diferente de cero, normalmente mayor que 100.
- *Mult*, *Gamma*: - El espacio de color de la solución de Radiosidad, es muchísimo más detallado de lo que puede expresarse con simples valores RGB de 24 bits. Cuando los Elementos se convierten en caras, sus valores de energía son convertidos a un color RGB usando los valores *Mult* y *Gamma*. Con el valor *Mult* puede multiplicarse el valor de la energía; con *Gamma* puede ajustarse el contraste de los valores de la energía.
- *Convergence*: - Cuando la cantidad de energía no disparada en un entorno es inferior a este valor, la resolución de Radiosidad se detiene. La energía inicial no disparada en un entorno se multiplica con el área de los Parches. Durante cada iteración, parte de la energía es absorbida, o desaparece cuando el entorno no es un volumen cerrado. En el sistema de coordenadas estándar de Blender, un emisor típico (como en los archivos de ejemplo) tiene un área relativamente pequeña. Por esta razón, el valor de convergencia es dividido por un factor de 1000 antes de la prueba.

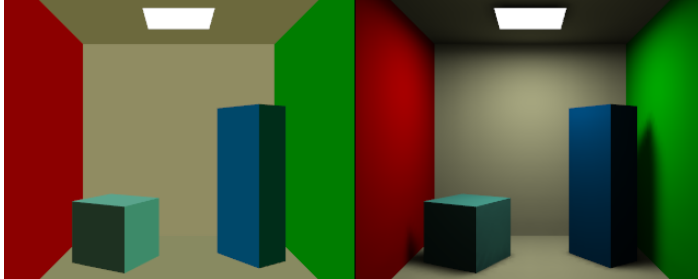
Ponga *Max Iterations* a 100 y seleccione el Contexto Scene y el Subcontexto Render (F10). Localice el botón conmutador *Radio* (*Habilitando la Radiosidad en los botones de Render.*) en el Panel *Render* y póngalo a 'on' para habilitar la Radiosidad, y luego ¡Render! (F12).



Habilitando la Radiosidad en los botones de Render.

El render tardará más que de costumbre; en la consola apreciará un contador subiendo. El resultado será bastante pobre (*Render de Radiosidad para mallas rudas (izquierda) y mallas refinadas (derecha).*, izquierda) porque el render automático de radiosidad ¡no realiza un refinado adaptativo! Seleccione todas las mallas, una tras otra, y en EditMode subdivídalas al

menos tres veces. La habitación, que es mucho mayor que las otras mallas, puede ser subdividida incluso cuatro veces. Aumente *Max Iterations* un poco, 300 o más. Intente renderizar una vez más (F12). Esta vez el renderizado tardará incluso más, pero los resultados serán mucho mejores, con sombras suaves y derrame de colores. (*Render de Radiosidad para mallas rudas (izquierda) y mallas refinadas (derecha).*, derecha).



Render de Radiosidad para mallas rudas (izquierda) y mallas refinadas (derecha).

Nota: En el renderizado de Radiosidad, Blender actúa como en un renderizado normal; esto significa que las texturas, curvas, superficies e incluso Objetos Dupliframe son correctamente tratados.

Radiosidad como Herramienta de Modelado

La radiosidad puede utilizarse como herramienta de modelado para definir Vértices de colores y luces. Esto puede resultar muy útil si lo que quieres es refinar tus modelos, o si pretendes utilizarlos en el Game Engine. Es más, el Modelado Radiosivo permite Refinamiento Adaptativo, lo que no permite el Renderizado Radiosivo. Hay unos pocos puntos importantes que hay que comprender para el trabajo práctico con Modelado Radiosivo: En Blender sólo las Mallas pueden actuar como entrada del Modelado Radiosivo. Esto es debido a que el proceso genera Vértices de colores... por tanto las entradas deben de ser Vértices. También es importante darse cuenta de que *cada* Cara de una Malla se convierte en un Parche, y por tanto en un emisor potencial de energía y reflector. Típicamente, los Parches largos envían y reciben más energía que los pequeños. Es importante, por tanto, tener un modelo de entrada equilibrado con Parches lo suficientemente grandes para crear una diferencia. Cuando se añaden Caras excesivamente pequeñas, casi nunca recibirán la suficiente energía para ser apreciadas por el método de "Refinamiento Progresivo", el cuál solo selecciona Parches con la suficiente cantidad de energía no disparada.

Objetos No-Malla: El método anterior trabaja sólo con Mallas, por lo tanto, hay que convertir las Curvas y Superficies en Mallas (CTRL+C) antes de aplicarlo.

Fase 1: Recolectando Mallas

Todas las Mallas seleccionadas y visibles en la Escena actual se convierten en Parches apretando el botón *Collect Meshes* del panel *Radio Tool* (*Botón Gourad*). Como resultado un nuevo panel, *Calculation*, aparece. En este momento Blender ha entrado en el modo de modelado con radiosidad, y las otras funciones de edición han sido bloqueadas hasta que el botón recién creado *Free Data* sea presionado. El texto *Phase* sobre los botones ahora pone

Init y muestra el número de Parches y Elementos. Tras recopilar todas las Mallas, éstas son dibujadas en un pseudo modo de iluminación que difiere de manera clara del dibujo normal. El panel *Radio Tool* (*Botón Gourad*) muestra tres botones: *Wire*, *Solid*, *Gour* que son tres opciones de modo de dibujo independientes del modo de dibujo indicado en una ventana 3D. El display Gouraud se muestra solamente después de que el Proceso Radiosivo ha comenzado. Presiona el botón *Gour* para obtener resultados más suaves en superficies curvas.



Botón Gourad

Fase 2: Límites de Subdivisión

Blender permite configurar los tamaños máximo y mínimos de los Parches y Elementos en las *Radio Tools* y paneles *Calculation* (*Botones Radiosivos para Subdivisión*).



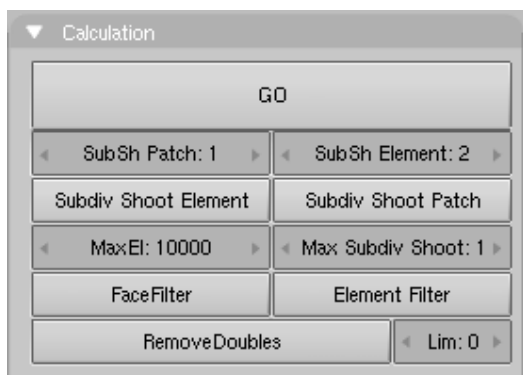
Botones Radiosivos para Subdivisión

- *Limit Subdivide* Los Parches son subdivididos respecto a los valores "PaMax" y "PaMin". Esta subdivisión es automáticamente realizada cuando ha comenzado una acción "GO".
- *PaMax, PaMin, ElMax, ElMin* El tamaño máximo y mínimo de un Parche o Elemento. Estos límites son usados durante todas las fases de Radiosidad. La unidad se expresa en un 0.0001 del tamaño de marco (boundingbox) del entorno completo. Por lo tanto, con la configuración por defecto de 500 y 200 de tamaño máximo y mínimo del tamaño del Parche, 0.05 del modelo completo (1/20) y 0.02 del modelo completo (1/50)
- *ShowLim, Z* Esta opción visualiza los límites de Parche y Elemento. Presionando la opción Z, los límites son dibujado y rotados de manera diferente. Las líneas blancas

muestran los límites de los Parches, las líneas magenta muestran los límites de los Elementos.

Fase 3: Subdivisión Adaptativa

Últimos ajustes antes de comenzar el análisis (*Botones de Radiosidad*).



Botones de Radiosidad

- *MaxEl* Máximo número de elementos permitidos. Puesto que los elementos son divididos automáticamente por Blender, la cantidad de memoria utilizada durante el tiempo de resolución puede ser controlada con este botón. Como regla general 20.000 elementos requieren 10Mb de memoria.
- *Max Subdiv Shoot* El número máximo de parches de disparo que son evaluados para el proceso de "subdivisión adaptativa" (descrito arriba). Si es cero todos los parches con 'Emit' son evaluados.
- *Subdiv Shoot Patch* El disparo de energía hacia el entorno permite la detección de errores que indican la necesidad de una mayor subdivisión de los parches. El resultado son parches menores y un tiempo de resolución mayor, pero a cambio se obtiene un mayor realismo de la solución. Esta opción puede ser también llevada a cabo de manera automática cuando la acción *GO* ha comenzado.
- *Subdiv Shoot Element* Disparando energía al entorno, y detectando fuertes variaciones de energía (frecuencias) dentro de un parche, los elementos de este parche son seleccionados para ser divididos otro nivel extra. La subdivisión es llevada a cabo solamente cada vez que llames a esta función. El resultados son elementos más pequeños, un mayor tiempo de resolución, y probablemente más aliasing, pero a cambio se obtiene un mayor nivel de detalle. Esta opción puede ser también llevada a cabo de manera automática cuando la acción *GO* ha comenzado.
- *SubSh P* El número de veces que el entorno es comprobado para detectar Parches que necesitan subdivisión.
- *SubSh E* El número de veces que el entorno es comprobado para detectar elementos que necesitan subdivisión.

- **GO** Con este botón comienza la Simulación Radiosiva. Las fases son :
 - *Limit Subdivide*. Cuando los Parches son demasiado largos , éstos son divididos.
 - *Subdiv Shoot Patch*. El valor de *SubSh P* define el número de veces que la función *Subdiv Shoot Patch* es llamada. Como consecuencia los Parches son subdivididos.
 - *Subdiv Shoot Elem*. El valor de *SubSh E* define el número de veces que la función *Subdiv Shoot Element* es llamada. Como consecuencia los Elementos son subdivididos.
 - *Subdivide Elements*. Cuando los Elementos son todavía mayores que el tamaño mínimo son subdivididos. Ahora, la cantidad máxima de memoria está normalmente asignada.
 - *Solve*. Este es el método actual de 'refianamiento progresivo'. El puntero del ratón visualiza el paso de la iteración, el número actual de Parches que dispararon su energía en el entorno. Este proceso continúa hasta que la energía no disparada del entorno es menor que el valor *Convergence* o cuando el máximo número de iteraciones han sido alcanzada.
 - *Convert to faces*. Los puntos son convertidos en triángulos o cuadrados con Aristas 'ancladas', para asegurar que sea posible mostrar un agradable Gouraud no-discontinuo.

Este proceso puede ser terminado con **ESC** durante cualquier fase.

Fase 4: Editando la Solución

Una vez que la solución de radiosidad ha sido computada aún se pueden ejecutar una serie de acciones. (*Post proceso de radiosidad*.).



Post proceso de radiosidad.

- *Element Filter* Esta opción filtra los elementos para eliminar el aliasing, para suavizar los contornos de las sombras, o para forzar colores ecualizados para la opción *RemoveDoubles*.
- *RemoveDoubles* Cuando dos elementos próximos muestran un color que difiere menos que *Lim*, los elementos son juntados. El valor *Lim* usado anteriormente es expresado en un estándar de 8 bits de resolución; un rango de color de 0-255.

- *FaceFilter* Los Elementos son convertidos en Caras para su visualización. *FaceFilter* fuerza un nivel extra de suavizado (smoothing) en el resultado mostrado, sin ningún cambio en los valores de los Elementos.
- *Mult, Gamma*: Tienen el mismo significado que en el Renderizado Radiosivo.
- *Add New Meshes* Las Caras de la solución Radiosiva son convertidas a Objetos de Malla con Vértices de colores. Un nuevo material que permite el renderizado inmediato es añadido. *Las Mallas de entrada permanecen sin cambios.*
- *Replace Meshes* Como antes, pero las Malla de entrada son borradas.
- *Free Radio Data* Todos los Parches, Elementos y Caras son liberados de Memoria. Siempre se debe realizar esta acción después de usar Radiosidad para ser capaz de volver al modo normal de edición.

Un jugoso ejemplo de radiosidad

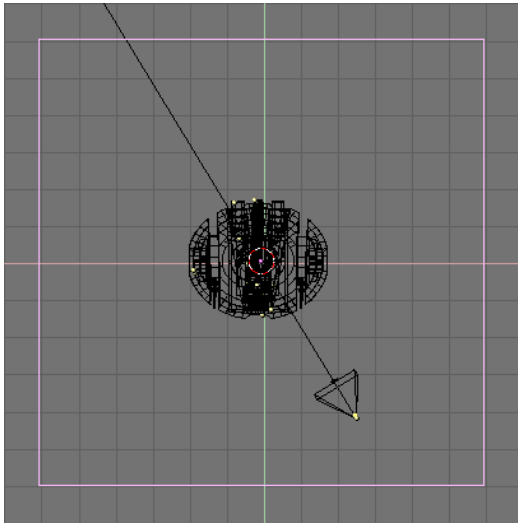
Para alejarnos de una vez de la árida teoría y ver que es lo que el modelado de Radiosidad puede realmente conseguir, veamos un ejemplo. Esto realmente le mostrará una verdadera escena de iluminación global, con resultados más suavizados que las 'luces focales de dupliverts'. Técnica mostrada en el capítulo de iluminación para conseguir algo como *Cylon Raider renderizado con radiosidad*.



Cylon Raider renderizado con radiosidad.

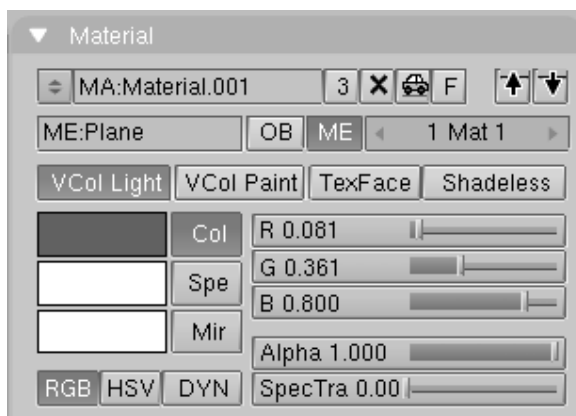
Configurando

Tenemos únicamente dos elementos en la escena al comienzo: una nave de asalto (si recuerda alguna película de ciencia ficción...) y una cámara. La nave de asalto tiene el material gris por defecto, excepto las ventanas principales de la cabina que son negras. Para esta técnica no necesitaremos ninguna lámpara. Lo primero que debemos hacer de añadir a la escena es un plano. Este plano será usado como el suelo en nuestra escena. Cambie el tamaño del plano como se ve en Figure 18-12 y sitúelo justo debajo de la nave. Deje un poco de espacio entre el plano y el plano de la nave. Esto le dará un buen toque de "levitación".



Añadir un plano

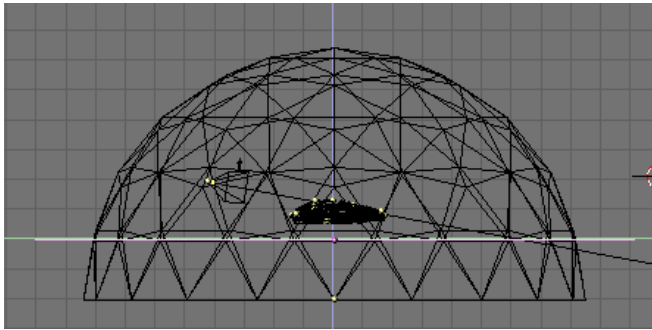
Después, tendrá que dar al plano un material y seleccionar un color para él. Lo intentaremos con un bonito azul. Puede usar la propiedad que ve en *Color del plano* para ello.



Color del plano

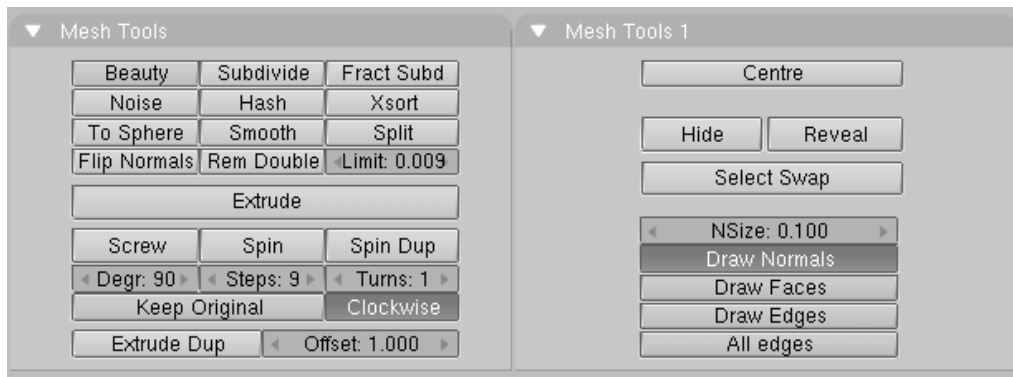
La cúpula del cielo

Queremos hacer un render GI, de modo que lo siguiente que tenemos que hacer es añadir una icoesfera. La esfera será nuestra fuente de luz en lugar de las lámparas típicas. Lo que vamos a hacer es usar sus caras como *emisores* que proyectarán luz para nosotros en multitud de direcciones, en lugar de en una sola como una única lámpara típica. Esto nos dará el efecto deseado. Para conseguirlo, añada una icoesfera con 3 subdivisiones. Mientras siga en el modo edición, use el modo selección **BKEY** para seleccionar la porción inferior de la esfera y eliminarla. Así nos quedaremos con nuestra cúpula. Cambie el tamaño de la cúpula para que encaje bien con la escena y hágala coincidir con el plano. Se debería parecer a *Cúpula del cielo*..



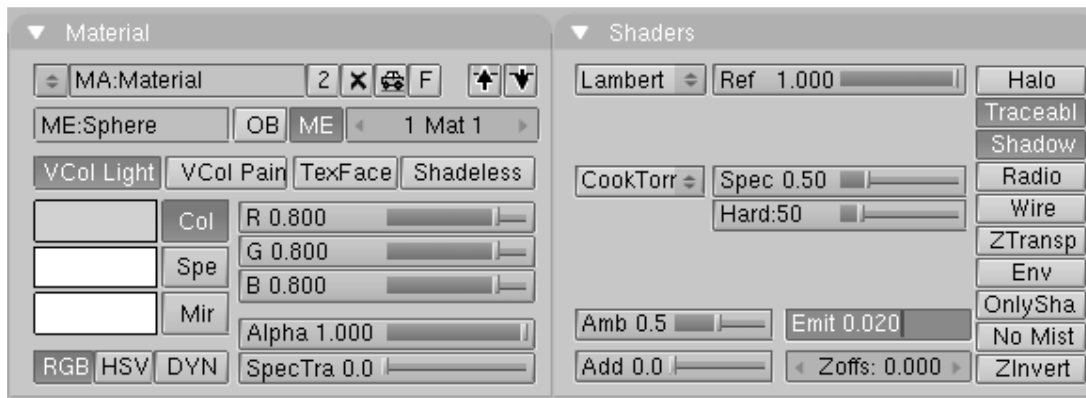
Cúpula del cielo.

Lo siguiente, es asegurarnos de que tenemos seleccionados todos los vértices de la cúpula, para hacer click en Edición de Botones (F9) y seleccionar *Dibujar Normales* (Draw Normals). Esto nos permitirá ver en qué dirección están "emitiendo" los vértices. Por defecto será hacia afuera, así que presione el botón *Flip Normals*, que cambiará el emisor del vértice para proyectarlo desde fuera hacia dentro de la cúpula (*Cambiando las normales.*)



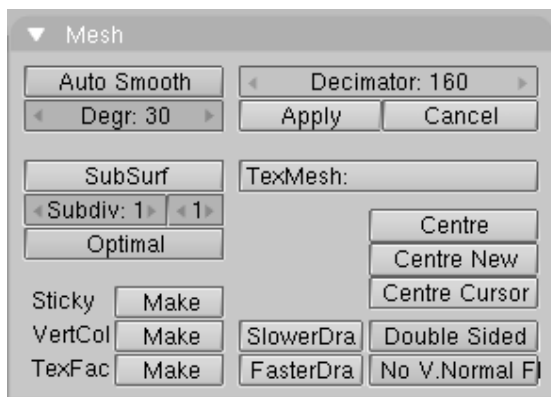
Cambiando las normales.

Ahora que hemos creado la cúpula, necesitamos un nuevo material. Cuando cree el material para la cúpula, cambie las siguientes propiedades en los Botones de Material (F5): *Add* = 0.000 *Ref* = 1.000 *Alpha* = 1.000 *Emit* = 0.020. El deslizador *Emit* es la clave aquí. Esta propiedad controla la cantidad de luz "emitida" desde nuestra cúpula. 0.020 está bien por defecto. ¡Recuerde que la cúpula es la parte más grande de la escena! ¡Tampoco queremos demasiada luz! Usted puede experimentar con esta propiedad para conseguir diferentes resultados. The Emit slider here is the key. This setting controls the amount of light "emitted" from our dome. 0.020 is a good default. Remember that the dome is the bigger part of the scene! you don't want too much light! But you can experiment with this setting to get different results. Cuanto menor sea ésta propiedad, piense que durará más el tiempo de resolución después. (*Material de la cúpula.*).



Material de la cúpula.

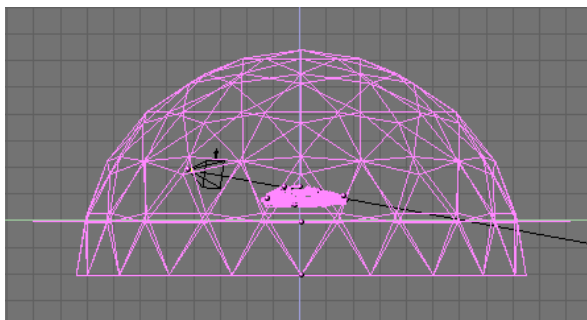
En este momento hemos creado todo lo que necesitamos para nuestra escena. El siguiente paso que haremos tiene que ser alterar la cúpula y el plano de "doble cara" a "una cara". Para conseguir esto, seleccionaremos la malla de la cúpula y volveremos a los Botones de Edición (F9). Haga click en el boton *Double Sided* (doble cara) y desactívelo. (*Configurando la cúpula y el plano como de 'una cara'*). Repita este proceso para el plano.



Configurando la cúpula y el plano como de 'una cara'.

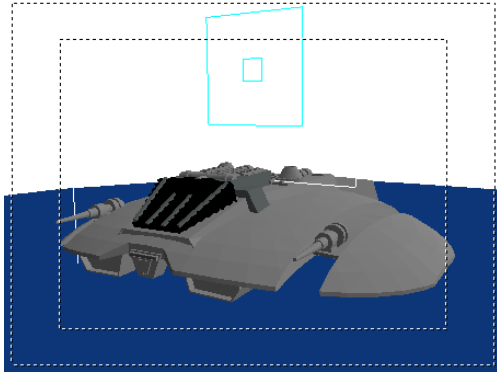
La solución de la radiosidad

Los siguientes pasos son el corazon y alma de la iluminación global. Vaya a la vista lateral con NUM 3 y use AKEY para seleccionar todas las mallas de nuestra escena. Ahora haga doble click en la cámara mientras mantiene SHIFT pulsado. No queremos que esté seleccionada. Debería parecerse a algo similar a *Seleccionando todas las mallas.*



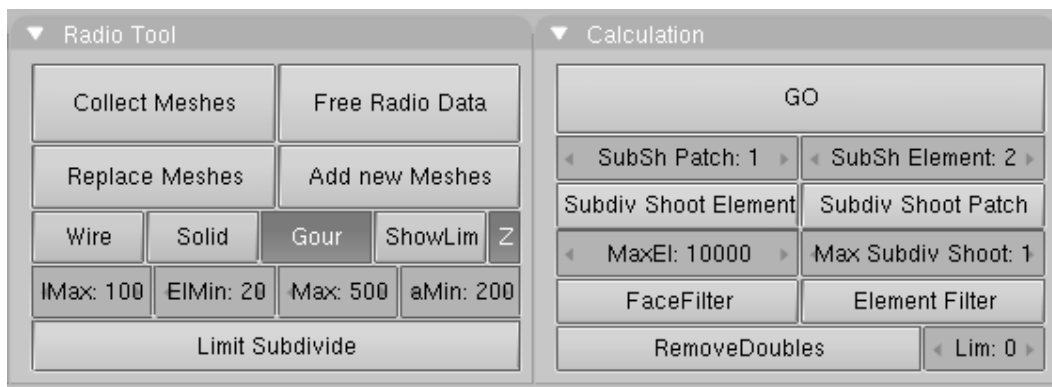
Seleccionando todas las mallas.

Después de seleccionar las mallas, vaya a la vista de cámara con NUM 0 y active el modo sombreado con **ZKEY** ahora podemos ver dentro de la cúpula. Seleccione el menu sombreado (**F5**) y el submeno de botones radiosidad [Image:Manual-Part-XII-RadButton.png]). En el panel *Radio Tool*, haga click sobre el botón *Collect Meshes*. Debería darse cuenta de un cambio de colores en la vista. Debería ser algo similar a *Preparando la solución de radiosidad*.



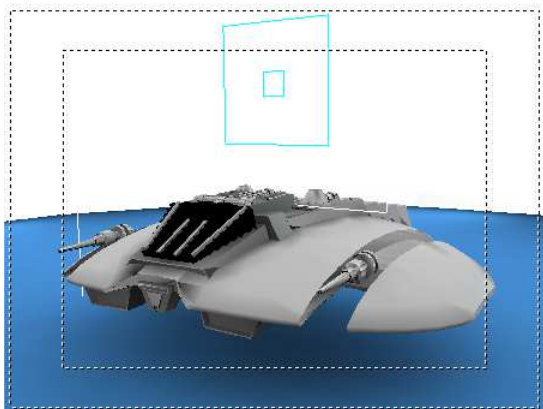
Preparando la solución de radiosidad.

Ahora, para mantener el suavizado de la nave de asalto como la malla original, cambiaremos de *Solid* a *Gour*. Esto devolverá a nuestra nave sus bellas curvas de nuevo, sobre lo mismo *Set Smooth* debería estar en los botones de Edición. También necesitará cambiar el valor de *Max Subdiv Shoot* a 1 (*Propiedades de radiosidad*). ¡No olvide este paso!



Propiedades de radiosidad.

Después de que haya seleccionado *Gour* y *Max Subdiv Shoot*, haga click en *Go* y espere. Blender empezará a calcular la parte emisora de la cúpula, cara por cara, de este modo "soluciona" el render. Mientras lo hace, verá el cambio en la escena como mas y mas luz se va añadiendo a la escena y las mallas están cambiadas. También se dará cuenta de que el cursor en Blender cambia a un contador como si fuera una animación. Deje que Blender trabaje, resolviendo el problema de Radiosidad. Dejando a Blender la cuenta en algún punto entre 50 y 500, dependiendo de la escenal, puede estar hecho en la mayoría de los casos. El tiempo de resolución depende de cuanto tiempo quiere permitirle trabajar... recuerde que puede presionar **ESC** en cualquier momento para parar el proceso. Éste es un área con el que se puede expiermentar para obtener diferentes resultados. Puede tomar desde 5 hasta 10 minutos y la velocidad de su sistema también determinará en gran meidda cuanto tiempo tomará el proceso. *Solución de radiosidad*. Es nuestra nave después de 100 iteraciones.

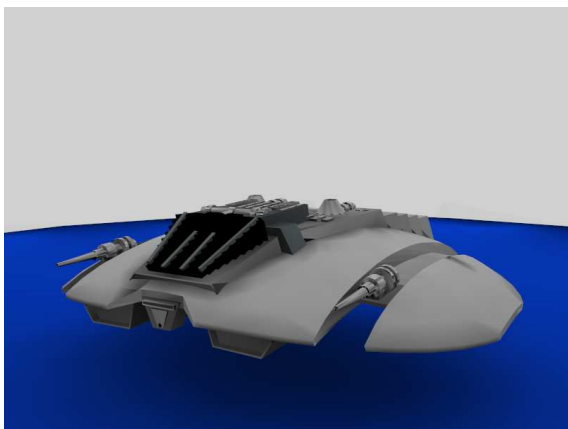


Solución de radiosidad.

Después de presionar la tecla **ESC** y parar la resolución, haga click en *Replace Meshes* (reemplazar mallas) (o *Add New Meshes* añadir nueva malla) y después *Free Radio Data*. Esto finaliza la resolución y reemplaza la escena anterior por la nueva escena de resolución Radiosidad.

Nota: Añadir en lugar de reemplazar las mallas es una forma de deshacer cambios. Usted mantiene las mallas antiguas y puede volver a ejecutar radiosidad de nuevo. Pero tiene que mover las mallas nuevas a una nueva capa y ocultar las viejas antes de renderizar.

Ahora estamos preparados para pulsar **F12** y renderizar (*Render de la solución de radiosidad.*).

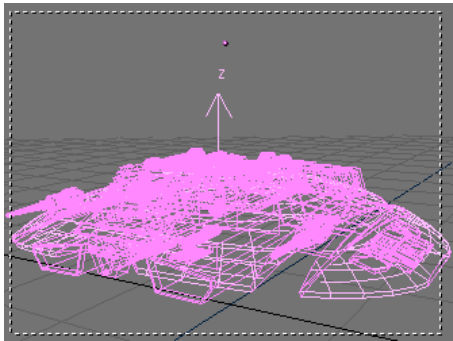


Render de la solución de radiosidad.

Texturizado

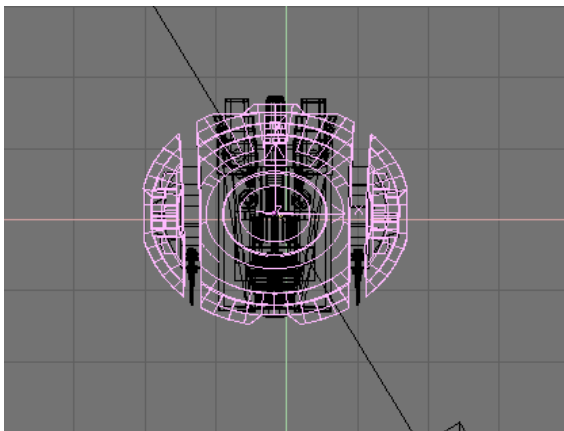
¡Allá vamos amigos! Ahora tiene un render muy limpio con una luz tenue de 360 grados usando radiosidad. Muy bonito... pero lo siguiente que tenemos que hacer es añadir texturas a la malla. Así que volvamos al área de la pantalla principal. Ahora intente seleccionar su malla. Se dará cuenta de que no solo se selecciona la nave, sino también el plano y la cúpula. Esto es debido a que la radiosidad ha creado una nueva malla única durante el proceso de resolución. Para añadir la textura, solo necesitamos la nave. Así, seleccione la malla y vaya al

Modo Edición. En este modo, podemos eliminar la cúpula y el plano ya que no les vamos a volver a necesitar. Puede usar **LKEY** para seleccionar los vértices adecuados y presionar **XKEY** para eliminarlos. Siga seleccionando y eliminando hasta que quede solo la nave. Debería aparecer algo como *La malla de la nave*.. Si quisieramos renderizar ahora, con **F12**, solo obtendríamos un fondo negro y nuestra nave. Es bonito... pero de nuevo, ¡queremos texturas!



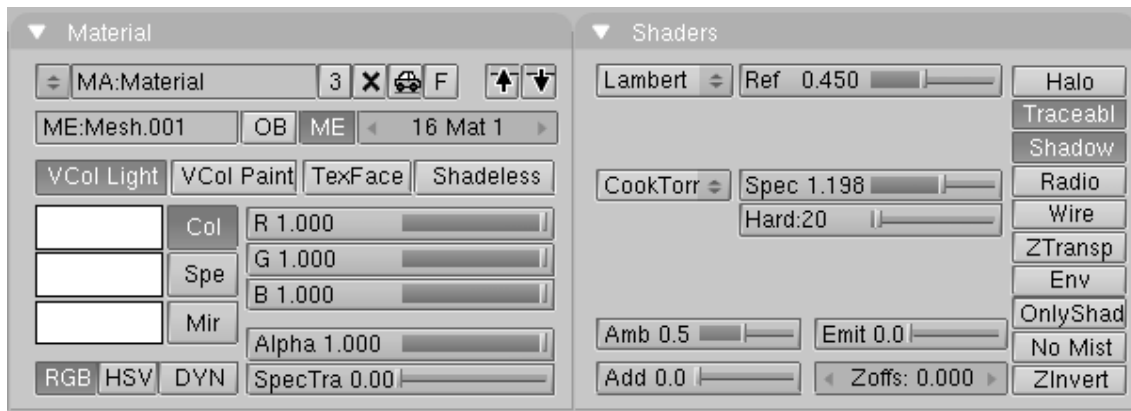
La malla de la nave.

Para añadir texturas a la malla, debemos separar las áreas sobre las que vamos a aplicar materiales y texturas. Para la nave, queremos añadir texturas a las alas y una sección media. Para hacerlo seleccione la malla de la nave, y vuelva al Modo Edición. Seleccione un vértice cerca del final del alaa y presione **LKEY** para seleccionar los vértices enlazados. Haga lo mismo con el otro lado. Ahora seleccione las áreas vistas en *Separando las partes de la nave que serán texturizadas*.. Cuando lo tenga, presione **PKEY** para separar los vértices seleccionados.



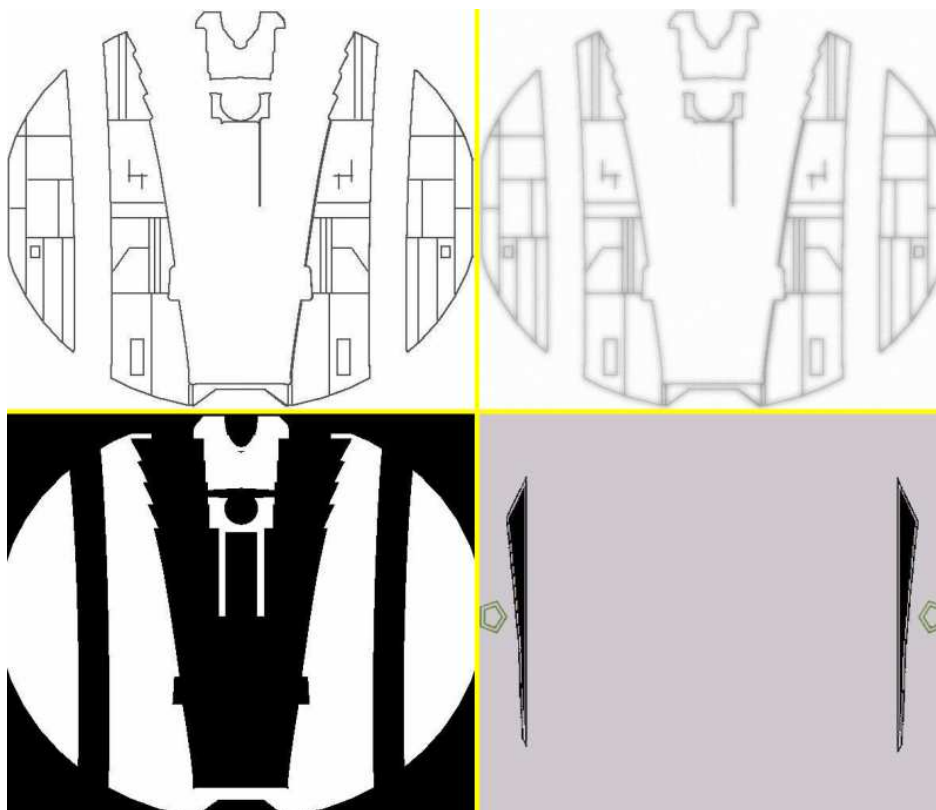
Separando las partes de la nave que serán texturizadas.

Ahora tenemos nuestra sección del ala separada y estamos preparados para añadir materiales y texturas. Queremos crear un nuevo material para esta malla. Para conseguir un resultado de aspecto metálico, podemos usar las propiedades que puede ver en *Material "Metálico"*..



Material "Metálico".

Es el momento de añadir las texturas. Queremos conseguir algunos resultados bien elaborados. Necesitaremos dos mapas de relieve para crear ranuras y dos máscaras para pintar y 'calcar'. Hay, de este modo, cuatro texturas para las alas de la nave que tienen que ser creadas, como se puede ver en *Cuatro texturas, desde la esquina superior izquierda, en el sentido de las agujas del reloj: RaiderBM, RaiderDI, Markings, Raider..*



Cuatro texturas, desde la esquina superior izquierda, en el sentido de las agujas del reloj: RaiderBM, RaiderDI, Markings, Raider.

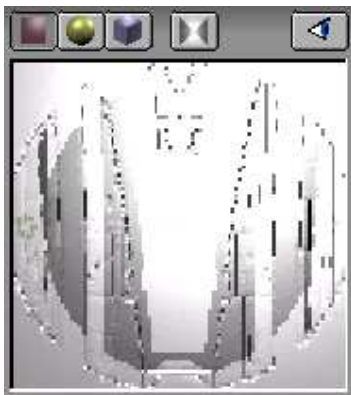
Las texturas deberían ir situadas en cuatro canales de material en la malla global de la nave. 'RaiderBM' y 'RaiderDI' deberían ponerse en negativo *NOR* (*Configuración de texturas*. abajo - clic dos veces y *NOR* se volverá amarillo). 'Raider' debería ponerse en negativo el *Ref* (*Configuración de texturas*. medio).

¿Qué material?: Una malla importada de una resolución de radiosidad tiene normalmente más de un material sobre ella. Es importante operar en el material "original" correcto.



Configuración de texturas.

El resultado es el plateado metálico deseado para la cabina de la nave. Finalmente la cuarta textura, 'Markings', se sitúa en *Col* en los botones de material (*Configuración de texturas*. arriba). Esto dará a la nave su propia marcas e insignias. Nuestra nave es totalmente plana, así que la proyección *Flat* (Plana) es adecuada. Cuando sea una forma más compleja será necesario algo de mapeado UV para conseguir buenos resultados. La previsualización del material para la malla debería ser algo como *Previsualización completa de material*..



Previsualización completa de material.

Nuestras texturas *no se verán* en la renderización por ahora (excepto markings) porque los tipos de textura "Nor" y "Ref" reaccionan a la luz, ¡y no hay fuente de iluminación en la escena! De este modo, ahora necesitamos añadir una lámpara o dos, teniendo en mente que nuestra nave ya está bien iluminada por la solución de radiosidad, de modo que la energía de las lámparas debe ser muy débil. Una vez tenga sus lámparas, intente un test de render. Experimente con las lámparas hasta obtener el resultado deseado. El render final (*Cylon Raider renderizado con radiosidad*.) muestra una bonita y bien iluminada nave de asfalto con texturizado suave.

PARTE XIII - HERRAMIENTAS ESPECIALES DE MODELADO

Capítulos

- Partículas
- Interacción de partículas
- Build and Waves
- DupliVerts
- DupliFrames

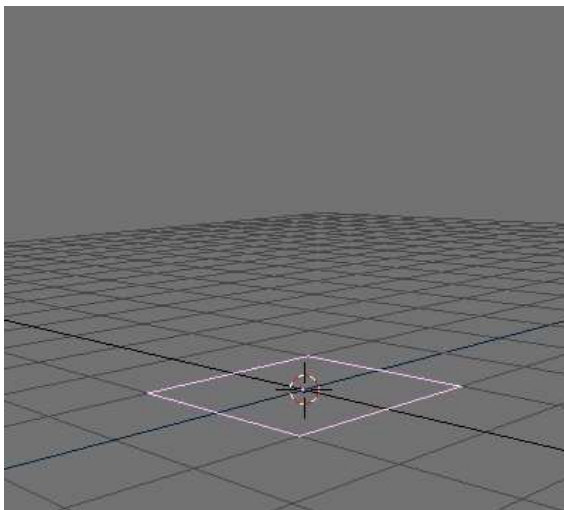
Partículas

Partículas sencillas


El sistema de partículas de Blender es rápido, flexible y potente. Cada Objeto de Malla puede servir como emisor de partículas. Los Halos pueden ser usados como partículas y con la opción DupliVert, también los objetos. Estos objetos con DupliVert pueden ser de cualquier tipo de Blender, por ejemplo, Mallas, Curvas Metaballs, o incluso focos. Las partículas pueden ser influenciadas por fuerzas globales para simular efectos físicos, como gravedad o viento. Con estas posibilidades se pueden generar humo, fuego, explosiones, fuegos artificiales o incluso bandadas de pájaros. Con las partículas estáticas se puede generar pelo, pieles, hierba o plantas.

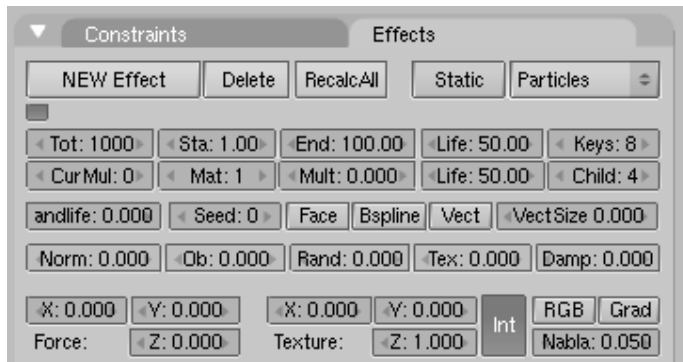
El primer sistema de partículas

Reinicia Blender a la escena por defecto, o crea una escena con un solo plano desde la vista superior. Este plano será nuestro emisor de partículas. Rota la vista hasta que tengas una buena visión del plano y el espacio sobre él. (**El emisor.**).



El emisor.

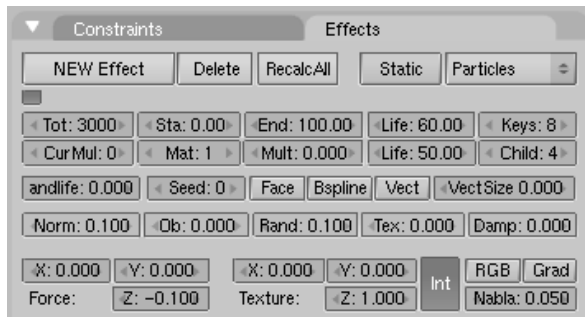
Cambia al panel **Effects** en el Object Context (F7 o ) y haz click en el botón **NEW Effect** en la mitad del Panel. Cambia la lista desplegable de MenuButton desde **Build** a **Particles**. Ahora se mostrarán los botones de partículas. (*Botones de partículas.*).



Botones de partículas.

En el NumButton **Norm:** ponga 0.100 con un click en la parte derecha del botón o use **SHIFT-LMB** para introducir el valor por teclado. Reprodzca la animación presionando **ALT-A** con el ratón sobre la ventana 3D. Verá una cadena de partículas ascendiendo verticalmente desde los cuatro vértices. Felicidades - ¡Acaba de crear su primer sistema de partículas en unos sencillos pasos!. Para crear un sistema un poco mas interesante, es necesario adentrarse mas en el sistema y sus botones (*Propiedades de partículas.*):

- El parámetro **Tot:** controla la cantidad total de partículas. En CPUs nuevas y rápidas podrá ampliar este número sin notar una disminución del rendimiento.
- El número total de partículas indicadas en el botón **Tot:** se crean uniformemente a lo largo de un intervalo de tiempo. Dicho intervalo está definido por los NumButtons **Sta:** y **End:**, que controlan el frame de inicio y el final en el que las partículas son generadas.
- Las partículas tienen un tiempo de vida, duran un número de frames, desde que se crean hasta que desaparecen. Puede cambiar el tiempo de vida de las partículas con el NumButton **Life:**.
- El NumButton **Norm:** es usado para dar a las partículas una velocidad inicial con un valor constante (0.1), direccionado a lo largo de las normales del vértice. Para hacer ésto más "aleatorio" puede modificar el NumButton **Rand:** hasta 0.1 también. Ésto también hace que las partículas empiecen con una variación de velocidad aleatoria.
- Use el grupo de NumButtons **Force:** para simular una fuerza constante, como viento o gravedad. Un valor de **Force: Z:** de -0.1 hará que las partículas caigan al suelo, por ejemplo.

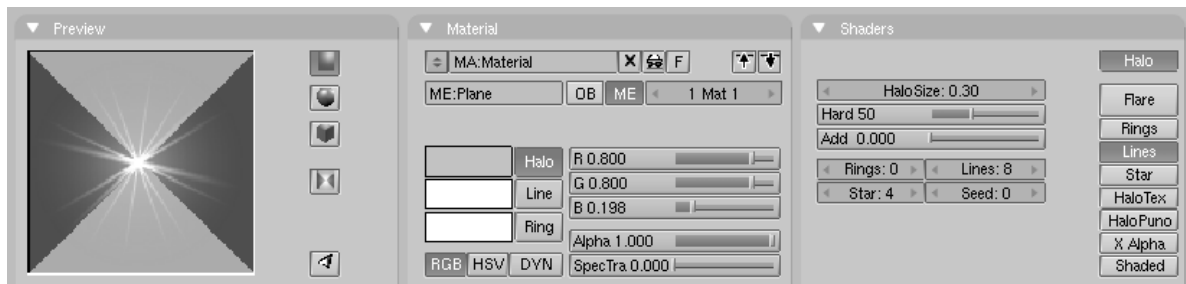


Propiedades de partículas.

Esto debería ser suficiente para que empezara, pero no tenga miedo de manipular algunos de los otros parámetros mientras está experimentando. Los cubriremos en detalle en secciones posteriores.

Renderizando un sistema de partículas

Es posible que haya intentado a renderizar una imagen con el ejemplo de antes. Si la cámara estaba correctamente alineada, habrá visto una imagen negra con unos puntos grisáceos en ella. Éste es el Halo-material estandar que Blender asigna a un sistema de partículas creado nuevo. Posicione la cámara de forma que obtenga una buena visión del sistema de partículas. Si desea añadir un ambiente sencillo, recuerde añadir algunas luces. Los Halos son renderizados sin luces, mientras no se hayan indicado, pero otros objetos necesitan luces para ser visibles. Vaya a los Botones de Materiales (F5) y añada un nuevo material para el emisor si no lo ha hecho antes. Haga click en el Botón "Halo" en el medio de la tabla (*Propiedades del Halo.*).



Propiedades del Halo.

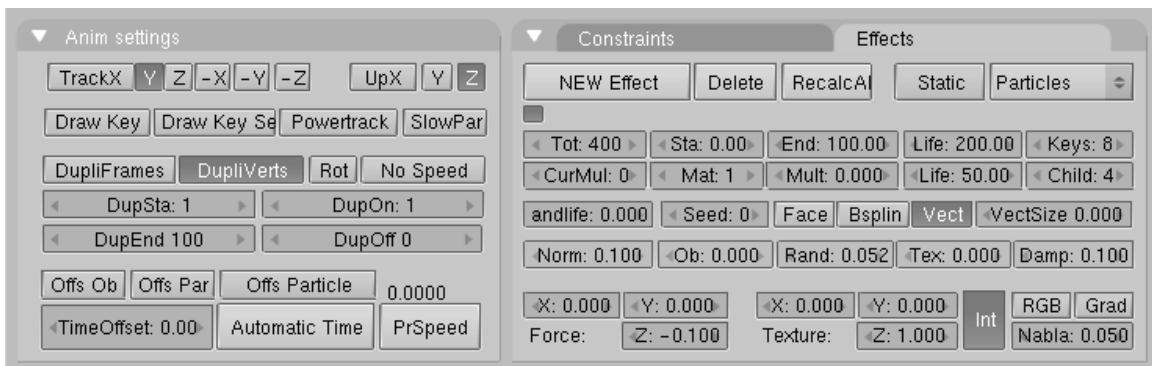
Los Botones de Material cambian a los Botones de Halo. Elija *Line*, y ajuste *Lines*: a un valor de su elección (puede ver el efecto directamente en en Material-Preview). Disminuya *HaloSize*: a 0.30, y elija un color para el Halo y para las líneas (*Propiedades del Halo.*). Ahora puede renderizar una imagen con F12, o una animación completa y ver miles de estrellas volando alrededor (*Estrellas disparándose.*).



Estrellas disparándose.

Objetos como partículas

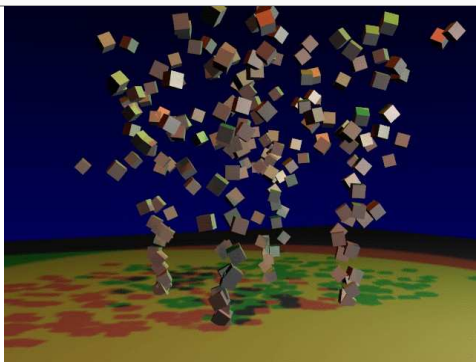
Es muy fácil crear objetos como partículas, es exactamente la misma técnica descrita en *modelado_especial_con_dupliverts*. Empiece creando un cubo, o cualquier otro objeto que desee, en su escena. Merece la pena pensar sobre la potencia de su ordenador, dado que vamos a tener tantos objetos como *Tot*: indique, en la escena. ¡Ésto significa tener tantos vértices como el objeto elegido por el valor de *Tot*! Escale el nuevo objeto disminuyendolo hasta que encaje con la escala general de la escena. Ahora seleccione el objeto, y haga **SHIFT-RMB** con el emisor y hagalo padre del cubo usando **CTRL-P**. Elija el emisor solo y seleccione la opción "DupliVerts" en el Panel *Anim Settings* del Object Context (**F7**). Los cubos con la opción "DupliVert" aparecerán inmediatamente en la ventana 3D.



Colocando las partículas con DupliVert.

Usted podría querer reducir el número de partículas presionando **ALT-A** (*Colocando las partículas con DupliVert.*). En la animación se dará cuenta que todos los cubos comparten la misma orientación. Esto puede ser interesante, pero también puede serlo tener los cubos orientados aleatoriamente. Ésto lo podemos hacer seleccionando la opción *Vect* en los parámetros de partícula, que causa que los dupli-objetos sigan la rotación de las partículas, resultando un movimiento más natural (*Colocando las partículas con DupliVert.*). Podemos ver un frame de la animación (*Renderizado de partículas con DupliVert.*).

Objeto original: Recuerde mover el objeto original fuera de la vista de cámara, ya que al contrario que una Malla de DupliVerts normal, las partículas con DupliVert también serán renderizadas.

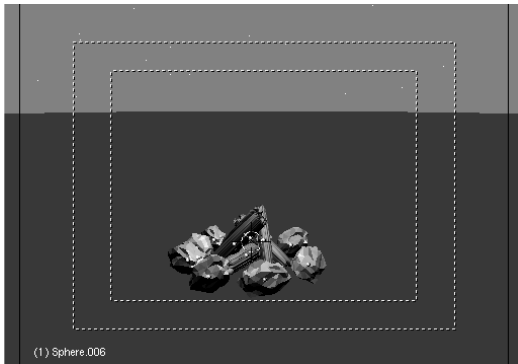


Renderizado de partículas con DupliVert.

Haciendo fuego con las partículas

El sistema de partículas de Blender es muy útil para hacer fuego y humo realista. Esto podría ser una vela, una hoguera, o una casa ardiendo. Es interesante conocer cómo el fuego esta condicionado por la física. Las llamas de un fuego son gases calientes. Ascenden porque su densidad es menos que la del aire frío circundante. Las llamas son calientes y brillan en el medio, y se van haciendo mas oscuras hacia su perímetro.

Prepare un escenario sencillo para nuestro fuego, con algunos trozos de madera, y algunas piedras. (*Configuración de la hoguera.*)

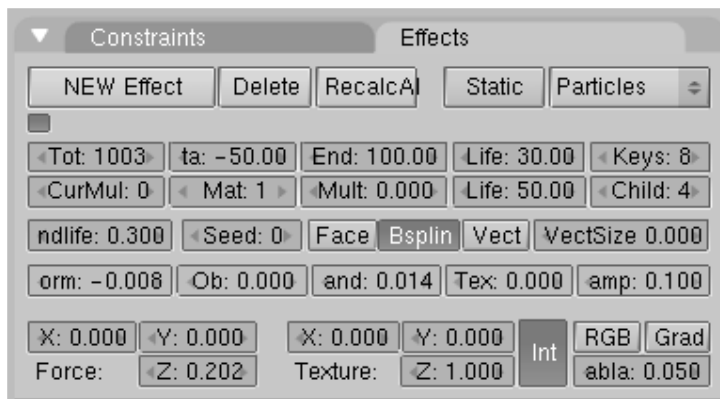


Configuración de la hoguera.

El sistema de partículas

Añada un plano en el medio del círculo de piedras. Este plano será nuestro emisor de partículas. Subdivida el plano una vez. Ahora podrá mover los vértices a una posición en la madera donde las llamas (partículas) se deberían originar.

Ahora vaya al Object Context F7 y añada un nuevo efecto de partículas al plano. Los números aquí indicados (*Configuración de las partículas de fuego.*) deberían valer para un fuego realista, pero alguna puede ser necesaria alguna modificación, dependiendo del tamaño del emisor actual.



Configuración de las partículas de fuego.

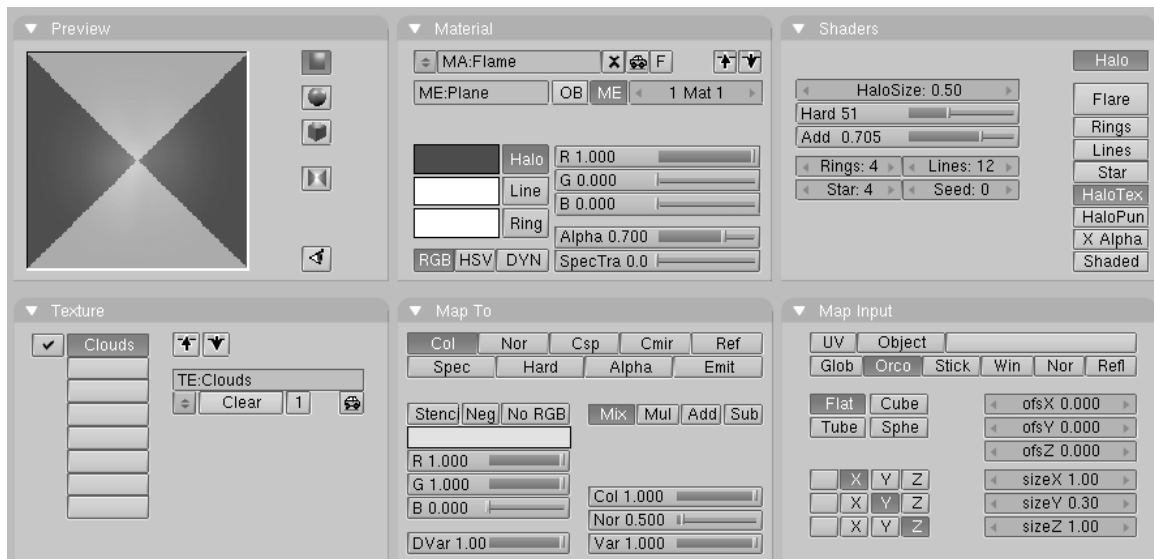
Algunas notas:

- Para tener el fuego ardiendo desde el principio de la animación haga *Sta*: negativo. Por ejemplo, intente -50. El valor de *End*: debería reflejar la longitud deseada para la animación.
- El valor de las partículas *Life*: es 30. Normalmente podría estar sobre 50 por ahora. Usaremos este parámetro después cuando ajustemos la altura de las llamas.
- Haga el parámetro *Norm*: un poco negativo como por ejemplo (-0.008) que resultará un fuego mas potente en su base.
- Use *Force*: Z: alrededor de 0.200. Si su fuego parece demasiado lento, ajuste este parámetro.
- Cambie *Damp*: a 0.100 para frenar las llamas al cabo de un rato.
- Active el botón *Bspline*. Esto usará un método de interpolación que dará mucho más movimiento de fluido.
- Para añadir algo de aleatoriedad a nuestras partículas, ajuste el parámetro *Rand*: alrededor de 0.014. Use el parámetro *Randlife*: para añadir aleatoriedad e la vida de las partículas; un valor muy alto nos dará una llama muy viva.
- Use alrededor de 600-1000 partículas en total para la animación (*Tot*:).

El la ventana 3D, obetendrá una primera impresión de como de realistas quedará el movimiento de las llamas. Pero lo más importante para nuestro fuego será el material.

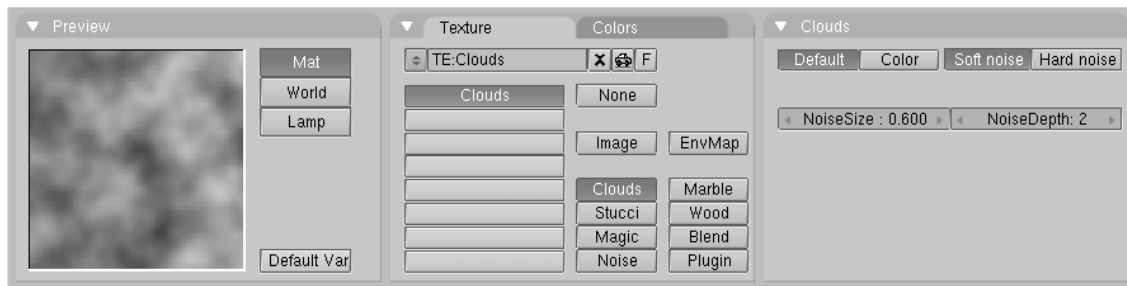
El material de fuego

Con el emisor de partículas seleccionado, vaya al panel de sombreado **F5** y añada un nuevo Material. Haga este nuevo material de tipo Halo activando el botón *Halo*. Active también *HaloTex*, situado debajo de dicho botón. Ésto nos permitirá usar una textura más adelante.



Material de Llamas.

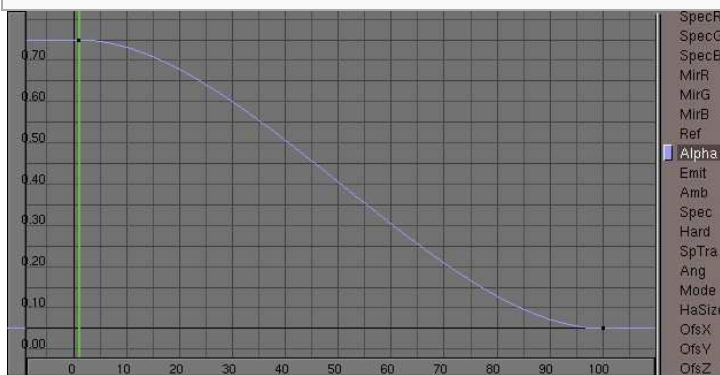
Asigne al material un color rojo completamente saturado con los deslizadores RGB (RGB-sliders). Disminuya el valor Aplha a 0.700; ésto hará las llamas un poco transparentes. Aumente el deslizador Añadir (*Add*) hasta 0.700, asi los Halos potenciarán a los demás, dando un brillo interior a las llamas, y mas oscuro en exterior. (*Material de Llamas*:).



Textura de las llamas.

Si ahora hace un render de prueba, verá que solo puede ver una brillante llama roja. Para darle un toque más realista, necesitaremos una textura. Mientras el emisor siga seleccionado, vaya al Panel de Texturas (Texture Panel) y añada una nueva textura del tipo Nube (*Cloud*), en los botones de textura (F6). Ajuste el Tamaño de Ruido (*NoiseSize:*) a 0.600. (*Textura de las llamas*). Vuelva a los botones de Material F5 y haga el color de textura a un amarillo con los deslizadores RGB en el lado derecho de los botones de material. Para ajustar los puntos amarillos en la textura de nube, disminuya el valor de Tamaño Y (*SizeY*) a 0.30. Un render de prueba ahora nos mostrará un bonito fuego. Pero seguimos necesitando crear partículas apagándose en lo alto del fuego. Podemos conseguir esto con una animación del *Alpha* y del *Tamaño del Halo* del material. Asegurese que su animación está en el frame 1 (**SHIFT-LEFTARROW**) y mueva el ratón hasta la ventana de materiales. Ahora presione la tecla **IKEY** y elija *Alpha* del menú emergente. Avance el barra de tiempos hasta el frame 100, e introduzca un valor *Alpha* de 0.0 e inserte otra clave para el *Alpha* con la tecla **IKEY**. Cambie una de las ventanas a una ventana IPO. Active el tipo de material IPO haciendo click en la sección del menú correspondiente. Ahora verá una curva del canal *Alpha* del material (*Material IPO del fuego*).

Nota: Una animación para un material de partículas siempre está mapeada desde los primeros 100 frames de animación hasta el período de vida de la partícula. Esto significa que cuando desvanecemos un material en un frame del 1 al 100, una partícula con un período de vida de 50, se desvanecerá en ese tiempo.



Material IPO del fuego.

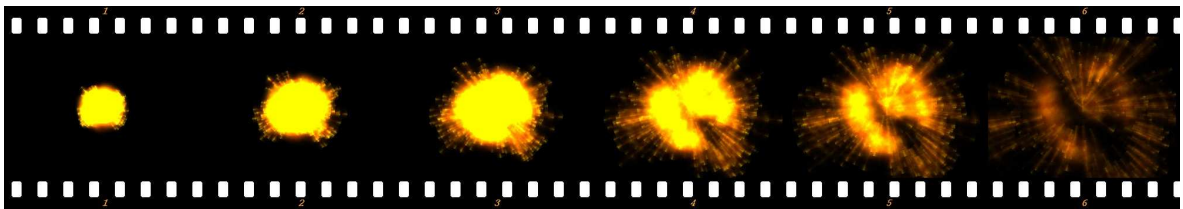
Ahora puede renderizar la animación. Quizá deba refinar algunos parámetros como el período de vida de las partículas. Puede añadir un mayor realismo a la escena animando las luces (o usando luces direccionales con sombra) y añadiendo un sistema de partículas de chispas al fuego. También es recomendado animar el emisor para obtener llamas mucho más vivas, o usar más de un emisor (*Render Final.*).



Render Final.

Una simple explosión

Esta explosión está diseñada para ser usada como textura animada, para insertarla en la escena actual o para ser usada como textura animada. Para un render estático, o una cámara lenta de una explosión, es posible que necesitemos hacer un poco más de trabajo para hacer que se vea realmente bien. Pero tenga en cuenta, que nuestra explosión sólo será vista durante medio segundo (*La explosión.*).

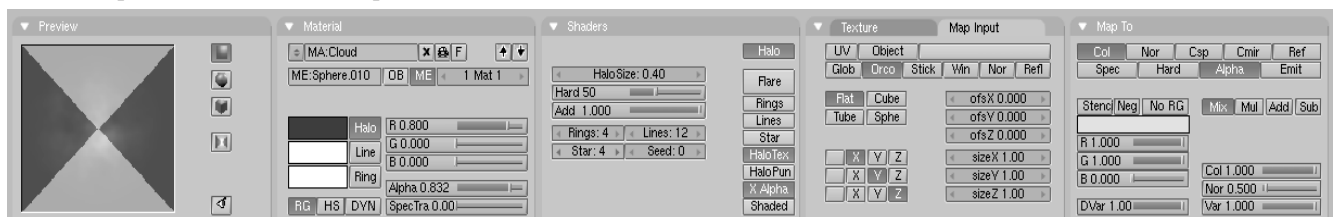


La explosión.

Como emisor para la explosión he elegido una IcoEsfera (IcoSphere). Para hacer la explosión un tanto irregular, he eliminado patrones de vertices con la función de selección circular en el modo Edición (Edit Mode). Para una escena concreta podría ser mejor usar un objeto como emisor, que esté formado de forma diferente, por ejemplo, el objeto actual que se quiera explotar. Mi explosión está compuesta por dos sistemas de partículas, uno para la nube de gases calientes, y otro para las chispas. He cogido una versión girada del emisor para generar las chispas. Además, he animado la rotación de los emisores mientras las partículas son generadas.

Los materiales

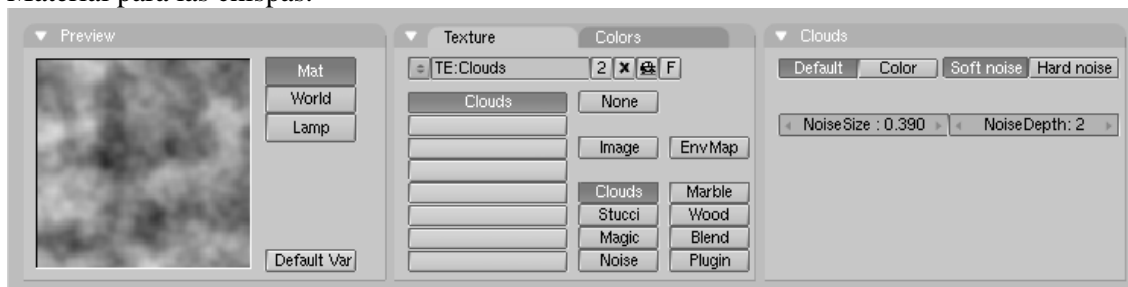
Las partículas para la explosión son de un material Halo muy claro, con una textura de nube para añadir aleatoriedad, las chispas también tienen un material similar, mire desde *Material para la nube de la explosión.* hasta *Textura de ambos.*



Material para la nube de la explosión.



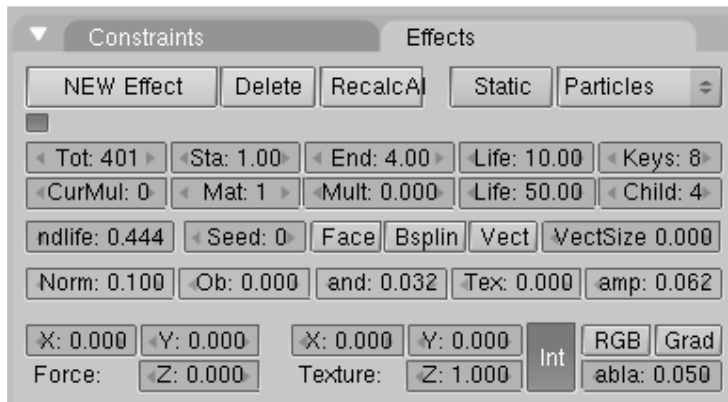
Material para las chispas.



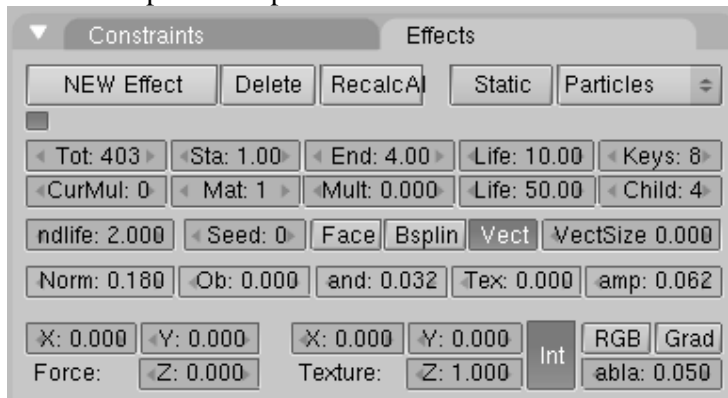
Textura de ambos.

Anime el valor Alpha del de las partículas del Halo de 1.0 a 0.0 en los primeros 100 frames. Esto hará que sean mapeadas al período de vida de las partículas, como es normal. Dese cuenta de la propiedad *Star* en el material de las chispas (Figure 20-17). Ésto modela un poco las chispas. También podríamos haber usado una textura especial para conseguir esto, de todas maneras, en este caso, la manera más fácil es usando la propiedad *Star*.

Los sistemas de partículas



Sistema de partículas para las nubes.

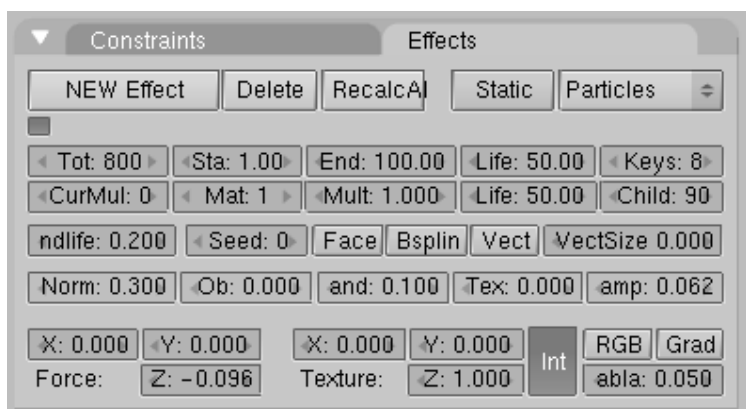


Sistema de partículas para las chispas.

Como puede ver en *Sistema de partículas para las nubes*, y en *Sistema de partículas para las chispas*, los parámetros son básicamente los mismos. La diferencia está en la propiedad Vect para las chispas, y una mayor cantidad en *Norm*: para provocar mayor velocidad a dichas chispas. También he situado el valor de *Randlife*: para las chispas a 2.000 dando una forma irregular. Le sugiero que empiece a experimentar, usando estos parámetros como inicio. Los valores a utilizar dependen de lo que quiera conseguir. Intente añadir más emisores para obtener escombros, humo, etc.

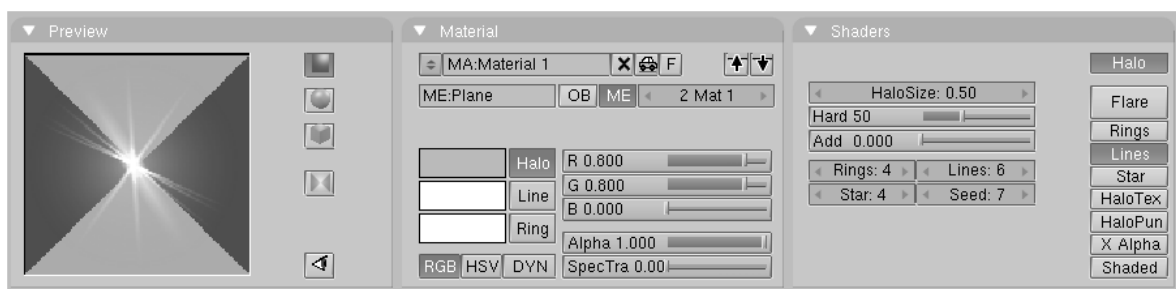
Fuegos Artificiales

Un botón que todavía no hemos usado es el *Mult*:. Toda la tercera línea de botones del panel está relacionada con ésto. Cree un plano y añada un sistema de partículas al mismo. Ajuste los parámetros de forma que obtenga algunas partículas volando en el cielo, y aumente el valor de *Mult*: a 1.0. Ésto provocará que el 100% de las partículas generen partículas hijas cuando su período de vida acabe. Así, cada partícula generará cuatro hijos. Entonces necesitamos aumentar el valor Hijos (*Child*:) a alrededor de 90 (Figure 20-21). Ahora debería ver un fuego artificial convincente hecho con partículas cuando previsualice la escena con ALT-A.



Botones de multiplicación de partículas

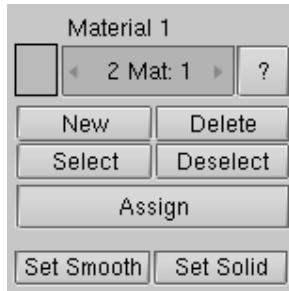
Cuando renderice los fuegos artificiales, no parecerán muy impresionantes. Ésto es debido al material estandar de Halo asignado por Blender. Por lo tanto el siguiente paso es asignar un mejor material. Asegurese que tiene el emisor seleccionado y vaya al panel de sombreado (Shading Context) y a los botones de material (F5). Añada un nuevo material con Botón de Menú, y seleccione el tipo *Halo*.



Material de los fuegos artificiales 1.

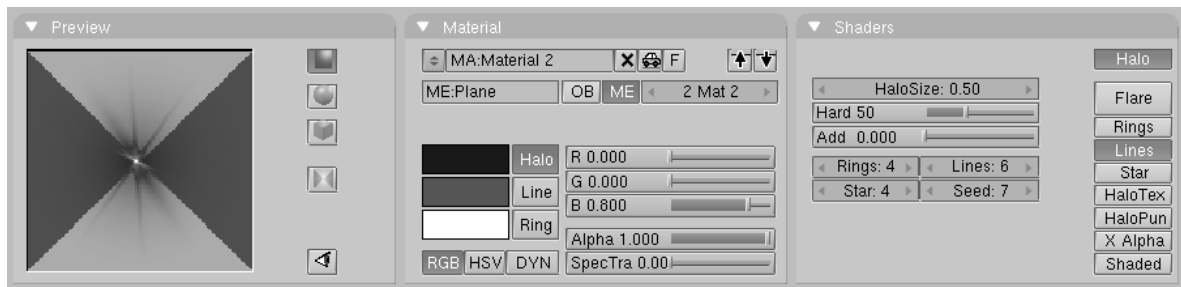
He usado un material halo muy claro; puede ver los parámetros en *Material de los fuegos artificiales 1.* La animación renderizada ahora se verá mucho mejor, pero aún queda algo que podemos hacer.

Mientras el emisor esté seleccionado, vaya al panel Edición (Editing Context) **F9** y añada un nuevo índice de material haciendo click en el botón *New* en el panel Enlace y materiales (*Link and Materials*) (*Añadiendo un segundo material al emisor.*).



Añadiendo un segundo material al emisor.

Ahora vuelva al panel de sombreado (Shading Context). Verá que el panel de datos del material ha cambiado de color a azul. El botón llamado 2 indica que este material es usado por dos usuarios. Ahora haga click en el botón 2 y confirme la ventana emergente. Cambie el nombre del material a "Material 2" y cambie el color del halo y las líneas (*Material 2.*).



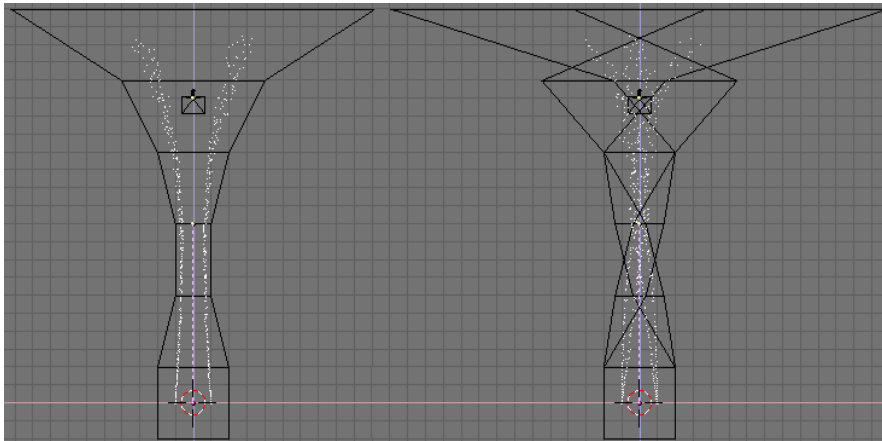
Material 2.

Vuelva a los parámetros de partículas y cambie el botón *Mat:* a "2". Renderice de nuevo y verá que la primera generación de partículas está usando el primer material y la segunda generación el segundo material. Mediante este método puede conseguir hasta 16 materiales para las partículas (Es el número máximo de índices para los materiales).

Más mejoras: Aparte de cambiar los materiales, también puede usar el los IPOs del material para animar las propiedades de cada uno de los materiales.

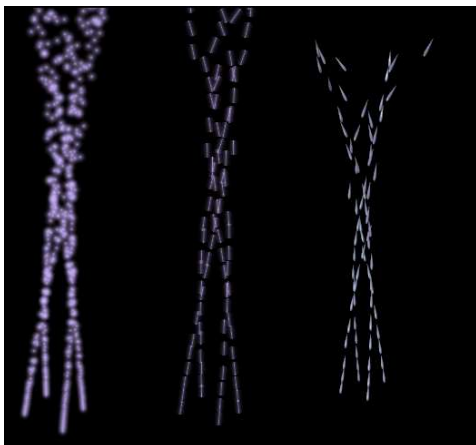
Controlando las partículas mediante celosía

El sistema de partículas de Blender es muy potente, y el camino de las partículas no solo estará determinado por las fuerzas sino canalizado por una celosía. Cree una simple maya cuadrada y añada un sistema de partículas en ella con una fuerza-z negativa y los parámetros generales así *Propiedades de partícula.*



Efectos de deformación de celosía.

El remolino es evidente, y por supuesto, puede conseguir mayores efectos rotando más la celosía, o usando una con más subdivisiones. Si le da al emisor un material Halo y renderiza verá algo como *Partículas normales, izquierda; Partículas vectoriales, centro; Y objetos con vertices duplicados siguiendo las partículas, derecha.* en la izquierda.

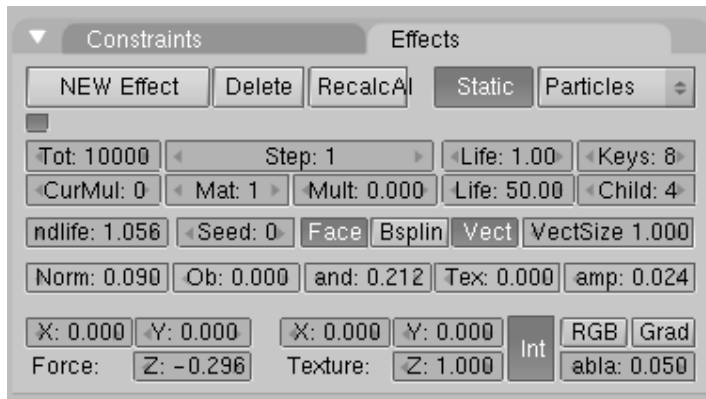


Partículas normales, izquierda; Partículas vectoriales, centro; Y objetos con vertices duplicados siguiendo las partículas, derecha.

Si selecciona el emisor, vaya a los botones de animación y presione el botón de partícula *Vect*. Las partículas cambiarán de ser puntos a segmentos, con una longitud y dirección proporcional a la velocidad de la partícula. Un renderizado debería dar un resultado como *Partículas normales, izquierda; Partículas vectoriales, centro; Y objetos con vertices duplicados siguiendo las partículas, derecha.* en el medio. Si ahora duplica los vertices (*DupliVert*) de un objeto al emisor, asociándolo (*Parent*) y haciendo click en el botón *DupliVert*, el objeto con vertices duplicados tendrá la misma orientación que el objeto original si las partículas son normales, pero será alineado con la dirección de las mismas si están seleccionadas como *vert*. Seleccionando el Objeto Original y jugando con los botones *Track* puede cambiar la orientación (*Partículas normales, izquierda; Partículas vectoriales, centro; Y objetos con vertices duplicados siguiendo las partículas, derecha.* a la derecha).

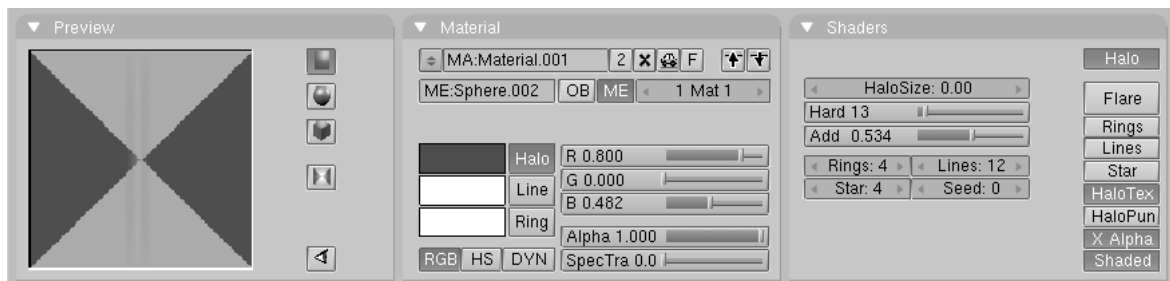
Partículas estáticas

Las partículas estáticas son muy útiles cuando creamos objetos como fibras, hierba, pelo y plantas. Intente crear un pequeño personaje, o simplemente una pelota, para probar las partículas estáticas. Intente por ejemplo, una pequeña 'bola de pelo'. El emisor no se renderiza, así que duplique la maya (o el objeto que quiera usar y conviértalo en una malla (ALT-C)). Podría ser buena idea subdividir la malla con un fractal para obtener algo de aleatoriedad en ella. Si termina con una malla demasiado densa, use "Eliminar duplicados (Remove Doubles)" con un límite elevado. Elimine las partes del círculo donde no quiera tener pelo. Ahora, asigne el sistema de partículas y cambie a la opción *Static*.



Propiedades de partículas estáticas.

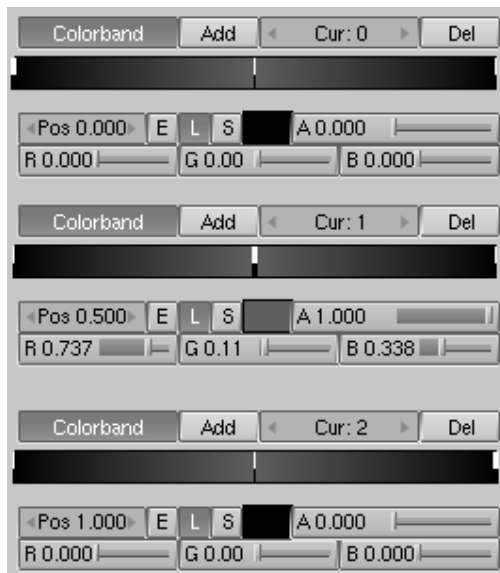
Use los parámetros que puede ver en *Propiedades de partículas estáticas*. Con la combinación de *Life* y *Norm* puede controlar la longitud del pelo. Use una fuerza en una dirección Z negativa para permitir que el pelo se combe. Active la propiedad *Face* para generar las partículas, no solo en los vértices, sino también distribuidas por las caras. Active también *Vect*; esto generará la fibra como partículas. La importancia del paso (*Step*) indica cuantas partículas por período de vida son generadas. Sitúelo en un valor pequeño para obtener curvas más suaves para las partículas, y asegúrese de no pasar por alto el valor de la propiedad *Rand*. Cuando renderice, verá que obtiene partículas muy difuminadas. El material usado para las partículas estáticas es muy importante, de modo que añada un material para el emisor en el panel de sombreado (Shading Context) (F5).



Propiedades de material.

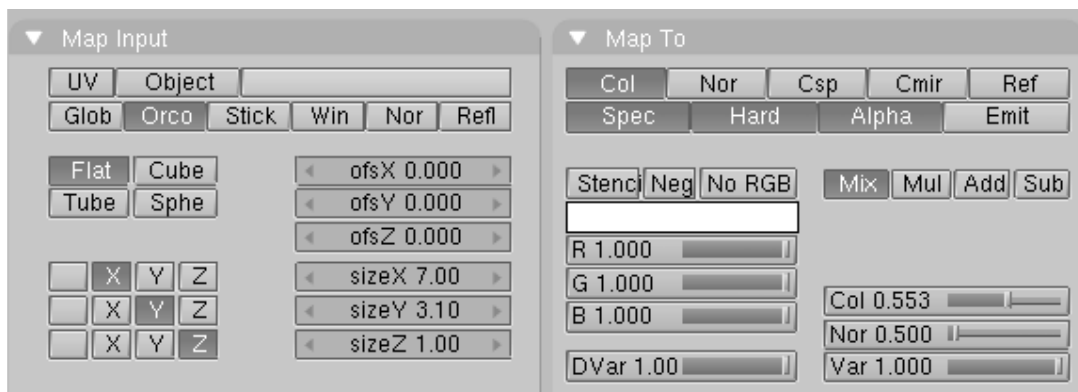
He usado un tamaño de Halo (*Halosize*) muy pequeño (0.001). En el botón numérico no se podrá ver así que para ajustarlo, haga click en el botón con **LMB** pulsado mientras mantiene **SHIFT** apretado. Active la opción sombreado (*Shaded*) para que las partículas se vean

influenciadas por las luces de la escena, después active *HaloTex*. Ahora vamos a usar una textura para sombrear los pelos (*Propiedades de material*.).



Propiedades de banda de colores de textura.

Vaya al subpanel de textura (Texture sub-context) (F6) y añada un nuevo tipo de textura *Blend*. Elija como subtipo *Lin*. Active la opción banda de color (*Colorband*) y ajuste los colores como se ve en *Propiedades de banda de colores de textura*.. Obtendrá un bonito degradado, desde transparente hasta púrpura, volviendo de nuevo a transparente.



Propiedades de textura en los botones de material.

Vuelva al subpanel de los botones de material y asegúrese que *Alpha* está activado en la salida del mapeado de textura (texture mapping output) a la derecha de los botones de material. Use *sizeX* y *sizeY* para definir la forma del halo en la previsualización del material como una peñea fibra (*Propiedades de textura en los botones de material*.). Si su pelo no es suficientemente denso, incremente el número de partículas con *Tot* o añada más emisores. También cambie los parámetros para los emisores adicionales para obtener algo de variación en los pelos. (*Resultado final*.).



Resultado final.

Interacción entre Partículas

Introducción

El sistema de partículas de Blender permite que las partículas interactúen de dos maneras - campos de fuerza (force fields) y desvío (deflection).

- Campos de Fuerza

Existen dos tipos de campos de fuerza:

- *Campo de Fuerza estándar* - Se comporta como un campo de fuerza gravitacional.
- *Campo de Vértice Vortex field* - Un campo de vértice produce un efecto parecido al de un tornado, haciendo que las partículas se muevan en forma de espiral alrededor del centro del vértice (vortex).
- *Viento* - Fuerza del viento.

Los campos de fuerza están disponibles para todos los tipos de objeto. Actualmente está disponible un campo basado en puntos (point-based field) con una pérdida (fall-off) esférica.

Los campos de fuerza pueden ser asignados a cualquier objeto. Las partículas serán entonces: atraídas si se usa un valor negativo de Fuerza (*Strength*), o repelidos si se utiliza un valor positivo de Fuerza (*Strength*). También hay un parámetro de Pérdida (*Fall-off*) que define cuanta fuerza se pierde de acuerdo a la distancia con respecto al origen del objeto.

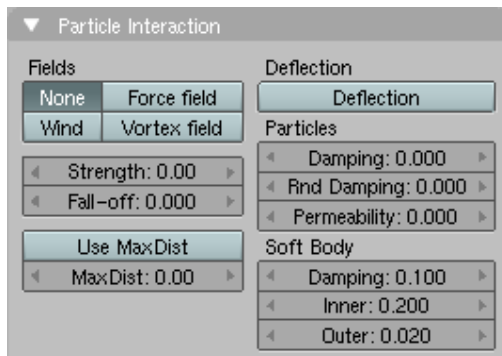
- Desvío (*Deflection*)

Permite configurar cualquier objeto de malla como un desviador (deflector) de partículas. Así, las partículas rebotarán sobre la superficie de la malla. Se puede controlar cuanto rebotan las partículas mediante el valor *Damping*, hacer aleatoria la manera en que las partículas rebotan con *Rnd Damping*, y se puede definir el porcentaje de partículas que pasarán a través de la malla, usando el parámetro Permeabilidad (*Permeability*).

Superficies normales: Hay que asegurarse que las normales de la superficie de la malla miran en dirección a las partículas si se quiere obtener un desvío correcto.

La interfaz

La configuración de la interacción de partículas se controla mediante el panel Interacción de Partículas (Particle Interaction), dentro del contexto Objeto (F7).



Panel Interacción de Partículas.

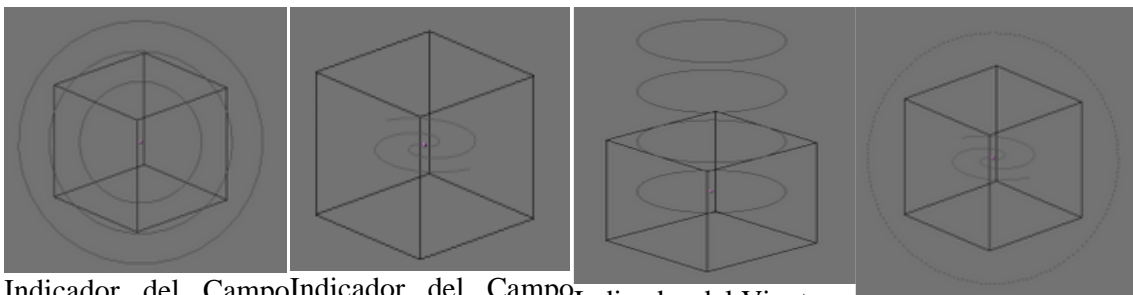
Campo de Fuerza

Campo de Fuerza/Vértice (Force/Vortex field)

- *Fuerza* (Strength) - Fuerza del efecto del campo.
- *Pérdida* (Fall-off) - Cuanto disminuye la fuerza con respecto a la distancia.

Cuando se agrega un campo de fuerza/vértice (force/vortex) a un objeto, el objeto mostrará un pequeño gráfico indicando que existe una interacción de partículas conectada al objeto.

- Los campos de fuerza mostrarán unos pequeños círculos grises.
- Los campos de vértice mostrarán el dibujo de una espiral.
- Las fuerzas del viento mostrarán una serie de círculos indicando la dirección y rango de la fuerza.
- Hay también un indicador de rango para el parámetro *MaxDist* que muestra un círculo alrededor del campo de fuerza del objeto.



Indicador del Campo de Fuerza.

Indicador del Campo de Vértice.

Indicador del Viento.

Indicador de Máxima Distancia.

Desvío

En el lado derecho del panel *Particle Interaction* tiene la configuración para desvíos (deflection).

Desviadores: Los desviadores (deflectors) no muestran indicadores gráficos como lo hacen los campos de fuerza.

Parámetros de Desvío

- *Damping* - Controla la cantidad de rebote que produce la superficie.
0.0 - Sin damping, las partículas rebotarán al máximo.
1.0 - Máximo damping, las partículas no rebotarán.
- *Rnd Damping* - Agrega el elemento aleatorio al rebote. Por ejemplo, con un valor de *Damping* de 1.0 y un valor de $\{\text{Literal|Rnd Damping}\}$ de 0.5, el damping variará entre los valores 1.0 y 1.5.
- *Permeabilidad* (Permeability) - Porcentaje de partículas que atravesarán la malla.
0.0 - No pasa ninguna partícula.
1.0 - Todas las partículas pasan a través del desviador (deflector).

Si se configura un desviador de partículas, hay que asegurarse de tener suficientes cuadros clave para que Blender calcule las colisiones con suficiente detalle. Si se ven partículas moviéndose a través del desviador o rebotando de manera extraña, es probable que haya pocos cuadros clave, o las partículas o el desviador se está moviendo demasiado rápido.

Notas:

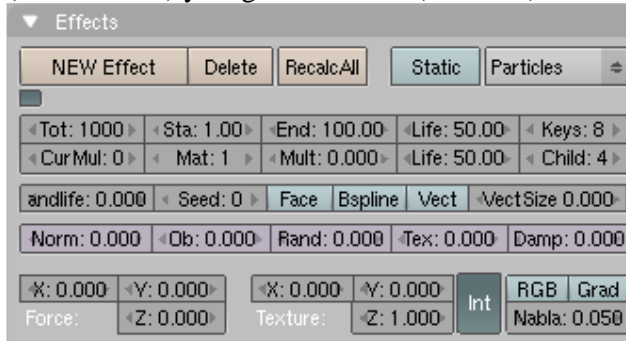
- Se pueden animar desviadores en movimiento, pero las partículas pueden atravesar la malla si el desviador se mueve demasiado rápido o si su malla es demasiado compleja. Esto puede resolverse parcialmente aumentando los Cuadros Clave (Keys) del emisor de partículas.
- Después de hacer cualquier cambio en los parámetros, es necesario seleccionar el emisor de partículas, volver a la pestaña *Efectos* (Effects) y presionar el botón *Recalca*. (Figure 20-37).
- Más cuadros claves (keys) significa un tiempo mayor de cálculo y un mayor uso de memoria. Véase Manual/PartXIII/Partículas para conocer cómo configurar los emisores de partículas.

Para todos los parámetros de los campos de fuerza y desviadores, excepto para el parámetro *MaxDist*, las claves Ipo pueden ser insertadas. Las curvas Ipo son editadas como tipos *Objeto Ipo* en la Ventana Ipo. Vea el Capítulo 14, Animación de Objetos no Deformados para más información sobre Animación e Ipo.

Ejemplo

Aquí hay un pequeño ejemplo que muestra el funcionamiento de los desviadores de partículas.

1. Comenzar con la escena por defecto, con el cubo.
2. En la vista de arriba (top), agregar una *Malla Círculo* (Circle) y aceptar los **32** vértices por defecto. Presionar **TAB** para salir del *Modo Edición*. Cambiar a la vista lateral y mover el círculo **5** unidades hacia arriba.
3. Con el círculo seleccionado, presionar **S**, y escalarlo a **0.1** en todas las direcciones.
4. Ahora, presionar **F7** e ir a la pestaña *Efectos* (Effects). Presionar *NUEVO Efecto* (NEW Effect) y elegir *Partículas* (Particles) de la Lista (Popup-list).

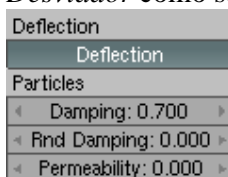


Effects panel.



Example frame 40.

5. Incrementar la vida de la partícula (particle life) de 50 a **100** cambiando el campo Vida (*Life*) a 100. En el campo de fuerza, en la sección más baja de la pestaña, asignar un valor de **-0.100** a *Force Z*. Si se presiona **ALT-A**, debería verse una animación de partículas moviéndose hacia abajo, en dirección al cubo. Ver: **Cuadro 40 del Ejemplo**. Hasta el momento, las partículas viajan en línea recta, atravesando el cubo.
6. Seleccione el cubo. Con el contexto *Efecto* (Effect) todavía activo, ir a la pestaña *Interacción de Partículas* (Particle Interaction) y configurar los parámetros del *Desviador* como se muestra en la figura.



Ejemplo de Configuración de Desviación.



Resultado final del Ejemplo.

7. Seleccionar el círculo y presionar *RecalcA* en la pestaña *Efectos* (Effects). Al correr la animación, ALT-A, se verá que las partículas rebotan en el cubo hacia arriba y hacia abajo. El problema es que el rebote está un poco alejado del cubo. Esto se debe a que hay muy pocos Cuadros Clave (Keys). En el campo de Cuadros Clave (*Keys*), ingresar un valor de **50**, y pulsamos *RecalcA*. Al correr nuevamente la animación, las partículas rebotaran en el cubo, o muy cerca de él.

8. Seleccionar el cubo e inclinarlo **10** grados en cualquier dirección. Seleccionar el círculo y recalcular las partículas. Al correr nuevamente la animación, las partículas rebotan en la dirección en la que se giró el cubo.

Ejemplo renderizado

Aquí he añadido un *Meta* objeto y configurado el círculo como padre. Después, he activado *DupliVerts* sobre el círculo. Puse un material con un *Alpha* bajo para simular un fluido. También, jugué con las configuraciones de *Rnd Damping*, y después rendericé todo...

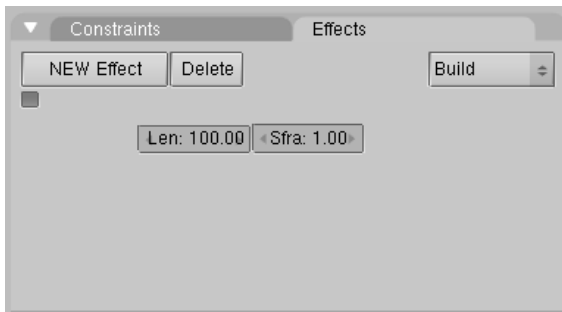


Resultado final modificado.

Build y Ondas

Efecto Build

El efecto Build trabaja sobre mallas que hacen aparecer las caras del objeto, unas detrás de otras, todo el tiempo. Si el material de la malla es tipo Halo, en vez del estandard, son los vértices de la malla, y no las caras, las que se solapan.



Efecto Build.

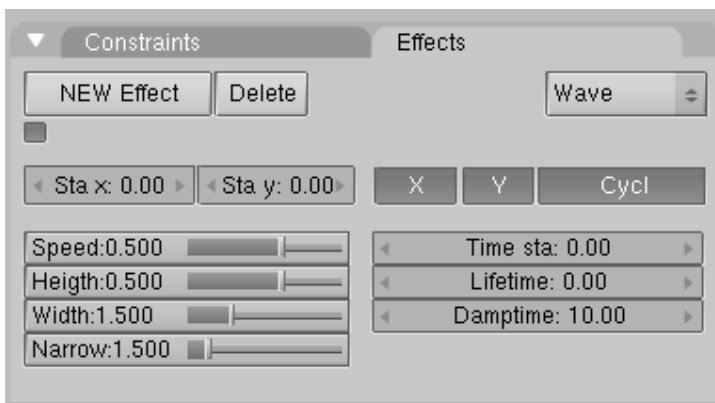
Las caras, o vértices, aparecen en el orden en el que han sido almacenados en memoria. Este orden puede ser alterado seleccionando el objeto con la tecla **CTRL-F** fuera del modo edición. Esto hace que las caras sean reordenadas en función de su propio valor (coordenada z) en la referencia local de la malla.

El efecto Build solo tiene dos controles numéricos (*Efecto Build.*):

- *Len* - Define cuantos frames tomara el efecto Build.
- *Sfra* - Define el frame inicial del proceso Build.

Efecto Onda

El efecto onda añade movimiento a la coordenada Z del objeto malla.



Panel de control de Onda.

La influencia del efecto onda viene generada por un punto de inicio dado, definido en los botones numéricos *Sta X* y *Sta Y*. Estas coordenadas son de la referencia local de la malla (*Origen de la Onda*.)



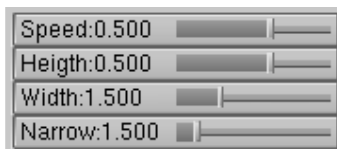
Origen de la Onda.

La difamación del efecto onda se origina en el punto de inicio dado y se propaga a lo largo de la malla con frentes de onda circulares, o con frentes de onda rectilíneos paralelos a los ejes X e Y. Esto esta controlado por los botones de selección X y Y. Si solo uno de los botones esta presionado es lineal, si lo están los dos, sera circular (*Tipo de frente de ondas*). La onda por si misma es como una perturbacion gaussiana que puede ser, o un pulso único o una serie de perturbaciones (tren de ondas), según si el botón *Cycl* esta presionado.



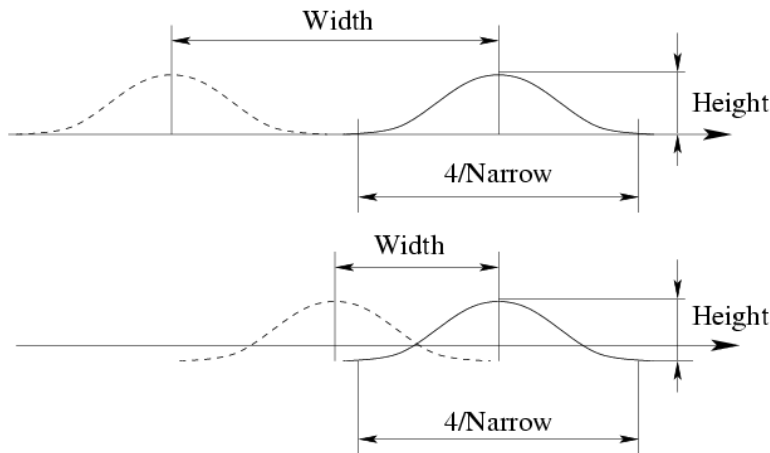
Tipo de frente de ondas.

La onda esta gobernada por dos series de controles, el primero define la forma de onda, y la segunda la duración del efecto. Para lo que concierne a la forma de onda, los controles son *Speed*, *Height*, *Width* y *Narrow* (*Controles de frente de onda*.).



Controles de frente de onda.

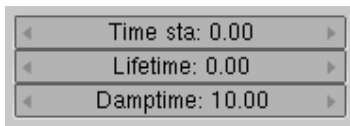
El control numérico *Speed* define la velocidad, en unidades por frame, de la perturbacion. El control numérico *Height* define la altura, en unidades de Blender sobre el eje Z, de la perturbación (Figure 21-6). Si el botón *Cycl* esta presionado, el botón numérico *Width* representa la distancia, en unidades de Blender, entre la mayor distancia de dos perturbaciones seguidas, y el el efecto total de la onda esta dado por la envoltura de todos los pulsos únicos (Figure 21-6). Esto tiene un efecto indirecto en la amplitud de la perturbación. Siendo las perturbaciones de forma Gaussiana, si los pulsos están muy cerca unos de otros, la envoltura no podría alcanzar el punto $z=0$ en ningún momento. Si esto es el caso, Blender en realidad reduce toda la onda de forma que el mínimo sea cero y, consecuentemente, el máximo es menor que el valor de la amplitud esperada, como se ve al final de *Características del frente de onda*. Al ancho real de cada pulso de estilo Gaussiano esta controlado por el botón numérico *Narrow*, cuanto mayor sea el valor mas se estrechara el pulso. El ancho real del área en la que un pulso único sea considerablemente diferente de cero viene dado, en unidades de blender por 4 dividido entre el valor de *Narrow*. Esto es, si *Narrow* es 1, el pulso tiene 4 unidades de ancho, y si *Narrow* es 4 el pulso tiene 1 unidades de ancho.



Características del frente de onda.

Los últimos controles de onda son los controles de tiempo. Los tres botones numéricos definen:

- *Time sta* El frame en el que la onda comienza;
- *Lifetime* el numero de frames que dura el efecto;
- *Damptime* es el numero adicional de frames en los que la onda poco a poco se va reduciendo en amplitud hasta llegar a cero. La reduccion se aplica a todas las perturbaciones y empieza en el primer frame despues de llegar al valor de *Lifetime*. Las perturbaciones desaparecen al llegar al valor en frames de *Damptime*.



Controles de tiempos de onda.

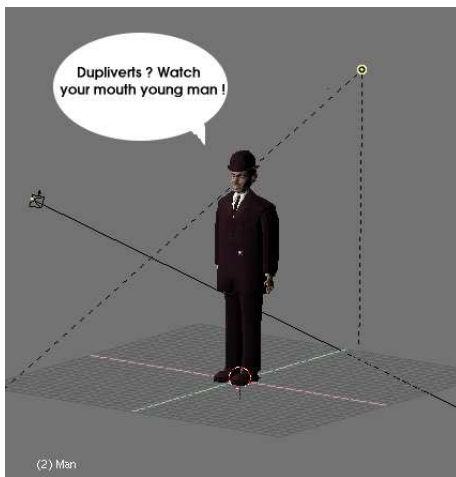
DupliVerts

DupliVerts no es una banda de rock ni un apalabra holandesa para algo ilegal (bueno, quizás sí lo sea) sino una contracción para *Duplicado en Vértices*, lo que significa la duplicación de un objeto base en la posición de los vértices de una malla (o incluso un sistema de partículas). En otras palabras, cuando usamos Dupliverts sobre una malla, una instancia del objeto base es ubicada en cada vértice de la malla. Existen de hecho dos enfoques diferentes para modelar usando DupliVerts. Se los puede usar como una herramienta para ubicar objetos, permitiéndonos modelar arreglos geométricos de objetos (por ej: las columnas de un templo griego, los árboles de un jardín, un ejércitos de soldados robots, los pupitres de un aula). El objeto puede ser de cualquier tipo de objeto soportado por Blender. El segundo enfoque se usa para modelar objetos a partir de una parte del mismo (por ej: las púas de un puerco espín, los mosaicos de un piso, los pétalos de una flor).

DupliVerts como una Herramienta de Ordenación

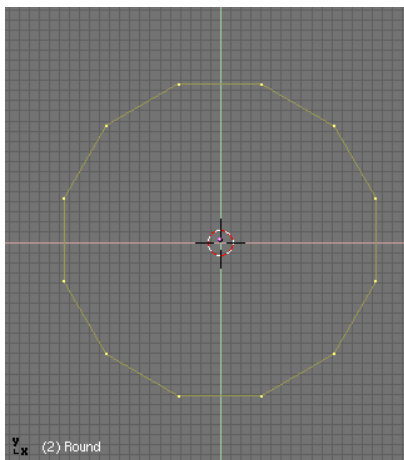
Todo lo que necesitamos es un objeto base (ej: el *árbol* o la *columna*) y una malla con sus vértices siguiendo el patrón que tenemos en mente. Para lo que sigue usaré una escena simple.

Consiste en una cámara, unas luces, un plano (para el piso) y un extraño hombrecillo que modelé basado en un personaje famoso de Magritte (*Una escena simple con la que jugar.*). Si el surrealismo no te apetece, encontrarás esta parte muy aburrida.



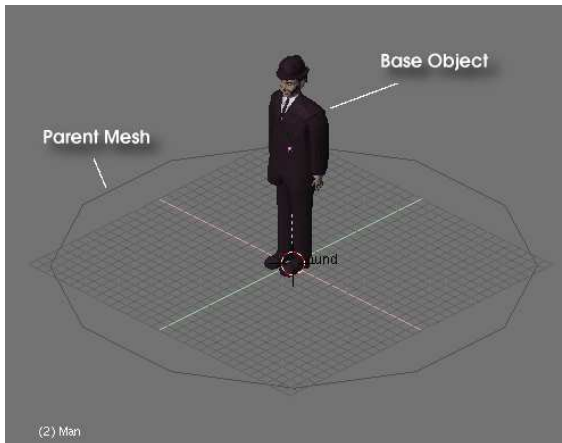
Una escena simple con la que jugar.

El hombre será mi *Objeto base*. Es una buena idea que este se encuentre en el centro de coordenadas y que todas las rotaciones que tenga hayan sido borradas. Moveremos el cursor al centro del objeto base, y desde la vista superior agregaremos una malla Circulo con 12 vértices más o menos. (*La malla padre puede ser cualquier primitiva.*).

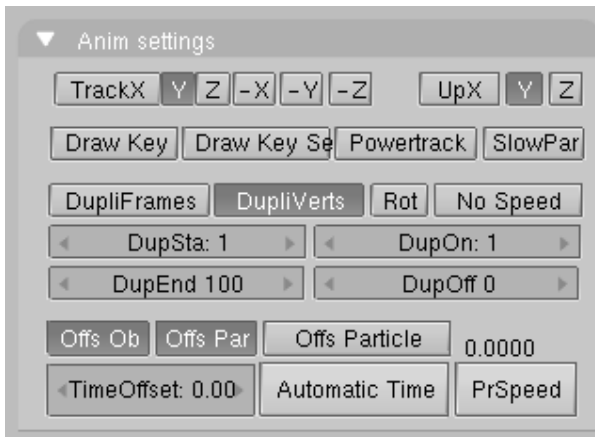


La malla padre puede ser cualquier primitiva.


Fuera de Modo Edición, seleccionaremos el objeto base, y añadiremos el círculo a la selección (el orden es importante aquí). Emparentaremos el objeto base al círculo usando **CTRL-P**. Ahora el círculo es el padre del personaje (*El hombre está vinculado con el círculo.*). Casi estamos.

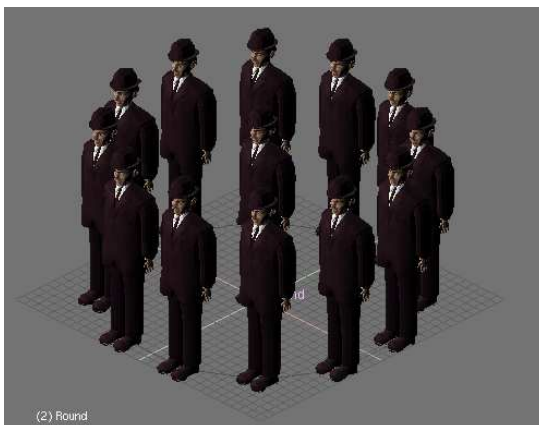


El hombre está vinculado con el círculo.



Los botones de Animación.

Seleccionaremos ahora el círculo, cambiaremos la ventana de botones al Contexto Objeto (via  o con F7) y pulsaremos el botón *DupliVerts* en el panel *Anim Settings* (Los botones de Animación.).



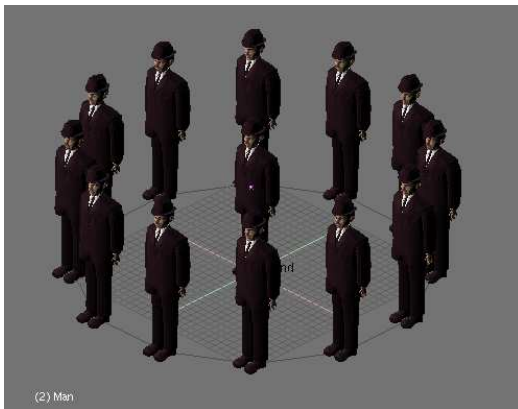
En cada vértice del círculo se coloca un hombre.

Wow, ¿ no es fabuloso ? No te preocupes por el objeto que aparece en el centro del círculo (En cada vértice del círculo se coloca un hombre.). Aunque aparezca en las vistas 3D no será rendereado. Podemos seleccionar ahora el objeto base y modificarlo (escalarlo, rotarlo, en

Modo Edición)[1] y estos cambios se verán en todos los objetos duplicados. Aún más interesante es observar que también podemos editar el círculo padre.

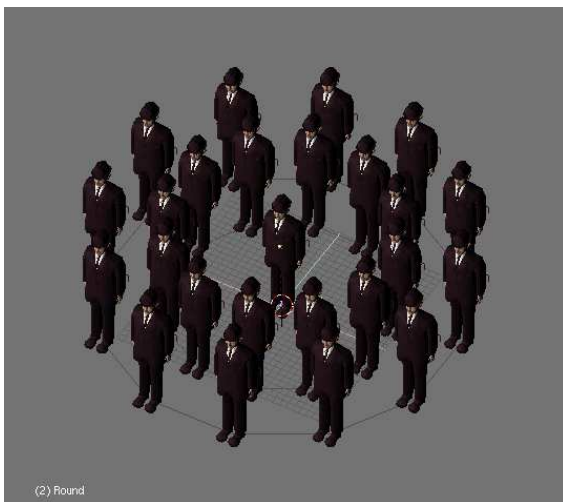
Nota: El objeto base no será renderizado si aplicamos dupliverts sobre una malla pero sí lo será si aplicamos dupliverts sobre un sistema de partículas!

Seleccionemos el círculo y escalémoslo. Podremos ver que los hombres misteriosos son escalados uniformemente con él. Entremos en Modo Edición para el círculo, seleccionemos todos los vértices **AKEY** y agrandémoslo hasta el triple. Salgamos de Modo Edición y los objetos duplicados se actualizarán (*Cambiando el tamaño del círculo en Modo Edición.*). Esta vez mantendrán su tamaño original pero la distancia entre ellos habrá cambiado. No solamente podemos escalar en Modo Edición, también podemos borrar o agregar vertices para modificar la disposición de los hombrecitos.



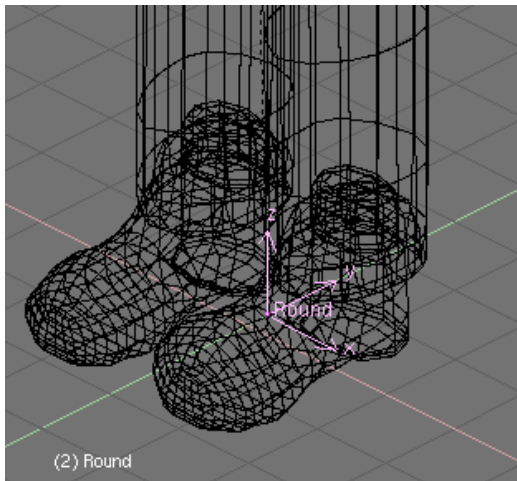
Cambiando el tamaño del círculo en Modo Edición.

Seleccionemos todos los vertices en Modo Edición y dupliquémoslos (**SHIFT-D**). Ahora escalemos los nuevos vertices hacia afuera para obtener un segundo círculo alrededor del original. Salgamos de Modo Edición, y aparecerá un segundo círculo de hombres (*Una segunda fila de hombres Magritte.*).



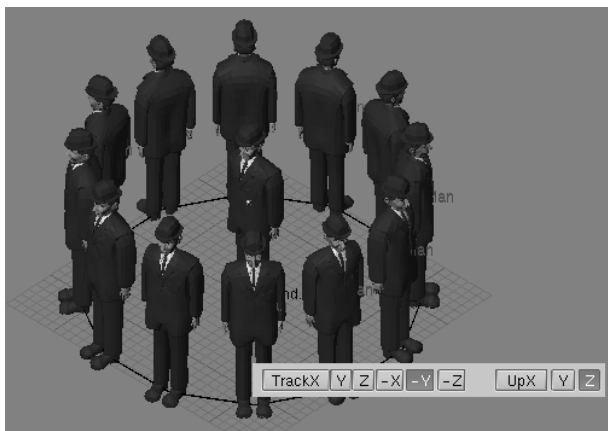
Una segunda fila de hombres Magritte.

Hasta ahora, todos los homrecitos Magritte están enfrentados hacia la cámara, ignorándose entre sí. Podemos obtener resultados más interesantes usando el botón *Rot* junto al botón *DupliVerts* en el panel *Anim Settings*. Con este botón activado, podemos rotar los objetos duplicados de acuerdo a las normales del objeto padre. Más precisamente, los ejes de los objetos duplicados serán alineados con la normal en la posición del vértice. Cual es el eje alineado (X, Y o Z) con la normal de la malla padre, depende de lo indicado en los botones *TrackX*, *Y*, *Z* y *UpX*, *Y*, *Z* en el panel *Anim Settings*. Probando esto con nuestros amigos surreales puede llevarnos a resultados rarísimos dependiendo de la configuración elegida. La mejor manera de entender qué ocurrirá es primero que nada alinear los ejes de los objetos "padre" y "base" con los ejes del espacio 3D. Esto lo podemos hacer seleccionando ambos objetos y pulsando **CTRL-A**, y haciendo click sobre *Apply Size/Rot?*.

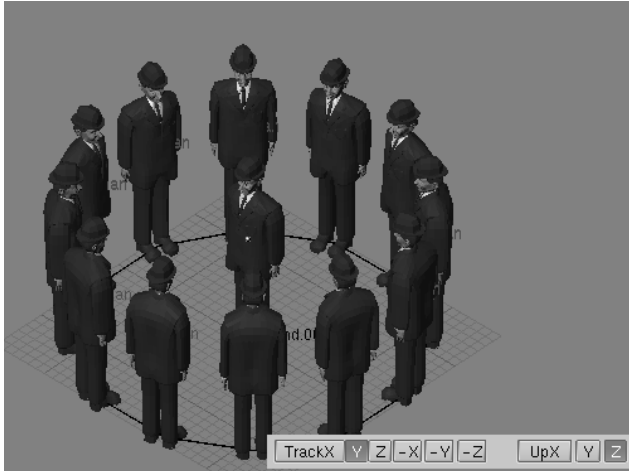


Muestra los ejes del objeto para que elijas el que quieras.

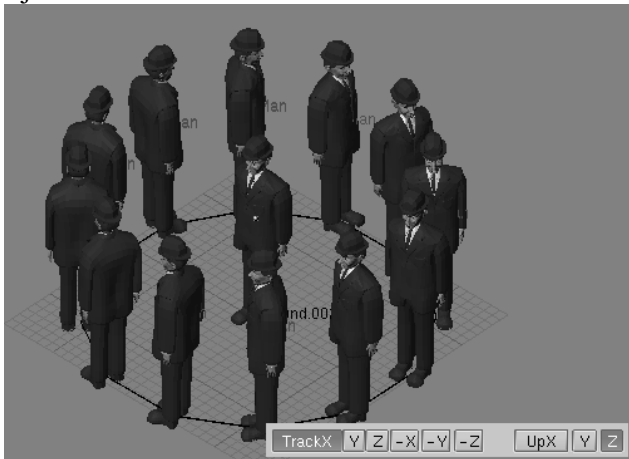
Luego haciendo visibles los ejes del objeto base y los ejes y normales del objeto "padre". (*Muestra los ejes del objeto para que elijas el que quieras.* - en este caso, siendo un círculo sin caras, se debe definir una cara primeramente para que la normal sea visible - de hecho para que la normal exista). Ahora seleccionaremos el objeto base (nuestro homrecito Magritte) y jugaremos un poco con los botones de Tracking. Observar los distintos alineamientos de los ejes con las sucesivas combinaciones de *UpX*, *Y*, *Z* y *TrackX*, *Y*, *Z* (*Eje Y Negativo está alineado al vértice normal (apunta al centro del círculo).*, *Eje Y Positivo está alineado al vértice normal.*, *Eje X Positivo está alineado al vértice normal.*, *Eje Z Positivo está alineado al vértice normal.* (horrible, ¿verdad?).



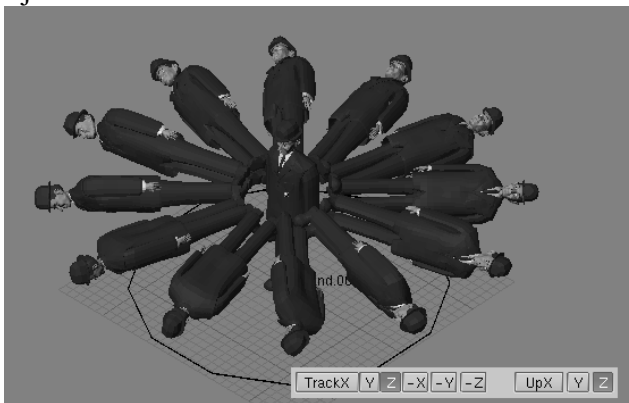
Eje Y Negativo está alineado al vértice normal (apunta al centro del círculo).



Eje Y Positivo está alineado al vértice normal.



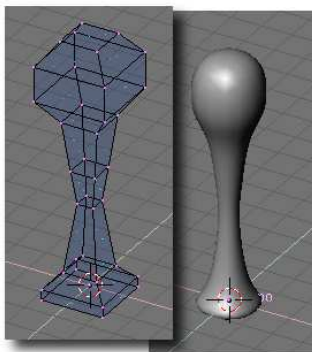
Eje X Positivo está alineado al vértice normal.



Eje Z Positivo está alineado al vértice normal. (horrible, ¿verdad?)

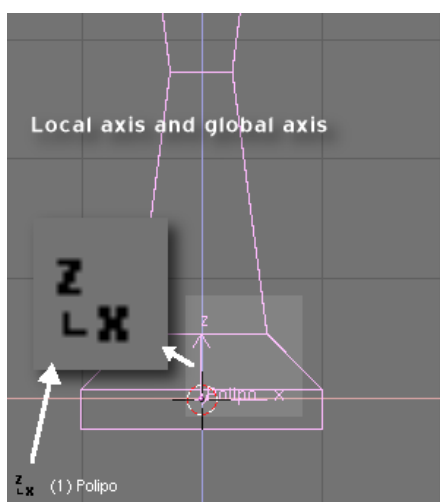
DupliVerts para modelar un objeto

Se pueden modelar objetos muy interesantes usando DupliVerts y una primitiva estándar. Comenzando con un cubo en la Vista Frontal y extrudando un par de veces he modelado algo que se parece a un tentáculo cuando activamos Subsurf (*Extraño tentáculo y versión subdivisión de caras.*). Después agregué una Icoesfera con 2 subdivisiones.



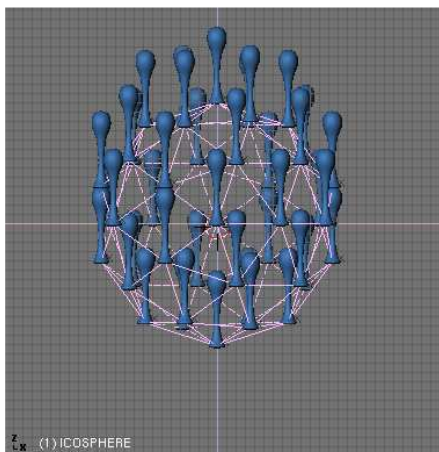
Extraño tentáculo y versión subdivisión de caras.

Tuve especial cuidado en asegurarme que el tentáculo estuviese en el centro de la esfera y que ambos objetos tuvieran alineados sus ejes con los del espacio 3D como dice líneas arriba (*Referencia local del tentáculo.*).

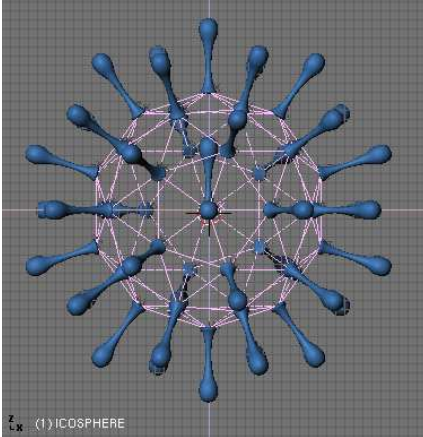


Referencia local del tentáculo.

Luego simplemente hago que la icoesfera sea "padre" del tentáculo. Selecciono la icoesfera solamente y activo *DupliVert* en el panel de *Anim Settings* (*DupliVerts sin rotar.*). Presionemos también el botón *Rot* para hacer girar los tentáculos (*DupliVerts rotados.*).

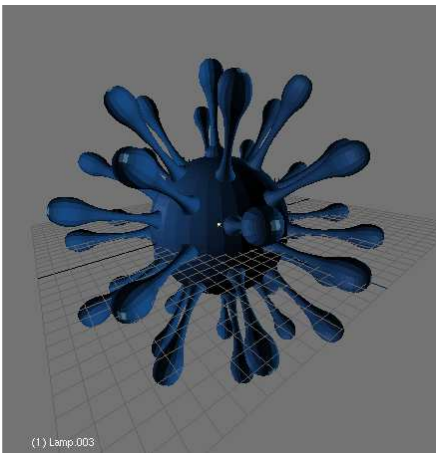


DupliVerts sin rotar.



DupliVerts rotados.

Para hacer que los tentáculos apunten hacia fuera, deberemos una vez más, observar cuidadosamente hacia dónde apuntan los ejes del objeto. Cuando aplicamos *Rot*, Blender intentará alinear uno de los ejes del tentáculo con el vector normal en el vértice de la malla padre. En el caso de los hombrecitos Magritte no le dimos importancia al círculo padre, pero aquí deberemos ocuparnos de la esfera, la que pronto descubrirás que no es *renderizada*. Quizás quieras *añadir* una esfera extra renderizable para completar el modelo. Podemos experimentar en Modo Edición con el tentáculo, moviendo sus vertices apartándolos del centro de la esfera, pero los centros de los objetos deberán permanecer siempre en el centro de la esfera para obtener una figura simétrica. Sin embargo, cuidate de no escalar restringido en un eje en Modo Objeto ya que esto llevará a resultados impredecibles en los objetos duplicados cuando activemos el botón *Rot*.



Nuestro modelo completado.

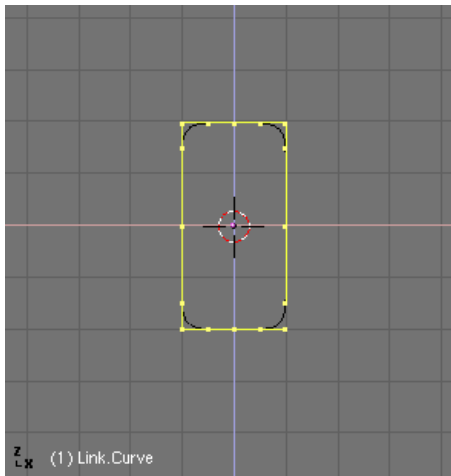
Cuando hayas terminado con el modelo y estés contento con el resultado, puedes seleccionar el tentáculo, presionar **SHIFT-CTRL-A** y hacer click en la opción *Make duplis real ?* para convertir las copias virtuales en masllas reales (*Nuestro modelo completado.*).

DupliFrames

Podemos pensar en DupliFrames de dos maneras distintas: como una herramienta de modelado o como una herramienta para disponer objetos. En un sentido, los DupliFrames son muy similares a los DupliVerts. La única diferencia es que con DupliFrames, disponemos o ubicamos nuestros objetos haciéndolos seguir una curva en lugar de usar los vértices de una malla. DupliFrames viene de DUPLICado en FRAMES (cuadros) y es una técnica de modelado muy útil para objetos que se repiten a lo largo de un recorrido, como los durmientes de una vía ferroviaria, las tablas de una cerca o los eslabones de una cadena, pero también sirve para modelar objetos curvos complejos como tirabuzones, conchas marinas y espirales.

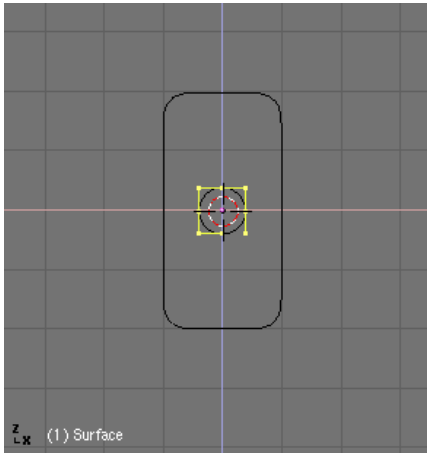
Modelando con DupliFrames

Vamos a modelar una cadena con sus eslabones usando DupliFrames. Lo primero viene primero. Para explicar el uso de DupliFrames para modelar, vamos a empezar modelando un eslabón. Para ello crearemos una Curva Círculo en la vista frontal (Bézier o NURBS da lo mismo). En Modo Edición, lo subdividimos una vez y movemos los vértices un poco para que se ajusten al contorno del eslabón (*Contorno del eslabón.*).



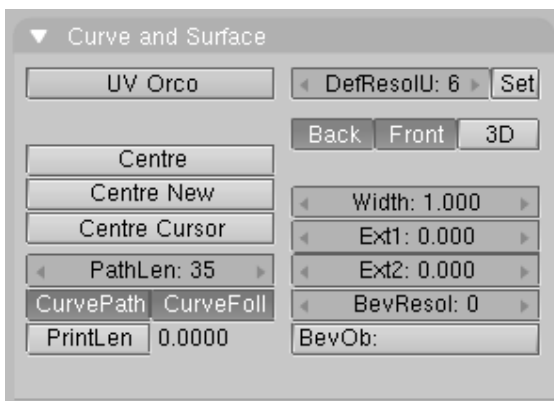
Contorno del eslabón.

Salimos de Modo Edición y añadimos una Superficie Círculo (*Sección transversal del eslabón.*). Las Superficies NURBS son ideales para esto, porque podemos cambiar la resolución fácilmente después de crearlas, y si lo necesitamos, podemos convertirlas a objetos malla. Es muy importante no confundir Curvas Círculo con Superficies Círculo. Las primeras servirán para dibujar la forma del eslabón, pero no nos permitirá realizar el paso de "skinning" luego. Las segundas servirán para trazar las secciones transversales de nuestro "skinning".



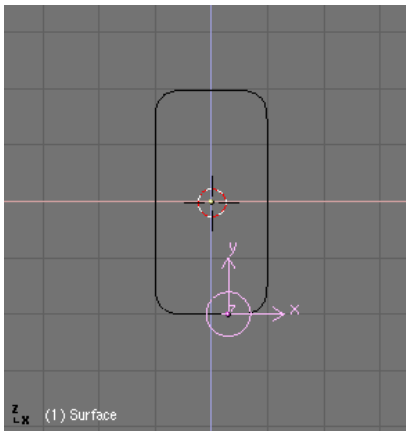
Sección transversal del eslabón.

Ahora emparentaremos la superficie círculo a la curva círculo (el contorno del eslabón) usando emparentado *Normal* (no confundir con *Curve Follow*). Seleccionaremos la curva y en los botones de Object Context y en el panel *Anim Settings* pulsaremos los botones *CurvePath* y *CurveFollow* (*Configuración de Curvas: Curve Path y Curve Follow.*).



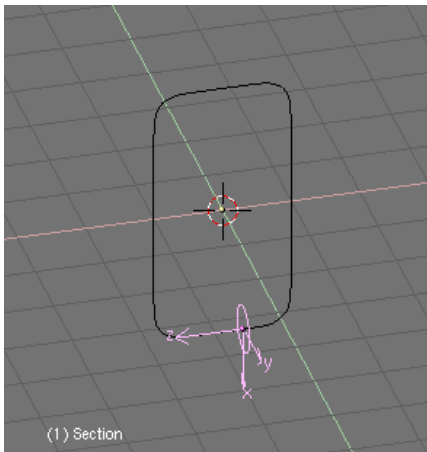
Configuración de Curvas: Curve Path y Curve Follow.

Es probable que la superficie círculo aparezca desplazada. Simplemente la seleccionaremos y pulsaremos **ALT-O** para borrar el origen (*Borrando el origen.*).



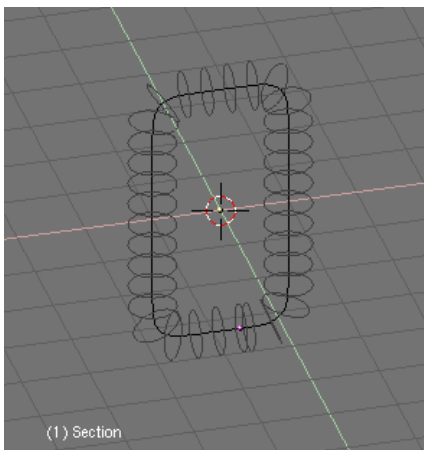
Borrando el origen.

Si pulsas **ALT-A** la superficie avanzará siguiendo la curva. Ahora quizás tengas que ajustar los ejes usando los botones *TrackX*, *Y*, *Z* y *UpX*, *Y*, *Z* en los botones de animación, para que la superficie vaya perpendicular al recorrido curvo. (*Siguiendo el eje correcto.*).



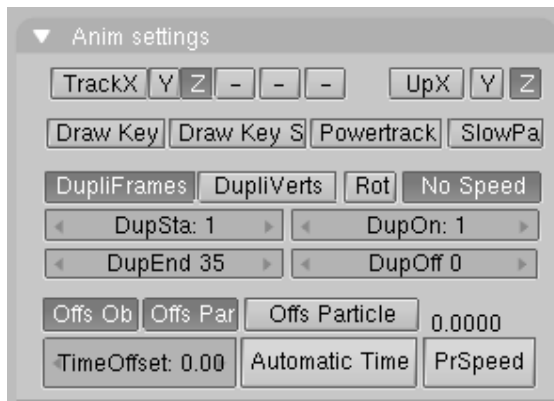
Siguiendo el eje correcto.

Seleccionemos ahora la superficie y vayamos al panel de *Anim Settings* y pulsemos *DupliFrames*. Un número de instancias de la sección circular aparecerán a lo largo del recorrido curvo. (*¡DupliFrames!*).



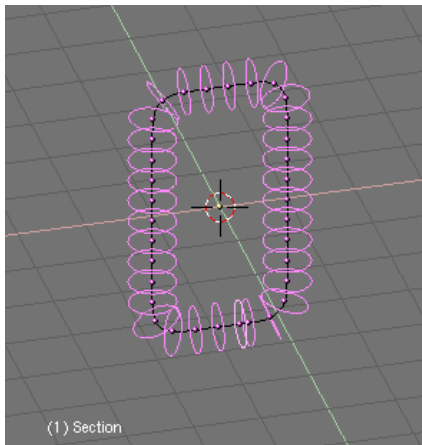
¡DupliFrames!

Podemos ajustar el número de círculos que queremos tener con los botones *DupSta*, *DupEnd*, *DupOn* y *DupOff*. Estos botones controlan el Inicio y Final de la duplicación, el número de duplicados por frame y también el Offset o separación entre duplicaciones. Si quieres que el eslabón se vea abierto en lugar de cerrado puedes probar un valor distinto para *DupEnd* (Valores para *DupliFrames*. Notar que "*DupEnd: 35*" terminará el eslabón antes del final de la curva.).



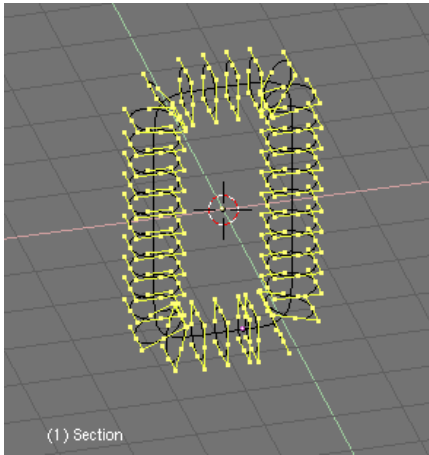
Valores para DupliFrames. Notar que "DupEnd: 35" terminará el eslabón antes del final de la curva.

Para convertir la estructura en un objeto NURB real, seleccionaremos la Superficie Circulo y pulsaremos **CTRL-SHIFT-A**. Un menú nos preguntará *OK? Make Dupli's Real (Haciendo Dupli Real)*.



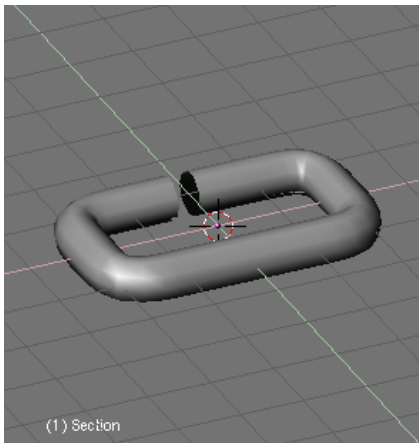
Haciendo Dupli Real

No deseccionemos nada. Tenemos una colección de NURBS que forman el contorno de nuestro objeto, pero aún no existe una "piel" que una los contornos (no existe un "skinning"), por lo que no podemos verlo en un render o en un preview sombreado. Para lograr esto necesitamos unir todos los anillos en un solo objeto. Sin deseleccionar ningún anillo pulsa **CTRL-J** y confirma el menú emergente. Ahora entra en Modo Edición para este objeto y pulsa **AKEY** para seleccionar todos los vertices (*Skineando el eslabón.*). Estamos listo para generar la superficie. Pulsa **FKEY** y Blender generará automáticamente el objeto sólido. Esta operación se conoce como "Skinning" y se describe con más detalle en Section *capítulo_skinning*.



Skineando el eslabón.

Cuando salgas del Modo Edición, podrás ver el objeto en modo sombreado. Sin embargo se ve muy oscuro. Para corregir esto, entremos otra vez en Modo Edición y seleccionemos todos los vertices, luego pulsemos **WKEY**. Elige *Switch Direction* en el menú y sal de Modo Edición. El objeto será representado correctamente ahora. (*Eslabón con piel*). El objeto que hemos creado es un objeto NURBS. Esto significa que aún podremos editarlo. Pero aún más interesante, también podremos controlar la resolución del objeto NURBS por medio de los botones de edición. Allí podemos establecer la resolución del objeto usando los botones *ResolU* y *ResolV*, así por ejemplo podremos usar una resolución baja mientras trabajamos con el objeto y luego usar una resolución alta al momento de hacer render. Los objetos NURBS también tienen la ventaja de ocupar poco espacio en disco cuando salvamos la escena. Compara el tamaño de una escena NURBS con la misma escena donde las NURBS se han convertido a malla (**ALT-C**). Para finalizar podemos borrar la curva que usamos para el contorno del eslabón ya que no la necesitaremos más.

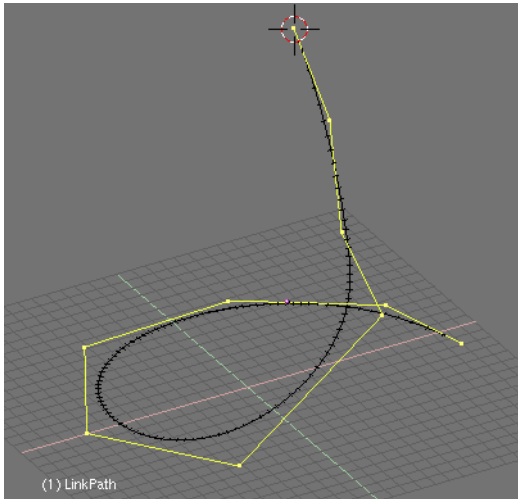


Eslabón con piel

Colocando objetos con DupliFrames

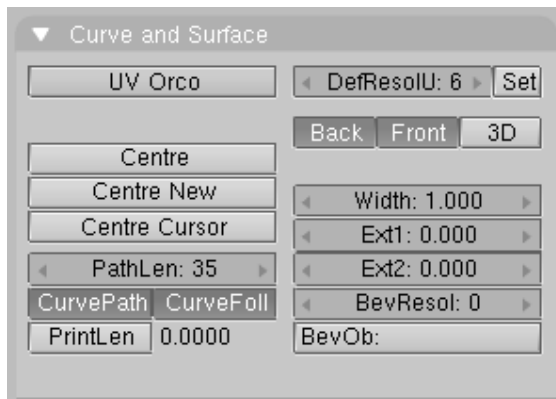
Ahora seguiremos con el modelado de la cadena propiamente dicha. Para esto, añadiremos una Curva Path (Camino) (podemos usar otro tipo de curva pero esta da el mejor resultado). En Modo Edición movemos sus vértices hasta lograr la forma deseada para la cadena (*Usando*

un camino Curva para modelar la cadena.). Si no usamos una Curva Path, deberíamos activar el botón 3D en los botones de edición para que la cadena se despliegue en el espacio 3D.



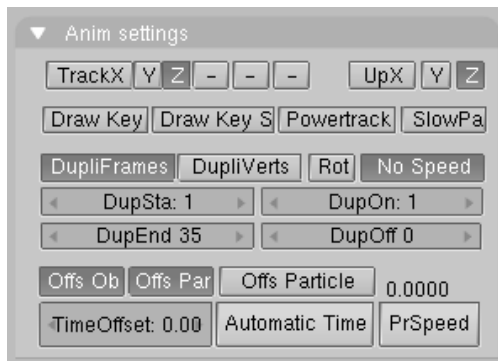
Usando un camino Curva para modelar la cadena.

Selecciona el objeto "Eslabón" que hemos modelado anteriormente y emprantémoslo a la curva "Cadena" usando "Normal Parent". Como estamos usando una Curva Path la opción *CurvePath* en los botones de edición aparecerá activada automáticamente, sin embargo no pasará lo mismo con la opción *CurveFollow* así que deberemos activarla manualmente (*Configuración de Curva.*).



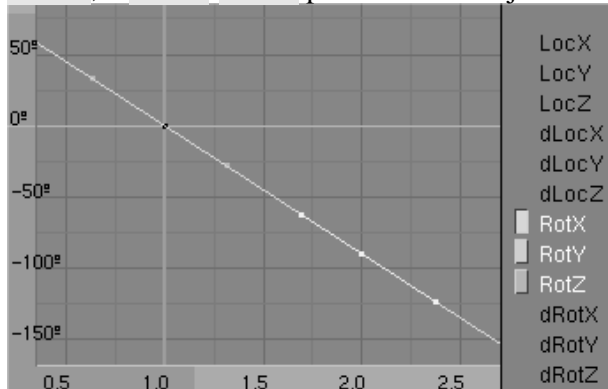
Configuración de Curva.

Si el eslabón aparece trasladado, seleccionalo y pulsa **ALT-O** para borrar el origen. Hasta ahora hemos hecho poco más que animar el eslabón a lo largo de la curva. Podemos verificarlo reproduciendo la animación con **ALT-A**. Ahora, con el eslabón seleccionado, una vez más iremos a Object Context y el panel *Anim settings*. Aquí, activaremos la opción *DupliFrames* como antes. Juguemos con los botones numéricos *DupSta:*, *DupEnd:* y *DupOf:*. Normalmente usaríamos *DupOf:* 0 pero en el caso de una cadena, si usamos *DupOf:* 0 los eslabones estarán demasiado cerca uno de otro y deberemos cambiar el valor *PathLen* de la curva camino a un valor menor en el Contexto de Edición y el panel *Curve and Surface* y luego cambiar correspondientemente el valor *DupEnd:* a ese número para el eslabón (*Ajustando los DupliFrames.*).



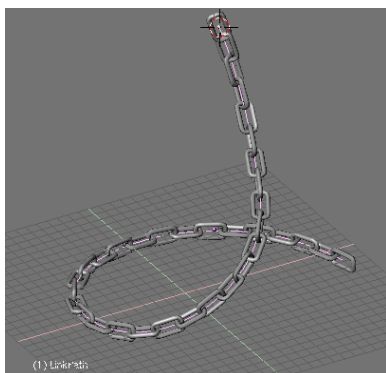
Ajustando los DupliFrames.

Necesitamos que los eslabones roten a lo largo de la curva, de forma que queden rotados 90 grados uno respecto del anterior. Para ello, seleccionaremos el eslabón y pulsaremos *Axis* en los botones de edición para visualizar los ejes del objeto. Insertaremos un keyframe de rotación en el eje paralelo a la curva. Nos moveremos 3 o 4 frames adelante y rotaremos alrededor de este eje pulsando **RKEY** seguido de **XKEY-XKEY** (XKEY dos veces), **YKEY-YKEY**, o **ZKEY-ZKEY** para rotar en el eje *local* X, Y o Z (*Rotando el eslabón*).



Rotando el eslabón.

Abriremos una ventana IPO para editar la rotación del eslabón alrededor del camino. Pulsaremos el *Modo Extrapolación* para que el eslabón rote continuamente hasta termianr el recorrido. Podemos editar la curva IPO de rotación de manera que el eslabón rote exactamente 90 grados cada uno,dos o tres eslabones (cada eslabón corresponde a un cuadro de animación). Usa **NKEY** para ubicar un nodo exactamente en X=2.0 e Y=9.0, que corresponden a 90 grados en un cuadro (desde el cuadro 1 al 2). ¡Ahora tenemos una bonita cadena (*Cadena con Dupliframe*).!



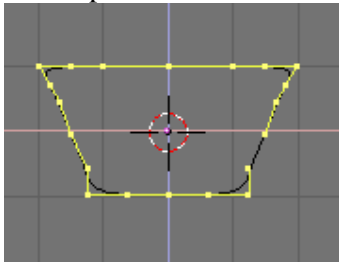
Cadena con Dupliframe.

Más modelado y animación

No estás limitado a usar Curvas Path para modelar tus objetos. Los hemos usado por conveniencia, sin embargo en algunos casos no hay necesidad de ellas. En la vista frontal, agregaremos una superficie Círculo (ya deberías saber por qué a estas alturas *Una superficie Círculo.*). Subdividiremos una vez para darle una forma más cuadrada. Moveremos y escalaremos un poco algunos vértices para darle una forma más trapezoidal (*Sección cruzada trapezoidal.*).

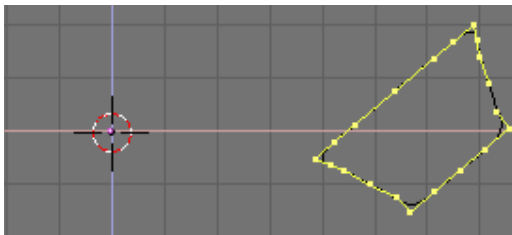


Una superficie Círculo.



Sección cruzada trapezoidal.

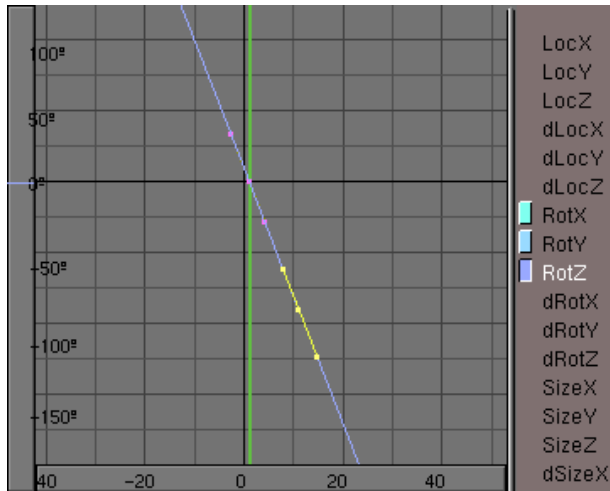
Luego rotaremos todos los vertices algunos grados. Seleccionaremos todos los vértices y los desplazaremos algunas unidades hacia la izquierda o la derecha algunas unidades en X (pero sin cambiar la ubicación en Z). Podemos mantener presionada **CTRL** para mover y escalar con precisión. Salgamos de Modo Edición (*Sección cruzada trapezoidal, rotada y trasladada.*).



Sección cruzada trapezoidal, rotada y trasladada.

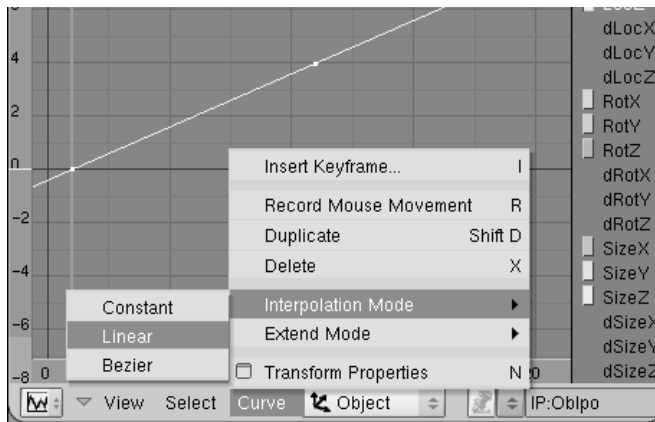
A partir de aquí la única cosa que haremos será editar las curvas IPO de animación. Podríamos llamar a lo que sigue como "Modelando con Animación" si te parece. Ya no entraremos al Modo Edición de la superficie para nada. Cambiemos a una vista superior. Insertemos un keyframe de rotación en el cuadro 1, avancemos 10 cuadros y rotemos la superficie 90 grados alrededor de su nuevo origen. Insertemos un keyframe más. Abramos

una ventana IPO y pongamos la IPO de rotación en Modo Extrapolación. (*Rotación IPO para la sección cruzada.*).



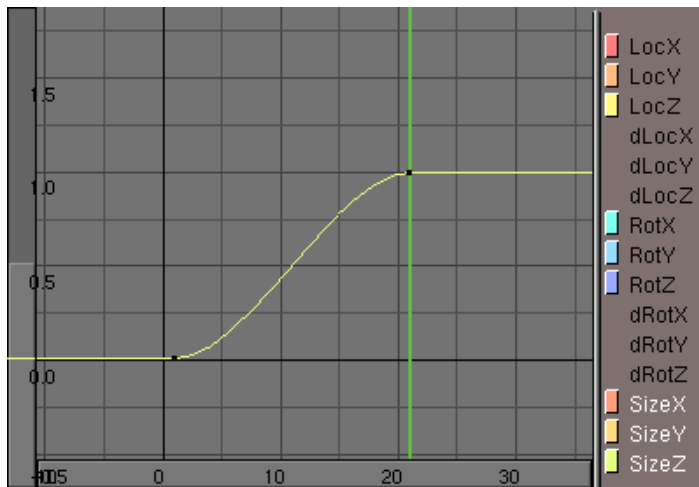
Rotación IPO para la sección cruzada.

Volvamos al cuadro 1 e insertemos un keyframe de Ubicación (Location). Cambiemos a la vista frontal. Avancemos 10 cuadros (solo pulsa **UPARROW**) y movamos la superficie en Z algunas unidades. Insertemos un nuevo keyframe para Ubicación. En la ventana IPO pondremos la curva *LocZ* en Modo Extrapolación (*Traslación IPO para la sección cruzada.*).



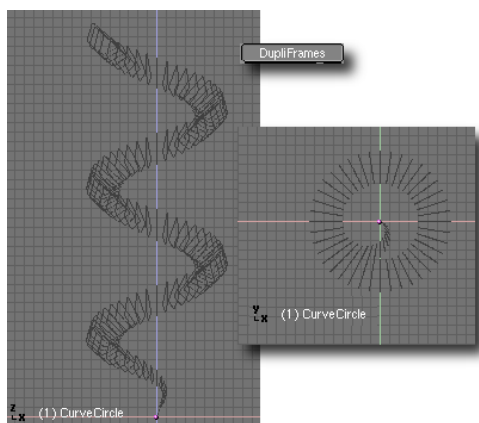
Traslación IPO para la sección cruzada.

Ahora, por supuesto, iremos a los botones de animación y pulsaremos *DupliFrames*. Podremos ver cómo la superficie va ascendiendo en espiral a través del espacio 3D formando algo parecido a un resorte. Se ve lindo, pero queremos más !. Desactivemos *DupliFrames* para continuar. En el cuadro 1 escalaremos la superficie casi hasta cero e insertaremos un keyframe para Size. Adelantaremos hasta el cuadro 41 y borraremos el tamaño con **ALT-S**. Insertaremos entonces un nuevo keyframe para el tamaño. Esta IPO en particular no la pondremos en Modo Extrapolación ya que no queremos que se agrande hasta el infinito ¿verdad? (*Tamaño IPO para la sección cruzada.*).

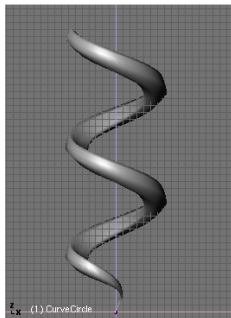


Tamaño IPO para la sección cruzada.

Si ahora activamos DupliFrames verás la silueta de un bonito tirabuzón. (*Usando un camino Curva para modelar la cadena.*). Una vez más los últimos pasos son: Make Duplis Real, unir las superficies, seleccionar todos los vértices y hacer el skinning, invertir la dirección de la normal si es necesario y salir de Modo Edición (*Usando un camino Curva para modelar la cadena.*).



Usando un camino Curva para modelar la cadena.



Usando un camino Curva para modelar la cadena.

Este fue un ejemplo más o menos sencillo. Con más trabajo de edición de curvas IPO podemos lograr modelos más complejos e interesantes. Solo usa tu imaginación.

PARTE XIV - SECUENCIADO

Una de las funciones que quizá sea subestimada en el editor de Blender es el Editor de Secuencia. Es un editor de vídeo completo que permite combinar múltiples canales de vídeo y efectos. Incluso aunque tenga un número limitado de operaciones, puede usarlas para crear ediciones de vídeo bastante potentes (especialmente cuando lo combina con el poder de animación de Blender). Además, es extensible mediante un sistema de plugin similar al sistema de plugins de Texturas.

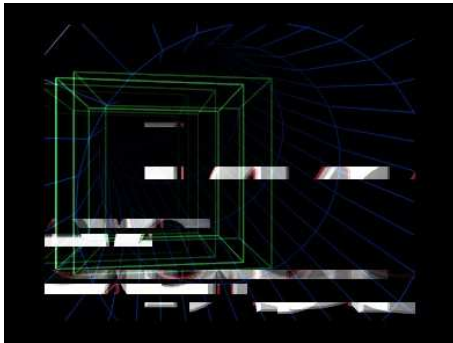
Capítulos

- Editor de Secuencia de Vídeo
- Editor de Secuencia de Audio

Editor de Secuencia de Vídeo

Conociendo el Editor de Secuencia

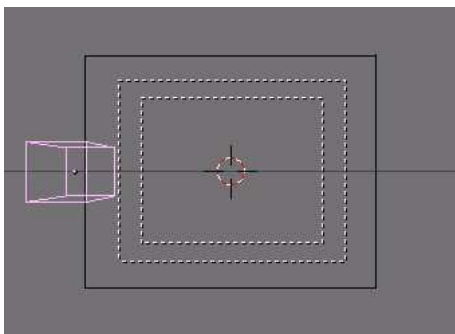
Esta sección muestra un ejemplo de edición de vídeo práctico mostrando la mayoría de características del Editor de Secuencia. Vamos a usar juntas unas cuantas de las animaciones de Blender para obtener algunos impresionantes efectos. Un fotograma de la animación editada resultante está en este *Resultado final*.



Resultado final.

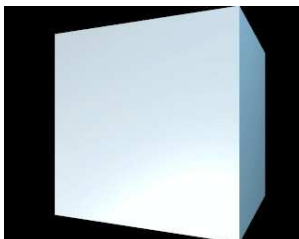
Primera animación: dos cubos

Podemos empezar con algo simple y ver donde nos lleva. Empiece con una escena vacía de Blender y elimine el plano por defecto. Divida la ventana 3D en dos y cambie una de las vistas a la cámara con NUM0. En la vista superior, añada un cubo y muévelo justo fuera del cuadrado punteado que indica la vista de cámara. (*Moviendo el cubo fuera de la vista de cámara*).

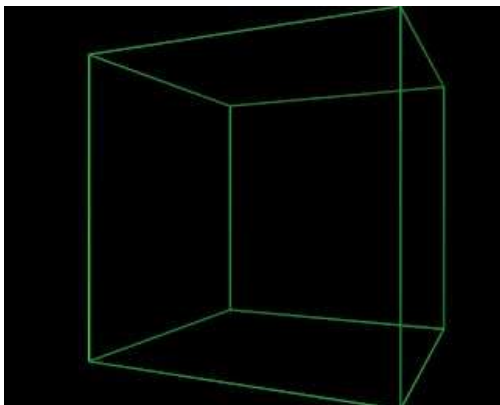


Moviendo el cubo fuera de la vista de cámara.

Queremos crear una animación simple del cubo entrando en la vista, rotando, y después desapareciendo. Defina el final de la animación en 61 (sitúe el valor *End:* en panel *Anim* del Menú Escena, Botones de render **F10**) e inserte un fotograma clave *LocRot* en el fotograma 1 con **I** y seleccionando *LocRot* del menú que aparece. Esto guardará las dos localizaciones y la rotación del cubo en estos fotogramas. Vaya al fotograma 21 (presione **UPARROW** dos veces) y mueva el cubo cerca de la cámara. Inserte otro fotograma clave. En el fotograma 41, mantenga el cubo en la misma posición pero rótelo 180 grados e inserte otro fotograma clave. Finalmente en el fotograma 61 mueva el cubo fuera del campo visual, hacia la derecha e inserte el último fotograma clave. Necesitaremos dos versiones de la animación: una con un material sólido y otra con *WireFrame* (modo alámbrico). Para el material, podemos usar un plano blanco iluminado por dos lamparas brillantes - una blanca y otra azul con un valor de energía de dos (*Un renderizado del cubo sólido...*). Para el cubo *WireFrame* alámbrico, ponemos el tipo de material a 'Wire' (alámbrico) y cambiamos el color a verde. (...y *uno del cubo WireFrame.*)

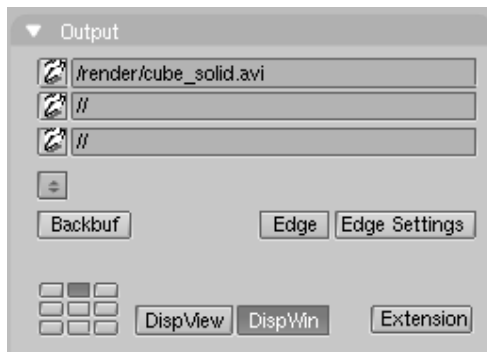


Un renderizado del cubo sólido...



...y uno del cubo *WireFrame*.

Introduzca un nombre de archivo apropiado (por ejemplo 'cubo_solido.avi') en el campo *Pics* (primer botón de texto en la parte superior) del menú Escena, submenú Render panel *Output* ((F10) (*Establezca un nombre de archivo de salida.*).




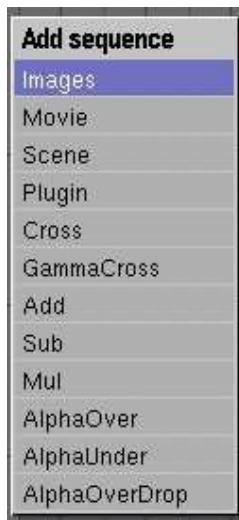
Establezca un nombre de archivo de salida.

Renderice la animación con el cubo sólido blanco. Esto lo guardará en su disco. Guárdelo como un archivo AVI. Use AVI Raw si es posible, ya que ofrece mayor calidad - la compresión debería ser el último paso en el proceso de edición - sin embargo, si dispone de poco espacio en disco, use AVI Jpeg o AVI Codec, la primera comprime menos y por lo tanto tiene mayor calidad. Ahora cambie el material al alámbrico verde, renderice la animación de nuevo guardando el resultado en 'cubo_alámbrico.avi'. Ahora tiene en su disco duro 'cubo_sólido.avi' y 'cubo_alámbrico.avi'. Será suficiente para nuestra primera edición de secuencia.

Primera secuencia: alambres retrasados

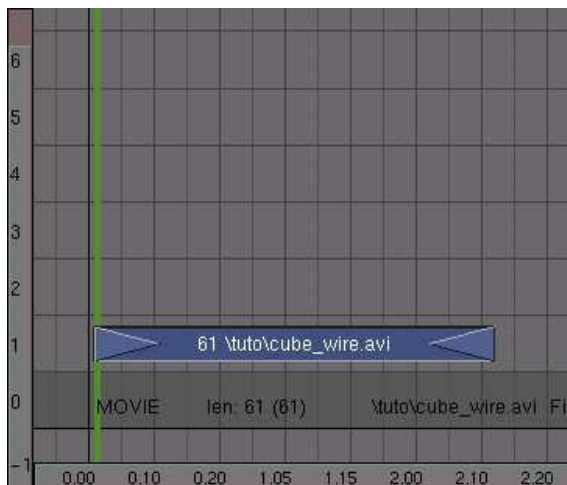
La primera secuencia usará solo la animación alámbrica - ambas - para crear un interesante efecto. Crearemos múltiples capas de vídeo, les daremos un pequeño desfase de tiempo y las añadiremos juntas. Esto simulará el efecto de 'cola difuminada' que puede ver en pantallas de radar.

Empiece con un archivo vacío de Blender y cambie la ventana 3D a un editor de secuencia presionando **SHIFT-F8** o seleccionando el icono del editor de secuencia  de la cabecera del menú tipo de ventana. Añada un vídeo a la ventana presionando **SHIFT-A** y seleccione *Movie* (*Añadiendo un corte de vídeo*) o con la entrada de menú *Add>>Movie*. En la ventana de selección de archivo que aparece seleccione la animación del cubo alámbrico que creó anteriormente.



Añadiendo un corte de vídeo.

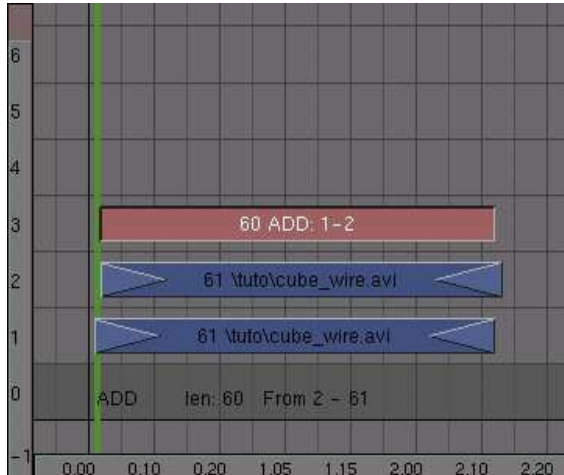
Después de que haya seleccionado y cargado el archivo de vídeo, verá un pequeño corte que la representa. Después de añadir el corte, estará automáticamente en el modo agarrar 'grab' y la franja sigue al ratón. El primer y último frame se muestran ahora en la barra. Miremos la pantalla del editor de secuencia mas de cerca. Horizontalmente vera el tiempo. Verticalmente, vera los canales de vídeo. Cada canal puede contener una imagen, un vídeo o un efecto. Encapando canales diferentes en la parte superior de cada uno y aplicando efectos, podrá mezclar diferentes orígenes juntos. Si selecciona un corte de vídeo, su tipo, longitud y nombre de archivo se mostraran en la parte inferior de la ventana. Mueva el corte de vídeo y deje que comience en el frame 1. Coloquelo en el canal 1, este está en la columna del final y presione **LMB** para finalizar (*Colocando el corte.*).



Colocando el corte.

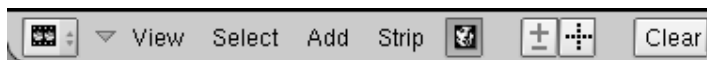
Entradas, salidas y fotogramas: Puede añadir frames lead-in (entradas) y lead-out (salidas) seleccionando los triángulos al inicio y al final del corte (se volverán color púrpura) y arrastrarlos fuera. En el mismo sentido, puede definir la 'longitud' en frames de una imagen estática.

Duplique el corte de vídeo con **SHIFT-D**, sitúe el duplicado en el canal 2 y cambie el primer frame a la derecha. Ahora tenemos dos capas de vídeo una encima de otra, pero solo una se verá. Para mezclar las dos capas necesita aplicarles un efecto. Seleccione ambos cortes y presione **SHIFT-A**. Seleccione **ADD** del menú emergente (*Mezclando dos cortes de vídeo.*). Otra forma sería usar el **Add>>Effect>>Add**.



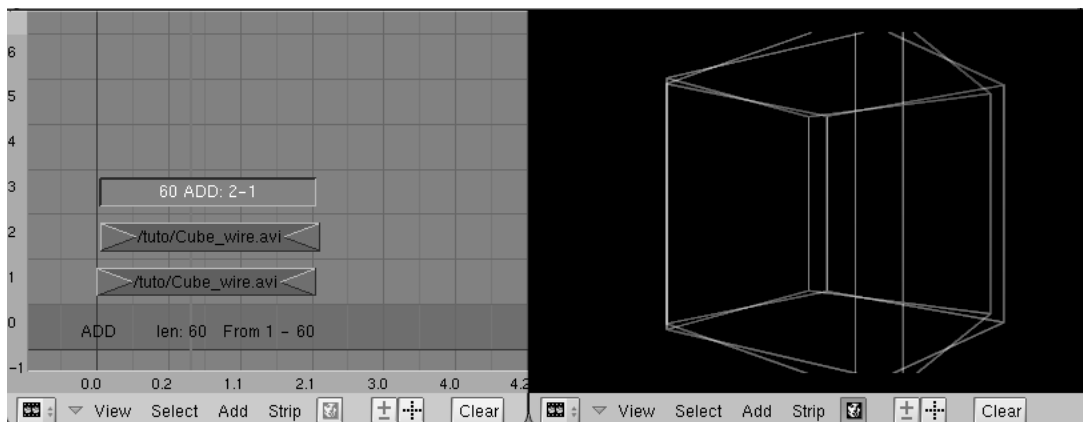
Mezclando dos cortes de vídeo.

Para ver que ocurre divida la ventana del editor de secuencia y seleccione el botón de imagen en la cabecera (*Botón previsualizar del editor de secuencia.*). Esto activará la visualización automática (*Añadiendo una ventana de previsualización.*). Si selecciona un frame en la ventana del editor de secuencia con los cortes, la previsualización será automáticamente actualizada (con todos los efectos aplicados).



Botón previsualizar del editor de secuencia.

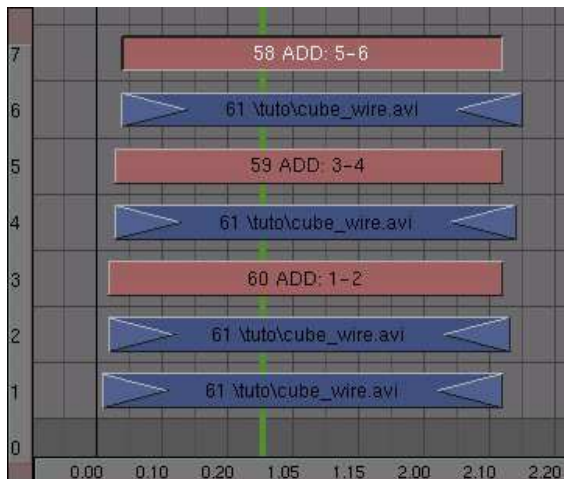
Si presiona **ALT-A** en la ventana de visualización, Blender reproducirá la animación. (El render de los efectos la primera vez toma bastante tiempo de proceso, de modo que no espere una previsualización en tiempo real).



Añadiendo una ventana de previsualización.

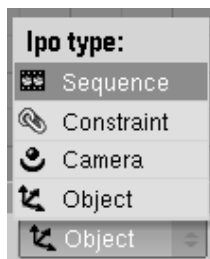
Previsualización sin ventana: Si no le gusta la ventana separada de render, cambie a los botones de render (F10) y seleccione 'DispView' en la parte inferior izquierda.

Ahora es el momento de añadir más caos a la animación. Duplique otra capa de vídeo y situela en el canal 4. Añadala al efecto ADD (añadir) existente en el canal de vídeo 3 con un nuevo efecto ADD (añadir). Repita esto hasta que tenga cuatro cubos alámbricos en la ventana de visualización (*Secuencia con 4 cortes de cubos alámbricos añadidos juntos.*).



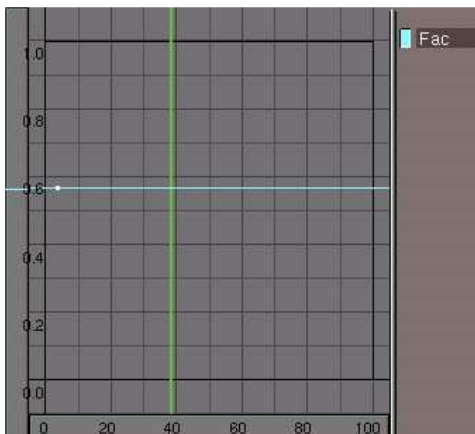
Secuencia con 4 cortes de cubos alámbricos añadidos juntos.

Ahora todos los cubos tienen el mismo resplandor, pero quisiéramos tener una disminución del brillo. Esto es fácil de conseguir: abra una ventana IPO en alguna parte (F6) y seleccione el icono de secuencia en su Menú IPO de clase (*Botón IPO de Secuencia.*).



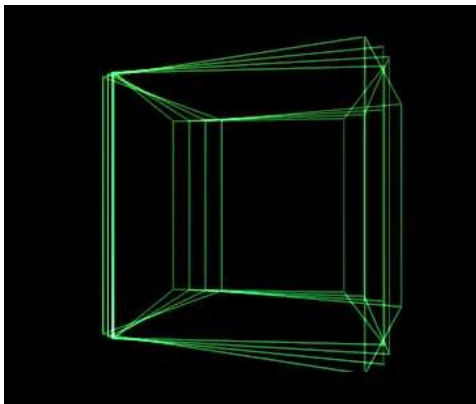
Botón IPO de Secuencia.

Seleccione el primer corte añadido (el del canal 3), presione **CTRL** y haga click **LMB** en la ventana IPO en un valor de 1. Esto pone el brillo de dicha operación de añadir al máximo. Repita el proceso para los otros dos cortes añadidos, pero disminuya el valor un poco en cada uno, alrededor de 0.6 y 0.3 (*Definiendo el brillo de unacapa con un IPO.*).



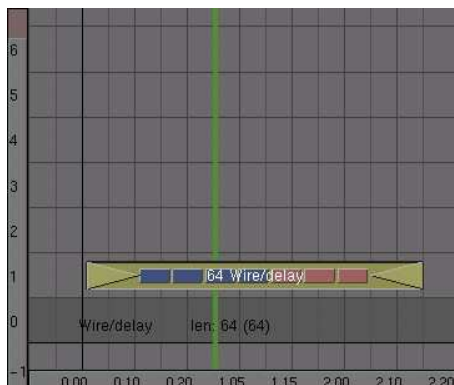
Definiendo el brillo de unacapa con un IPO.

Dependiendo de los valores ADD (añadir) que haya puesto, su resultado debería ser algo parecido a lo que se ve en *Cuatro cubos alambricos combinados con efectos de fundido..*



Cuatro cubos alambricos combinados con efectos de fundido.

Ahora tenemos 7 cortes ¡y habíamos empezado solo con nuestra secuencia!. Puede pensar que la pantalla puede volverse saturada. Para hacer nuestro proyecto mas manejable, seleccione todos los cortes (A y B, presione M y ENTER o haga click en el popup *Make Meta*. Otra manera que podría usar sería en la entrada de menú *Strip>>Make Meta Strip*. Los cortes serán combinados en un meta-corte, y pueden ser copiados como un todo. Con el meta-corte seleccionado, presione N e introduzca un nombre, por ejemplo 'desfase/alambico', para recordar mejor lo que es. (*Nombrado el Meta Corte.*).



Nombrado el Meta Corte.

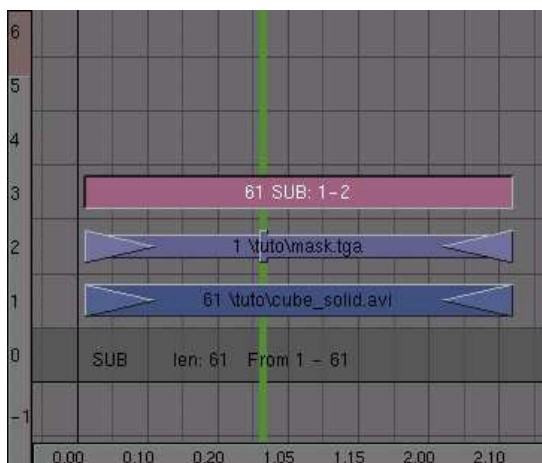
Segunda animación: un cubo solido desfasado

Es el momento de usar algunas mascaras. Queremos crear dos áreas en las que la animación se reproduzca con un frame de diferencia. Esto crea un efecto visual estilo cristal muy interesante. Empiece creando una imagen en blanco y negro como en *Mascara de animación*.. Puede usar un programa de dibujo o hacerlo en Blender. El camino mas fácil para hacerlo en Blender es crear un material blanco con un valor de emisión de 1 o un material blanco sin sombra en algunos círculos biselados. De esta forma no necesita crear ninguna lampara. Guarde la imagen como mask.tga.



Mascara de animación.

Cambie al editor de secuencia y mueva el meta-corte que hemos creado antes, fuera del camino (le reposicionaremos despues). Añada la animación del cubo solido (**SHIFT-A**>>*Movie*). Despues, añada la imagen de mascara. Por defecto una imagen estática tiene una longitud de 50 frames en el editor de secuencia. Cambielo para que coincida con la longitud de la animación del cubo con **RMB** y **RMB** con las flechas salientes en los laterales del corte de imagen con el botón derecho del ratón. Ahora seleccione ambos cortes (presione **SHIFT**), presione **SHIFT-A** y añada un efecto (sustraer) **SUB**. (*Substrayendo la mascara del vídeo.*).



Substrayendo la mascara del vídeo.

En la ventana de previsualización ahora verá el efecto; las áreas donde la máscara es blanca se habrán eliminado del dibujo (*Máscara eliminada.*).



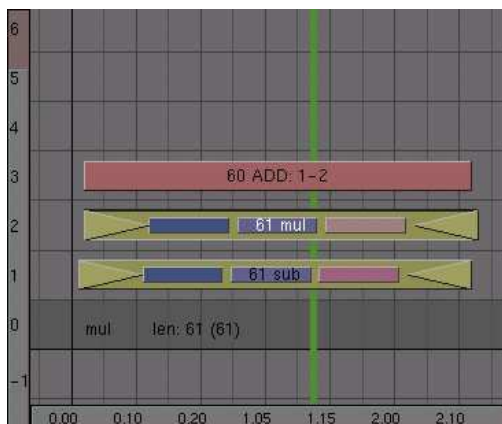
Máscara eliminada.

El efecto ya esta preparado; seleccione los tres cortes y conviértalos en un META corte presionando M. Ahora repita los pasos anteriores, excepto que ahora no usa el efecto *SUB* sino el efecto *MUL* (multiplicar) (*Máscara multiplicada*). Ahora solo verá la imagen original donde la máscara es blanca. Cambie los tres cortes de éste efecto en un meta corte de nuevo.



Máscara multiplicada.

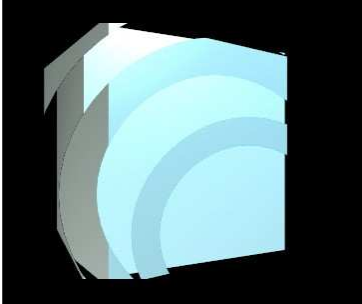
Para el último paso tenemos que combinar los dos efectos juntos. Mueva uno de los meta cortes por encima del otro y apliquele un desfase de un frame. Seleccione ambos cortes y añada un efecto *ADD* (añadir) (*Añadiendo los dos efectos*).



Añadiendo los dos efectos.

En la ventana de previsualización ahora verá el resultado de la combinación de la animación y la máscara (*Dos capas cambiadas en el tiempo*). Cuando esté listo, seleccione los dos meta cortes y el efecto *ADD* y conviértalos en un nuevo meta corte. (¡Es cierto! Puede tener meta cortes de meta cortes!)

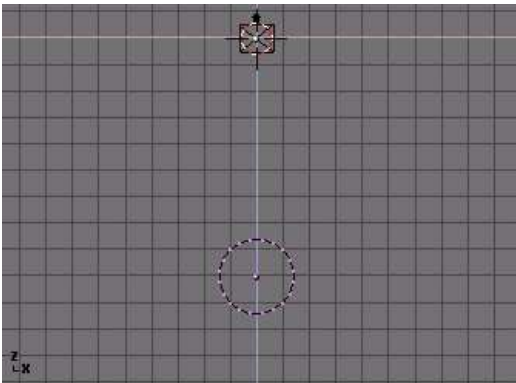
Entrando en un meta corte: Para editar los contenidos de un meta corte, selecciónelo y presione TAB. El meta corte 'explotará' para mostrar sus componentes y el fondo se volverá verde amarillento para indicar que está trabajando dentro de un meta corte. Presione TAB de nuevo para volver al modo de edición normal.



Dos capas cambiadas en el tiempo.

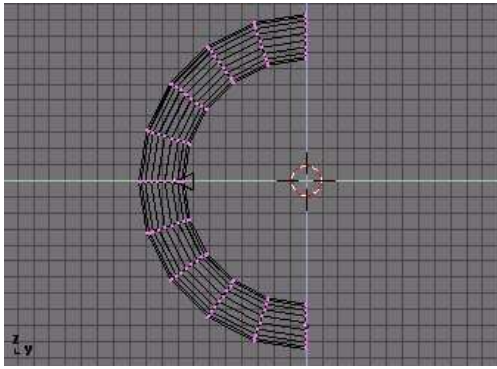
Tercera animación: Un túnel

Ahora queremos un tercer 'efecto' para enriquecer más allá nuestra animación; un 'túnel' 3D para usarlo como efecto de fondo. Es realmente fácil de crear. Primero guarde su trabajo actual - ¡lo necesitará después! Comience una nueva escena (CTRL-X) y elimine el plano por defecto. Cambie a la vista frontal (NUM1). Añada un círculo de 20 vértices aproximadamente 10 unidades por debajo de la línea de $z=0$ (la línea rosa en su pantalla) (*Añadiendo un círculo de 20 vértices.*).



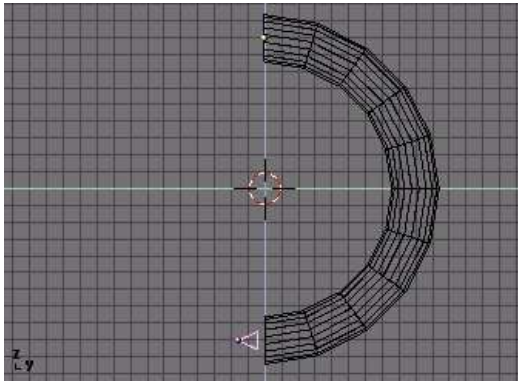
Añadiendo un círculo de 20 vértices.

Mientras estamos en el Modo Edición, cambie a la vista lateral (NUM3) y apunte el cursor al origen de forma aproximada al punto $x,y,z=0$ y presionando SHIFT-S. Seleccione *Curs>>Grid*. Queremos convertir el círculo en un tubo circular, o toroide. Para ello, usaremos la función Revolución. Vaya al menú edición (F9) e introduzca un valor de 180 en el Botón numérico *Degr* e introduzca '10' en el numérico *Steps* en el panel *Mesh Tools* (herramientas de malla). Presionando *Spin* ahora rotará los vértices seleccionados alrededor del cursor 180 grados en 10 pasos. (*Revolucionando el círculo alrededor del cursor.*).

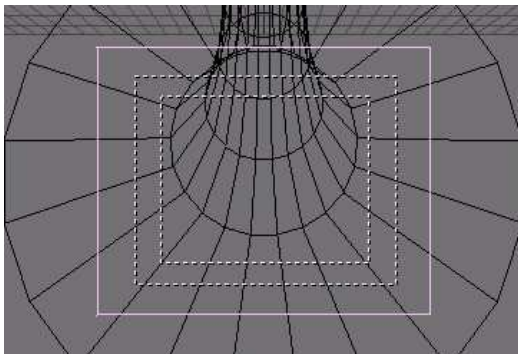


Revolucionando el círculo alrededor del cursor .

Salga del modo edición (TAB). Con las opciones por defecto, Blender siempre rota y escala alrededor del centro del objeto que se muestra como un pequeño punto. Este punto es amarillo cuando el objeto está sin seleccionar y rosa cuando lo está. Mientras el cursor esté en el origen, presione el botón *Center Cursor* en la ventana de Botones de Edición para mover el centro del objeto a la posición actual del cursor. Ahora presione **R** y rote el tubo 180 grados alrededor del cursor. Ahora es el momento de mover la cámara dentro del túnel. Abra otra ventana 3D y cambie a la vista de cámara (NUM0). Posicione la cámara en la ventana vista lateral para que coincida *Visión de cámara del interior del túnel.*, la vista de cámara ahora debería coincidir *Cámara dentro del túnel.*



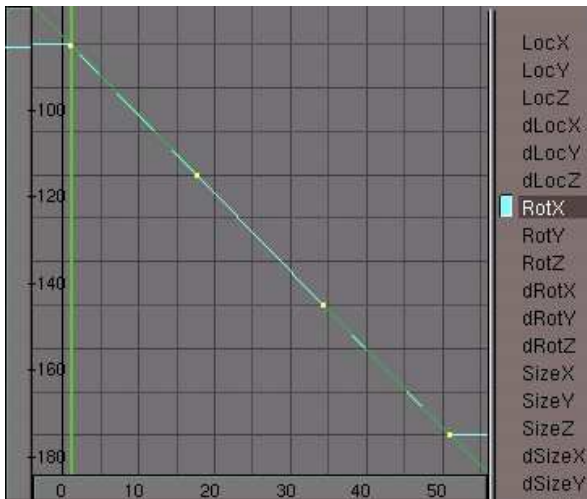
Cámara dentro del túnel.



Visión de cámara del interior del túnel.

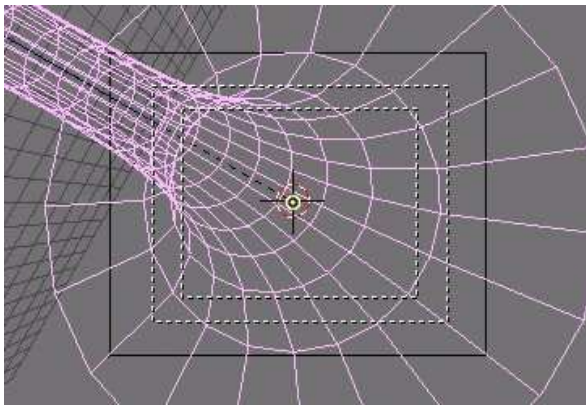
Para quitarnos algún problema, me gustaría renderizar la escena como una animación en bucle. Así, podré añadir tantas copias como quiera a la compilación final de vídeo. Hay dos cosas que debemos tener en mente cuando creamos animaciones en bucle. Primero, asegúrese de que no hay 'salto' en su animación cuando se repite. Para ello, debe tener cuidado creando

los KeyFrames y estableciendo la longitud de la animación. Cree dos KeyFrames: uno con la actual rotación del tubo en el frame 1, y otro con una rotación de 90 grados (presione **CTRL** mientras rota) en el frame 51. En su animación el frame 51 ahora es lo mismo que el frame 1, de modo que cuando renderice necesitará dejar fuera el frame 51 y renderizar desde el 1 hasta el 50. Dese cuenta de que el número 90 grados no está elegido al azar, debido a que el túnel es periódico, con periodo 18, entonces debe rotarlo por un múltiplo de 18, y 90 lo es, para garantizar que el frame 51 es exactamente el mismo que el 1. Segundo, para conseguir un movimiento *lineal* necesita eliminar las entradas y salidas de la rotación. Se pueden ver en la Ventana IPO del tubo después de insertar los Keyframes de rotación. El suavizado IPO empieza y termina como una función coseno. Queremos que sea recto. Para ello seleccionamos la curva de rotación, entramos en el modo edición (TAB) y seleccionamos todos los vértices (A) y presione V ('Vector') para cambiar la curva en una línea. (*Rotación IPO del túnel sin entrada y salida.*).



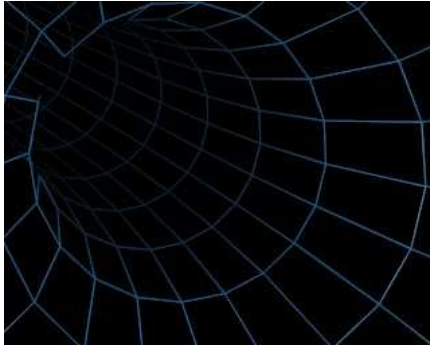
Rotación IPO del túnel sin entrada y salida.

Para crear un efecto más dramático, seleccione la cámara mientras está en el modo vista de cámara (*Rotando la cámara para conseguir un efecto más dramático.*). La misma cámara se muestra como un cuadrado sólido. Presione **R** y rotela un poco. Si ahora reproduce la animación debería repetirse sin vetas.



Rotando la cámara para conseguir un efecto más dramático.

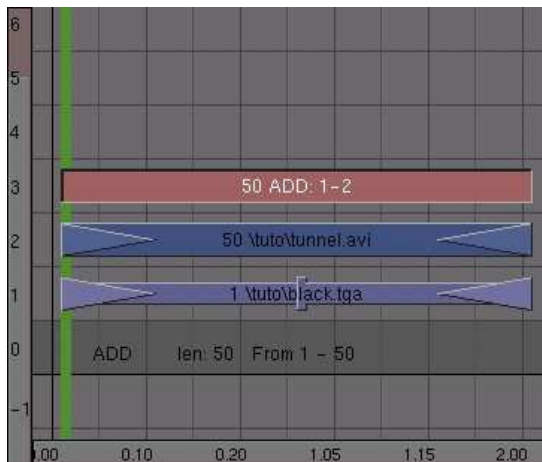
Para el toque final, añada un material alámbrico azul al tubo y añada una pequeña lámpara en la posición de la cámara. Jugando con los valores *Dist* de la lámpara (distancia de atenuación) puede hacer que el final del tubo desaparezca en la oscuridad sin tener que trabajar con nieblas. (*Un túnel estupendo.*). Cuando esté satisfecho con el resultado, renderice la animación y guárdela como 'tunnel.avi'.



Un túnel estupendo.

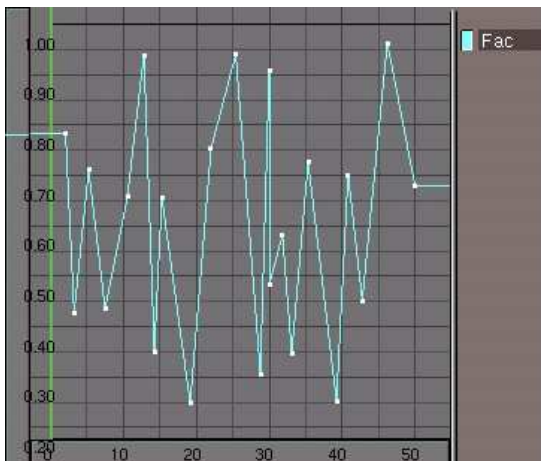
Segunda secuencia: Usando el túnel como telón de fondo

Recargue su archivo de compilación de vídeo de Blender. El túnel que creamos en el paso anterior será usado como telón de fondo para toda la animación. Para hacerlo más interesante he modificado un efecto *ADD* para cambiar el túnel en un telón pulsante. Prepare una imagen completamente negra y llámela 'black.tga' (inténtelo presionando **F12** en un archivo vacío de Blender. Guárdelo con **F3**, pero asegúrese que tiene seleccionado el formato de archivo TGA en la ventana de botones de render). Añada ambos black.tga y la animación del túnel y combínelos con un efecto *ADD*. (*Aplicando el efecto de telón.*).



Aplicando el efecto de telón.

Ahora, con el efecto *ADD* seleccionado, abra una ventana *IPO* y seleccione el botón editor de secuencia en la cabecera. Desde los frames 1-50, dibuje una línea irregular manteniendo presionado **CTRL** y con el botón izquierdo. Asegúrese de que los valores están entre 0 y 1 (*Añadiendo aleatoriedad con un IPO irregular.*).



Añadiendo aleatoriedad con un IPO irregular.

Cuando esté listo, eche un vistazo al resultado en una ventana de previsualización y convierta la animación en un meta-corte. ¡Salve su trabajo!

Cuarta animación: un logo saltarín

Vayamos a crear algo de aleatoriedad y caos. Coja un logo (podríamos simplemente añadir un objeto de texto) y haga que salte a través de la pantalla. De nuevo, el modo más fácil de hacer ésto es añadiendo vértices directamente en la ventana IPO (seleccione primero el canal *LocX*, *LocY* o *LocZ*, pero en este momento puede que necesite ser un poco más precavido con los valores mínimos y máximos para cada canal. No se preocupe por el aspecto de cada uno demasiado - el próximo paso hará que sea muy reconocible de cualquier manera. (*Logo saltarín.*).



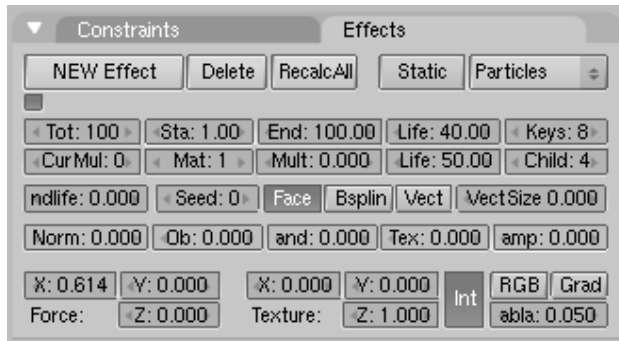
Logo saltarín.

Guarde la animación como 'jumpylogo.avi'.

Quinta animación: Barras de partículas

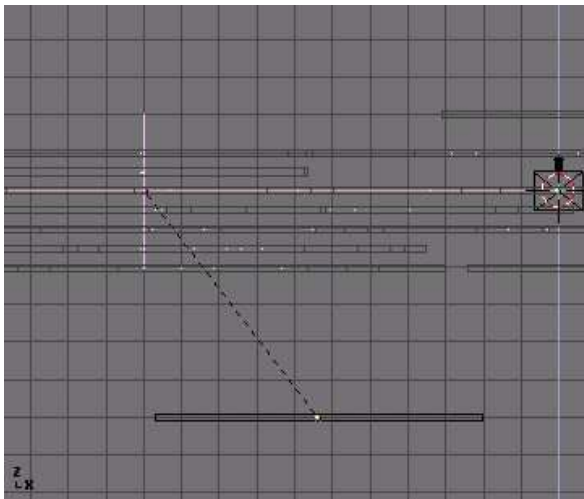
Nuestro último efecto usará una máscara animada. Combinandolo con el logo del paso anterior, conseguiré un efecto rayado que introduzca el logo en nuestra animación. Esta máscara la haremos usando un sistema de partículas. Para aplicar uno cambie a la vista lateral, añada un plano a su escena y mientras está seleccionado cambie al menú de objeto (**F7**). En la ficha *Effects* del panel *Constraints*. Seleccione *New effect* (Nuevo efecto) y cambie el efecto

por defecto *build* a *Particles*. Cambie las propiedades del sistema como se indica en *Propiedades del sistema de partículas.*




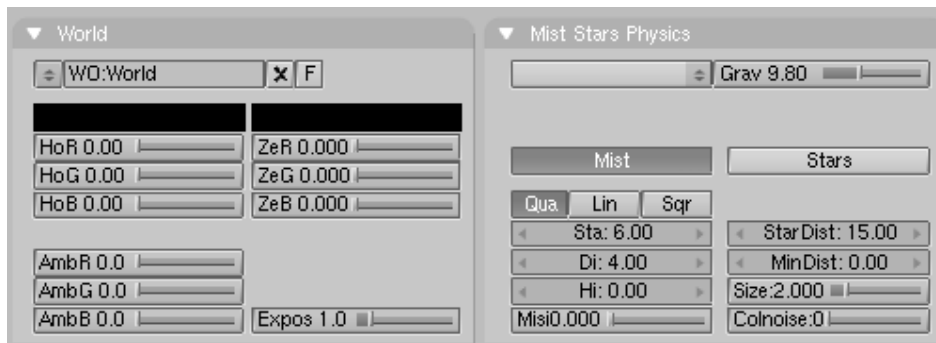
Propiedades del sistema de partículas.

Presione **TAB** para entrar en el modo Edición, seleccione todos los vértices y subdivida el plano dos veces presionando **WKEY** y presionando *Subdivide* del menú emergente. Lo siguiente será cambiar a la vista frontal y añadir otro plano. Escalo a lo largo del eje X para convertirlo en un rectángulo (presione **S** y mueva su ratón horizontalmente. Entonces haga click **X** o **MMB** para escalarlo sobre el eje indicado solamente). Aplique al rectángulo un material blanco con un valor de emisión de uno. Ahora necesita cambiar las partículas en rectángulos usando la función de duplicado de vértices (*dupliverts*). Seleccione el rectángulo, después el emisor de partículas y relacionelos. Seleccione solo el plano y en el menú Objeto y panel *Anim Settings* (Propiedades de animación) seleccione el botón *DupliVerts*. Ahora cada partícula es remplazada por un rectángulo (*Rectángulos con vértices duplicados.*).



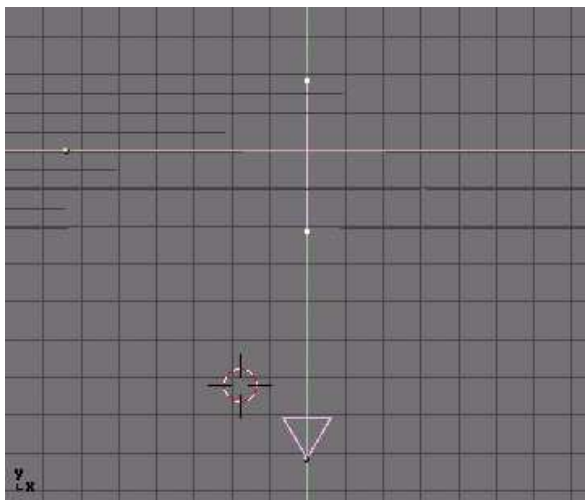
Rectángulos con vértices duplicados.

Añadiré ahora algo de niebla como un truco rápido para dar a cada rectángulo un sombreado diferente de gris. Ve a la ventana de botones de "World" (Entorno) con **F5** para cambiar al menú de sombreado, y haga click en el botón  y seleccione *Add New* en el panel *World* (Entorno). Las propiedades de entorno aparecerán. Por defecto, el cielo será renderizado como un gradiente entre azul y negro. Cambie los colores de horizonte (HoR, HoG, HoB) a negro puro (*Configurando la niebla.*).



Configurando la niebla.

Para activar el render de la niebla active el botón Mist (Niebla) en el medio de la pantalla. Cuando usamos niebla, tenemos que indicar cual será la distancia de la cámara a la que funcionará. Seleccione la cámara, cambie al menú Edición y active *ShowMist* en el panel *Camera*. Ahora cambie a la vista superior y vuelva al menú de sombreado (F5) y los botones de entorno. Retoque los parámetros *Sta:* y *Di:* (Comienzo y distancia, respectivamente) para que la niebla cubra toda la anchura de la cadena de partículas (*Configurando la niebla.* y *Configurando los parámetros de niebla.*).



Configurando los parámetros de niebla.

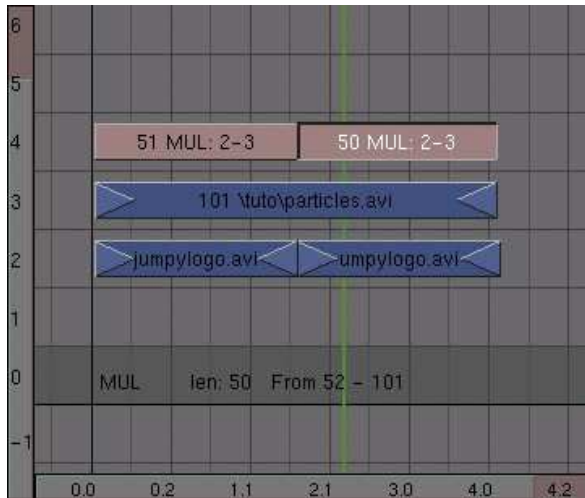
Configure la longitud de la animación en 100 frames y rendericela a disco. Llame al archivo 'particles.avi' (*Rectángulos de partícula renderizados.*).



Rectángulos de partícula renderizados.

Tercera secuencia: Combinando el logo y las barras de partículas

Ahora sabéis lo que hay que hacer: recargar el archivo de proyecto, cambiar a la ventana del editor de secuencia y añadir tanto 'particles.avi' como 'logo.avi' a vuestro proyecto. Combinelos juntos con un efecto *MUL*. Dado que la animación del logo son 50 frames y la de las partículas son 100 frames, necesitará duplicar la animación del logo una vez y aplicar un segundo efecto *MUL* en él. (*Use dos veces el logo de la animación. y Animación de partículas combinado con animación de logo.*).



Use dos veces el logo de la animación.

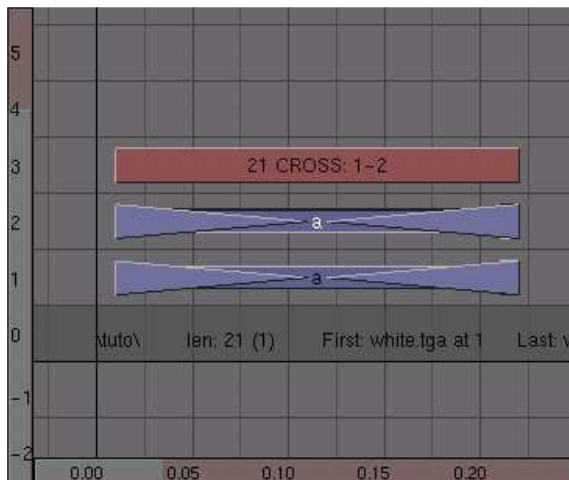
Combine estos tres cortes en un metacorte. Si se ve con valor, puede hacer algunas copias y darles algo de desfase de tiempo como hicimos con el cubo alámbrico.



Animación de partículas combinado con animación de logo.

Sexta animación: Haciendo zoom al logo

Si quisiera combinar todas sus animaciones, lo más que conseguiría sería una compilación de vídeo realmente revuelta, pero si es la presentación de su compañía desearía presentar el logo de una forma más reconocible. La parte final de nuestro montaje será hacer una animación del logo que haga un zoom muy despacio. Prepárelo y guárdelo como 'zoomlogo.avi'. Prepare también una imagen blanca y guardela como 'white.tga'. Ahora usaremos el efecto *CROSS* para hacer primero una rápida transición de blanco a negro, y después a blanco de nuestra animación de logo. Finalmente, una transición a negro que concluirá el montaje. Comenzaremos situando black.tga en el canal 1 y white.tga en el canal 2. Haga a los dos de 20 frames de duración. Seleccione ambos y aplique un efecto "cross". El efecto "cross" (cruce) cambiará gradualmente la imagen resultante de la capa 1 a la capa 2. En este caso, el resultado será una transición de negro a blanco (*Transición negro-blanco.*).



Transición negro-blanco.

Ahora, añade un duplicado de white.tga a la capa 1 y sitúela justo a la derecha de black.tga. Hagalo alrededor de la mitad de largo que el original. Sitúe la animación del zoom de logo en la capa 2 y añade un efecto "cross" entre las dos. En este momento, la animación se ve como un flash blanco seguido de la animación del zoom del logo. (*Imagen de transición vídeo blanco.*).

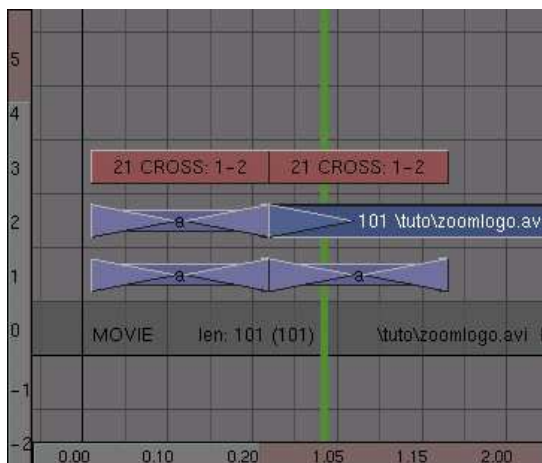
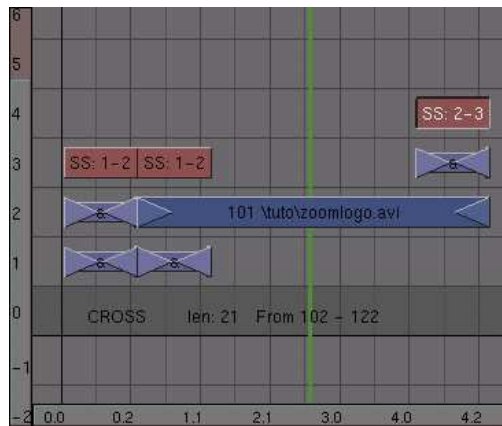


Imagen de transición vídeo blanco.

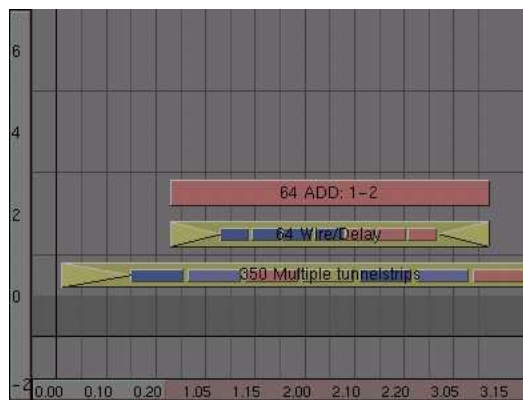
La última cosa que necesita hacer es asegurarse de que la animación tendrá una bonita transición al negro casi al final. Añade un duplicado de black.tga y aplique otro efecto "cross". Cuando esté list, transforme todo en un metacorte strip (*Transición vídeo-negro.*).



Transición vídeo-negro.

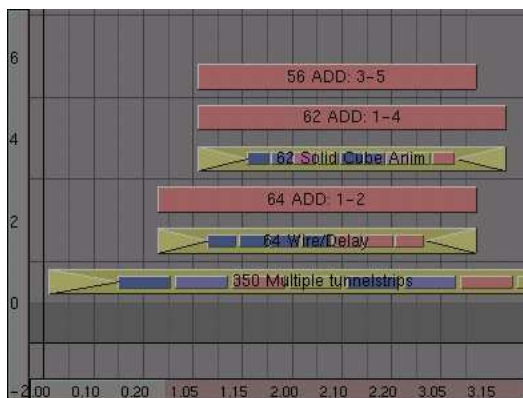
Ensamblándolo todo

¡Ya estamos terminando nuestro trabajo! Es el momento de añadir alguna de nuestras compilaciones que hicimos anteriormente y ver como se ve el trabajo. Lo más importante a recordar mientras crea su compilación final es que cuando renderice su animación, el editor de secuencia solo 've' la capa superior de vídeo. Esto significa que tiene que asegurarse de que sea un corte listo para usarse, o un efecto como *ADD* (añadir) que combine algunas capas subyacentes. La base de la compilación será el túnel variante. Añada algunos duplicados del meta corte del túnel y sitúelos en el canal uno. Combinelos en un metacorte. No se preocupe por la longitud exacta de la animación aun; siempre podrá duplicar mas cortes de túnel. Encima de ellas, sitúe el cubo alámbrico desfasado en el canal 2. Añada canal 1 a canal 2 y sitúe el efecto add (añadir) en el canal 3 (*Combinando el túnel y el cubo alámbrico.*).



Combinando el túnel y el cubo alámbrico.

Ahora también queremos añadir el cubo sólido a la animación. Sitúelo en el canal 4, superponiéndolo a la animación alámbrica en el canal 2. Añadalo a la animación del túnel en la capa uno. Aquí es donde las cosas se empiezan a volver algo complicadas; si quisiera dejar algo como esto, la animación en el canal 5 (el cubo sólido junto con el tubo) debería ignorar la animación en el canal 2 (el cubo alámbrico) y el cubo alámbrico se hará visible tan pronto como el cubo sólido se muestre. Para solucionar esto, añada el canal 3 al canal 5 (*Combinando el túnel, y los cubos alámbrico y sólido.*).



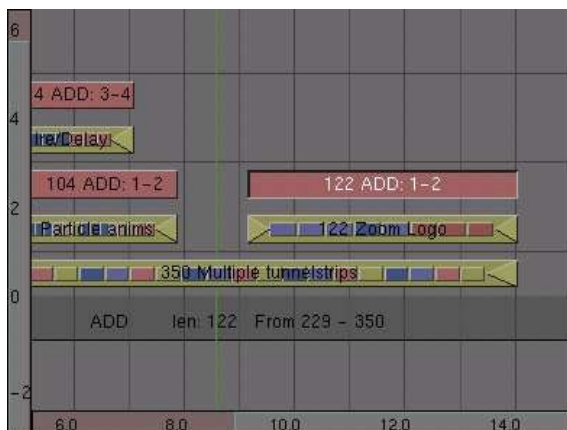
Combinando el túnel, y los cubos alámbrico y sólido.

También es posible que necesite aplicar algunas operaciones de "añadir" extra para arreglar algunas partes desaparecidas del vídeo. Esto se verá mucho más después de que haya renderizado la secuencia final. Deslice la ventana del editor de secuencia un poco a la izquierda y añada el metacorte con la animación del logo con partículas en ella. Coloque este corte en la capa 2 y sitúe el efecto añadir (add) en la capa 3. Para dar algo de variación, duplique la animación alámbrica y combínela con el "add" (añadido) en la capa 3 (*Añadiendo la animación del logo con partículas.*).



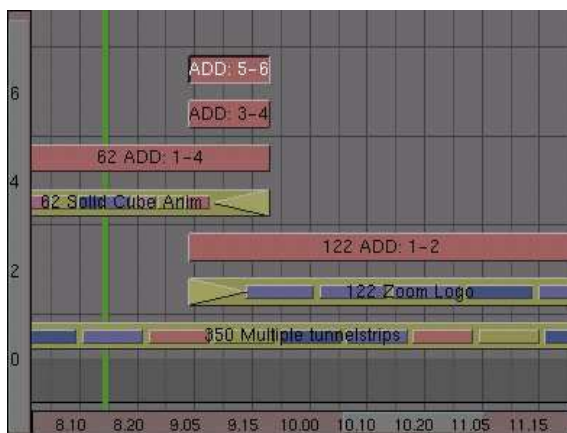
Añadiendo la animación del logo con partículas.

Ahora vaya al final del corte de la animación del túnel. Debería haber suficiente sitio para colocar la animación del logo ampliado al final y después mantener algo de espacio libre (*Añadiendo la animación del logo con zoom.*). Si no es así, seleccione el corte del túnel, presione **TAB** y añada un duplicado de la animación al final. Presione **TAB** de nuevo para salir del modo edición "meta".



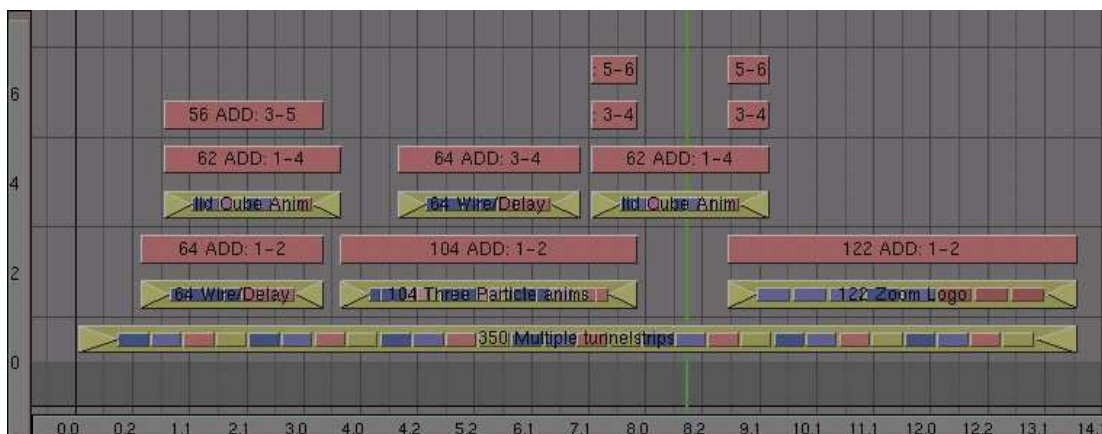
Añadiendo la animación del logo con zoom.

Si sigue habiendo algo de espacio a la derecha, podemos añadir una copia de la animación del cubo sólido. Para conseguir que se muestre correctamente, tendrá que aplicarle dos canales "add" (añadir): uno para combinarla con la animación del logo con partículas y otra para combinarlo con la animación del logo con zoom (*El último detalle.*).



El último detalle.

La secuencia final shows the complete sequence.



La secuencia final.

Conclusión

Ahora estamos preparados para renderizar la composición final de vídeo. Para decirle a Blender que use la información del editor de Secuencia mientras renderiza, seleccione el botón *Do Sequence* (hacer secuencia) en la ventana de botones de renderizado. Después de ésto, renderice y guarde su trabajo de animación como antes (asegúrese de no sobrescribir ninguna de sus secuencias AVI).

Plugins para el Editor de Secuencia

Como hemos dicho antes, Blender es ampliable mediante un sistema de plugins, y podemos encontrar dos tipos de plugins: De texturas y de secuencia. Los plugins de secuencia trabajan con cortes de una manera muy similar a las operaciones convencionales *ADD*, *CROSS* etc. Debe tener al menú un corte seleccionado y presionar **SHIFT-A**>>Plugin o la entrada de menú *Add*>>*Effect*>>Plugin. Esto abre una ventana de selección de archivos en la cual podrá seleccionar el plugin deseado. Las funcionalidades de los plugins varían mucho, de modo que no es posible describirlos aquí. Al contrario que los plugins de texturas, los plugins de secuencia no tienen botones en ninguna ventana, así que sus parámetros se suelen configurar mediante **N**.

Editor de Secuencia de Audio

Desde Blender 2.28 hay una herramienta de secuencia de Audio (aún limitada). Puede añadir archivos WAV vía el menú **SHIFT-A** y seleccionando la entrada *Sound*. Un corte verde de audio será creado. De momento no hay características de mezclado de 'alto nivel'. Puede colocar tantos cortes de audio como desee, y el resultado será la mezcla de todas ellas. Puede dar a cada corte su propio nombre y ganancia (en dB) vía el menú *NKEY*. Esto también le permite silenciar o dar balance; -1 totalmente a la izquierda, +1 totalmente a la derecha. Un IPO de 'Volumen' puede ser añadido al corte en la Ventana IPO como hemos hecho para cortes de efectos. El canal *Fac* es el volumen aquí. Los frames 1-100 corresponden con toda la longitud de la muestra, 1.0 es el volumen total, y 0.0 para completo silencio. Blender todavía no puede mezclar sonido en el resultado final del Editor de Secuencia. El resultado del Editor de Secuencia seguirá siendo un archivo de vídeo, si el botón *ANIM* en el panel *Anim* del menú escena está activado, el submenú render es usado como se describe, o un archivo WAV separado, conteniendo la secuencia entera de audio, en el mismo directorio que el archivo de vídeo y con el mismo nombre pero extensión .WAV. El archivo de audio es creado con el botón *MIXDOWN* en el botón *Sequencer* (secuenciador) del menú escena, submenú sonido. Puede mezclar vídeo y audio después con un programa externo. La ventaja de usar el editor de secuencia de Blender radica en la sencilla sincronización debida a el secuenciamiento de frames y sonido en la misma aplicación.


PARTE XV - EXTENDIENDO BLENDER

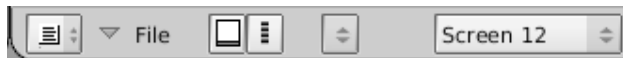
Al contrario de muchos programas con los que puedes estar familiarizado, Blender no es monolítico y estático. Puedes extender sus funcionalidades incluso sin tener que modificar las fuentes y recompilarlas. Hay dos maneras de realizar dicha extensión: Scripts en Python y Plugins, siendo el primero de los dos el preferido y mayormente usado. Esta parte describirá ambas técnicas.

Capítulos


- Scripts en Python
- Sistema de plugins de Blender

Scripts en Python

Blender tiene una característica muy poderosa que a menudo es pasada por alto. Esta característica es un intérprete de Python totalmente funcional. Esto le permite a cualquier usuario añadir funcionalidades a Blender escribiendo un simple script de Python. Python es un lenguaje de programación interpretado, interactivo, y orientado a objetos. Incorpora módulos, excepciones, tipeado dinámico, tipo de datos dinámicos de muy alto nivel y clases. Python combina un gran poder con una sintaxis muy sencilla. Está expresamente diseñado para ser usado como una extensión para las aplicaciones que necesiten una interfaz programable, y esto es por lo que Blender lo utiliza. De las dos maneras que tiene de extender Blender, siendo la otra los plugins binarios, la tratada aquí los script de Python es la más poderosa, versátil, más sencilla de comprender y robusta. Realmente es preferible usar Python que escribir un plugin. Realmente los script de Python tenían unas funcionalidades limitadas en Blender 2.25, la última de las versiones que liberó NaN. Cuando Blender se hizo código abierto muchos de los desarrolladores que se movían alrededor de la Fundación lo eligieron para trabajar y, junto con el cambio de UI, el API de Python es probablemente la parte de Blender que ha tenido un desarrollo más grande. Una total reorganización de lo que existía fue llevada a cabo y muchos nuevos módulos fueron añadidos. Dicha evolución está aún en proceso y una mejor integración está por venir en las próximas versiones de Blender. Blender tiene una ventana texto junto con los otros tipo de ventana la cual es accesible vía el botón  Text Editor del menú Tipos de Ventana o a través de **SHIFT-F11**. Una ventana de texto recién abierta es gris y está vacía, y tiene una barra de herramientas muy sencilla (*Barra de herramientas de texto*). De izquierda a derecha están el botón estándar de selección de ventana y el menú ventana. Luego está el botón de pantalla completa, seguido por un botón que muestra/oculta el número de líneas del texto y el botón Menú.



Barra de herramientas de texto.

El botón Menú () permite seleccionar que buffer de texto tiene que ser mostrado, y también le permite crear un nuevo buffer o cargar un fichero de texto. Si elige cargar un fichero en la ventana texto temporalmente se crea una ventana de selección de ficheros, con las funciones usuales. Una vez que el buffer de texto está en la ventana texto, esta se convierte

en un editor de texto muy simple. Escribir en el teclado provoca que aparezca texto en el buffer de texto. Como siempre con presionar **LMB** arrastrar y liberar **LMB** se selecciona el texto. Los siguientes comandos del teclado son aplicables:

- **ALT-C** o **CTRL-C** - Copia el texto marcado al portapapeles;
- **ALT-X** o **CTRL-X** - Corta el texto marcado al portapapeles;
- **ALT-V** or **CTRL-V** - Pega el texto marcado desde portapapeles en la posición del cursor en la Ventana Texto;
- **ALT-S** - Guarda el texto como un fichero de texto, una Ventana de Selección de Ficheros aparecerá;
- **ALT-O** - Carga un texto, una Ventana de Selección de Ficheros aparecerá;
- **ALT-F** - Emergerá la herramienta buscar;
- **SHIFT-ALT-F** o **RMB** - Emergerá el Menú Fichero para la Ventana de Texto;
- **ALT-J** - Emergerá un panel donde puede especificar un número de línea para situar el cursor;
- **ALT-P** - Ejecuta el texto como un script de Python;
- **ALT-U** - Deshacer;
- **ALT-R** - Rehacer;
- **CTRL-R** - Reabrir (recargar) el buffer actual;
- **ALT-M** - Convierte el contenido de la ventana de texto en texto 3D (max. 100 caracteres);

El portapapeles de Blender que corta/copia/pega está *separado* del portapapeles de Windows así que *no podrá* copiar desde o hacia Blender. Para acceder a su portapapeles de Windows utilice **SHIFT-CTRL-C** **SHIFT-CTRL-V**. Para eliminar un buffer de texto tan sólo presione el botón 'X'. próximo al nombre del buffer, tal y como hace para los materiales, etcétera. La más importante combinación de teclas es **ALT-P** la cual hace que el contenido del buffer sea pasado al intérprete de Python de Blender. La siguiente sección le mostrará un ejemplo de script de Python. Antes de que continúe tenga en cuenta que Blender viene con un intérprete de Python muy sencillo con unos pocos módulos totalmente específicos de Blender, que están descritos en la ****REFERENCIA****.

Otros usos para la ventana de texto: La ventana de texto se utiliza también cuando quieres compartir tus ficheros .blend con la comunidad o con tus amigos. Puedes usarla para escribir un texto aclaratorio de los contenidos de tu fichero. Esto es mucho más interesante que tener dicho fichero en un archivo separado. Asegurate de mantenerlo visible mientras salves tu fichero!. Si compartes el archivo con la comunidad y quieres hacerlo bajo alguna licencia puedes escribir dicha licencia en una ventana texto.

Para tener acceso a los módulos estándar de Python necesitarás tener instalado Python. Puedes descargarlo de <http://www.python.org>.

Configurando la variable del sistema PYTHONPATH

Asegurate de comprobar en <http://www.blender.org> cual versión *exacta* de Python es con la que fue construida Blender para evitar posibles problemas de compatibilidad. Blender también debe conocer *donde* se ha realizado dicha instalación. Esto se hace definiendo una variable entorno **PYTHONPATH**.

Configurando PYTHONPATH en Win95,98,Me

Una vez que haya instalado Python en, digamos, *C:\PYTHON22* debe editar el fichero *C:\AUTOEXEC.BAT* con su editor de textos favorito y añadir una línea:

```
SET
PYTHONPATH=C:\PYTHON22;C:\PYTHON22\DLLS;C:\PYTHON22\LIB;C:\PYTHON22\LIB\LIB
-TK
```

y reiniciar el sistema.

Configurando PYTHONPATH en WinNT,2000,XP

Una vez que haya instalado Python en, digamos, *C:\PYTHON22*. Vaya al icono "Mi PC" en el escritorio, **RMB** y seleccione *Properties*. Seleccione la pestaña *Advanced* y presione el botón *Environment Variables*. Debajo de la caja de Variables del Sistema, (la segunda caja), pinche en *New*. Si no eres administrador posiblemente no seas capaz de hacer esto. En este caso pinche en *New* en la caja superior. Ahora, en la caja "Variable Name" (Nombre de variable), escriba *PYTHONPATH* y en la caja "Variable Value" (Valor de la variable), escriba:

```
C:\PYTHON22;C:\PYTHON22\DLLS;C:\PYTHON22\LIB;C:\PYTHON22\LIB\LIB-TK
```

Pulse OK repetidamente para salir de todos los diálogos. Dependiendo del sistema operativo quizás tengas que reiniciar.

Configurando PYTHONPATH en Linux y otros UNIXes

Normalmente tendrás Python ya instalado. Si no fuese así, hazlo. Ahora solo tendrás que descubrir donde está instalado. Esto es sencillo, solo arranque una shell interactiva de Python simplemente abriendo una shell y escribiendo *python* en ella. Escriba los siguientes comandos:

```
>>> import sys
>>> print sys.path
```

y anote lo que se muestre por pantalla que debería ser algo como

```
['', '/usr/local/lib/python2.2', '/usr/local/lib/python2.2 /plat-linux2',
'/usr/local/lib/python2.0/lib-tk', '/usr/local/lib/python2.0/lib-dynload',
'/usr/local/lib/python2.0/site-packages']
```

Añada esto a su fichero favorito *rc* (fichero de configuración) como una variable de entorno. Por ejemplo, añada en su *.bashrc* la siguiente línea

```
export PYTHONPATH=/usr/local/lib/python2.2:/usr/local/lib/python2.2/plat-
linux2:/usr/local/lib/python2.2/lib-tk:/usr/local/lib/python2.2/lib-
dynload:/usr/local/lib/python2.0/site-packages
```

todo en una única línea. Abra una nueva shell, o salga e introduzca de nuevo el login.

Un ejemplo funcional de Python

Ahora que ha visto que Blender es extensible vía script de Python y que tiene los elementos básicos para el manejo de scripts y sabe como lanzarlos, antes de estrujar su cerebro con la API de Python echemos un vistazo a un pequeño ejemplo. Realizaremos un pequeño script para producir polígonos. Este script duplica lo que hace `SPACEAdd>>Mesh>>Circle`, pero creará polígonos rellenos, no sólo con el perímetro exterior. Para hacer el script aún más completo este exhibirá una interfaz gráfica de usuario (GUI) escrita completamente a través de la API de Blender.

Cabeceras, importación de los módulos y variables globales

Las primeras 32 líneas de código son mostradas abajo.

Cabecera del script

```
001 #####
002 #
003 # Script demostración para la Guia de Blender 2.3
004 #
005 #####S68
006 # Este script genera poligonos. Es bastante inútil
007 # ya que puedes hacer lo mismo con ADD->Mesh->Circle
008 # pero es un buen ejemplo, y además los
009 # polígonos están 'rellenos'.
010 #####
011
012 #####
013 # Importación de los módulos
014 #####
015
016 import Blender
017 from Blender import NMesh
018 from Blender.BGL import *
019 from Blender.Draw import *
020
021 import math
022 from math import *
023
024 # Parámetros del Polígono
025 T_NumberOfSides = Create(3)
026 T_Radius        = Create(1.0)
027
028 # Events
029 EVENT_NOEVENT = 1
030 EVENT_DRAW    = 2
031 EVENT_EXIT    = 3
032
```

Después de necesarios comentarios que indican lo que hace el script se encuentra la importación de los módulos de Python (líneas 016-022). *Blender* es el módulo principal de la API de Python de Blender. *NMesh* es el módulo que proporciona acceso a las mallas de Blender, mientras que *BGL* y *Draw* dan acceso a las funciones y constantes de OpenGL y a la interfaz de ventanas de Blender, respectivamente. El módulo *math* es el módulo matemático de Python, pero dado que ambos módulos 'math' y 'os' están incluidos dentro de Blender no

necesita una completa instalación de Python para usarlos!. Los polígonos se definen a través del número de lados y de su radio. Estos parámetros tienen valores que deben ser definidos por el usuario a través del GUI, líneas (025-026), con lo que se deben crear dos 'botones genéricos', con sus valores por defecto. Finalmente, los objetos GUI trabajarán con eventos y también los generarán. Los identificadores de los eventos son enteros que serán definidos por el programador. Es una buena práctica utilizar nombres fáciles de recordar para los eventos, como se hace en la líneas (029-031).

Dibujando el GUI

El código fuente que se encarga del dibujo del GUI está en la función *draw* (dibujo) (*Dibujando el GUI*).

Dibujando el GUI

```
033 #####
034 # Dibujando el GUI
035 #####
036 def draw():
037     global T_NumberOfSides
038     global T_Radius
039     global EVENT_NOEVENT,EVENT_DRAW,EVENT_EXIT
040
041     ##### Titles
042     glClear(GL_COLOR_BUFFER_BIT)
043     glRasterPos2d(8, 103)
044     Text("Demo Polygon Script")
045
046     ##### Parameters GUI Buttons
047     glRasterPos2d(8, 83)
048     Text("Parameters:")
049     T_NumberOfSides = Number("No. of sides: ", EVENT_NOEVENT, 10, 55,
210, 18,
050                             T_NumberOfSides.val, 3, 20, "Number of sides of out
polygon");
051     T_Radius          = Slider("Radius: ", EVENT_NOEVENT, 10, 35, 210,
18,
052                             T_Radius.val, 0.001, 20.0, 1, "Radius of the
polygon");
053
054     ##### Draw and Exit Buttons
055     Button("Draw",EVENT_DRAW , 10, 10, 80, 18)
056     Button("Exit",EVENT_EXIT , 140, 10, 80, 18)
057
```

Las líneas (037-039) proporcionan acceso a los datos globales. El verdadero meollo del asunto comienza a partir de las líneas (042-044). La ventana de OpenGL es inicializada, y su actual posición configurada a x=8, y=103. El origen de esta referencia es la esquina inferior izquierda de la ventana de scripts. Entonces el título *Demo Polygon Script* es mostrado. Una cadena de caracteres es escrita en las líneas (047-048), entonces se crean los botones para introducir los parámetros. El primero (líneas 049-050) es un botón numérico, exactamente como esos que hay en la ventana de botones de Blender. Para comprender todos los parámetros por favor diríjase a la referencia de la API. Básicamente está la etiqueta del botón, el evento generado por el botón, su localización (x,y) y sus dimensiones (anchura, altura), su valor, que es un dato perteneciente al objeto mismo, los valores máximos y mínimos

permitibles y un texto que aparecerá como ayuda mientras sitúa el cursor sobre el botón, como una etiqueta de ayuda. Las líneas (051-052) definen un botón numérico con un deslizador, con una sintaxis muy similar. Las líneas (055-056) finalmente crean el botón *Draw* (Dibujar) el cual creará los polígonos y un botón *Exit* (Salir).

Manejando eventos

El GUI no se ha dibujado, y no trabajará, hasta que se escriba un manejador de eventos apropiado y se registre (*Manejando eventos*).

Manejando eventos

```
058 def event(evt, val):
059     if (evt == QKEY and not val):
060         Exit()
061
062 def bevent(evt):
063     global T_NumberOfSides
064     global T_Radius
065     global EVENT_NOEVENT, EVENT_DRAW, EVENT_EXIT
066
067     ##### Manages GUI events
068     if (evt == EVENT_EXIT):
069         Exit()
070     elif (evt == EVENT_DRAW):
071         Polygon(T_NumberOfSides.val, T_Radius.val)
072         Blender.Redraw()
073
074 Register(draw, event, bevent)
075
```

Las líneas (058-060) definen el manejador de eventos del teclado, en este caso respondiendo a la **Q** con una sencilla llamada a *Exit()*. Hay más interesantes líneas en (062-072) donde se manejan los eventos del GUI. Cada vez que un botón del GUI es usado esta función es llamada, con el número de evento definido en los parámetros del botón. El núcleo de esta función es una estructura de "selección" que ejecuta diferentes códigos dependiendo del número de evento. Como última llamada, se invoca a la función *Register*. Esto efectivamente dibuja el GUI y comienza el bucle que captura los eventos.

Manejando Mallas

Finalmente, *Cabecera del script* muestra la función principal, la que crea el polígono. Es una simple edición de la malla, pero muestra muchos puntos importantes de la estructura interna de datos de Blender.

Cabecera del script

```
076 #####
077 # Cuerpo principal
078 #####
079 def Polygon(NumberOfSides,Radius):
080
081     ##### Creates a new mesh
082     poly = NMesh.GetRaw()
083
```

```

084     ##### Populates it of vertices
085     for i in range(0,NumberOfSides):
086         phi = 3.141592653589 * 2 * i / NumberOfSides
087         x = Radius * cos(phi)
088         y = Radius * sin(phi)
089         z = 0
090
091         v = NMesh.Vert(x,y,z)
092         poly.verts.append(v)
093
094     ##### Adds a new vertex to the center
095     v = NMesh.Vert(0.,0.,0.)
096     poly.verts.append(v)
097
098     ##### Connects the vertices to form faces
099     for i in range(0,NumberOfSides):
100         f = NMesh.Face()
101         f.v.append(poly.verts[i])
102         f.v.append(poly.verts[(i+1)%NumberOfSides])
103         f.v.append(poly.verts[NumberOfSides])
104         poly.faces.append(f)
105
106     ##### Creates a new Object with the new Mesh
107     polyObj = NMesh.PutRaw(poly)
108
109     Blender.Redraw()

```

La primera línea importante aquí es la número (082). Aquí hay un nuevo objeto malla, donde se crea el *poly* polígono. El objeto malla está constituido de una lista de vértices y de una lista de caras, además de otras interesantes cosas. Para nuestros propósitos los vértices y la lista de caras será lo que necesitaremos. Por supuesto la malla recientemente creada ésta vacía. El primer bucle (lines 085-092) calculará la localización x,y,z de los vértices *NumberOfSides* necesarios para definir el polígono. Para un plano su coordenada 'z' será cero para todos los vértices. La línea (091) llama al método *Vert* de *NMesh* para crear un nuevo objeto vértice de coordenadas (x,y,z). Tal objeto es entonces añadido (línea 096) en la lista de *verts* vértices de la malla *poly* polígono. Finalmente en las líneas (095-096) un último vértice es añadido al centro. Las líneas (099-104) conectan los vértices para crear las caras. No será necesario crear todos los vértices primero y luego las caras. Podrás crear una nueva cara tan pronto como todos los vértices estén disponibles. La línea (100) crea una nueva cara. Un objeto cara tiene su propia lista de vértices *v* (hasta 4) definidos. Las líneas (101-103) añaden 3 vértices a la lista *f.v* originalmente vacía. Los vértices son dos vértices adyacentes del polígono y el vértice central. Estos vértices deben de ser tomados de la lista de *verts* vértices de la malla. Finalmente la línea (104) añade la cara recién creada a la lista de *faces* caras de nuestra malla *poly* polígono.

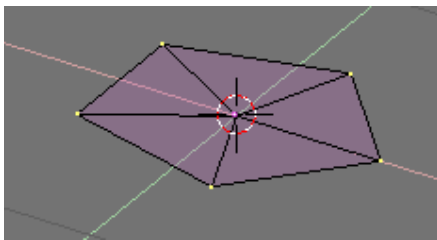
Conclusiones

Si crea su fichero *polygon.py* conteniendo el código anteriormente descrito y lo carga dentro de una ventana de texto de Blender, como ha aprendido en la sección anterior y presiona **ALT-P** en dicha ventana para lanzarlo verá como la ventana desaparece y se vuelve gris. En la esquina inferior izquierda el GUI se dibujará (*El GUI de nuestro ejemplo.*).



El GUI de nuestro ejemplo.

Seleccionando, por ejemplo, 5 vértices y un radio de 0.5, y presionando el botón *Draw* aparecerá un pentágono en el plano xy de la ventana 3D (*El resultado de nuestro script de ejemplo.*).



El resultado de nuestro script de ejemplo.

Referencia de Python

La documentación de referencia del lenguaje de programación Python para Blender es un libro en si mismo. Por razones de espacio no se incluye en este documento.

Scripts de Python

Hay más de 100 scripts para Blender disponibles en Internet. Como los plugins, los scripts son muy dinámicos, cambiando su interfaz, sus funcionalidades y su localización en Internet muy rápidamente, así que para una lista actualizada de ellos por favor dirijase a uno de estos dos sitios, www.blender.org o www.elysiun.com.

Sistema de plugins de Blender

Esta sección muestra una referencia detallada de como escribir plugins de Secuencia y de Textura para Blender.

Escribiendo un plugin de Textura

En esta sección escribiremos un plugin de textura básico y después enseñaremos a usarlo. Los elementos básicos detrás de un plugin de textura consisten en que tu estás dando algunos valores de entrada, tales como posición, etc y alguna información más. Entonces dependiendo del tipo de plugin de textura se te devolverá información sobre la intensidad, el color y/o las normales. Tanto los archivos necesarios para desarrollar los plugins como unos pocos plugins se pueden encontrar en el directorio `blender/plugins`. Alternativamente puede obtener más

plugins de <http://www.cs.umn.edu/~mein/blender/plugins>. Los plugins son utilizados (cargados/llamados) en Blender a través de `dlopen()`. Para aquellos que no están familiarizados con el sistema `dlopen` diremos que este permite a un programa (Blender) usar un objeto compilado como si éste fuese parte del programa mismo, similar a cuando se enlazan librerías dinámicamente, exceptuando que el objeto a cargar se determina cuando se está ejecutando el programa. La ventaja de usar el sistema `dlopen` para los plugins consiste en que es muy rápido acceder a una función, y además no hay sobrecarga en la interfaz del plugin, lo cual es crítico cuando (como en el caso de los plugins de textura) el plugin se puede llamar varios millones de veces en un único render. La desventaja de este sistema es que el plugin funciona como si realmente fuese parte de Blender, con lo que si el plugin se muere, Blender muere. Los ficheros de cabecera que se encuentran en el subdirectorio `plugin/include/` de la instalación de Blender están documentados mostrando las funcionalidades que se proporcionan a los plugins. Dichas funcionalidades incluyen las funciones de la librería `Imbuf` para cargar y trabajar con imágenes y buffers de imagen, y funciones de ruido y turbulencia para un texturizado consistente.

Especificación

- **#include <plugin.h>**

Todos los plugins de Blender deben incluir este fichero cabecera, que contiene todas las estructuras y definiciones necesarias para trabajar correctamente con Blender.

- **char name[]="Tiles";**

Un string (cadena de caracteres) conteniendo el nombre del plugin, este valor se usará para indicar el título de la textura en la ventana de botones de textura

- **define NR_TYPES 2 char stnames[NR_TYPES][16]= {"Square", "Deformed"};**

A los Plugins se les permite tener subtipos separados para variaciones mínimas en algoritmos - por ejemplo la textura "clouds (nubes)" que está por defecto en Blender tiene los subtipos "Default" y "Color". `NR_STYPES` debería ser definido con el número de subtipos que el plugin requiera, y además a cada subtipo se le tiene que dar un nombre. Cada plugin debe tener al menos 1 subtipo y el nombre de ese subtipo.

- **VarStruct varstr[]= {...};**

"varstr" contiene toda la información que Blender necesita para mostrar los botones del plugin. Los botones para los plugin pueden ser de tipo numérico para introducir datos, o de tipo texto para introducir comentarios u otra información. Se pueden utilizar como máximo 32 variables en los plugins. Cada estructura `VarStruct` consta de un tipo, un nombre, un rango de información y una nota emergente sobre la herramienta.

El tipo define el tipo de dato para cada botón, y la manera en la que se muestra dicho botón. Para botones numéricos este valor debería ser una combinación (ORed) de `INT` o `FLO` para el formato del número y `NUM`, `NUMSLI` o `TOG` para el tipo de botón. Los botones de texto son de tipo `LABEL`.

El nombre es lo que se mostrará en el botón y está limitado a 15 caracteres.

El rango de información consiste en 3 números reales que definen los valores máximo, mínimo y "por defecto" para el botón. Para los botones TOG el mínimo es configurado cuando el botón se haya presionado y el máximo cuando está sin presionar.

La nota emergente (tool tip) es una cadena de caracteres que se mostrará cuando el puntero esté encima del botón (si el usuario tiene los tool tips activos). Tiene un límite de 80 caracteres y se tiene que poner a NULL (") si no se utiliza.

- **typedef struct Cast {...};**

La estructura Cast se usa en la llamada a la función doit, y simplifica el acceso de cada plugin a los valores de los datos. La estructura Cast debería contener, en este orden, un entero o numero real por cada botón definido en varstr, incluidos los botones de texto. Típicamente deberían tener el mismo nombre que el botón para permitir una búsqueda o una referencia sencilla.

- **float result[8];**

El vector result se usa para enviar y recibir información al plugin. Los valores de result están definidos de esta manera:

Result Indice	Significado	Rango
result[0]	Valor de la intensidad	0.0 to 1.0
result[1]	Valor del color rojo	0.0 to 1.0
result[2]	Valor del color verde	0.0 to 1.0
result[3]	Valor del color azul	0.0 to 1.0
result[4]	Valor del color transparente (alpha)	0.0 to 1.0
result[5]	Valor del desplazamiento normal en X	
result[6]	Valor del desplazamiento normal en Y	
result[7]	Valor del desplazamiento normal en Z	

El plugin siempre devuelve el valor de la intensidad. Devolver el color (RGB) o la normal es opcional, y se debería indicar con el "return" de la funcion doit(), siendo "1" para (RGB) y "2" para (Normal).

Antes de que se llame al plugin, Blender asigna el resultado de renderizar normalmente en result[5], result[6] y result[7].

- **float cfra**

El valor cfra actualizado por Blender a su valor actual antes de que todos los renders se hagan. Este valor es el numero de frame +/- .5 dependiendo de las configuraciones del campo.

- **plugin_tex_doit prototype**

La funcion plugin_tex_doit deberia ser prototipada para que pueda ser usada por la funcion getinfo. No necesita cambiar esta linea.

- **plugin_tex_getversion**

Esta funcion deberia estar en cada plugin para que pueda ser cargado correctamente. No debería cambiar esta función.

- **plugin_but_changed**

Esta función se usa para pasar información sobre que botones cambia el usuario en la interface. La mayoría de los plugins no necesitan esta función, solo cuando el interface permita al usuario modificar algunas variables que furze al plugin a recalcularse (una tabla hash aleatoria por ejemplo).

- **plugin_init**

Si fuese necesario los plugins pueden utilizar esta funcion para inicializar datos internos. NOTA: Esta función de iniciacion se puede llamar varias veces si el mismo plugin de textura se copia. En esta función, no inicialice datos globales especificos a una unica instancia de un plugin.

- **plugin_getinfo**

Esta función se usa para comunicar información a Blender. Nunca deberías cambiarla.

- **plugin_tex_doit**

La función doit es la responsable de devolver información a Blender sobre pixeles requeridos.

Los Argumentos

- **int stype**

Este es el número de subtipos seleccionados, mira las entradas NR_TYPES y char stypes anteriores.

- **Cast *cast**

La estructura Cast que contiene los datos del plugin, mira la entrada **Cast** anterior.

- **float *texvec**

Esto es un puntero a 3 floats, los cuales son las coordenadas para las cuales un valor de la textura debe retornarse.

- **float *dxt float *dyt**

Si los punteros son no-NULL apuntarán a 2 vectores (2 arrays de 3 floats) que definen el tamaño del valor textura requerida en el espacio pixel Apuntan solo a no-NULL cuando el OSA esté activado, y se usan para calcular adecuadamente el antialiasing.

La función doit debería rellenar el array result y devolver 0, 1, 2 o 3 dependiendo de que valores hayan rellenado el array. La función doit debería siempre rellenarse con el valor intensidad. Si la función asigna el valor "color" debería devolver 1, si asigna el valor "normal" debería devolver 2 y si asigna todo debería devolver 3.

Interacción Texturas/Materiales

Blender es algo diferente de la mayoría de paquetes 3D en lo que se refiere a la separación lógica entre materiales y textura En Blender las texturas son objetos que devuelven ciertos valores, son generadores de señales de hecho. Los materiales controlan el mapeado de texturas en los objetos, lo que está afectado, en que cantidad, en que manera, etc. Plugins adecuadamente diseñados deberían incluir únicamente variables que afecten a la señal devuelta no al mapeado de ella. Es mejor incluir botones para controlar el escalado, el rango, los ejes, etc únicamente cuando hagan a la textura más fácil de usar (en caso del botón tamaño en el plugin Tiles) o aceleran el cálculo (los subtipos Intensity/Color/Bump en el plugin Clouds2). De otra manera los botones del material hacen a esos botones redundantes, y la interfaz incrementa innecesariamente su complejidad.

Un plugin de textura genérico:

```
#include "plugin.h"

/* Nombre de la textura */
char name[24]= "";

#define NR_TYPES 3
char stnames[NR_TYPES][16]= {"Intens","Color", "Bump"};
/* Estructura para los botones,
 * butcode name default min max 0
 */

VarStruct varstr[]= {
    {NUM|FLO, "Const 1", 1.7, -1.0, 1.0, ""},
};

typedef struct Cast {
    float a;
} Cast;

float result[8];
```



```

float cfra;
int plugin_tex_doit(int, Cast*, float*, float*, float*);

/* Funciones fijas */
int plugin_tex_getversion(void) {
    return B_PLUGIN_VERSION;
}

void plugin_but_changed(int but) { }

void plugin_init(void) { }

void plugin_getinfo(PluginInfo *info) {
    info->name= name;
    info->stypes= NR_TYPES;
    info->nvars= sizeof(varstr)/sizeof(VarStruct);

    info->snames= stnames[0];
    info->result= result;
    info->cfra= &cfra;
    info->varstr= varstr;

    info->init= plugin_init;
    info->tex_doit= (TexDoit) plugin_tex_doit;
    info->callback= plugin_but_changed;
}

int plugin_tex_doit(int stype, Cast *cast, float *texvec, float *dxt,
    float *dyt) {

    if (stype == 1) {
        return 1;
    } if (stype == 2) {
        return 2;
    }
    return 0;
}

```

Nuestras modificaciones:

El primer paso es tener un plan. Para qué sirve este plugin y como los usuarios interactuarán con él. Para este ejemplo crearemos una textura sencilla que crea un simple patrón de ladrillos. Ahora copiaremos nuestro plugin generico al archivo cube.c y rellenaremos los espacios vacíos. Es siempre una buena idea añadir algunos comentarios. Lo primero de todo es decir a los usuarios qué hace el plugin, de donde pueden obtener una copia, con quién deberían contactar para mejorarlo o para informar de fallos y cual es la licencia del código. Cuando uses comentarios asegurate de usar `/* */`. Los plugins están en C y algunos compiladores de C no aceptan los comentarios con el estilo de C++ (`//`).

```

/*
Descripción: Este plugin es un ejemplo de plugin de textura que crea
un simple patrón de ladrillos.

```

```

Toma dos valores, el tamaño brick y el tamaño mortar.
El tamaño brick es el tamaño de cada ladrillo.
El tamaño mortar es el tamaño que hay entre los ladrillos (cemento).

```

```

Autor: Kent Mein (mein@cs.umn.edu)
Sitio web: <nowiki>http://www.cs.umn.edu/~mein/bender/plugins

```

Licencia: Dominio Público
 Última modificación: Tue Oct 21 0:57:13 CDT 2003
 */
 </nowiki>

Lo siguiente que necesitamos será rellenar el nombre, realmente debería ser el mismo que tu fichero .c, preferiblemente descriptivo, con menos de 23 caracteres, sin espacios y todas las letras en minúsculas.

```
char name[24]= "cube.c";
```

Vamos a mantener este plugin simple, solo tiene un tipo que está relacionado con la intensidad. Así que necesitamos lo siguiente:

```
#define NR_TYPES 1
char stnames[NR_TYPES][16]= {"Default"};
```

Para nuestro interfaz vamos a permitir a la gente cambiar el tamaño del ladrillo, el espaciado entre ellos y los valores de intensidad devueltos por el ladrillo y el espaciado entre ladrillos. Para ello necesitamos modificar varstr y Cast. Cast debería tener una variable por cada entrada que haya en varstr.

```
/* Estructura para los botones,
 * butcode      name      default  min  max  Tool tip
 */
VarStruct varstr[]= {
  {NUM|FLO,  "Brick",      .8,      0.1,   1.0, "Size of Cell"},
  {NUM|FLO,  "Mortar",     .1,      0.0,   0.4, "Size of boarder in cell"},
  {NUM|FLO,  "Brick Int",  1,       0.0,   1.0, "Color of Brick"},
  {NUM|FLO,  "Mortar Int", 0,       0.0,   1.0, "Color of Mortar"},
};

typedef struct Cast {
    float brick,mortar, bricki, mortari;
} Cast;
```

Ahora rellenaremos la función plugin_tex_doit. Basicamente queremos romper nuestra textura en "bloques" que estarán formados de un ladrillo y del espaciado a lo largo de las aristas del ladrillo (cemento) y entonces determinar si estamos en el ladrillo o estamos en el cemento. El código siguiente hace esto.

```
int plugin_tex_doit(int stype, Cast *cast, float *texvec, float *dxt,
  float *dyt) {
  int c[3];
  float pos[3], cube;

  /* configura el tamaño de nuestro bloque */
  cube = cast->brick + cast->mortar;

  /* necesitamos determinar donde estamos dentro del ladrillo actual. */
  c[0] = (int)(texvec[0] / cube);
  c[1] = (int)(texvec[1] / cube);
  c[2] = (int)(texvec[2] / cube);

  pos[0] = ABS(texvec[0] - (c[0] * cube));
  pos[1] = ABS(texvec[1] - (c[1] * cube));
  pos[2] = ABS(texvec[2] - (c[2] * cube));
```

```

/* Descubrir si estamos en una posición de cemento o no. */
if ((pos[0] <= cast->mortar) || (pos[1] <= cast->mortar) ||
    (pos[2] <= cast->mortar)) {
    result[0] = cast->mortari;
} else {
    result[0] = cast->bricki;
}
return 0;
}

```

Una cosa que destacar, la función ABS está definida en una cabecera que está en `plugins/include`. En ese lugar hay otras funciones de uso común por lo que sería aconsejable que echara un vistazo allí.

Compilando:

`bmake` es una utilidad simple (shell script) para ayudar a la compilación y desarrollo de plugins, y puede encontrarse en el subdirectorio `plugins` del directorio de instalación de Blender. Se invoca mediante: `bmake (nombre_plugin.c)` e intentará enlazar las librerías adecuadas y compilar adecuadamente para su sistema el archivo C especificado. Si usted está intentando desarrollar plugins en una máquina windows, `bmake` puede no funcionar en este caso; debería usar `lcc`. Puede usar lo siguiente para compilar un plugin con `lcc`, asumiendo que usted tiene sus plugins en `c:\blender\plugins`: Aquí tiene un ejemplo de cómo se compilaría el plugin de textura `sinus.c`. Abra una ventana DOS y haga lo siguiente: (Nota: Asegúrese de que el directorio `lcc\bin` está en su path)

```

cd c:\blender\plugins\texture\sinus
lcc -Ic:\blender\plugins\include sinus.c
lcclnk -DLL sinus.obj c:\blender\plugins\include\tex.def
implib sinus.dll

```

Escribiendo un plugin de Secuencia

En esta sección escribiremos un plugin de secuencia básico y luego seguiremos los pasos para usar un plugin de secuencia. Las bases de un plugin de secuencia son: se reciben unas entradas (1-3 buffers de imagen de entrada, así como alguna otra información), y se produce como resultado un buffer de imagen de salida. Todos los archivos necesarios para el desarrollo de plugins así como unos pocos plugins de ejemplo pueden encontrarse en el directorio `blender/plugins`. Puede conseguir adicionalmente un puñado de plugins en <http://www.cs.umn.edu/~mein/blender/plugins>.

Especificación:

- **#include <plugin.h>**

Todos los plugins de Blender deben incluir este fichero cabecera, que contiene todas las estructuras y definiciones necesarias para trabajar correctamente con Blender.

- **char name[]="Blur";**

Un string (cadena de caracteres) conteniendo el nombre del plugin, este valor se usará para indicar el título de la textura en la ventana de botones de textura.

- **VarStruct varstr[]={...};**

"varstr" contiene toda la información que Blender necesita para mostrar los botones del plugin. Los botones para los plugin pueden ser de tipo numerico para introducir datos, o de tipo texto para introducir comentarios u otra información. Se pueden utilizar como máximo 32 variables en los plugins. Cada estructura VarStruct consta de un tipo, un nombre, un rango de información y una nota emergente sobre la herramienta.

El tipo define el tipo de dato para cada botón, y la manera en la que se muestra dicho botón. Para botones numericos este valor debería ser una combinación (ORed) de INT o FLO para el formato del numero y NUM, NUMSLI o TOG para el tipo de botón. Los botones de texto son de tipo LABEL.

El nombre es lo que se mostrará en el botón y está limitado a 15 caracteres.

El rango de información consiste en 3 numeros reales que definen los valores maximo, minimo y "por defecto" para el botón. Para los botones TOG el minimo es configurado cuando el boton se haya presionado y el maximo cuando está sin presionar.

La nota emergente (tool tip) es una cadena de caracteres que se mostrará cuando el puntero esté encima del botón (si el usuario tiene los tool tips activos). Tiene un límite de 80 caracteres y se tiene que poner a NULL (") si no se utiliza.

- **typedef struct Cast {...};**

La estructura Cast se usa en la llamada a la función doit, y simplifica el acceso de cada plugin a los valores de los datos. La estructura Cast debería contener, en este orden, un entero o numero real por cada botón definido en varstr, incluidos los botones de texto. Típicamente deberían tener el mismo nombre que el botón para permitir una búsqueda o una referencia sencilla.

- **float cfra**

El valor cfra actualizado por Blender a su valor actual antes de que todos los renders se hagan. Este valor es el numero de frame +/- .5 dependiendo de las configuraciones del campo.

- **plugin_seq_doit prototype**

La funcion plugin_seq_doit deberia ser prototipada para que pueda ser usada por la funcion getinfo. No necesita cambiar esta linea.

- **plugin_seq_getversion**

Esta funcion deberia estar en cada plugin para que pueda ser cargado correctamente. No debería cambiar esta función.

- **plugin_but_changed**

Esta función se usa para pasar información sobre que botones cambia el usuario en la interface. La mayoría de los plugins no necesitan esta función, solo cuando el interface permita al usuario modificar algunas variables que fuerze al plugin a recalcularse (una tabla hash aleatoria por ejemplo).

- **plugin_init**

Si fuese necesario los plugins pueden utilizar esta funcion para inicializar datos internos. NOTA: Esta función de iniciación se puede llamar varias veces si el mismo plugin de textura se copia. En esta función, no inicialice datos globales especificos a una unica instancia de un plugin.

- **plugin_getinfo**

Esta función se usa para comunicar información a Blender. Nunca deberías cambiarla.

- **plugin_seq_doit**

La función de secuencia doit es responsable de aplicar el efecto del plugin y de copiar los datos finales en el buffer de salida.

Los argumentos

- **Cast *cast**

La estructura Cast, que contiene los datos del plugin; véase la entrada **Cast** más arriba.

- **float facf0**

El valor de la curva IPO del plugin para el offset del primer campo. Si el usuario no ha hecho una curva IPO, su rango va entre 0 y 1 para la duración del plugin.

- **float facf1**

El valor de la curva IPO del plugin para el offset del segundo campo. Si el usuario no ha hecho una curva IPO, su rango va entre 0 y 1 para la duración del plugin.

- **int x int y**

La anchura y altura de los buffers de imagen, respectivamente.

- **int x int y**

Un apuntador al primer buffer de imagen al que el plugin está vinculado. Será siempre un buffer de imagen válido.

- **Imbuf *ibuf2**

Un apuntador al segundo buffer de imagen al que el plugin está vinculado. Los plugins que usan este buffer deberían comprobar si es un buffer NULL, pues el usuario puede no haber vinculado el plugin a dos buffers.

- **Imbuf *out**

El buffer de imagen de la salida del plugin.

- **Imbuf *use**

Un apuntador al tercer buffer de imagen al que el plugin está vinculado. Los plugins que usan este buffer deberían comprobar si es un buffer NULL, pues el usuario puede no haber vinculado el plugin a tres buffers.

Estructura de imagen ImBuf La estructura ImBuf contiene siempre 32 bits de datos de píxel ABGR. Las estructuras ImBuf son siempre iguales en tamaño, indicado por los valores de **x** e **y**.

Interacción con el usuario Blender no tiene ninguna manera de saber cuántas entradas espera el plugin, por lo que le es posible a un usuario adjuntar sólo una entrada a un plugin que espera dos. Por esta razón es importante comprobar siempre los buffers que usa el plugin para asegurarnos de que son todos válidos. Los plugins de secuencia deberían incluir también una etiqueta de texto describiendo el número de entradas requeridas en la interfaz de botones.

Plugin de Secuencia Genérico:

```
#include "plugin.h"
char name[24]= "";

/* estructura para los botones,
 * butcode name default min max 0
 */

VarStruct varstr[]= {
    { LABEL, "In: X strips", 0.0, 0.0, 0.0, "" },
};

/* La estructura cast es para entrada en la función doit principal
   Varstr y Cast deben tener las mismas variables en el mismo orden */

typedef struct Cast {
    int dummy; /* debido al botón 'etiqueta' */
} Cast;

/* cfra: el fotograma actual */

float cfra;

void plugin_seq_doit(Cast *, float, float, int, int,
    ImBuf *, ImBuf *, ImBuf *, ImBuf *);

int plugin_seq_getversion(void) {
    return B_PLUGIN_VERSION;
}

void plugin_but_changed(int but) {
}

void plugin_init() {
}
```

```

void plugin_getinfo(PluginInfo *info) {
    info->name= name;
    info->nvars= sizeof(varstr)/sizeof(VarStruct);
    info->cfra= &cfra;

    info->varstr= varstr;

    info->init= plugin_init;
    info->seq_doit= (SeqDoit) plugin_seq_doit;
    info->callback= plugin_but_changed;
}

void plugin_seq_doit(Cast *cast, float facf0, float facf1, int xo, int yo,
    ImBuf *ibuf1, ImBuf *ibuf2, ImBuf *outbuf, ImBuf *use) {
    char *in1= (char *)ibuf1->rect;
    char *out=(char *)outbuf->rect;
}

```

Nuestras modificaciones:

El primer paso es diseñar un plan de juego. Qué hará este plugin, cómo interactuarán con él los usuarios. Para este ejemplo crearemos un filtro simple que tendrá un botón de deslizamiento para una intensidad de 0-255. Si alguno de los componentes R, G o B de un píxel en la imagen fuente es menos de la intensidad seleccionada, devolverá negro y alfa; de lo contrario devolverá lo que haya en la imagen. Ahora copiaremos nuestro plugin genérico a `simpfilt.c` y rellenaremos los vacíos. Siempre es una buena idea añadir algunos comentarios. Primero, explicar a los usuarios lo que hace el plugin, dónde pueden obtener una copia, a quién deberían contactar para fallos/mejoras, y cualquier restricción de licencia sobre el código. Al usar comentarios, debemos asegurarnos de usar comentarios del estilo `/* */`. Los plugins están en c, y algunos compiladores de c no aceptan comentarios del estilo `//`.

```

/*
Descripción: Este plugin es un ejemplo de plugin de secuencia que filtra
píxeles
de baja intensidad. Funciona con un Strip como entrada.
Autor: Kent Mein (mein@cs.umn.edu)
Sitio web: http://www.cs.umn.edu/~mein/blender/plugins
Licencia: Dominio Público
Última modificación: Sun Sep  7 23:41:35 CDT 2003
*/

```

Después necesitamos rellenar el Nombre, debería ser el mismo que el archivo `.c`. Preferiblemente descriptivo, menos de 23 caracteres, sin espacios, y todo minúsculas.

```
char name[24]= "simpfilt.c";
```

Cast y varstr necesitan estar sincronizados. Queremos una barra de desplazamiento por lo que haremos lo siguiente:

```

varStruct varstr[]= {
    { LABEL,      "In: 1 strips", 0.0, 0.0, 0.0, ""},
    { NUM|INT,    "Intensity", 10.0, 0.0, 255.0, "Our threshold value"},
};

```

```
typedef struct Cast {
    int dummy;
    int intensity;
} Cast; /* debido al botón 'etiqueta' */
```

Ahora necesitamos rellenar `plugin_seq_doit`. Básicamente queremos pasar por cada píxel y si RGB es menor que la intensidad seleccionada, coloca el píxel de salida a: 0,0,0,255; si no, devuelve el mismo valor que el píxel de entrada para esa posición.

```
int x,y;

for(y=0;y<cast->intensity) &
    (inl[1] > cast->intensity) &
    (inl[2] > cast->intensity)) {
    out[0] = out[1] = out[2] = 0;
    out[3] = 255;
} else {
    out[0] = inl[0];
    out[1] = inl[1];
    out[2] = inl[2];
    out[3] = inl[3];
}
}
```

Terminamos pues con `simpfilt.c`.

Compilando:

bmake es una utilidad simple (shell script) para ayudar a la compilación y desarrollo de plugins, y puede encontrarse en el subdirectorio *plugins/* del directorio de instalación de Blender. Se invoca mediante: `bmake (nombre_plugin.c)` e intentará enlazar las librerías adecuadas y compilar el archivo C especificado adecuadamente para su sistema. Si usted está intentando desarrollar plugins en una máquina windows, *bmake* puede no funcionar en este caso; debería usar *lcc*. Puede usar lo siguiente para compilar un plugin con *lcc*, asumiendo que usted tiene sus plugins en `c:\blender\plugins`: Aquí tiene un ejemplo de cómo se compilaría el plugin de secuencia `sweep.c`. Abra una ventana DOS y haga lo siguiente: (Nota: Asegúrese de que el directorio `lcc\bin` está en su path)

```
cd c:\blender\plugins\sequence\sweep
lcc -Ic:\blender\plugins\include sweep.c
lcclnk -DLL sweep.obj c:\blender\plugins\include\seq.def
implib sweep.dll
```