



Aprende en **24** horas

Blender & YafRay

Diseño Gráfico 3D con Software Libre



Carlos González Morcillo
Escuela Superior de Informática
Universidad de Castilla-La Mancha

Sobre este libro...

¿Aprender Blender y Yafray en 24 horas?



Es sorprendente el ritmo de desarrollo del Software Libre en general y de Blender en particular. Cada pocos meses aficionados, profesionales y entusiastas de los gráficos 3D podemos disfrutar de una nueva versión con características y funcionalidades sorprendentes. Como docente, eso me obliga a tener que revisar y rehacer gran cantidad de material docente todos los años. :-)

Este libro que tienes en tus manos (o en tu pantalla) es en realidad una recopilación de los tutoriales prácticos realizados para la asignatura *Animación para la Comunicación* impartida en la Escuela Superior de Informática de la Universidad de Castilla-La Mancha en el curso 2005/2006. Probablemente encuentres referencias en el texto a conceptos que fueron explicados en las clases teóricas... Nada que una búsqueda en Google o en la Wikipedia no puedan arreglar.

Si el subtítulo de "Aprende en 24 horas" te ha parecido mentira... en realidad tienes razón. Es mentira. No podrás aprender a usar las herramientas en tan poco tiempo. Lo he titulado así porque aproximadamente teníamos poco más de 20 horas de prácticas en la asignatura. Eso se traduce en que los alumnos se llevaban "tarea" para casa. El tiempo que te requerirá realizar estos tutoriales depende de tu destreza, tu experiencia previa, tu capacidad de visión 3D, etc... pero por lo menos serán 50 ó 60 horas de práctica. Puedes completar los conocimientos con el manual oficial de Blender y con la multitud de tutoriales existentes en la red.

Al igual que otros años, este libro es Libre bajo licencia Creative Commons 2.5. Puedes copiarlo donde quieras, subirlo a tu página web (agradeceré que hagas un *mirror*), regalárselo a tu vecino, etc... Siempre que me nombres como autor del mismo, no realices obras derivadas o lo vendas. Si eres docente, estaré encantado de que lo uses en tus clases y me comentes la experiencia.

Y nada más, un saludo y disfruta con tu aprendizaje en el mundo de la síntesis de imagen tridimensional por computador. Si tienes algún comentario, contacta conmigo en Carlos.Gonzalez@uclm.es.

Happy Blending!

- Sesión 01:** Introducción al Interfaz
- Sesión 02:** Operadores de Modelado Básico
- Sesión 03:** Técnicas de Modelado Avanzado
- Sesión 04:** Modelado de un Personaje
- Sesión 05:** Materiales y Texturas
- Sesión 06:** Animación Básica
- Sesión 07:** Animación Jerárquica
- Sesión 08:** Animación basada en Esqueletos
- Sesión 09:** Esqueletos Avanzados + Adenda
- Sesión 10:** Renderizado Realista
- Sesión 11:** Animación No Lineal

LICENCIA:

© 2005, 2006 Carlos Gonzalez Morcillo. Se permite la copia y distribución de la totalidad o parte de esta obra sin ánimo de lucro. Toda copia total o parcial deberá citar expresamente el nombre del autor. Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra. Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor. Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.

Documento: Blender y Yafray: Diseño Gráfico 3D con Software Libre.

Fecha de Publicación: 31 de Julio de 2006

Autor: Carlos González Morcillo

Email: Carlos.Gonzalez@uclm.es

Web: <http://www.inf-cr.uclm.es/www/cglez>

Institución: Escuela Superior de Informática – Universidad de Castilla-La Mancha

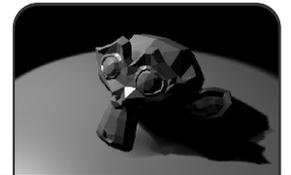
Descripción: Apuntes de la parte práctica de la asignatura "Animación para la Comunicación" del curso 2005/2006.



LICENCIA CREATIVE COMMONS
Reconocimiento-NoComercial-SinObrasDerivadas 2.5 España

Usted es libre de copiar, distribuir y comunicar públicamente la obra, bajo las condiciones siguientes:

- Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador.
- No comercial. No puede utilizar esta obra para fines comerciales.
- Sin obras derivadas. No se puede alterar, transformar o generar una obra derivada a partir de esta obra.



Introducción al Interfaz de Blender

Animación para la Comunicación · Escuela Superior de Informática · Universidad de Castilla-La Mancha

➤ Esta primera práctica nos servirá como toma de contacto con la herramienta. Personalizaremos el entorno, añadiremos algunos objetos básicos y, finalmente, renderizaremos una escena sencilla.

El interfaz que presenta Blender al arrancar es el mostrado en la Figura 1. Todas las ventanas de Blender son personalizables, de tal forma que podemos dividir la pantalla como nos resulte más cómodo y situar en cada porción el tipo de ventana que queramos. Además, las órdenes sobre el interfaz de usuario son coherentes (si una combinación sirve para desplazar los elementos de la ventana 3D, hará lo mismo sobre una ventana de botones o de texto).

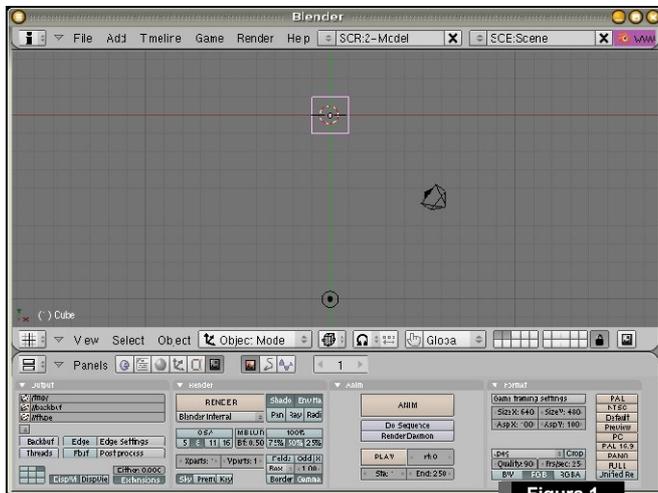


Figura 1

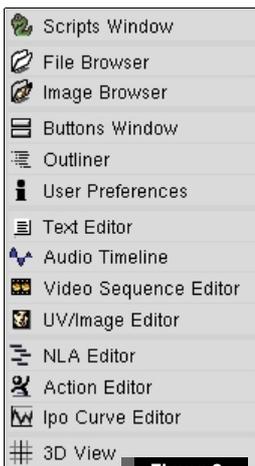


Figura 2

Por defecto Blender arranca la pantalla con tres divisiones; la ventana superior de tipo **User Preferences**, la intermedia de tipo **3D View** y la inferior de tipo **Buttons Window**. Podemos cambiar el tipo de ventana pinchando en el botón y seleccionando alguno de los tipos que se muestran en la figura 2.

La vista 3D nos muestra un cuadrado de color rosa. Este cuadrado es un cubo que crea Blender al comenzar a trabajar. El color rosa indica que está seleccionado. El círculo inferior es el foco por defecto, y la pirámide negra de la derecha representa la cámara. El resto de cuadrados de la escena forman la cuadrícula que nos servirá para colocar los objetos de la escena con precisión.

El círculo de color rojo y blanco situado en el centro de la ventana es el puntero 3D. Todos los elementos que se añadan a la escena lo harán donde se encuentre el **puntero 3D**; además podrá servirnos como punto y para situar elementos con precisión en próximas prácticas.

Antes de empezar a trabajar, personalizaremos el entorno de Blender. Dividiremos la ventana 3D con la distribución que se muestra en la Figura 3. Para ello, nos situaremos con el ratón en la zona de división de dos ventanas (por ejemplo, entre la zona de división entre la ventana 3D y la ventana de Botones) y con la ventana 3D iluminada (seleccionada), haremos **Ctrl** y elegiremos **Split Area**. Partiremos la pantalla hasta conseguir una disposición del área de trabajo como se muestra en la Figura 3. Podemos eliminar las cabeceras, así como situarlas en la parte superior de las ventanas haciendo **Ctrl** sobre ellas y seleccionando **No Header**.

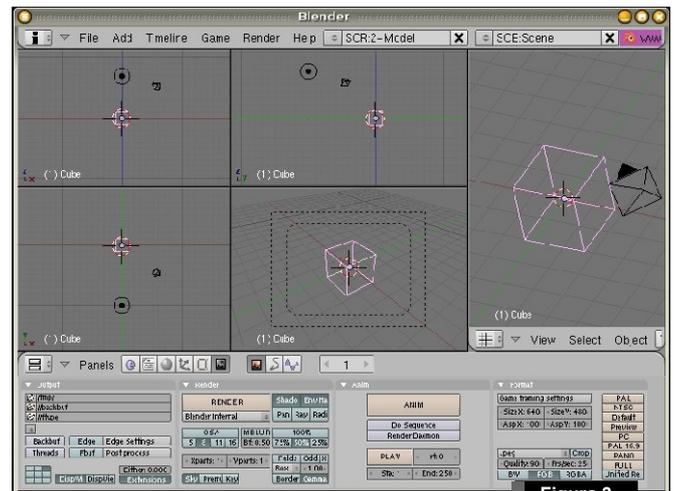


Figura 3

Para rotar el punto de vista de una ventana 3D, arrastraremos con **Ctrl** pinchado. En ratones de dos botones, con **Alt** pulsada, arrastraremos el ratón con **Ctrl**. Podemos hacer zoom en cualquier ventana (vista 3D, ventanas de botones, etc...) manteniendo **Ctrl** pulsado, y arrastrando **Ctrl**. Si estamos trabajando sobre una vista 3D, también podemos hacer zoom utilizando la rueda central con la que cuentan algunos ratones. En ratones de dos botones, mediante **Ctrl** **Alt** **Ctrl**. Para desplazar el punto de vista (horizontal y verticalmente), utilizaremos **Shift** **Ctrl** **Ctrl**. De igual forma, con ratones de dos botones utilizaremos **Alt** **Shift** **Ctrl**. Si estamos trabajando en Linux y tenemos un ratón de dos botones, cambiaremos en la combinación de teclas anterior la tecla **Alt** (utilizada para arrastrar una ventana) por la tecla de función de Windows **Win** (que por fin utilizaremos para algo útil ;-)

Cambiaremos el modo de vista de cada vista 3D mediante atajos de teclado o los botones destinados a tal efecto. Pulsando **N** en el **teclado numérico**, cambiaremos entre proyección ortográfica y perspectiva. Con **0** tendremos la vista de la cámara. **1**, **2**, **3** (también desde el el menú **View** en la cabecera de la ventana 3D) nos darán las vistas de la escena de planta, alzado y perfil respectivamente.

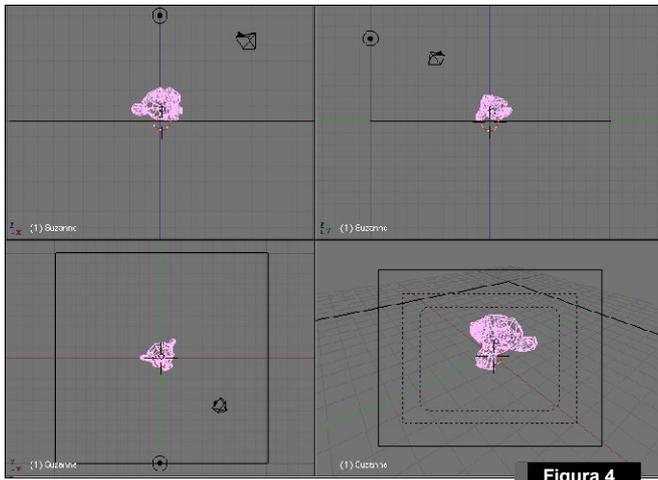


Figura 4

Una vez configuradas las vistas como se muestra en la Figura 3, vamos a personalizar nuestra escena. Con el cubo por defecto seleccionado (seleccionamos objetos con **Left Click** sobre ellos), en color rosa, pulsaremos **Del** y lo quitaremos de la escena. Realizaremos la misma acción con la lámpara de la escena. Añadiremos un objeto básico de Blender; por ejemplo, la cabeza de *Suzanne* (*Suzanne* es la mascota de Blender). Para insertar nuevos elementos a la escena, pulsaremos **Spacebar**, y elegiremos **Add/ Mesh/ Monkey**. Cuando insertamos un objeto en la escena, lo hará en modo de edición de vértices. Para volver al modo de edición de objeto, pulsaremos **Tab**.

Con los objetos seleccionados, podemos realizar las tres operaciones básicas de transformación sobre ellos:

- **Rotación:** Pulsando una vez (no mantener pulsada) la tecla **R** (*Rotation*) y desplazando con el ratón, rotamos el objeto.
- **Traslación:** Pulsando una vez la tecla **G** (*Grabber*) y desplazando con el ratón.
- **Escalado:** Pulsando una vez la tecla **S** (*Scale*) y desplazando el ratón.

Estas operaciones pueden realizarse también empleando manipuladores 3D. Se activan pinchando en el icono **3D Manipulator** situado en la cabecera 3D de la ventana. Una vez activado, elegimos el tipo de transformación (traslación **Translate**, rotación **Rotate** o escalado **Scale**) y si queremos que se realicen respecto del sistema de referencia global, local o perpendicular a la vista. Aparecerán sobre el objeto unos ejes que podemos desplazar mediante **Left Click**.

Las operaciones anteriores, permiten el uso de ciertos modificadores para conseguir mayor precisión en la orden a realizar. Podemos ejecutar estos modificadores mientras transformamos el objeto con una operación de rotación, escalado o traslación:

- **Limitación del eje:** Si mientras realizamos la transformación, pulsamos la tecla **X**, **Y** o **Z**, estamos limitando la transformación a que se realice en un único eje. La primera vez que pulsemos la tecla forzaremos a que la transformación se realice sobre el eje correspondiente al Sistema de Referencia Global, y si pulsamos una segunda vez lo limitamos sobre el sistema de referencia local del objeto.

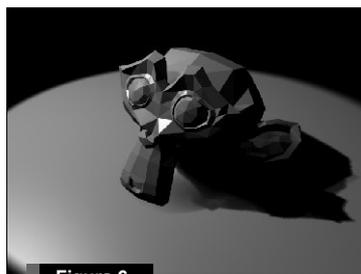


Figura 6

Por defecto, las transformaciones básicas son dependientes del punto de vista (es decir, si rotamos un objeto desde la vista en perfil, rotaremos desde un eje. Si lo realizamos desde la vista en planta, desde otro diferente). En las vistas de planta, alzado y perfil podemos ver un pequeño eje de coordenadas cartesianas en la parte inferior izquierda de cada ventana 3D.

- **Precisión:** Si mantenemos pulsada la tecla **Shift** mientras realizamos una transformación, lo haremos en modo de precisión. La tecla **Ctrl** permite realizar las transformaciones en unidades discretas (ajustándonos a la rejilla o de 5 en 5 unidades).

Todas las operaciones anteriores pueden realizarse de forma numérica, si con el objeto seleccionado pulsamos **N**. Aparecerá una ventana en la que podemos indicar con precisión los valores numéricos de las transformaciones a aplicar. Los valores numéricos pueden introducirse por teclado si hacemos click con **Left Click** sobre una entrada de texto mientras mantenemos pulsada la tecla **Shift**.

Ayudándonos de las vistas de planta, alzado y perfil, situaremos el puntero 3D debajo de la cabeza del mono. Recordemos que la posición del puntero 3D se cambia pinchando con **Left Click** sobre la vista. Una vez situado correctamente el puntero, añadiremos un plano a la escena que servirá de "suelo" al objeto anterior. Para ello, y con el puntero del ratón situado en la vista superior, pulsaremos **Spacebar**, **Add/ Mesh/ Plane**. Saldremos del modo de edición de vértices del objeto con **Tab**, y lo escalaremos para que cubra el suelo de la escena. Debemos obtener una configuración similar a la Figura 4.

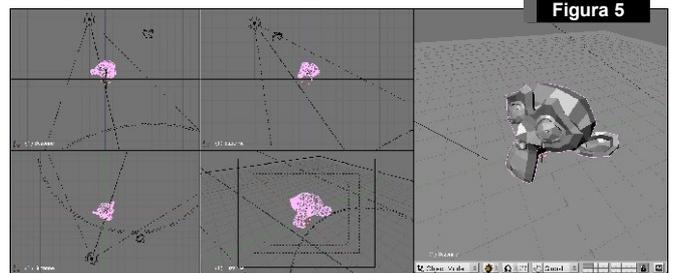
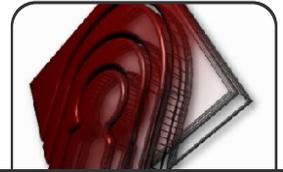


Figura 5

El Render se realiza a través del botón del grupo de Escena **Render** (también mediante la tecla **F12**), **Render Buttons**, y pinchando en el botón marrón de **RENDER** o bien con el atajo de teclado **F12**... ¿Por qué no sale nada?. No hemos iluminado la escena. Ocultaremos la ventana de render pulsando **O**. Añadiremos una luz de tipo **Spot** en la posición que muestra la Figura 5. Es importante direccionar bien este tipo de lámpara, ya que la escena será iluminada en el área que abarca el foco. Para añadir el foco, situaremos el cursor 3D en la posición adecuada. Añadiremos la luz mediante **Spacebar**, **Add/ Lamp/ Spot**. Cambiaremos la intensidad del foco con el botón de gestión de lámparas **Lamp Properties** del grupo de sombreado **Shading** (o pulsando **L**). Aumentamos el valor de energía a 1.5. Orientamos el foco para que apunte correctamente al objeto (utilizando la rotación **R**).

Renderizamos y si el resultado es similar al mostrado en la Figura 6, guardamos la imagen pulsando **S**. El formato de la imagen que salvaremos se selecciona en el apartado del botón de render **Render Properties**, dentro de la pestaña **Format**. Elegiremos, por ejemplo, JPG conservando la calidad de la imagen en un valor alto (Quality entre 80-95). Es necesario indicar la extensión de forma explícita en la imagen que guardemos desde Blender (por ejemplo "resultado.jpg").



Operadores de Modelado Básico

Animación para la Comunicación · Escuela Superior de Informática · Universidad de Castilla-La Mancha

➤ Como continuación del trabajo realizado en la primera sesión, comenzaremos a realizar los primeros objetos 3D partiendo de una imagen 2D. Veremos el uso de curvas y operadores básicos de modelado.

Ejecutamos Blender y eliminamos el cubo y la lámpara creados por defecto. Pinchamos en **View** (en la cabecera de la vista 3D) y elegimos **Background Image**. Pinchamos luego en **Use Background Image**. La ventana que nos aparece (ver Figura 1) nos permite cambiar la imagen de fondo, variar su situación (**X Offset** e **Y Offset**) y el nivel de transparencia (**Blend**). Cargamos como fondo la imagen *logouclm.jpg*. Hecho esto, cerramos la ventana **Background Image**.

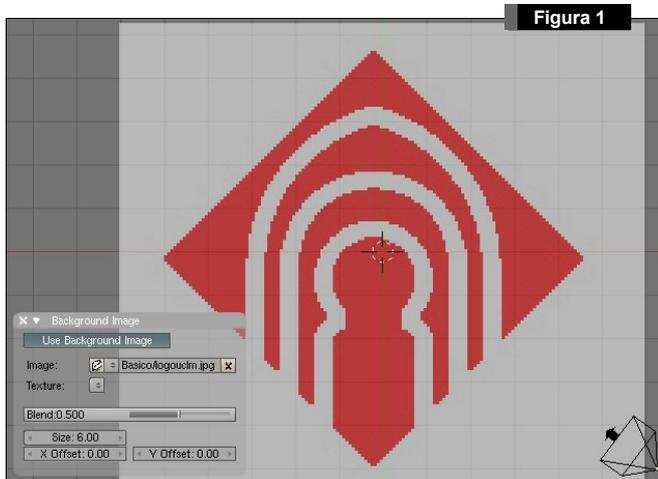


Figura 1

Vamos a trazar el contorno del logotipo por medio de una curva. Para ello, añadiremos una a la escena con **Add/ Curve / Bezier Curve**. Con la curva añadida, y sin salir del modo de edición (sin pulsar **Tab**), vamos a los botones de edición (**A** o **F7**) y convertimos la curva a polígono, pinchando en el botón **Poly** del grupo **Curve Tools**. Esto nos permitirá trabajar de forma más cómoda. Cuando hayamos ajustado los puntos de control al logotipo, convertiremos de nuevo a curva de Bezier, ajustando la tensión en los tramos que lo requieran.

Por defecto, tenemos todos los vértices seleccionados (todos en color amarillo). Pulsamos la tecla **A** y los deseccionamos todos (aparecen en rosa). Ahora desplazaremos los vértices al contorno exterior del logotipo (seleccionamos cada vértice de forma individual con **G**), y desplazamos pinchando una vez la tecla **C**. La curva original está formada por 6 vértices. Los situaremos en las posiciones 1 .. 6 tal y como muestra la figura 2. Podemos realizar zoom en la vista 3D (**Ctrl** + **G**) o usando la rueda central del ratón, para tener una mayor precisión a la hora de situar los vértices.



Figura 2

Para añadir vértices: manteniendo pulsado **Ctrl**, haremos **M**. De esta forma, añadiremos los vértices 7 y 8. Para cerrar la figura (es decir, añadir la arista que une 8 con 1), pulsaremos la tecla **C**. Podríamos pensar que hacen falta más vértices, por ejemplo, entre 6 y 7. Sin embargo, no es necesario ajustar mejor la superficie en las secciones curvas, ya que lo haremos por medio de la tensión en la curva tipo Bezier.

Seleccionamos de nuevo todos los vértices, pulsando la tecla **A** (deben quedar todos en color amarillo). Volvemos a los botones de edición **Tab** y convertimos la curva a tipo Bezier. Deben aparecer unos segmentos de color verde. Estos son los segmentos de tangente, que especifican el grado de tensión en la pendiente de cada sección de curva. Deseleccionamos todos los vértices y elegimos uno de los que requieren curvatura (por ejemplo el 6).

Ajustaremos los puntos de tangente hasta obtener la geometría que se ajuste bien a la imagen de fondo. Cuando hayamos terminado, volveremos al modo de edición de objeto **Tab**, y pulsaremos la tecla **Z** para obtener la representación sombreada, tal y como muestra la figura 3 (ventana izquierda). Volvemos a la representación con líneas pulsando de nuevo la tecla **Z**.

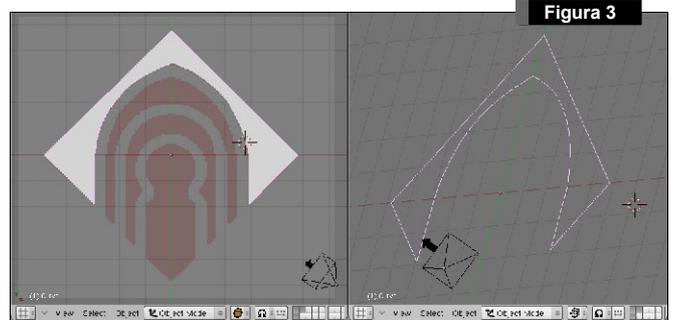


Figura 3

Añadiremos otras curvas de Bezier a la escena y procederemos de forma similar, convirtiendo a polígono, ajustando puntos y volviendo a convertir a Bezier para modelar las otras tres secciones que definen el logotipo. Cuando hayamos terminado, podemos quitar la imagen de fondo pulsando de nuevo **View/ Background Image**... Si hemos salido del modo de edición al añadir cada curva, tendremos que cada curva creada es un objeto independiente. En Blender, podemos unir objetos en una única entidad si seleccionamos todos con **A** (manteniendo **Shift** pulsada), y después pulsando **Ctrl** **J** **Join Selected [Tipo de Objeto]**, siendo **Curves** el tipo de objeto en este caso. También podemos separar objetos unidos, seleccionando la parte a separar y pulsando la tecla **P**, con el objeto en modo de edición de vértices.

Pasaremos a realizar una extrusión sobre la curva para obtener el modelo 3D. Rotaremos la vista para tener un ángulo mejor a la hora de ver la operación realizada. Elegimos en modo de selección de objeto, cambiamos los valores de extrusión y redondeado tal y como se muestra en la figura 4. Este panel se encuentra dentro de los botones de edición (dentro del grupo **Curve and Surface**). Podemos cambiar los valores en los botones de inserción, pinchando y arrastrando sobre ellos, o bien manteniendo pulsada la tecla **Shift** y haciendo clic sobre ellos.

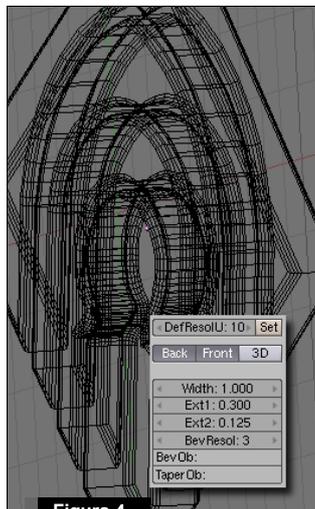


Figura 4

Vamos a aplicarle un material sencillo al modelo. Seleccionamos el logo y pinchamos en el botón de materiales, tal como muestra la figura 5 (o bien pulsamos **M**). Como no hay ningún material en la escena, tendremos que añadir uno nuevo, que luego aplicaremos al logotipo. Se pueden crear tantos materiales como sean necesarios, pero en este ejemplo utilizaremos sólo uno. Para crear un nuevo material, pincharemos en el botón **Add New**.



Figura 5

De todas las propiedades que pueden tener los materiales, únicamente vamos a definir el color. Ajustaremos los valores del color tal y como se indica en la figura 6. En próximas sesiones de prácticas, veremos con más profundidad las opciones de materiales y texturas en Blender. Podemos seleccionar el color del material de una forma más intuitiva, pulsando sobre el cuadro de color situado a la izquierda del botón **Col**.



Figura 6

Daremos un nombre al material que hemos creado; en el botón que hemos pinchado para añadir el material, tendremos una caja de texto. Manteniendo pulsado **Shift**, pinchamos con **Enter** y tecleamos un nombre para el material, por ejemplo, **UCLMrojo**. Podríamos aplicar el mismo material a otros objetos de la escena, y que todos compartan las mismas propiedades. Así, si cambiamos las propiedades del material, todos los objetos de la escena que lo tengan sufrirán los cambios. Esto se realiza seleccionando el material de la lista desplegable. Podemos activar el modo de visualización sólida con **Z**.

Vamos a renderizar la escena. Situaremos la cámara para obtener una vista como la que se muestra en la figura 7. Añadiremos un foco de iluminación de tipo **Spot**. Para ello, pulsamos **Spacebar** **Add/ Lamp/ Spot**. Orientamos el foco para que apunte al logo 3D, rotándolo con ayuda de las vistas superior, frontal y lateral. La situación del foco deberá ser similar a la mostrada en la figura 7. Probamos a renderizar la escena **R**. Si el objeto sale muy oscuro, podemos ajustar la intensidad del foco en los botones de foco **F**, en el campo **Energy** a un valor superior. Ajustaremos las opciones de renderizado como se indica en la figura 8.

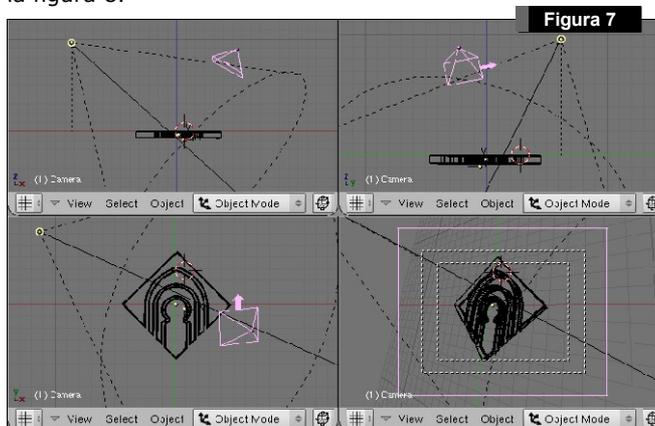


Figura 7

Activamos el botón de **Shadows** para que el motor de render calcule las sombras de la escena. **OSA (OverSampling)** con valor 16 hace que el antialiasing se aplique con la mayor intensidad. Seleccionamos uno de los tamaños predefinidos pinchando en **PC (640x480)**. Con **100%** indicamos que el renderizado se haga al tamaño total elegido.

Cuando hacemos pruebas es interesante poner este valor al **50%** o al **25%** para que la generación de la imagen lleve menos tiempo. Con **RGBA**, indicamos que la imagen resultado debe tener canal **Alpha** de transparencia. Así, tendremos que seleccionar un formato que soporte canal **Alpha** (como **Targa; *.TGA**). El canal **Alpha** codifica, en tonos de grises, qué parte de la imagen es de objeto y qué parte es el fondo. Un valor de negro indica que no hay ningún objeto en ese pixel de la imagen (es el fondo), mientras que blanco indica objeto sólido. Los tonos de gris indican objetos semitransparentes.

El resultado de renderizar la escena debe ser similar al mostrado en la figura 9. La figura 10 se corresponde con el canal Alpha de la imagen resultado (este canal puede verse abriendo el fichero con un programa de procesamiento de imágenes como GIMP). Podemos cambiar el color de fondo accediendo a las propiedades del mundo y cambiando el color del horizonte (**HoR, HoG, HoB**) (figura 11).



Figura 8

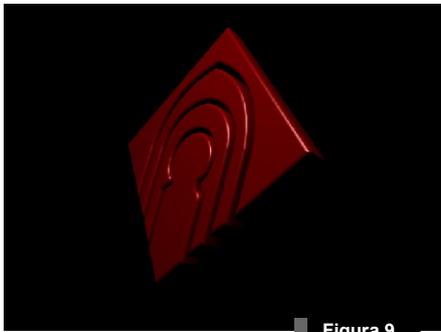


Figura 9

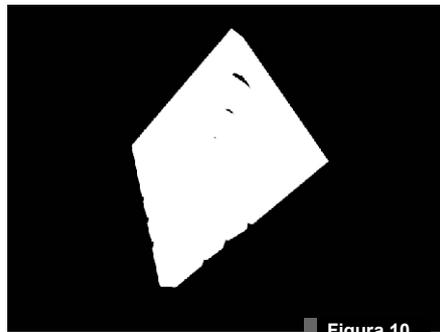


Figura 10

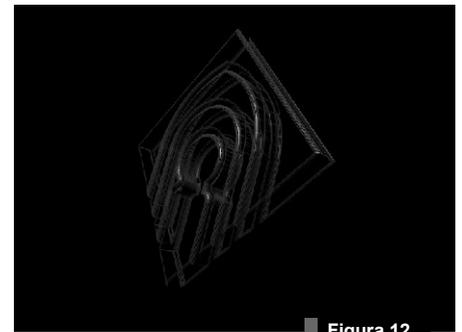


Figura 12

Guardamos la imagen como **solido.tga**. Salvamos también el fichero de blender y cambiamos el material. Nos vamos al menú de materiales y asegurándonos que estamos trabajando con **MA:UCLM Rojo**, cambiamos el color del material en las tres componentes (**R,G,B**) a 0.300. Además, activamos el botón de **Wire** (situado en el centro). Renderizamos de nuevo la escena; debemos obtener un resultado similar al de la figura 12. Sin haber cambiado ninguna opción de renderizado, ni la posición de la cámara, guardamos la imagen resultante como **alambre.tga**.



Figura 11

Salvamos el proyecto de blender y abrimos GIMP, el programa de composición y retoque fotográfico GNU. Cargamos las imágenes **solido.tga** y **alambre.tga**. Comprobamos que el fondo de la imagen no es negro; está formado por "cuadritos". Esto indica que GIMP ha importado las imágenes con el canal de transparencia correctamente.

Las capas actúan como planchas de plástico transparentes. El orden de las capas importa (al igual que si colocamos una transparencia delante de otra). Renombraremos la capa de la imagen sólida. Para ello, nos situamos sobre la capa y haciendo **Editar atributos de capa**, le pondremos como nombre "**solido**". Añadiremos una capa nueva, que esté rellena de color blanco. Para ello, pincharemos en el botón **Editar**, y le pondremos como nombre **Fondo**. La capa se ha situado delante del logo. Pincharemos en la flecha "abajo", como indica la figura 13, para ordenar las capas.

Seleccionamos la ventana donde se ha abierto el fichero **alambre.tga**. Hacemos sobre la imagen, seleccionamos todo (**Ctrl A**) y seleccionamos **Editar/ Copiar** (o bien pulsamos **Ctrl C**). Después, nos vamos a la imagen **solido**, y pegaremos el logo de alambre como una nueva capa. Para ello, **Editar/ Pegar** (o **Ctrl V**). Vemos que aparece en la ventana de capas como una selección flotante. Para generar una capa nueva con la imagen, pincharemos de nuevo en el botón de nueva capa **Editar/ Nueva Capa**. Renombramos la capa como "**Alambre**".

Situamos la capa **alambre** entre la capa **solido** y la capa **fondo**. En la ventana de



Figura 13

Selección de brocha (figura 15), pinchamos para seleccionar una redonda que tenga un radio grande (por ejemplo, **Circle Fuzzy 19**).

Ajustamos la opacidad de la goma (en la ventana de opciones de herramienta) a 25 aproximadamente. Hecho esto, borramos una parte de la capa **solido** (la derecha, por ejemplo). Hacemos otra pasada consiguiendo un efecto de degradado, como se muestra en el resultado final.

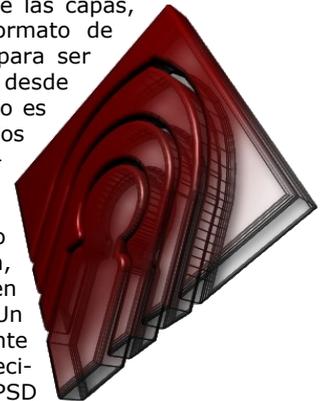
Para terminar, añadimos un efecto de sombra a la capa **alambre**. Seleccionamos la capa, nos vamos a la imagen y hacemos **Script-Fu/ Sombra/ Sombra Base**. Podemos dejar los parámetros tal y como están y aceptamos.



Figura 14

Vemos que, debajo de la capa **alambre** se ha creado una capa llamada **Drop-Shadow** (ver figura 14). Podemos variar la opacidad de la capa de sombra generada. Si no nos gusta el resultado, eliminamos la capa (pinchando en el botón del cubo de basura), y probamos a cambiar los parámetros del script. Para terminar, pinchamos con **Editar/ Pegar** sobre la imagen (o sobre el menú **Archivo** de la ventana de la imagen), y guardamos el fichero con formato XCF (el formato nativo de GIMP), que permite almacenar el fichero tal y como estamos trabajando (capas, canales, etc...).

Si queremos utilizar la imagen desde otras aplicaciones, tendríamos que salvarlo en un formato como JPG. En este caso GIMP nos advierte que perderemos la información acerca de las capas, canales, etc.... El formato de GIMP está pensado para ser utilizado únicamente desde esta herramienta y no es portable. Incluso los propios desarrolladores de GIMP desaconsejan tratar de utilizar este formato desde otra aplicación, ya que está en continuo cambio. Un formato ampliamente usado, con especificación pública, es PSD de Adobe Photoshop.





Técnicas de Modelado Avanzado

Animación para la Comunicación · Escuela Superior de Informática · Universidad de Castilla-La Mancha

➤ En esta sesión utilizaremos técnicas avanzadas de modelado en Blender, como NURBS, superficies de revolución, barrido, etc. Construiremos un modelo real con un nivel de detalle medio.

Una técnica muy empleada en modelado tridimensional es la *rotoscopia*. Partimos de fotografías o bocetos del objeto a modelar desde diferentes puntos de vista y los utilizamos como plantilla para construir el objeto 3D. El objeto de esta práctica lo construiremos empleando esta técnica. Dividimos el interfaz de Blender tal y como se muestra en la figura 1, y cargamos como imagen de fondo de cada ventana una fotografía de la plancha a modelar. Cargamos una imagen de fondo en una ventana 3D desde la cabecera de la ventana, en **View/ Background Image**. Es importante cargar en cada vista el fichero adecuado; en la vista del plano **ZX** cargamos **Lateral.jpg**, en la vista **ZY** la imagen **Frontal.jpg** y en la vista **YX** el fichero **Planta.jpg**.

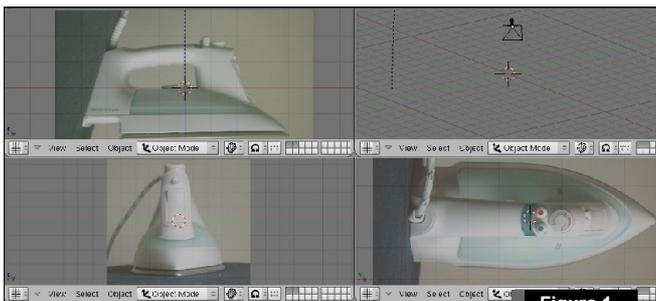


Figura 1



Nota: En las vistas de planta, alzado y perfil (Side, Front, Top), puedes ver un pequeño sistema de coordenadas en la parte inferior izquierda de la ventana. Estos ejes nos permiten identificar fácilmente el plano de trabajo al que nos referíamos anteriormente.

Por defecto Blender inserta las imágenes con la misma escala y en la posición 0,0. Puede ser necesario ajustar los parámetros de (figura 2) tamaño de la imagen **Size**, desplazamiento horizontal **X Offset** y desplazamiento vertical **Y Offset** para que la posición y el tamaño de las imágenes en las tres vistas sea coherente. Este tamaño lo ajustaremos según vayamos trabajando con la escena y veamos las necesidades de ajuste. El parámetro **Blend** indica la transparencia de la imagen; si trabajamos con imágenes con colores muy vivos, puede ser recomendable establecer este valor por debajo de 0.5.



Figura 2

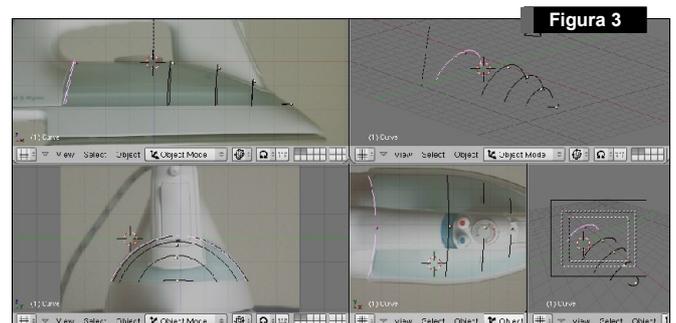
Construcción de la Base: NURBS

Para comenzar, utilizaremos la técnica del *Skinning* para construir la base de la plancha. Mediante un conjunto de perfiles, definiremos la base del objeto a modelar, y los

convertiremos en una única superficie NURBS. Empleando las 3 vistas anteriores, añadimos a la escena una superficie curva NURBS (**Spacer**, **Add/ Surface/ NURBS Curve**). Duplicamos la curva mediante (**Shift + D**) 4 veces y situamos los puntos de control, escalando las curvas duplicadas para definir el contorno como se muestra en la figura 3. **¡Cuidado!**, no añadir mediante **Add/ Curve/ NURBS Curve**, ya que queremos definir posteriormente una superficie.



Importante: Blender permite utilizar dos modos de duplicado; uno en el que se duplica el objeto, obteniendo una copia independiente del original (se realiza mediante (**Shift + D**)), y otro modo en el que los objetos duplicados comparten las propiedades como la malla poligonal, materiales, etc... En este segundo modo (pulsando (**Alt + D**)), si realizamos un cambio en la malla de uno de los objetos duplicados, todos sufrirán el cambio.



Con el contorno de la figura 3 definido, unimos todas las curvas en un único objeto, seleccionándolas todas y pulsando (**Ctrl + J**) **Join Selected NURBS**. Pasamos a modo de edición de objeto (**Tab**), y seleccionamos todos los puntos de control de la curva (**A**), y pulsamos la tecla (**E**). Con esto, los perfiles son convertidos en una superficie como se muestra en la figura 4.

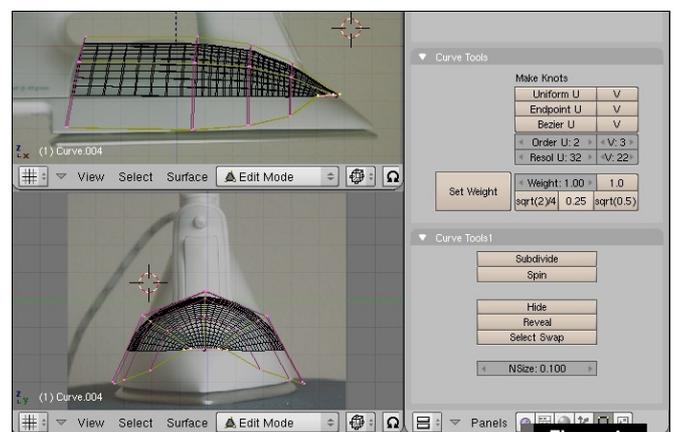


Figura 4

Podemos ajustar algunos parámetros de la superficie generada (figura 4). En los botones de edición , podemos definir el orden de la curva (**Order U,V**) en la representación paramétrica U, V. El orden es la profundidad de cálculo de la curva. Si tenemos orden 1 trabajamos con un punto, orden 2 es lineal, 3 es cuadrático... También se puede definir la resolución (**Resol U,V**); el número de puntos interpolados entre cada par de puntos de control de la superficie.

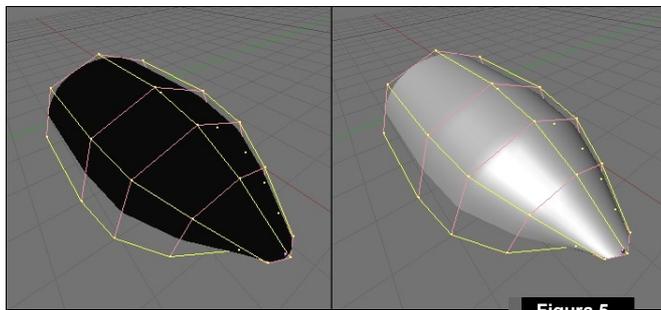


Figura 5

Si pulsamos la tecla **Z**, la malla deberá aparecer en modo sombreado. Es posible que Blender no haya calculado correctamente la cara "exterior" de la curva (esto se aprecia si la superficie generada es de color negro, como en la figura 5, parte izquierda). Podemos invertir la cara exterior de la superficie, seleccionando todos los vértices que forman la NURBS (tecla **A**), y con **W** **Switch Direction**. Debemos obtener una superficie de color gris, como la figura 5 derecha.

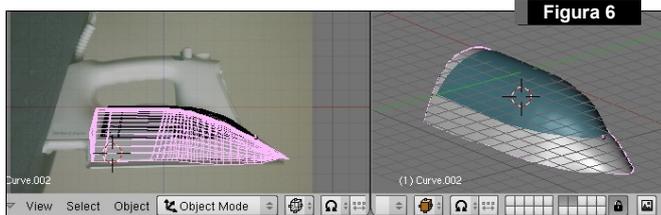


Figura 6

Utilizando la misma técnica, construimos la base inferior de la plancha (ver figura 6). En algunas ocasiones, puede interesarnos convertir la NURBS a malla poligonal (para aplicar algunos operadores que no están permitidos con curvas). Podríamos realizar la conversión con **Alt C**. No obstante, para el modelado de esta práctica, vamos a dejar la superficie de tipo NURBS.

Esta técnica es muy eficiente para la construcción de superficies curvas, que mediante polígonos sería muy costoso. Sin embargo, para otras partes de la plancha (como el asa), es más directo utilizar directamente técnicas poligonales.

Modelado poligonal del Asa

Comenzaremos añadiendo un cubo a la escena, como se muestra en la figura 7. Escalamos el cubo y rotamos las caras laterales para conseguir que se ajuste lo mejor posible a las vistas lateral y frontal.

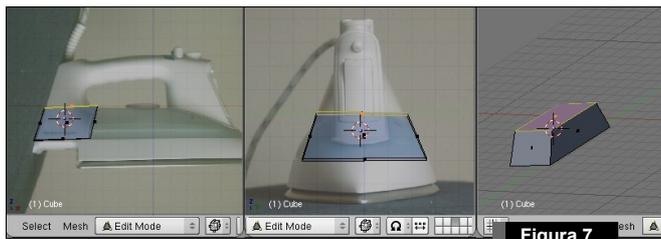


Figura 7

Blender permite trabajar en modo de edición de objeto a nivel de vértice, arista y cara. Indicamos en qué nivel queremos trabajar en la cabecera de una ventana 3D. Cuando estamos en modo de edición de objeto, aparecen 3 iconos (ver figura 8), correspondientes al modo de vértice, aristas y cara. Por ejemplo, para escalar y rotar las caras del cubo insertado de la figura 7 hemos trabajado a nivel de cara.

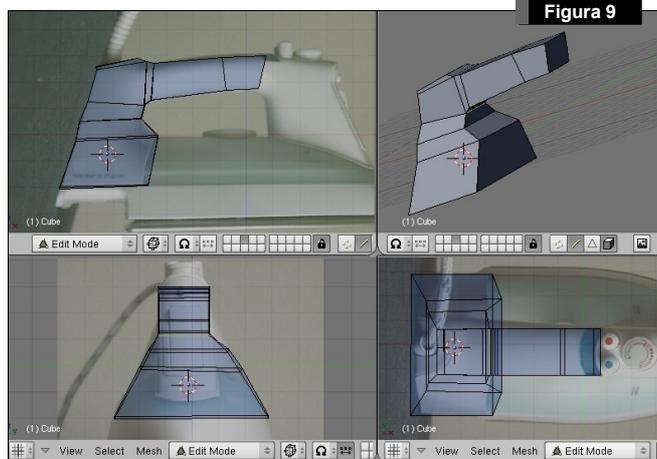


Figura 9

Con la cara superior seleccionada (como se muestra en la figura 7), realizaremos sucesivas operaciones de **extrusión** para generar nuevas caras que definan el objeto. Para realizar la extrusión, con la cara seleccionada, pulsamos **E**, y nos pregunta **Extrude Region** o **Individual Faces**. Si seleccionamos un grupo de caras, la primera opción (**Region**), realizará la extrusión conservando la agrupación, mientras que la segunda tratará el vector normal de cada cara de forma individual, obteniendo múltiples salientes en la superficie. Si hemos seleccionado sólo una cara, ambas opciones son equivalentes.

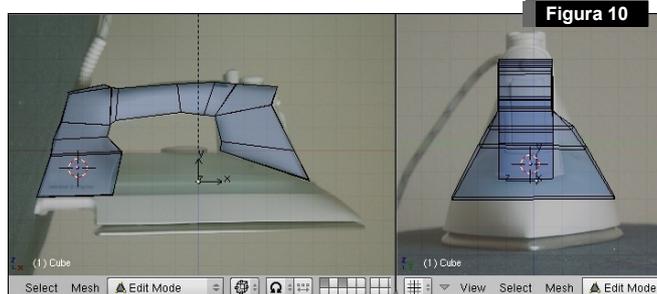


Figura 10

Continuamos añadiendo nuevos segmentos al objeto (mediante extrusiones), tal y como se muestra en la figura 9. Puede ser útil cambiar a modo de edición de aristas, para desplazar (tecla **G**) las aristas situadas en el puente del asa y conseguir una forma mucho más ajustada a la fotografía. En general, bastará con realizar extrusiones, y rotar y desplazar las caras intermedias para conseguir que el objeto 3D se ajuste correctamente a las fotografías base.

Finalmente obtendremos un objeto como se muestra en la figura 10. La representación de las caras en modo de edición de objeto en color azul (como se muestra en las figuras anteriores) se consigue activando en botón **Draw Faces** del grupo **Mesh Tools 1**, dentro de los botones de edición (ver figura 11, parte derecha).

Vamos a suavizar la forma del asa obtenida. Para ello, primero añadiremos más caras al objeto, que nos permita suavizar las aristas. Para ello, con todas las caras seleccionadas **A**, pinchamos dos veces en el botón **Subdivide** del panel **Mesh Tools** de los botones de edición



Figura 8

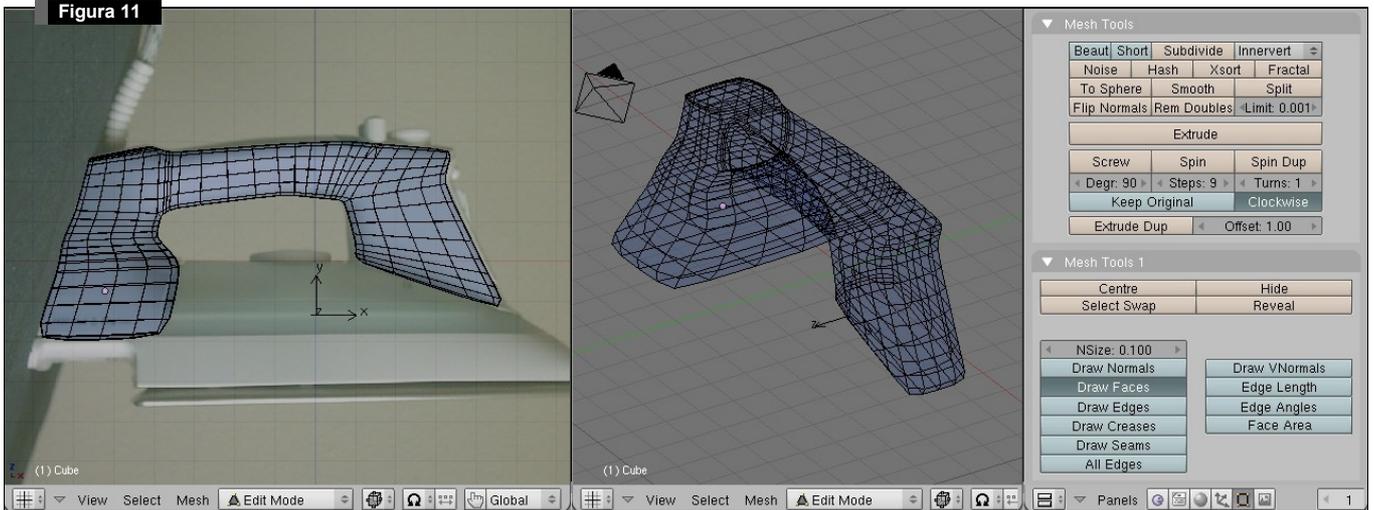


Figura 11

(ver figura 11). Las caras añadidas no suavizan el modelo; únicamente incorporan un mayor nivel de detalle. Para suavizar la forma pinchamos sucesivas veces sobre el botón **Smooth** (situado en el mismo grupo que Subdivide), hasta conseguir un resultado similar al de la figura 11.

De igual forma se ha modelado la pieza de la figura 12. Cuando la complejidad de la escena aumenta (como en este ejemplo), es interesante tener **vistas locales** de los objetos que estamos editando. Para ello, sobre una ventana 3D podemos seleccionar entre vista local de los objetos seleccionados y vista global en **View/ Local View** o **Global View**. También se puede realizar esta operación con la tecla **0** del teclado numérico.

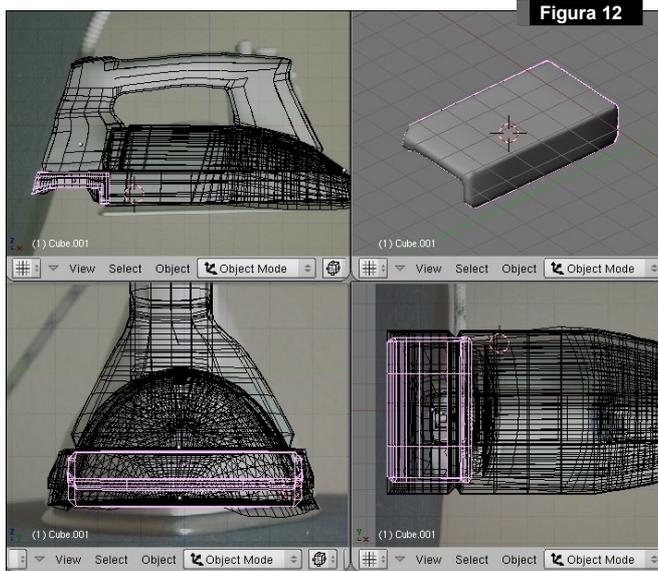


Figura 12

Vamos a añadir ahora la parte inferior de la plancha, utilizando una curva de Bezier y la misma técnica que empleamos en la segunda práctica (modelado de un logotipo). Esta curva la convertiremos a malla para realizar operaciones booleanas con otros objetos.

Operaciones Booleanas

Añadimos una curva de Bezier (**Spacebar** **Add/ Curve/ Bezier Curve**), que convertimos a Polígono mediante el botón Poly del panel Curve Tools de los botones de edición. Ajustamos los puntos de control, y convertimos de nuevo a Bezier, para obtener un contorno como el mostrado en la figura 13.

Ajustamos los parámetros de extrusión de la curva (como hicimos en la práctica de la sesión anterior) como se muestra en la figura 13. Esto nos dará la base de la

plancha. Salimos del modo de edición de objeto (**Tab**) y convertimos la superficie a malla (**Alt+C**) (**Convert Curve to Mesh**).

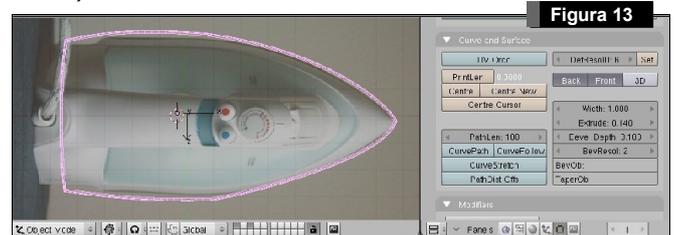


Figura 13

Añadimos un conjunto de cilindros pequeños que atraviesen la base de la plancha, como se muestra en la figura 14 (añadimos uno y lo duplicamos con **Shift+D**). Unimos todos los cilindros en un mismo objeto. Para ello, los seleccionamos todos (manteniendo **Shift**) pulsada, vamos seleccionándolos y finalmente pulsamos **Ctrl+U**. Vamos a realizar la operación booleana de diferencia; seleccionamos primero el objeto del que vamos a restar el segundo (primero la base y luego los cilindros). Pulsamos **W**, **Difference**.

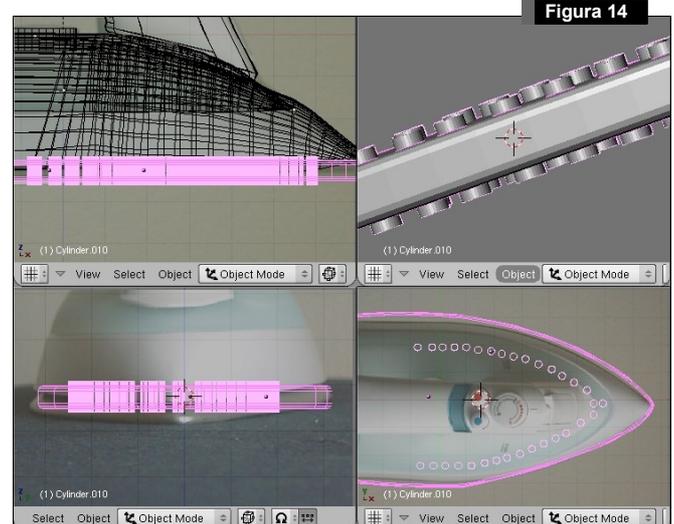


Figura 14

Con esto, obtenemos una base con los agujeros realizados. Podemos eliminar los objetos originales, o moverlos a otra parte de la escena donde no molesten.

Nota: Algunas veces la conversión de superficies curvas a mallas poligonales no se realiza de forma consistente y produce un fallo en el uso de operadores booleanos (diciendo que "ambas mallas tienen que ser objetos cerrados"). Podemos arreglar esta inconsistencia seleccionando el objeto convertido de

superficie curva, entrando en modo de edición de vértices (**Tab**), seleccionamos todos los vértices (**A**) y pinchamos en el botón **Rem Doubles** (ver figura 11). El parámetro de **Limit** situado a su derecha nos indica la separación máxima entre vértices para que éstos sean considerados como duplicados. Hecho esto, el programa nos indica el número de vértices que ha eliminado y seguramente se haya arreglado la inconsistencia en la malla poligonal. Si aún así la malla no forma una superficie cerrada, podemos intentar aumentar el valor del parámetro **Limit**.

Modelado por Barrido

El cable vamos a modelarlo con la técnica del barrido (*Sweeping*). Para esto, tendremos que definir un contorno y un camino por el que pasaremos este contorno. El camino será un objeto de tipo path (**Spoobar** **Add/ Curve/ Path**). Situaremos el path como se muestra en la figura 15. Podemos añadir nuevos puntos al path como si fuera una curva de Bezier (como vimos en la segunda práctica, con **Ctrl** **M**).

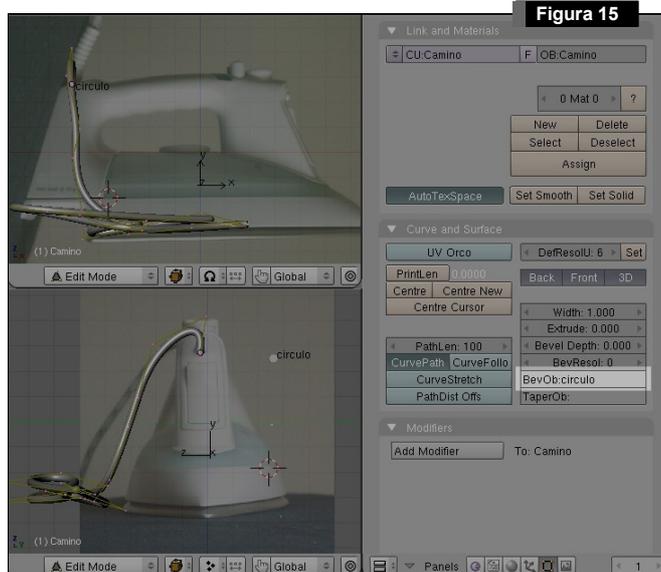


Figura 15

El objeto a "barrer" será un círculo de Bezier (**Spoobar** **Add/ Curve/ Bezier Circle**). Nombraremos el círculo insertado como queramos en el campo **OB**: de los botones de edición (pestaña de **Link and Materials**); por ejemplo "circulo". Ahora seleccionamos el path, y en modo de edición de objeto, ponemos el nombre del objeto "circulo" en el campo **BevOb** de la pestaña **Curve and Surface** (figura 15).

Hecho esto, debemos obtener un resultado como el mostrado en la figura 16. Si modificamos el círculo (por ejemplo, cambiamos el tamaño), la superficie barrida reflejará estos cambios. Al igual que con las superficies curvas vistas antes, podemos convertir la superficie a malla poligonal con **Alt** **C**.

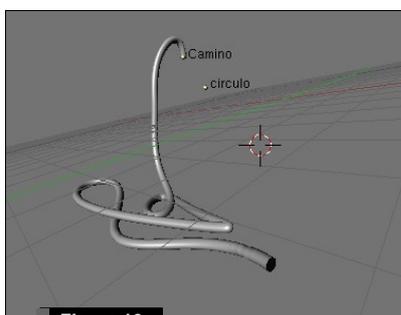


Figura 16

Superficies de Revolución

El enchufe vamos a modelarlo mediante una superficie de revolución. Para ello, añadimos a la escena una curva de Bezier, la convertimos a Poly y modificamos el contorno hasta obtener una figura como la mostrada en la figura 17.

Convertimos la curva a malla poligonal (**Alt** **C**), y pasamos a modo de edición de objeto (**Tab**). Trabajaremos a nivel de vértices (cambiar el modo en la cabecera de la ventana

3D), seleccionamos todos los vértices (**A**). Situamos el puntero 3D del ratón (**M**) en la posición del eje donde queremos realizar la revolución. Como se muestra en la figura 17, el eje está situado en el extremo derecho de la silueta.

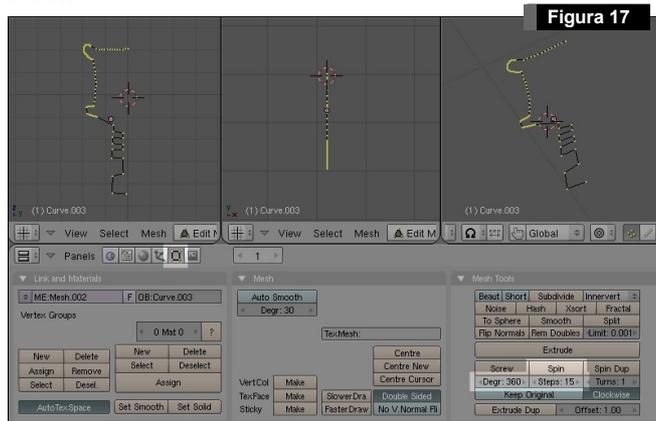


Figura 17

En la pestaña **Mesh Tools** tenemos el campo **Degr** donde indicaremos el número de grados que queremos realizar la revolución (en nuestro caso, un giro completo; 360) y el número de pasos intermedios que queremos que se calculen **Steps**; a mayor número de pasos, más suavidad. Hemos puesto 15 pasos intermedios. Con esto, pinchamos en el botón **Spin**, y el cursor del ratón cambia de forma, preguntándonos sobre qué vista queremos realizar la revolución. Tendremos que indicar una vista que sea perpendicular al eje de revolución. En la figura 17, queremos realizar la revolución sobre el eje **Z**, por tanto elegiremos la vista de en medio, definida sobre el plano **YX** (perpendicular al eje Z). Cuando pinchemos sobre la vista **M**, obtendremos una superficie de revolución como la mostrada en la figura 18.

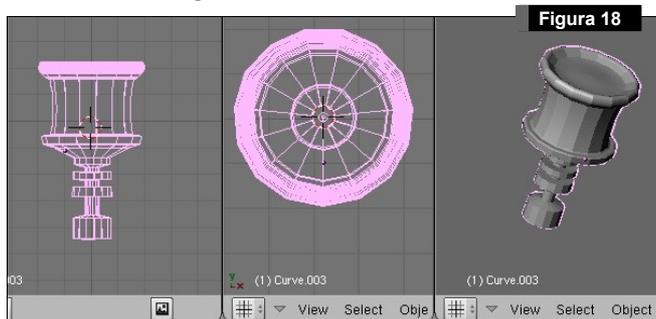


Figura 18

Si la figura de revolución obtenida no es esa, asegúrate que el cursor 3D de la escena está colocado correctamente, como se ha explicado en el párrafo anterior. La revolución se realiza teniendo en cuenta la vista sobre la que se pincha después de haber seleccionado **Spin**, y la posición del cursor 3D. Si Blender no muestra los botones del grupo **Mesh Tools**, recuerda que debes convertir la superficie a malla con las teclas **Alt** **C**.

En la siguiente parte de la etapa utilizaremos una herramienta de copia de objetos.

Duplicados de Revolución

En esta fase del modelado, añadiremos los cilindros y objetos auxiliares del modelo. En la parte central, queremos construir una malla que sea copia en espejo circular. Para ello, añadimos un plano que vamos a duplicar mediante **SpinDup**, y lo escalamos tal y como muestra la figura 19.

Con el plano añadido seleccionado, en modo de edición de objeto y el cursor 3D situado en el centro de revolución

(ver figura 19), elegimos el número de pasos de revolución (en este caso **Steps** es 10), y los grados (igual que antes, **Degr** 360). pinchamos en el botón **SpinDup**.

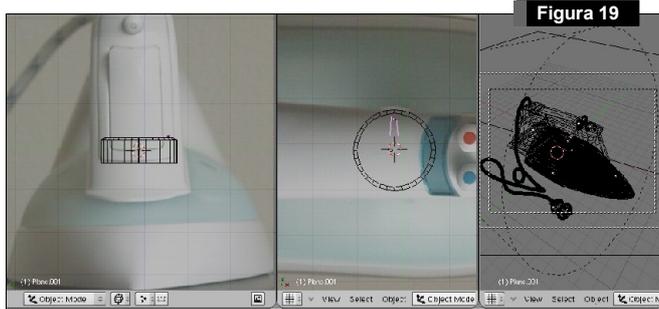


Figura 19

Al igual que ocurría con el operador Spin, blender nos pregunta la vista sobre la que queremos realizar la revolución. Tendremos que elegir la vista superior (figura 20, ventana central). Obtendremos un resultado como el mostrado en la figura 20.

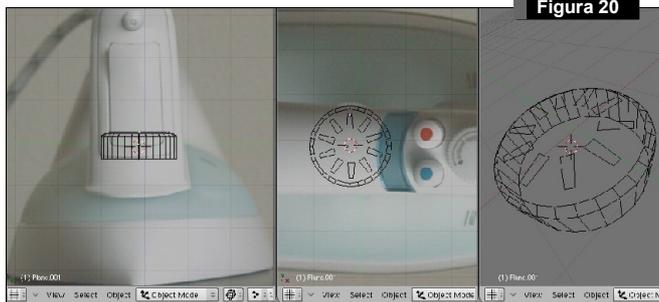


Figura 20

Separación de mallas

Para el compartimento frontal, hemos duplicado y separado parte de la malla original del asa. En modo de edición de objeto, y con selección de caras, seleccionamos el grupo de caras que se indica en la figura 21. Con las caras seleccionadas, las duplicamos con **Shift+D**. Como hemos añadido nuevas caras en modo de edición de objeto, estas caras forman parte del mismo objeto. Queremos que formen un objeto independiente que podamos rotar y manejar más cómodamente. Para ello, con las caras aún seleccionadas, pulsamos **P Separate/ Selected**.

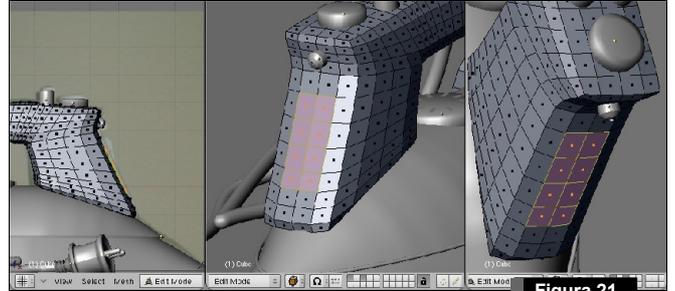


Figura 21

Finalmente, con operaciones de extrusión y rotado, construimos la pieza frontal de la plancha (ver figura 22).

En próximas sesiones de prácticas aprenderemos a manejar otras características de modelado con Blender, como superficies de subdivisión y metasuperficies (muy empleadas en modelado orgánico), modelado mediante técnicas de animación, etc...

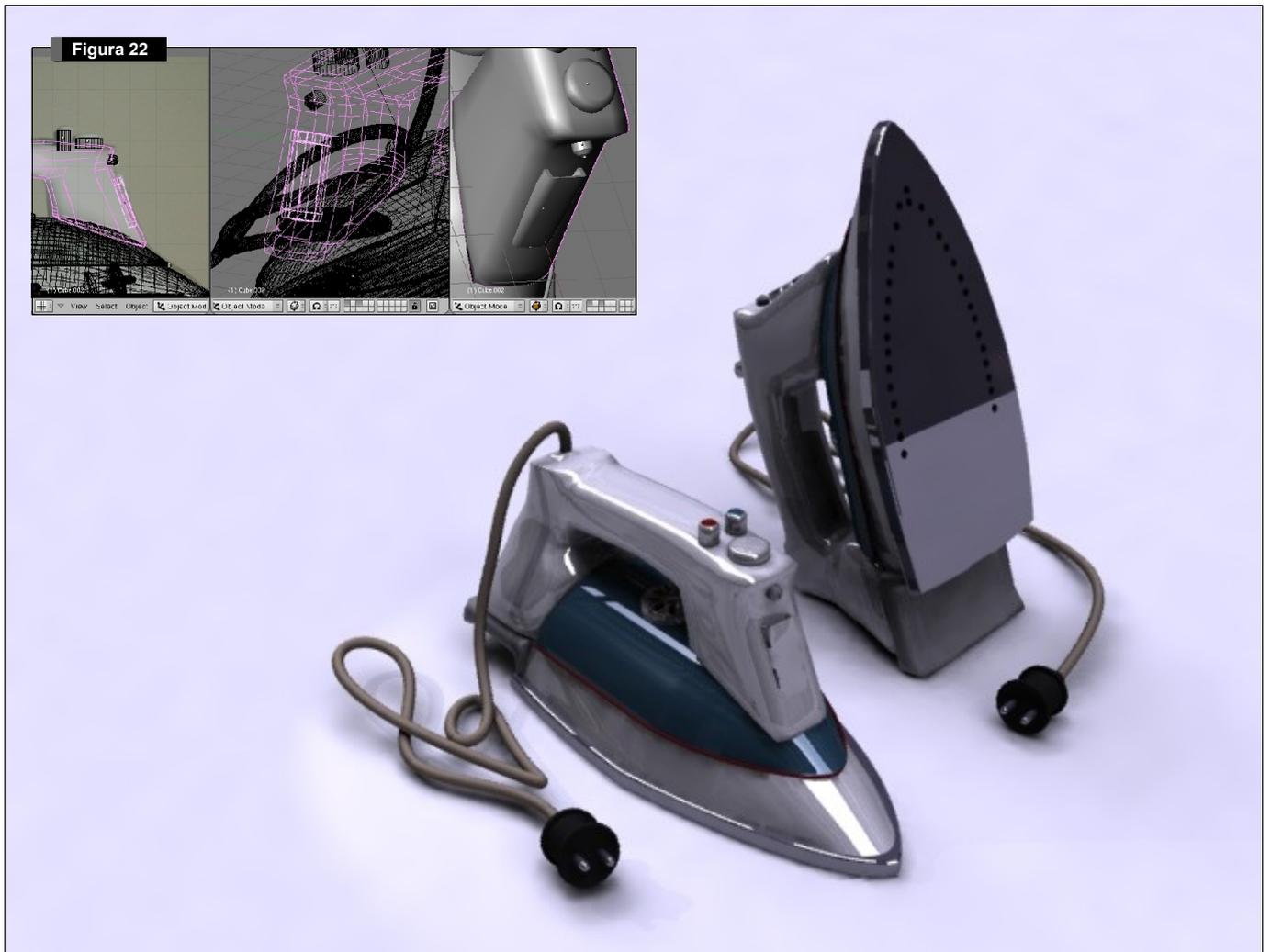


Figura 22



Modelado de un Personaje

Animación para la Comunicación · Escuela Superior de Informática · Universidad de Castilla-La Mancha

➤ Una técnica muy empleada en modelado de personajes es el uso de Superficies de Subdivisión con Rotoscopia. En esta sesión construiremos un sencillo personaje mediante este tipo de mallas poligonales.

Partiremos de dos vistas del personaje realizadas a mano y posteriormente escaneadas (figura 1). Cargamos los bocetos como fondo de dos ventanas 3D (en el menú **View/ Background Image** de la cabecera). Como puede verse en la figura 2, ha sido necesario el ajuste de la escala de las imágenes para que el tamaño en la escena de las mismas sea el mismo. En este caso, la vista lateral se ha escalado en 1.66 unidades y la frontal en 5 unidades.

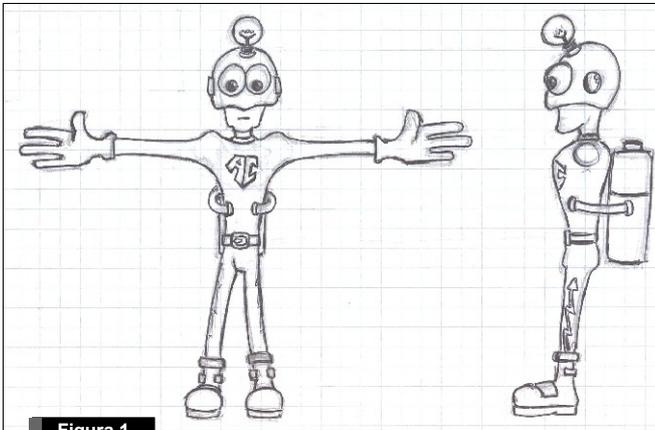


Figura 1

Tendremos que crear al menos 3 ventanas 3D (dos para poner las vistas frontal y lateral del modelo, y al menos otra donde rotaremos la vista para trabajar más cómodamente con el modelo. En la ventana 3D de la izquierda ponemos la vista frontal (perpendicular al plano XZ, accesible mediante **1** del teclado numérico). En la ventana siguiente, pondremos la vista lateral, perpendicular al plano YZ (**2** del teclado numérico).

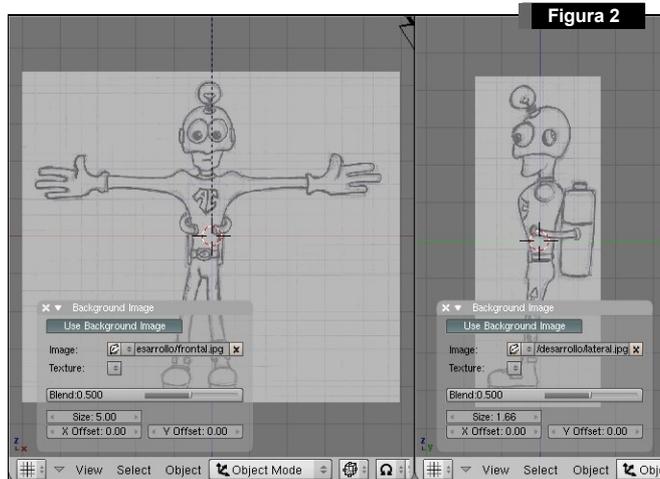


Figura 2

Debemos obtener una configuración similar a la mostrada en la figura 2. En la realización de los bocetos y su recorte se debe tener cuidado para que la posición de los elementos coincidan en altura.

Modelado con Superficies de Subdivisión

Comenzamos añadiendo un cubo a la escena. Con el cubo seleccionado, vamos a los botones de edición **E** y añadimos un Modificador de tipo **Subsurf**, para que la superficie sea de Subdivisión (figura 3).

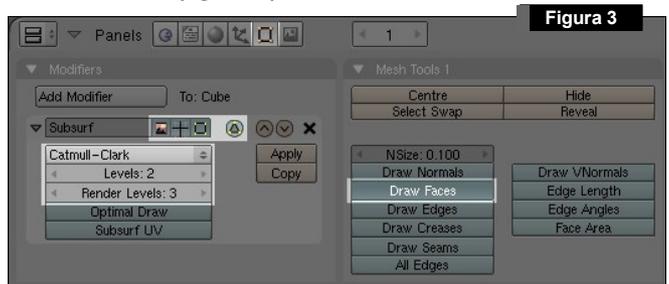


Figura 3

La lista desplegable del modificador que hemos añadido nos permite elegir entre el esquema de subdivisión simple (donde no hay fase de recolocación de vértices) o superficies de **Catmull-Clark** (las que nos interesan para esta práctica). Las cajas de texto inferiores sirven para especificar el nivel de subdivisión mientras trabajamos interactivamente **Levels**, y el que se alcanzará en la etapa de render **Render Levels**. En la figura 3 vemos que el primer valor es inferior, que es lo habitual; tendremos un valor de subdivisión mayor en la etapa de render. Podemos indicar en los cuadros que aparecen al lado de la caja Subsurf (ver figura 3) si se aplicará en la etapa de render, en la representación de objeto y en la etapa de edición del objeto. Finalmente, el botón situado un poco más a la derecha hace que únicamente se represente la superficie límite (controlada mediante los parches poligonales de la red de control). Es recomendable tener este botón activado porque permite un modelado mucho más intuitivo con este tipo de superficies.

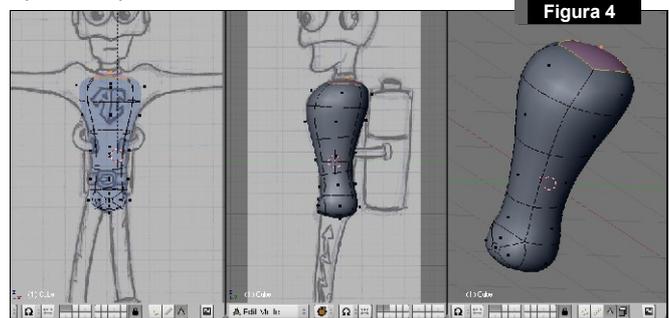


Figura 4

Puede resultar cómodo activar el botón **Draw Faces** de la pestaña **Mesh Tools 1** (figura 3), para tener una representación clara de la cara que tenemos seleccionada. También es útil cambiar entre representación **Wireframe** y **Sólida** (tecla **Z**) mientras estamos modelando. La representación wireframe nos permite ver el fondo a mientras modelamos, pero la sólida oculta las caras ocultas respecto del punto de vista, lo que resulta muy cómodo en el modelado de muchas zonas. En la figura 4 las ventanas

de la derecha tienen representación sólida mientras que la frontal utiliza wireframe. Partiendo del cubo, seleccionando la cara superior hemos realizado 4 extrusiones. Se ha cambiado entre modo de trabajo a nivel de caras y aristas (seleccionándolo en la cabecera de la ventana 3D, ver figura 5) para ir ajustando la posición de las aristas para que la superficie se ajuste a los contornos que muestra la figura 4. Es fundamental tener presente las caras de las que vamos a realizar próximas extrusiones (por ejemplo, los laterales de la última extrusión superior servirán para sacar los brazos).



Figura 5

En la figura 6 se muestra uno de los pasos seguidos para conseguir el contorno anterior. Partiendo de la posición superior, se ha trabajado en modo de edición de aristas y se ha desplazado la arista seleccionada para que se ajuste al contorno. Puede ser necesario en algunas ocasiones rotar caras para que se ajusten mejor a la forma perseguida.

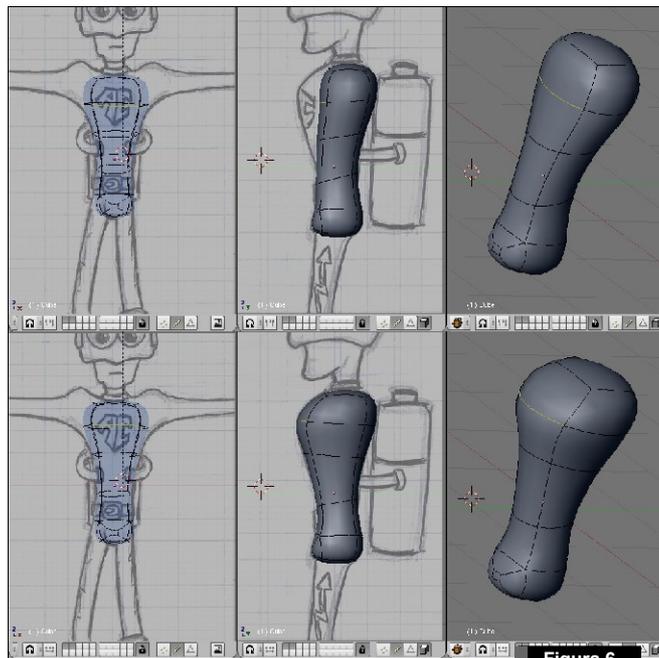


Figura 6

Realizamos una extrusión para sacar los hombros (ver figura 7). En este caso, ha sido necesario escalar la cara respecto del eje Y (coordenadas globales) para que el brazo no saliera achatado. Esta operación fue también necesaria en las piernas por el mismo motivo. Realizamos nuevas extrusiones para crear el brazo.

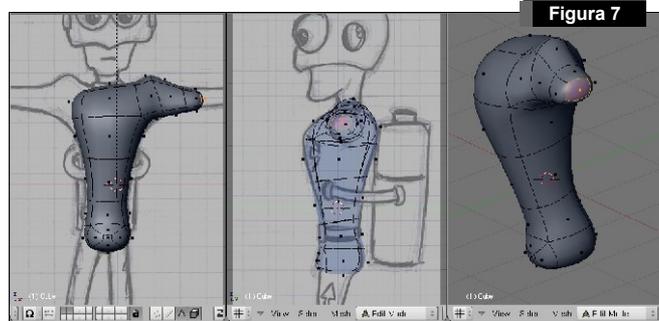


Figura 7

Las articulaciones del personaje requerirán un mayor nivel de detalle. Es por esta razón por la que se han realizado sucesivas extrusiones en los brazos (cuando con una extrusión hubiera sido suficiente para conseguir la forma del brazo completo). En la figura 8 puede verse el proceso seguido para su modelado.

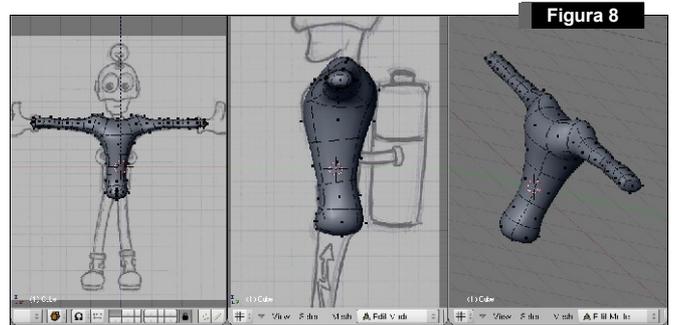


Figura 8

De las caras laterales de la parte inferior del tronco, se sacarán las piernas (ver figura 9). Al realizar las extrusiones, tendremos que rotar las caras para sacar correctamente el resto de caras. Sin embargo, como se ve en la figura 9, esto no es suficiente. Tendremos que entrar en modo de selección de aristas y desplazar las aristas inferiores para que el puente creado entre las piernas sea menor.

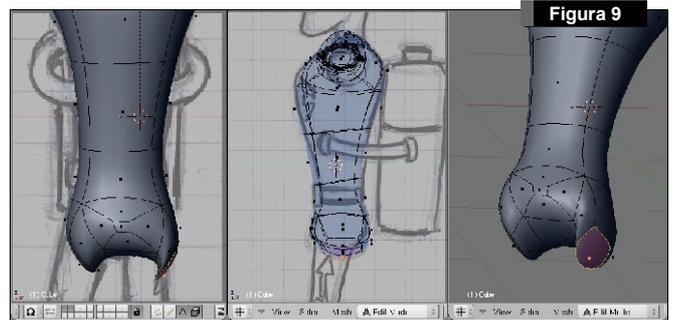


Figura 9

Ajustando correctamente esta distancia y la rotación de las nuevas caras creadas, obtendremos un resultado como el que se muestra en la figura 10.

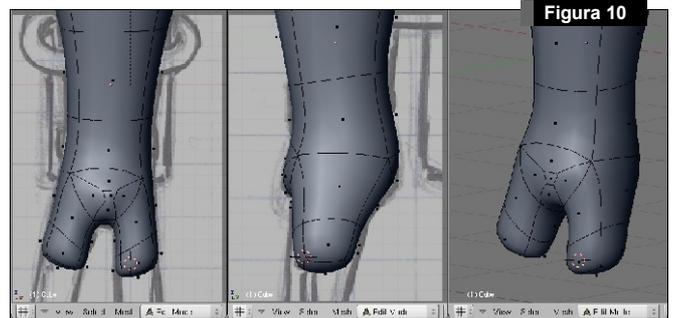


Figura 10

De forma análoga a como añadimos más nivel de detalle que el necesario para el modelado en los brazos, en la zona de las rodillas realizamos sucesivas extrusiones que nos definirán las articulaciones correctamente (ver figura 11) cuando añadamos el esqueleto interno en próximas sesiones de prácticas.

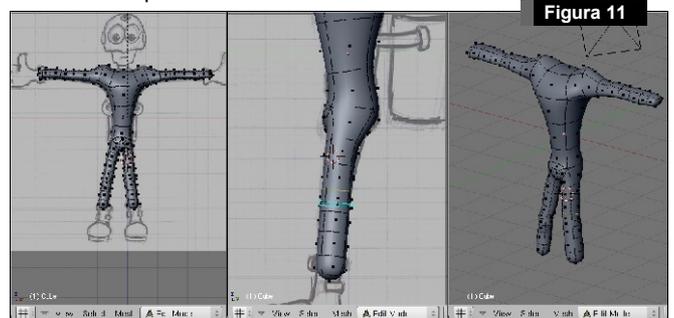


Figura 11

El modelado de las botas fue bastante sencillo. Partimos de un cubo al que realizamos dos extrusiones en horizontal para la puntera y una más para rematar el talón. La parte donde une con las piernas se ha realizado de nuevo con extrusiones (figura 12), ajustando el tamaño de las caras y posicionando manualmente algunas aristas.

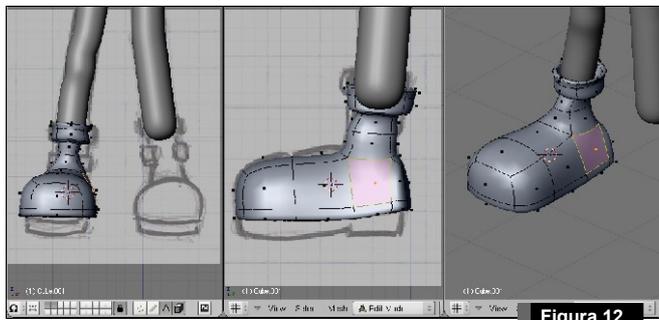


Figura 12

Cuando terminemos una bota, podemos duplicarla con efecto de espejo (para que siga la forma del otro pie) mediante **Shift+D** para duplicar y **Spacebar** **Transform/ Mirror/ XYZ Local**. Son ejes locales al objeto, por lo que deberemos activar la visualización del sistema de coordenadas local a la bota (en los botones de objeto, el botón **Axis**, situado en la pestaña **Draw**, dentro del grupo **Draw Extra**). En el ejemplo de la práctica se ha realizado el mirror respecto del eje Z (figura 13).

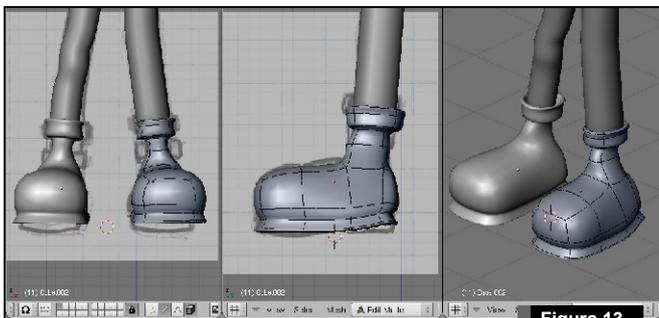


Figura 13

Modelado de la cabeza

Hemos comentado anteriormente la importancia en el modelado de las articulaciones y músculos que tendrá el personaje. También hay que pensar en facilitar el modelado de todos los elementos del personaje. Por ejemplo, la figura 14 es un claro ejemplo de lo que **no** se debe hacer. Nos hemos centrado en crear la forma de la cabeza con el menor número de polígonos posibles (lo cual en general es bueno), pero nos dificulta mucho ahora la creación de la boca.

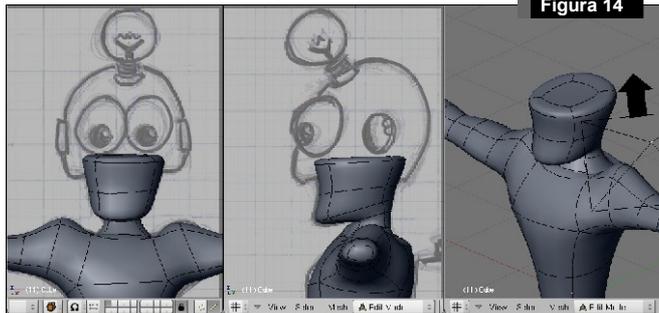


Figura 14

En el caso de la figura 14 habría que crear una cara en la zona donde tenemos una arista. Esta operación no es inmediata, y tendríamos que añadir nuevos vértices auxiliares e insertando la nueva cara manualmente. Una mejor planificación (como la mostrada en la figura 15) evita esta situación.

En el caso de la figura 15, se ha partido de un cubo al que se ha aplicado dos extrusiones horizontales (dejando la zona original en el centro). A la nueva región (formada por 3 caras), se ha aplicado 4 extrusiones verticales (para darle toda la altura hasta la zona donde irá el casco). Para terminar el modelado, se ha extruido 2 veces para sacar la zona de las orejas (si tuviera) y el rostro.

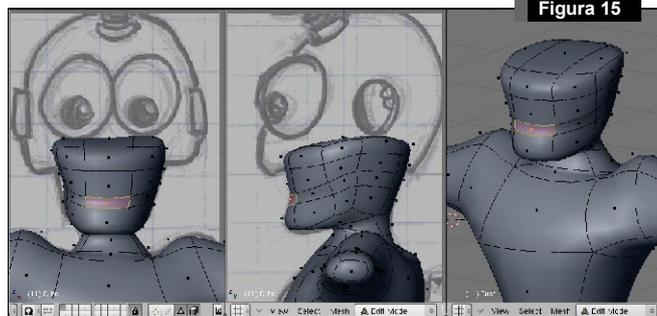


Figura 15

Aquí se comprueba claramente que ha quedado un polígono específico para la boca, que además se podrá manejar muy cómodamente con las 4 aristas que la definen. Para realizar la cavidad de la boca, bastará con extruir hacia el interior y ajustar las aristas para completar la forma que se muestra en la figura 16.

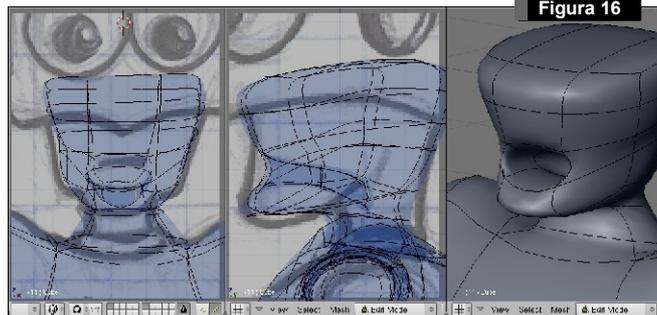


Figura 16

El casco se ha realizado a partir de un cubo. También es una superficie de subdivisión. La posición de las aristas y número de subdivisiones puede verse en la figura 17.

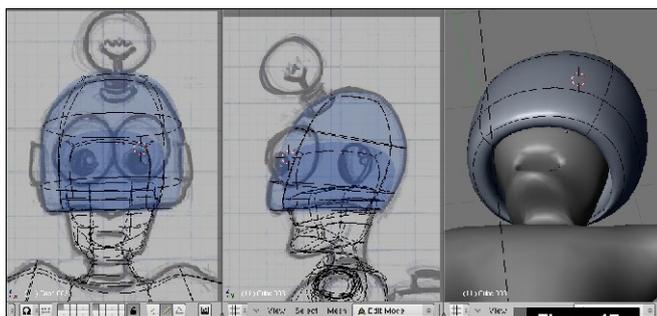


Figura 17

Las manos se han creado siguiendo los principios de diseño de *cartoons*; tienen 4 dedos y son extremadamente grandes en relación con el resto del cuerpo. Esto da mayor expresividad al personaje.

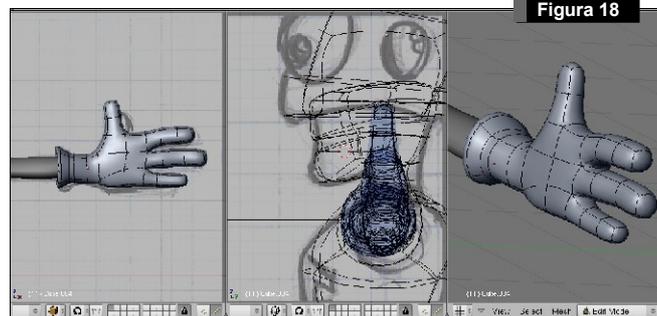


Figura 18

En la figura 18 se muestra la disposición en parches. Como puede comprobarse, hasta la zona del pulgar, todo se ha realizado partiendo de extrusiones de un único cubo. Después, se han realizado dos extrusiones partiendo de la cara superior e inferior del cubo para sacar la palma, y de ahí los dedos. Se pueden añadir más subdivisiones en las zonas donde habrá articulación en los dedos. La figura 19 resume el proceso seguido para la creación de la mano.

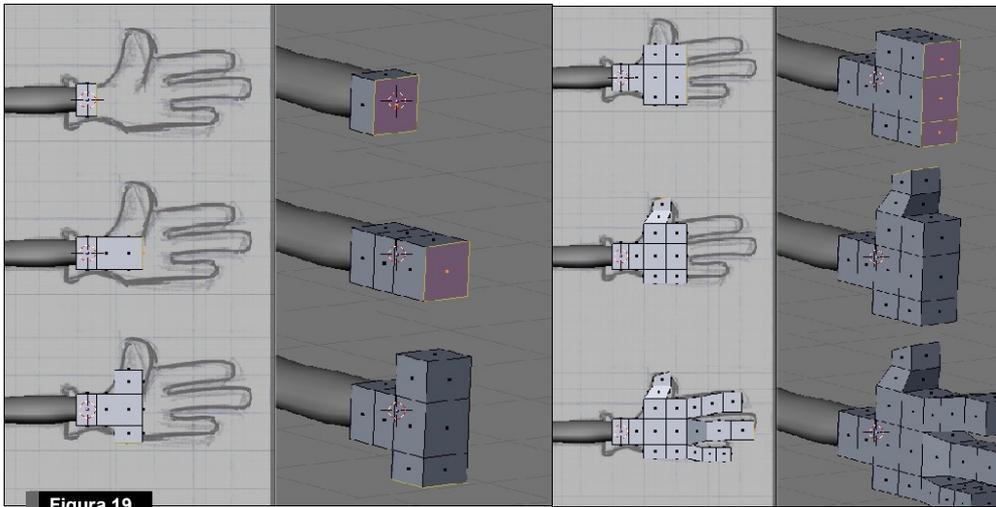


Figura 19

Al igual que realizamos con el modelo de la bota, la mano puede duplicarse y realizar un mirror para crear la otra. Pasamos a ver cómo se han realizado los ojos. Se ha partido de una **UVSphere**, a la que en modo de selección de vértices se ha seleccionado la zona de la córnea y se ha separado  como un objeto independiente. Se ha duplicado la córnea, y se ha rotado para cambiar la zona cóncava para obtener el iris. El modelo estará formado por tres piezas, como muestra la figura 20. Estas piezas se juntarán para formar un único objeto (mediante  **Join Selected Meshes**).

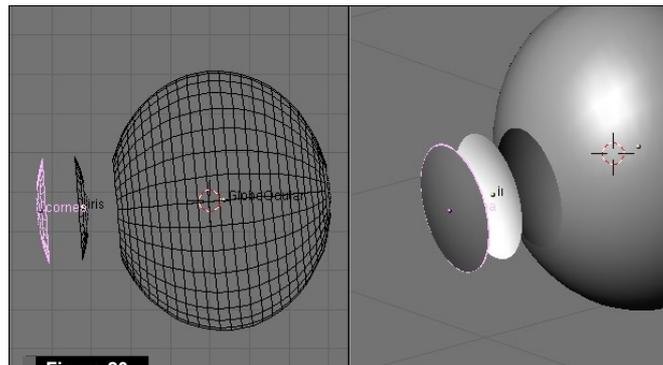


Figura 20

Los materiales que forman las piezas del ojo son muy sencillos. El iris utilizará una textura de imagen sin propiedades especiales. La córnea será transparente y tendrá un pequeño IOR para hacer efecto lente. Las propiedades de la córnea pueden verse en la figura 21. Veremos el trabajo con materiales en profundidad en la siguiente sesión de prácticas.

Faltarían por añadir los detalles adicionales del modelo (detalles de las botas, la bombilla del casco, los dientes, la lengua, el cinturón, etc...). Estos modelos han sido creados con técnicas explicadas en sesiones anteriores, por lo que no dedicaremos tiempo en explicarlos en esta sesión. En la próxima sesión dedicada a materiales y texturas veremos, entre otras cosas, cómo se ha texturizado el cuerpo del personaje mediante la técnica de UV Mapping. El resultado del modelado geométrico del personaje puede verse en la figura 22.

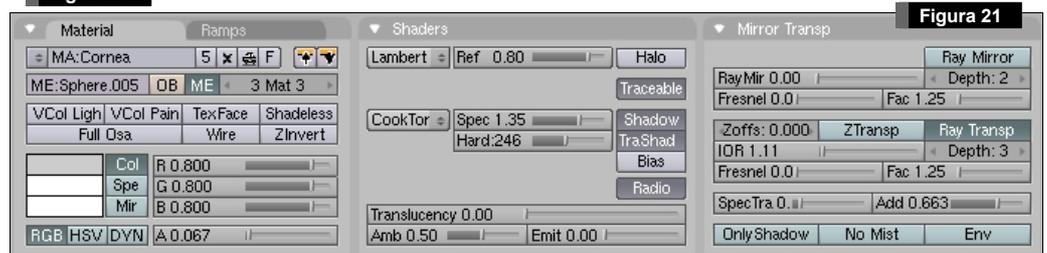


Figura 21

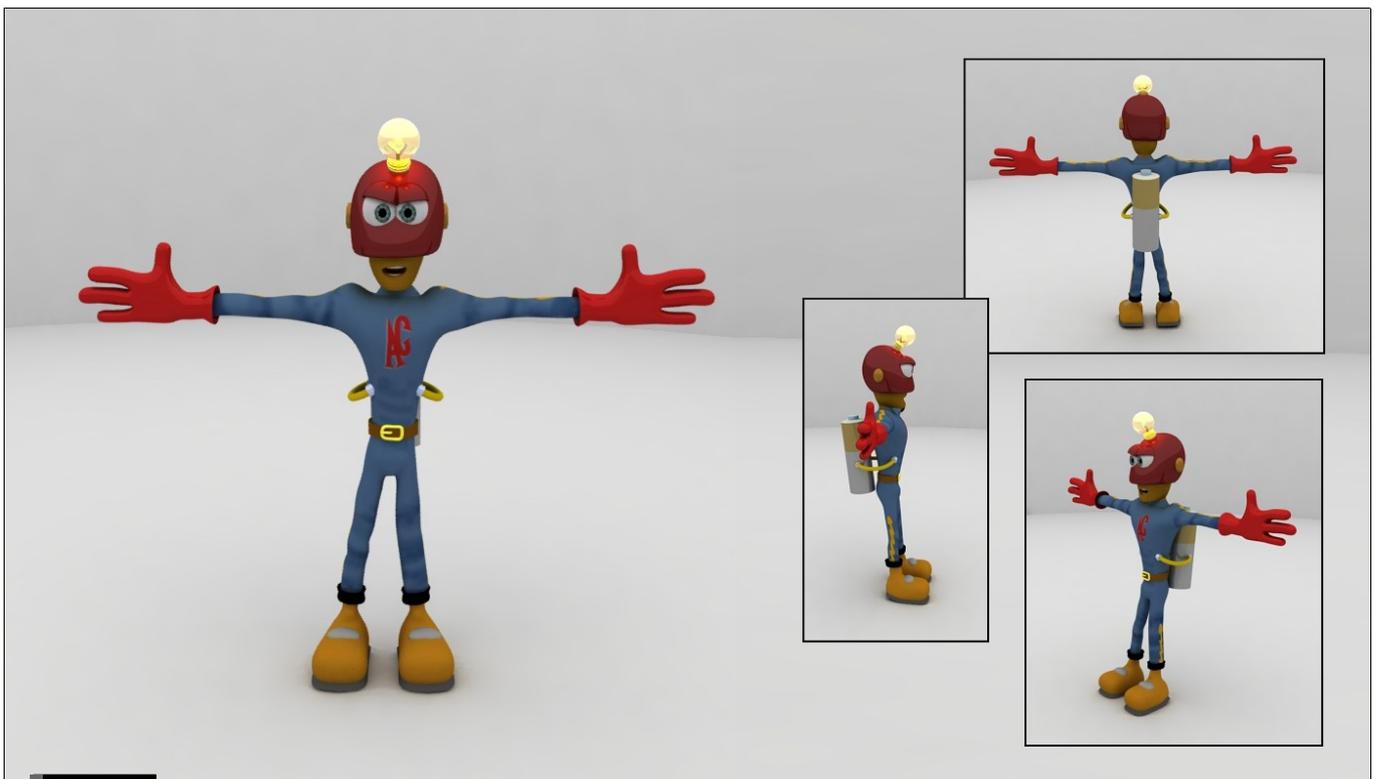


Figura 22



➤ Para obtener resultados realistas es necesario establecer correctamente los parámetros de materiales y texturas. En esta sesión trabajaremos con texturas procedurales, imágenes y capas de texturas.

Vamos a comenzar modelando los elementos de la escena a los que después aplicaremos materiales y texturas. Vamos a modelar una escena como la que se muestra en la figura 1. En esta escena, la base de los botes de spray han sido modelados mediante superficies de subdivisión. La cadena ha sido realizada mediante una técnica avanzada de modelado, basada en IPOs de animación, y algunos elementos de relieve (como las estrías de la tapa del bote de pintura) han sido realizados mediante **bump mapping**.

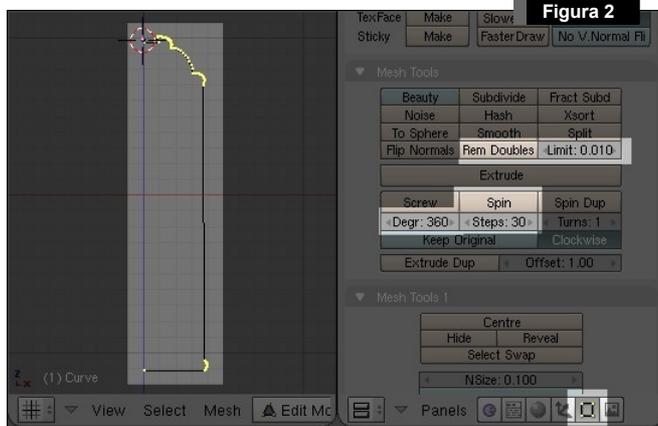


Figura 2

guramos los parámetros de revolución, indicando que vamos a realizar el giro de 360°, con 30 pasos intermedios (ver figura 2). Debemos obtener un modelo como se muestra en la figura 3. En este modelo existe un problema, y es que hay una columna de vértices que están duplicados (el contorno original y el último paso de revolución coinciden en la misma posición). Para eliminar vértices repetidos, en modo de edición de vértices (Tab), los seleccionamos todos (A) y establecemos un valor pequeño de la variable **Limit** situada al lado del botón **RemDoubles**. Esta variable nos indica qué distancia vamos a considerar la que nos distinga vértices. Si indicamos un valor grande, Blender eliminará más vértices que considerará repetidos. Con el valor de **0.01** en **Limit** y todos los vértices del modelo seleccionados, pinchamos en RemDoubles y Blender nos indicará el número de vértices eliminados (los correspondientes a la columna de vértices repetidos).

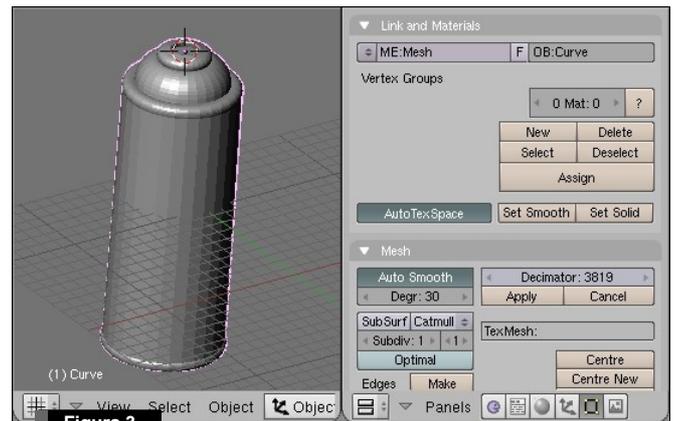


Figura 3

Creamos un perfil como el que se muestra en la figura 2, añadiendo una curva de bezier y ajustando los puntos de control, como hemos visto en sesiones anteriores. Antes de realizar la operación de **Spin**, tendremos que convertir el contorno curvo a poligonal, mediante (Alt)C. Hecho esto, nos vamos a los botones de edición (E), y entrando en modo de edición de vértices, los seleccionamos todos (A) y confi-



Figura 1



Figura 4

Pasamos a indicar el sombreado que queremos aplicar a esta superficie; suave o plano. Un sombreado suave interpolará el valor de suavizado calculando el vector normal en cada punto de la superficie, mientras que un sombreado plano sólo lo hará en el centro de cada cara. Con el objeto seleccionado, indicamos a blender que queremos que aplique un sombreado suave, pinchando en el botón **Set Smooth** de la pestaña **Link and Materials** de los botones de edición. El sombreado plano se consigue pinchando en **Set Solid** del mismo grupo de botones. En la figura 4 tenemos un ejemplo de cada tipo de sombreado aplicado en blender.

Sin embargo, nos interesa que se aplique un tipo de sombreado en cada parte del modelo, diferenciando zonas claramente. Para ello, después de pinchar en Set Smooth podemos activar el botón **Auto Smooth** de la pestaña **Mesh**, indicando en **Degr** el valor que hará que Blender calcule los bordes de la transición entre zonas cuyos vectores normales formen un ángulo superior a esa cantidad (en el ejemplo de la figura 3 es de 30°).

El resto del modelado es muy sencillo, dos cilindros adicionales para crear el difusor del spray y un cilindro más para realizar la tapadera.

Importación de objetos

La cadena de la escena se ha modelado utilizando una técnica que emplea las curvas de animación para repetir elementos de modelado a lo largo de un camino llamada **DupliFrames**. Estudiaremos esta técnica en próximas sesiones.

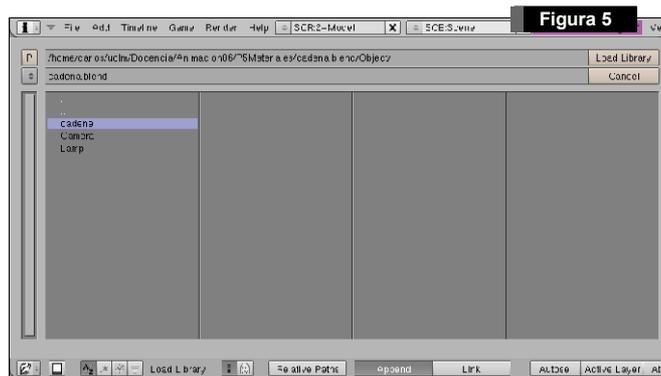


Figura 5

Para importar un objeto que está en otro archivo, vamos al menú **File / Append...** Seleccionamos con el fichero "cadena.blend", y entramos en la categoría Object. Ahí, podemos seleccionar con todos los objetos que sean necesarios (en nuestro caso sólo elegiremos el objeto "cadena". Con el objeto seleccionado (figura 5), pinchamos en el botón **Load Library**. Podemos importar de otros ficheros prácticamente cualquier dato del DNA (formato de ficheros interno de Blender). Bastará con acceder a la categoría correspondiente (o importar todos los objetos de una categoría seleccionando el nombre de la categoría).

Ajustando Materiales

Añadimos dos planos a la escena que servirán como suelo y como pared. Vamos a explicar algunas propiedades de los materiales, dentro de los botones **Shading** / **Material**:

Pestaña Material (figura 6)



Figura 6

- **MA:** El nombre del material; en el caso de la figura 6 "graffiti".
- El botón de la derecha del nombre del material indica el número de usuarios de ese material (en el caso de la figura 6, hay dos objetos que comparten este material). Si aparece únicamente la X indica que el material sólo lo utiliza un objeto. Podemos pinchar sobre ese número para crear una copia del material que sólo utilizará el objeto actual. Así, podemos cambiar fácilmente las propiedades de un material heredado sin afectar a las propiedades de los demás objetos.

- La **X** elimina el material.
- El botón del **coche** hace que Blender asigne automáticamente un nombre al material.
- **F:** Hace que se cree un usuario ficticio para ese material y que Blender lo salve dentro del fichero .blend (en el caso de que un material no esté siendo utilizado por nadie; es decir, se haya eliminado), Blender no lo guardará en el fichero.
- **VcolLight** y **VcolPaint:** utilizados con vertex colours.
- **TexFace:** Si está activo, los colores se toman de la textura asignada en el editor de UV Mapping.
- **Shadeless:** Si está activo, se ignora la información de sombreado en la fase de render.
- **Strands:** Opciones de textura para sistemas de partículas. Las utilizaremos en próximas sesiones.
- **Full Osa:** Fuerza a blender a que calcule todas las pasadas de oversampling.
- **Wire:** Renderiza el material en modo alambre.
- **Col:** Color base de la superficie
- **Spec:** Color de especularidad.
- **Mir:** Color de mirror; afecta únicamente en el caso de usar mapas de entorno.
- **A:** Alpha; nivel de transparencia de la superficie (entre 1, totalmente opaco y 0 totalmente transparente)

Pestaña Shaders (figura 7)

- La primera lista desplegable permite elegir el tipo de sombreado difuso. El parámetro principal y compartido entre todos los métodos es la cantidad de Reflexión de luz **Ref** de la superficie).
- La segunda lista desplegable permite elegir el método de sombreado especular. Según el método permitirá un conjunto de parámetros; aunque seguro que tenemos **Spec:** Cantidad de brillo especular y **Hard** dureza del brillo. Puede comprobarse en la ventana de previsualización cómo afectan estas propiedades al material.
- **Translucency:** Cantidad de reflexión difusa en la cara posterior del objeto.

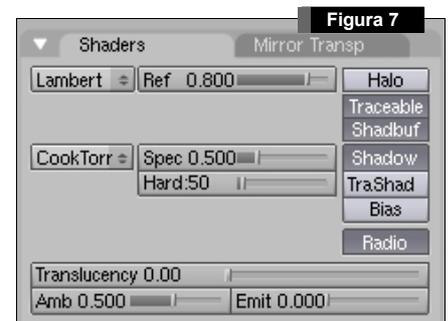


Figura 7

- **Amb:** Cantidad de color ambiental que se aplica. Este color se indica en las propiedades del mundo.
- **Emit:** Cantidad de luz emitida por el objeto.
- **Traceable:** Si está activo, el material arroja sombras.
- **Shadbuf:** Si está activo, el material recibe sombras con Shadow Buffer.
- **Shadow:** Si está activo, el material recibe sombras.
- **TraShadow:** Si está activo, el material recibe sombras semitransparentes.
- **Radio:** El material está activo para el cálculo de radiosidad.

Pestaña Mirror Transp (figura 8).

Útil si se utiliza renderizado mediante trazado de rayos (en las propiedades de renderizado, habrá que activar el botón **Ray**).

Si activamos el botón **RayMirror**, el material se comportará como un espejo:



Figura 8

- **RayMirr:** Indica la cantidad de reflexión de la luz.
- **Depth:** La profundidad de rebotes en el trazado de rayos (en este caso, 2).

Si activamos el botón **RayTransp**, el material será transparente; además habrá que bajar el parámetro alpha por debajo de 1).

- **IOR:** Índice de refracción; mide la diferencia de densidad entre superficies.
- **Depth:** Profundidad de rebotes de luz.

Si empleamos el motor de render *Yafray*, las propiedades mostradas en este panel cambiarán. Veremos cómo utilizar *Yafray* como motor de render en próximas sesiones de prácticas.

Pestaña Texture (figura 9).

Permite gestionar en capas las texturas que se aplican a un objeto. La pila de texturas está situada en la parte izquierda. Pueden estar activas (botón verde activado) o no. A la derecha aparece el nombre de la textura que se aplica en cada capa, y se pueden añadir, o eliminar texturas. El número que aparece en el botón indica el número de objetos que están utilizando esa textura. La textura que se aplica finalmente al objeto resulta de la composición de todas las texturas con sus propiedades correspondientes.

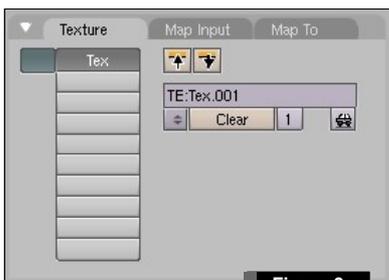


Figura 9

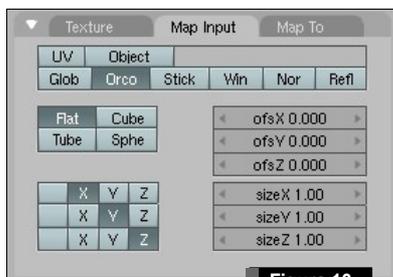


Figura 10

Pestaña MapInput (figura 10):

Cada textura en Blender tiene asociadas unas coordenadas

3D. Por defecto, vamos a trabajar con las coordenadas originales del objeto **Orco**.

El grupo de botones inferior indican la forma en la que las coordenadas de textura 3D se proyectarán en 2D; empleando un tipo de mapeado plano (**Flat**), cúbico (**Cube**), esférico (**Sphe**) o cilíndrico (**Tube**).

Los botones inferiores con etiquetas **XYZ** indican las nuevas coordenadas **XYZ** sobre las que se aplicará la textura. Normalmente, cada coordenada es mapeada con su análoga en **XYZ** (tal y como muestra la figura 15), pero podemos cambiar este mapeado, según las coordenadas de textura. Si activamos el primer botón sin etiqueta, desactivamos el mapeado en esa coordenada.

Los botones de la zona derecha permiten especificar traslaciones adicionales en las coordenadas de la textura (un **offset**), y un valor de escalado extra en la textura (**size**).



Figura 11

Pestaña MapTo (figura 11):

La primera fila de botones indican las propiedades sobre las que va a actuar la textura; algunas de estas propiedades son:

- **Col:** La textura afecta al color básico del material, según el factor indicado en la caja deslizante Col, y el valor de mapeado (la caja de selección superior).
 - **Mix:** el color de la textura se mezcla con el valor del material.
 - **Multiply:** el color de la textura multiplica al valor del material.

En caso de ser una textura procedural y no de imagen, el valor de color se indica en la caja situada encima de los selectores de color RGB.

- **Nor:** La textura afecta al vector normal. Puede ser un valor positivo o negativo (es un botón de tres estados). Si está activo, habrá bump mapping, con el factor indicado en la caja deslizante Nor.
- **Csp:** La textura afecta al valor de reflexión especular.
- **Cmir:** Afecta al color de reflexión (espejo).
- **Alpha:** Afecta al valor de transparencia Alpha.
- **Emit:** Afecta a la cantidad de emisión de luz.

Se pueden consultar todos los parámetros de texturas y materiales en el manual oficial de Blender.

Texturas de imagen

Seleccionamos el plano del suelo, y añadimos un material nuevo. Nos vamos a la pestaña de **Texture** y pinchamos en **Add New**. Aparecerá un nombre de textura por defecto; hecho esto, pinchamos en el botón de Texture Buttons, accesible con **T**, y cargamos una textura de imagen (elegimos **Image** en la lista desplegable **Texture Type**) y pinchamos en **Load Image**, cargando la imagen *suelo.jpg*. Vamos a configurar algunos parámetros (ver figura 12):

- En **TE:** Podemos indicar el nombre de la textura.
- **Extend:** Si está activo, indicamos que la textura se extienda para que ocupe toda la superficie donde vamos a aplicarla.
- **Repeat:** Si está activo, podemos indicar que la textura se repita un número de veces en el eje X y en el eje Y.



Esto es muy útil cuando tenemos texturas que se pueden repetir.

Si renderizamos la escena, el suelo deberá aparecer correctamente texturizado. Procedemos de forma similar para añadir la textura del muro, empleando la textura **graffiti.jpg**.

Creando grupos de vértices

El bote de spray en realidad es un único objeto con dos materiales distintos; uno para la etiqueta (figura 13), y otra sin textura, únicamente un material para el resto del objeto. Además, la textura de la etiqueta está formada por dos capas, una con la imagen y otra de "ruido", que simulará el papel ligeramente arrugado.



Figura 13

seleccionamos el resto de caras **W** **Select Swap** y pinchamos de nuevo en **Assign**. Con el botón **Select** podemos ver las caras que forman cada grupo, y con **Deselect**, deseleccionamos las caras contenidas en un grupo. Si todo ha ido bien, cuando pinchemos en **Select** estando en el primer grupo de vértices tienen que seleccionarse un conjunto de caras, y cuando pinchemos en **Select** estando en el segundo grupo, el resto de caras.

Hecho esto, vamos a los botones de material y seleccionamos el grupo con el que queremos trabajar. En la pestaña **Material**, la lado del botón verde **ME**, aparece un botón con el mismo formato que el que hemos visto antes. Seleccionamos el grupo de vértices que contienen la parte superior e inferior del bote y establecemos las propiedades del material como muestra la figura 15.

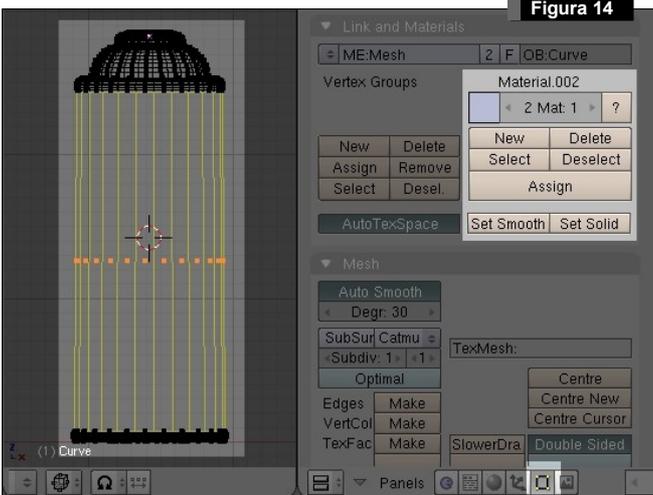
Como puede verse, se ha dado un alto nivel de reflexión especular, y propiedades de reflejo **Ray Mirror**. El color base de la superficie es gris.

Seleccionamos ahora el otro grupo de vértices en las propiedades de material, y vamos a añadir una textura de imagen, como vimos en el apartado anterior. Esta textura va a estar formada por dos capas; una con la imagen y otra que servirá para modificar la rugosidad de la superficie.

Para añadir la textura con la imagen, elegimos el tipo de textura **Image**, y cargamos la imagen (ver figura 17). En las propiedades de mapeado, tendremos que ajustar la imagen para que el mapeado cilíndrico sea correcto. Como se puede ver en la figura 16, se ha especificado un **offset** en **X**, el tipo de coordenadas de mapeado **"Tube"**, y se han cambiado las coordenadas de mapeado, debido a que la superficie de revolución con la que hemos creado el bote de spray tenía el sistema de coordenadas local cambiado respecto del origen global.

Ahora nos situamos en la segunda capa de texturas (pestaña **Texture**) y añadimos otra nueva. En esta ocasión

Figura 14



En modo de selección de caras, seleccionamos las caras que forman la parte central del bote de spray (donde vamos a aplicar la etiqueta), y en la pestaña de **Link and Materials** (figura 14) creamos un grupo nuevo de vértices. Para ello, pinchamos en **New** y luego en **Assign**. Con esto, tenemos un grupo de vértices que contiene las caras seleccionadas. Debemos tener en la parte superior una etiqueta que indica **2 Mat: 1**. Esto nos indica que tenemos dos grupos de vértices y que estamos trabajando con el primero. Pinchamos en el triángulo de la derecha de esta etiqueta para irnos al segundo grupo de vértices y asignarle el resto de caras. Cuando la etiqueta cambie a **2 Mat: 2**,

Figura 15





Figura 16



Figura 18

va a ser de tipo **Clouds** (figura 19). Queremos que tenga un ruido bastante marcado, como papel arrugado. En las propiedades del material, en la pestaña **Map To** hemos activado las opciones **Nor** (para que afecte a la rugosidad de la textura) y a la reflectividad de la superficie (los pliegues brillarán un poco más). El tipo de mapeado (en **Map Input**) sel podría haber definido como en la textura de imagen, pero como el fin de esta textura es únicamente añadir ruido, no se ha prestado atención en estos detalles. Si queremos que la rugosidad del papel se note más, tenemos que aumentar el factor **Nor** (en la figura 18 está a 0.50) a un valor superior.

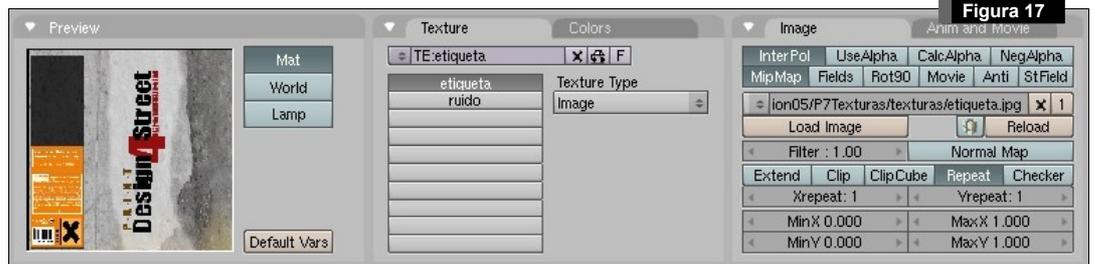


Figura 17

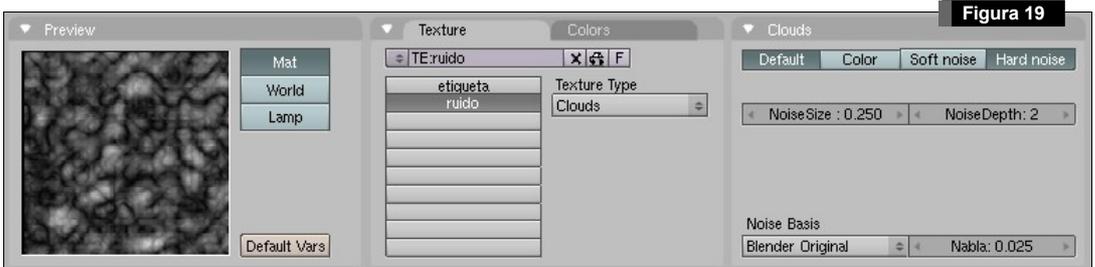


Figura 19



Figura 20

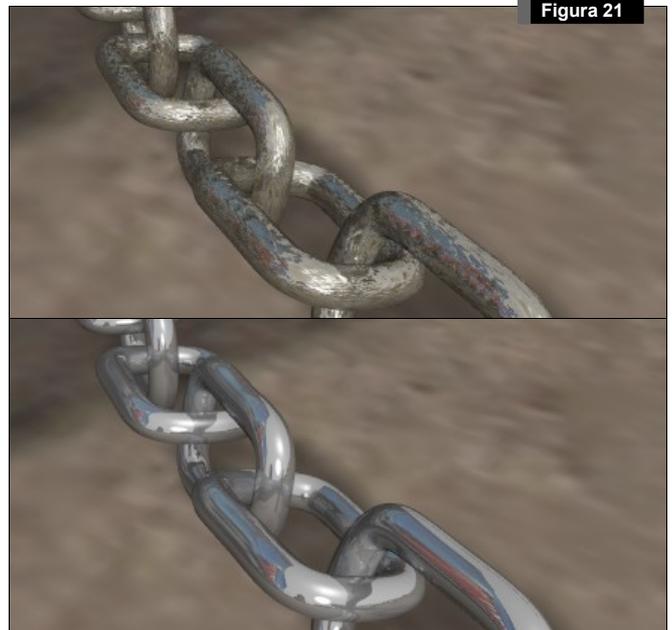


Figura 21



Figura 22

Las cadenas tienen un material con un color base, y propiedades de **RayMirror**. A la cadena se ha añadido también una textura para añadir ruido a la superficie, pero en este caso se ha definido que la textura afecta al color de la superficie y al valor del vector normal.

Podemos ver en la figura 20 las propiedades de mapeado de la superficie. En la figura 21 hay una comparativa del renderizado de las cadenas antes y después de aplicar la textura. Puede observarse los huecos de color marrón que se han conseguido, simulando el efecto del óxido. En este caso, un mismo mapa de textura se ha utilizado para modular varias propiedades.

Para realizar el render se ha añadido una fuente de luz de tipo **Spot** y una luz de tipo **Hemi** que añade iluminación ambiental. Para conseguir los reflejos realistas de **Raytracing** tenemos que activar el botón **Ray** en las propiedades de renderizado (ver figura 27).

Texturizado con UV Mapping

Nos centraremos ahora en el texturizado mediante UV Mapping. Este tipo de texturizado permite el posicionamiento exacto de una imagen a cada cara del modelo. Sin embargo tiene un problema a la hora de construir una representación de la malla tridimensional sobre la que podamos pintar cómodamente; la distorsión de las caras. Por suerte, disponemos de un método que minimiza la distorsión, el LSCM. Recuperamos el modelo del torso del personaje de la sesión anterior, y emplearemos este método de texturizado para dibujarle el traje.

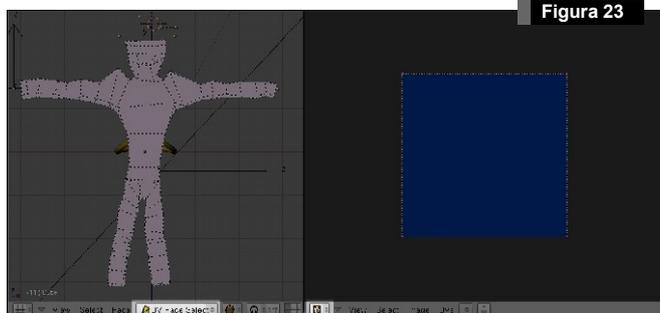


Figura 23



Figura 24

Accedemos al modo de trabajo de selección de caras para UVMapping mediante la tecla **F** o en la lista desplegable de la ventana 3D (ver figura 23). Cambiamos una de las áreas de trabajo a ventana **UV/Image Editor**.

Si pulsamos la tecla **U** en la ventana 3D (estando en modo de selección UV) nos aparece una lista de posibles modos de proyección sobre la imagen 2D. Para elegir el que mejores resultados ofrece en modelos complejos, tenemos que marcar primero las costuras (**seams**). Estas

costuras son zonas donde permitimos a blender que "corte" el modelo (imaginemos que queremos obtener un conjunto de trozos de piel del modelo, y le indicamos a blender por dónde meter la tijera ;-)).

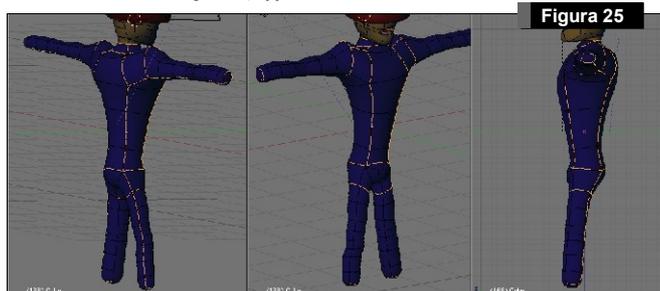


Figura 25

Volvemos al modo de edición de objeto y seleccionamos las aristas por donde queremos recortar el modelo (figura 24). Con cada grupo de aristas seleccionadas pulsamos **Ctrl+F**. Nos aparecerá un menú donde nos pregunta si queremos crear o eliminar la costura. Crearemos tantas costuras como creamos necesarias para que al partir el modelo sea cómodo el pintado. La figura 25 muestra cómo se ha partido el modelo para realizar esta práctica.

Volvemos al modo de trabajo con mapas UV (tecla **F**). Seleccionamos todas las caras **A** y en la cabecera de la ventana **UV/Image Editor** seleccionamos **UVs/ LSCM**

Unwrap. Con esto debemos tener una presentación como se muestra en la figura 26.

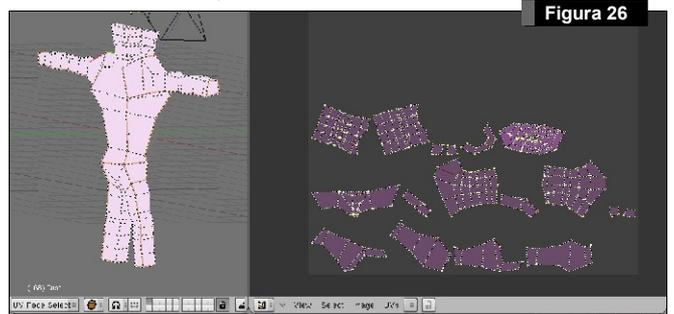


Figura 26

Exportaremos esta disposición de las caras, y trabajaremos sobre ellas en GIMP. Para esto, utilizaremos el script de exportación integrado en blender en el menú de la ventana UV (**UVs / Save UV Face Layout**). Le daremos una resolución mayor (por ejemplo, 1024 para que el modelo quede texturizado con mayor calidad). Esto nos guardará una imagen TGA que podemos abrir con el GIMP para añadir color. En la figura 27 se muestra la textura tal y como es exportada en Blender y después de ser pintada. Es recomendable que nos "salgamos" a la hora de pintar la textura, porque así tenemos la seguridad que la zona de unión entre caras queda perfectamente cubierta de color.

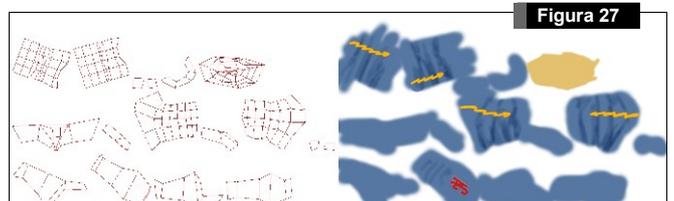


Figura 27

Cuando tengamos la textura pintada en GIMP, la guardamos y la importaremos en blender desde el menú **Image** de la cabecera de la ventana **UV/Image editor**. Elegiremos la imagen, y tendremos una representación como muestra la figura 28.

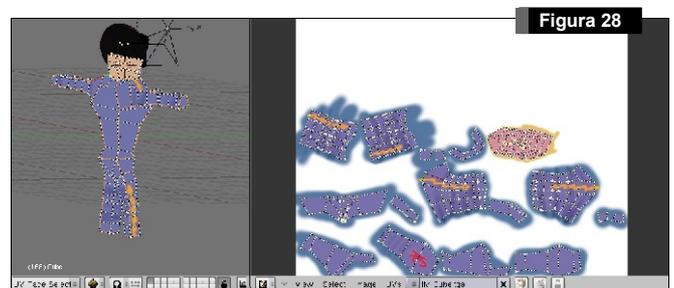


Figura 28



Figura 29

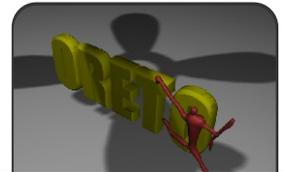
Hecho esto, si activamos el modo de sombreado **Textured** (figura 29) en la ventana 3D deberá aparecer el mapa UV aplicado. Finalmente, añadiremos una textura de tipo imagen y cargaremos la textura que hemos utilizado para el mapeado UV (esto es necesario si vamos a realizar el render final con Yafaray, que utilizaremos en próximas sesiones de prácticas). Tendremos que activar el botón **UV** en la pestaña **Map Input**.

Esta técnica de UV Mapping se utiliza mucho ya que permite indicar de una forma muy precisa cómo queremos situar la textura. En muchas ocasiones no es necesario utilizar un método de separación de la malla tan complejo, y se podrán asignar las coordenadas UV desplazando los vértices directamente sobre la ventana de **UV/Image Editor**.



PRÁCTICA Animación Básica

Carlos González Morcillo
(Carlos.Gonzalez@uclm.es)



29 Marzo 2006

Animación para la Comunicación · Escuela Superior de Informática · Universidad de Castilla-La Mancha

➤ El mecanismo básico de animación en Blender utiliza curvas de interpolación IPO. Este tipo de curvas, basadas en la representación de Bezier se emplean para animar las propiedades de los objetos.

Vamos a realizar una animación muy sencilla que maneje directamente las curvas IPO (InterPOLation Curves). Comenzaremos realizando un ventilador de techo que colocaremos en la parte superior de nuestra escena, que animaremos posteriormente. Añadimos un toroide **Add/ Surface/ NURBS Donut**. Utilizando la técnica de modelado que vimos en la sesión 2, modelamos una de las aspas (recordemos que añadimos una *curva de Bezier* **Add/ Curve/ Bezier Curve**, que convertimos a tipo Poly, ajustamos los puntos de control, cerramos **C** y luego de nuevo a *Bezier*). Debemos obtener una escena como se muestra en la figura 1.

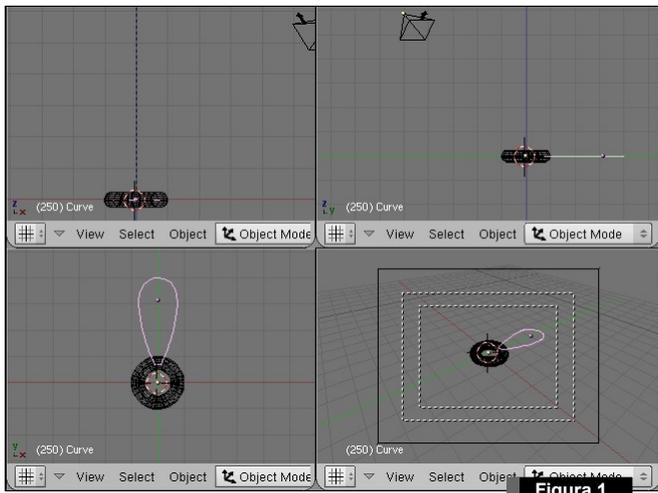


Figura 1

Queremos que nuestro ventilador tenga 4 aspas. Duplicaremos el objeto 3 veces, obteniendo 3 objetos clonados, que tendrán las mismas propiedades. Pero antes, vamos a cambiar el centro del aspa.

Todos los objetos en Blender tienen un sistema de referencia local. Podemos activar la visualización de este sistema de referencia local en los botones de objeto **7**, activando el botón **Axis** de la pestaña **Draw**. Con esto, el sistema de coordenadas local del objeto se dibuja como se muestra en la figura 2, con su origen dibujado como un pequeño círculo de color rosa. Se puede indicar también que se muestre otra información interesante, como el nombre del objeto (botón **Name**), o la caja límite (**Bounds**).

El Sistema de Referencia Local

Si aplicáramos una rotación al aspa, por defecto la realizará sobre el origen de su sistema de referencias local. Podemos cambiar este origen de coordenadas en los botones de edición **7**, seleccionando los siguientes botones (figura 3) de la pestaña **Curve and Surface** (si la malla es poligonal se realizará en la pestaña **Mesh** del mismo grupo de botones):

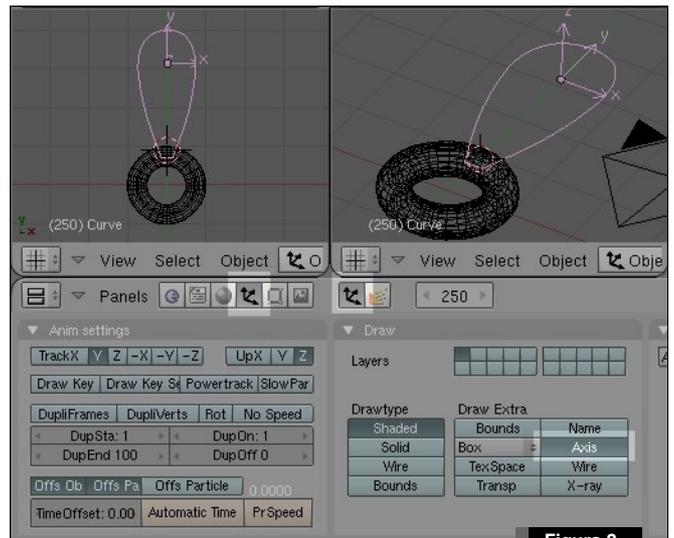


Figura 2

- **Centre:** Ajustamos la posición del objeto para que el centro del sistema de coordenadas local coincida con el centro geométrico del objeto. Esta operación no cambia la situación del origen de coordenadas, sino que produce un desplazamiento en la malla del objeto.
- **Centre New:** Sitúa el origen de coordenadas en el centro geométrico del objeto. Esta operación únicamente sitúa el nuevo origen de coordenadas.
- **Centre Cursor:** Sitúa el origen de coordenadas del objeto en la posición actual del puntero 3D. Mediante esta operación podemos situar el centro del objeto en cualquier punto del mundo.



Figura 3

El puntero 3D juega un papel fundamental en el posicionamiento de objetos y el centro de sus sistemas de coordenadas locales. Existen una serie de operaciones que nos permiten posicionar el puntero 3D y los objetos respecto de otros elementos de la escena. Para desplazar el puntero 3D al centro de un objeto seleccionado pulsaremos **Shift + S + 4** (también es accesible mediante **Spacer Add/ Transform/ Snap/ Cursor → Selection**). En caso de estar trabajando a otro nivel de detalle (por ejemplo, seleccionando un vértice), el puntero 3D se situará exactamente en esa posición. También es muy interesante modificar la posición de un objeto a donde se encuentra el puntero 3D; de forma similar, con el objeto seleccionado pulsamos **Shift + S + 2** (o **Spacer Add/ Transform/ Snap/ Selection → Cursor**). Mediante estas operaciones y las relativas al centro del objeto podemos situar cualquier objeto de la escena con precisión.

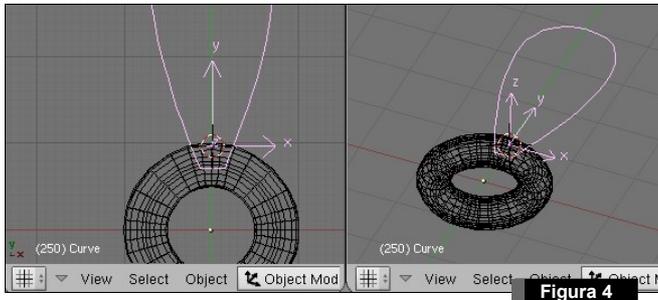


Figura 4

Situamos el nuevo origen de coordenadas del asa en el punto que se muestra en la figura 4. Ahora podemos rotar el objeto sobre este punto. Duplicamos el objeto para obtener clones (que comparten las propiedades) mediante **[Alt] D**. Modificamos la extrusión (**Extrude = 0.120**, **Bevel Depth = 0.05**) de una de las aspas, para obtener el resultado que muestra la figura 5. Como los 4 objetos son clones, basta con modificar los valores de extrusión de uno de ellos, y el resto lo obtendrán automáticamente.

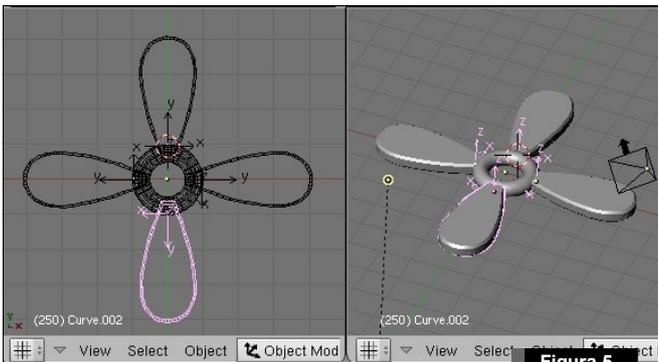


Figura 5

Relación de Parentesco

Tal como tenemos el objeto del ventilador, si lo quisiéramos desplazar por la escena, tendríamos que seleccionar cada uno de los objetos que lo componen y moverlos a la posición deseada. Si queremos simular el movimiento circular del mismo, tendríamos que rotar individualmente cada elemento que forma el objeto (o bien, combinar las mallas).

La mejor forma de realizarlo es mediante relaciones de parentesco. En una relación Padre-Hijo entre objetos, el objeto hijo hereda las propiedades del objeto padre, pero no a la inversa. Es decir, si desplazamos el objeto padre, el objeto hijo se desplazará también, pero si desplazamos el hijo, no le ocurrirá nada al objeto padre. Análogamente ocurrirá lo mismo si escalamos o rotamos el objeto.

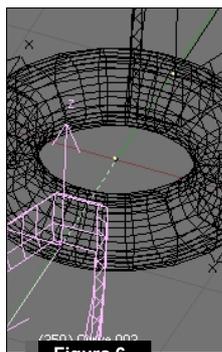


Figura 6

Para establecer una relación de parentesco Padre-Hijo, **primero** seleccionamos el objeto **hijo** y después, con **[Shift] G** pulsado seleccionamos el objeto **padre**, y pulsamos **[Ctrl] P**, y aceptamos la pregunta **Make Parent**. Así, hacemos que cada una de las aspas sean hijas del toroide. Blender representa las relaciones de parentesco mediante una línea punteada que une los centros de coordenadas del objeto hijo y el padre (ver figura 6). Si nos equivocamos al establecer una relación de parentesco, podemos eliminarla seleccionando el objeto hijo y pulsando **[Alt] P**.

Hecho esto, si rotamos o desplazamos el donut, se aplicará la misma operación a las 4 aspas, pero si aplicamos una transformación a las aspas, no se aplicará al toroide.

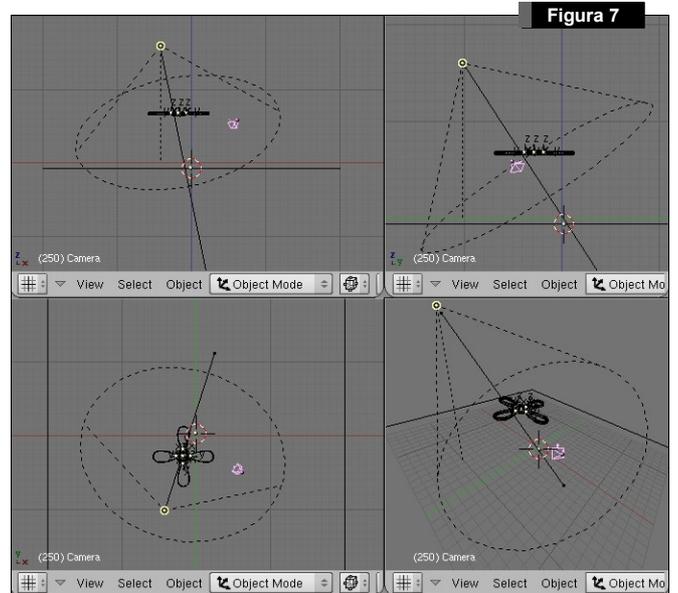


Figura 7

Añadimos una lámpara de tipo **Spot** a la escena, y un plano que sirva de suelo a la escena. Situamos el ventilador en la parte superior de la escena sin que se vea desde la cámara (ver figura 7), de forma que, si renderizamos la vista, obtengamos una sombra sobre el suelo, como muestra la figura 8.

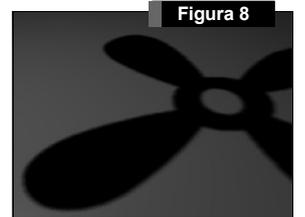


Figura 8

Curvas de Animación IPO

Vamos a realizar una animación cíclica de rotación en el ventilador. Las animaciones se descomponen en imágenes estáticas (en cine denominadas Fotogramas) o **Frames**. Blender dispone de una barra de animación situada en la cabecera de la ventana de botones (ver figura 9). El valor de ese cuadro de texto contiene el frame actual.



Figura 9

Podemos desplazarnos por los frames de la escena con los cursores, o bien introduciendo un valor numérico (**[Shift] G**). Podemos avanzar y retroceder 1 frame con los cursores **[Left]** o **[Right]**, o avanzar y retroceder 10 frames con **[Left]** o **[Right]**.

Cambiamos una ventana 3D al tipo **IPO Curve Editor**, y tendremos una representación como la mostrada en la figura 10.

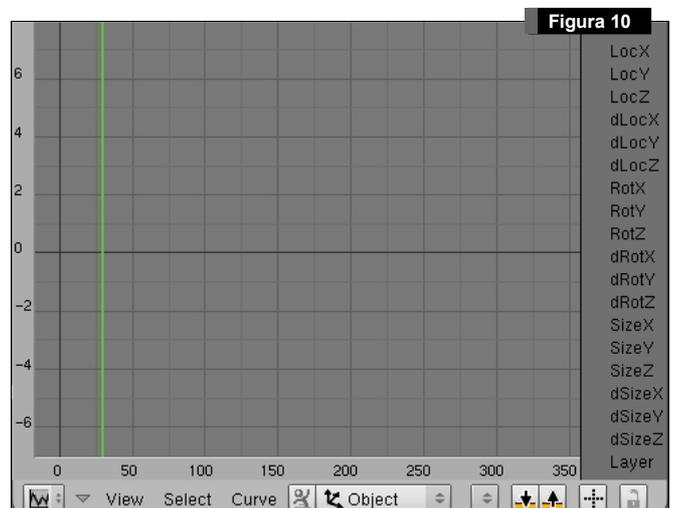


Figura 10

En el eje de abscisas se representan los frames, y en el de ordenadas los valores que pueden tomar los objetos respecto de las magnitudes indicadas en la columna de la derecha (Rotación, Localización, Materiales, etc...). La línea verde vertical (que podemos arrastrar para ver la evolución de nuestra animación), representa el frame actual. En la parte central de la cabecera de esta ventana aparece por defecto el elemento **Object** (ver figura 10). Existen multitud de propiedades animables en cada entidad de la escena; podemos cambiar de un tipo de entidades a otros seleccionando diferentes modos de trabajo en esta lista desplegable. En esta sesión de animación trabajaremos con algunas de ellas, aunque prácticamente cualquier propiedad puede ser animada en Blender.

Nos situamos en el primer frame de la animación, y con el donut seleccionado, definiremos en el frame 1 un punto clave de rotación. Para ello, con el puntero del ratón situado en una ventana 3D, pulsamos **U**. Aparecerá un menú preguntándonos qué atributos queremos manejar en nuestro frame clave. Le indicamos que sólo vamos a utilizar la rotación (**Rot**).



Importante: Dependiendo de en qué ventana tengamos el puntero del ratón cuando insertemos un frame clave mediante la tecla **U**, podremos manejar una serie de propiedades u otras. Si pulsamos **U** con el puntero del ratón sobre la ventana de materiales, trabajaremos con este tipo de propiedades, si estamos en una ventana 3D, manejaremos rotaciones, traslaciones, etc...

En la ventana de curvas IPO habrá aparecido una recta de color azul, con las entidades **RotX**, **RotY**, **RotZ** iluminadas en la zona derecha. Avanzamos hasta el frame 50, rotamos el ventilador respecto del eje Z y definimos otra clave de rotación **U**. Podemos visualizar la animación en una ventana 3D mediante **Alt+A** (Pulsamos **Esc** si queremos parar la animación). Si queremos ver la animación en todas las ventanas utilizamos la combinación **Alt+Shift+A**.

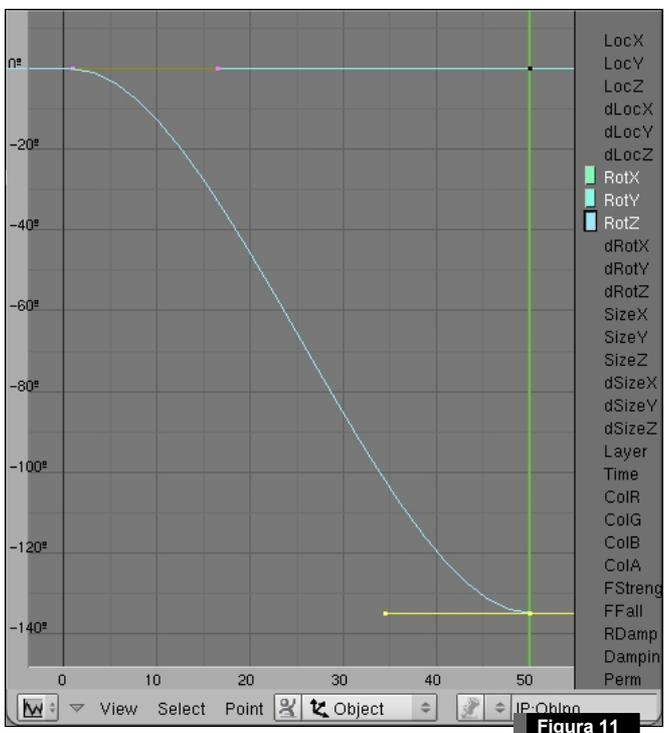


Figura 11

Dependiendo de cuánto hayamos rotado el objeto, tendremos una representación de la curva de animación parecida a la mostrada en la figura 11. Con la tecla **Home** hacemos que Blender ajuste la ventana a la mejor

representación posible. Queremos que la animación se pueda repetir en bucle, por lo que deberá rotar 360° y además mantener una velocidad constante. Para esto, editaremos las curvas de movimiento manualmente. Seleccionamos la curva **RotZ**, y pasamos a modo de edición de vértices **Tab**. Elegimos el punto de control de la derecha (el que está situado en el frame 50), y pulsamos la tecla **N** para darle un valor numérico exacto. Nos interesa que en el eje Y (**VertexY**) tenga un valor de -36 (correspondiente a 360°). Seleccionamos el punto de control del frame 1, y le damos un valor en el eje Y de 0.

Por defecto, las animaciones en blender tienen una duración de 250 frames. Esto puede cambiarse en los botones de Render **F0**, dentro de la pestaña **Anim**, el intervalo viene dado por las cajas **Sta: 1** y **End: 50**.

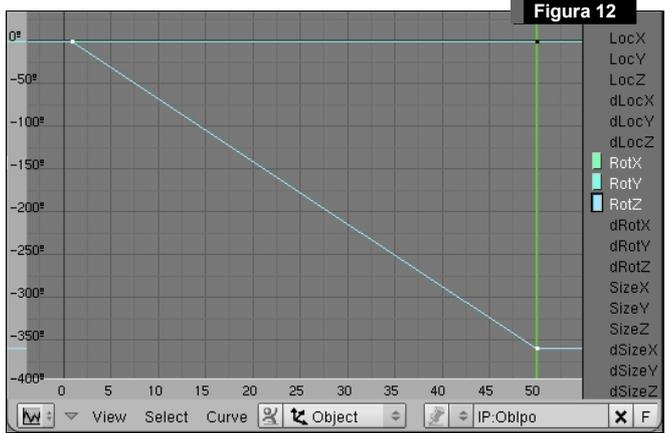


Figura 12

Con estos valores, visualizamos la animación con **Alt+A**. Podemos comprobar que el movimiento no es lineal, sino que tienen una parte de aceleración y frenado al inicio y al final del intervalo. No es el movimiento que buscamos, ya que no se repite perfectamente. Tenemos que ajustar los puntos de tangente en la curva IPO para que no haya zona de aceleración en el intervalo (ver figura 12).

Esta animación está definida dentro del intervalo 1:50. Si queremos que la animación se repita aunque trabajemos con intervalos mayores, podemos indicarlo seleccionando **Curve/ Extend Mode/ Cyclic** en la cabecera de la ventana de curvas IPO. Mediante esta opción, la animación se repetirá cíclicamente aunque estemos trabajando con intervalos de frames mayores.

Hecho esto, podemos generar nuestra primera animación. En los botones de render, configuramos las siguientes opciones como muestra la figura 13.

Indicamos el nombre del directorio donde vamos a guardar el render, el intervalo de frames en **Sta:** y **End:**, como formato podemos elegir **AVI Jpeg** (o, si estamos bajo sistema operativo *Microsoft Windows*, algún codec en particular) o secuencia de imágenes TGA (recomendable en secuencias que requieran alta calidad o postprocesado), el formato, por ejemplo, **320x240**. Una vez elegidas las opciones, pinchamos en el botón grande de **ANIM**.



Nota: Cuando generemos la animación en el formato final con el que trabajemos en postproducción, utilizaremos como formato de salida un conjunto de frames sin pérdida (como por ejemplo, secuencia de TGAs). Esto nos permite repetir de forma individual algún fotograma que salga con error y tener un mejor control sobre la animación final.

El botón de **PLAY** nos permite visualizar la animación después de haber sido generada (también podemos abrirla con cualquier reproductor de vídeo instalado en el sistema).



Figura 13

El reproductor integrado de blender permite avanzar y retroceder a nivel de frame (mediante los cursores) , o volver a reproducir el vídeo con **Enter**. Mediante **+** y **-** del teclado numérico podemos hacer zoom en el vídeo mientras se reproduce.

Para finalizar la práctica, vamos a importar objetos de otro fichero como realizamos en la sesión anterior, que también animaremos mediante frames clave. Para importar elementos que están guardados en otro fichero de blender, elegimos **File/ Append**. Elegimos el fichero *oreto.blend*, entramos dentro del nodo de Objetos (**Object**) con , y elegimos con los objetos **Logotipo** y **Texto**. Estos objetos se importarán en la capa donde se encuentren del fichero original (en este caso, en la segunda capa).

El sistema de Capas

Blender permite la gestión de los objetos de la escena en capas. Cada capa puede ser vista como un bloque contenedor que puede estar visible o no. Si está visible, todos los objetos que estén en esa capa formarán parte de la escena.

En todas las cabeceras 3D, blender permite gestionar las capas activas. Si el botón correspondiente a una capa está pulsado, la capa está activa (ver figura 14).



Figura 14

Podemos tener varias capas activas a la vez (hasta el máximo de 20 capas), seleccionándolas con **Shift** pulsado. Podemos mover un objeto de una capa a otra, seleccionando el objeto y pulsando **M**, blender nos pregunta la capa a la que queremos mover el objeto seleccionado.

Como los objetos del fichero anterior están situados en la capa 2, al menos tendremos que tener la capa 1 y 2 activas. El uso de capas se hace imprescindible cuando la complejidad de la escena aumenta, realizando la escena en diferentes capas y activándolas todas únicamente en el render final.

Situamos los objetos en la posición inicial que se muestra en la figura 15. Situamos en el frame 75 los objetos en la posición indicada en la figura anterior, e insertamos en cada uno (**Texto** y **Logotipo**) un frame clave respecto de la

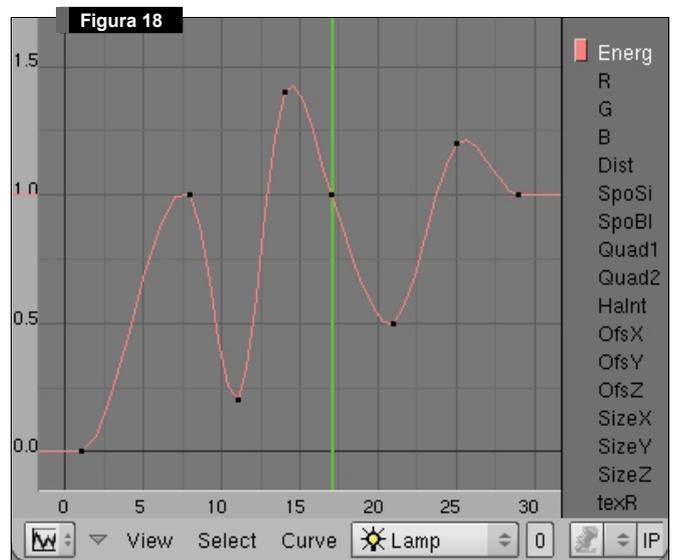


Figura 18

posición y la rotación (**LocRot**). Nos vamos al frame 50 y colocamos los objetos como muestra la figura 16, e insertamos de nuevo una clave de tipo **LocRot**.

Para conseguir la iluminación de la escena hemos añadido un nuevo foco de tipo Spot, que no arroja sombras (tiene desactivados los botones **Ray Shadow** y **Buf. Shadow**).

El efecto del parpadeo del foco al principio de la animación se ha conseguido añadiendo frames clave al foco respecto de su energía. Esto se realiza pulsando la tecla **I** con el puntero del ratón situado sobre la zona de propiedades del foco (ver figura 17). Aparecerá un menú donde elegiremos la propiedad **Energy** sobre la que añadiremos frames clave.



Figura 15

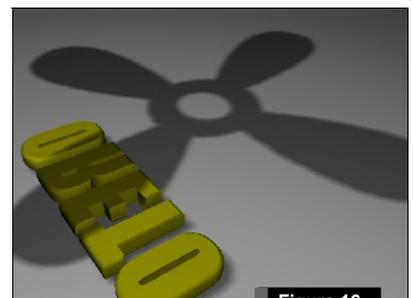


Figura 16

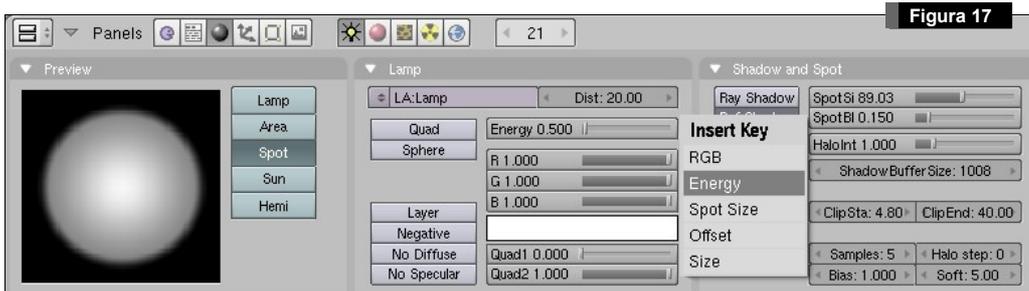
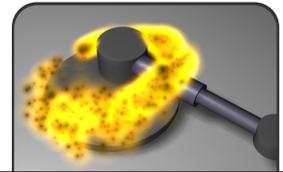


Figura 17

Podemos ver la curva IPO asociada a la energía del foco en la figura 18. En este caso, sólo tenemos una entidad a representar, cuyo valor oscila entre 0 y 1.4.

Finalmente generamos la animación con el nuevo intervalo 1:100. El resultado debería ser similar al mostrado en el vídeo "resultado.avi".



➤ En esta sesión aprenderemos los conceptos básicos de animación jerárquica en Blender, y añadiremos un sistema de partículas con simulación de colisiones con el resto de elementos de la escena.

Comenzaremos modelando el objeto que se muestra en la figura 1. Todos los componentes del modelo deben ser independientes. Recordemos que para esto, debemos salir del modo de edición de vértices (Tab) cada vez que añadamos un objeto a la escena. Creamos la escena con 5 cilindros, rotando y escalando hasta conseguir la configuración mostrada, y un plano que servirá como "suelo" de la escena.

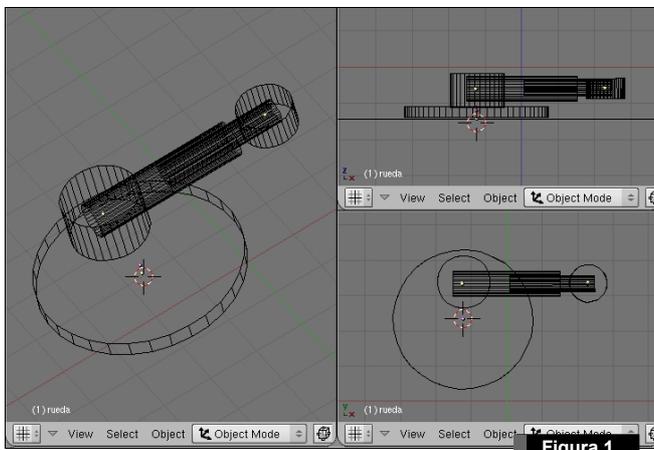


Figura 1

Renombraremos los objetos que forman la escena, con el fin de poder identificarlos más fácilmente, tal y como se muestra en la figura 2. Para ello, seleccionaremos en el modo de edición de objeto cada uno de ellos, y teclaremos el nombre en la sección **OB:** de los botones de edición (I), accesibles también con (F2).

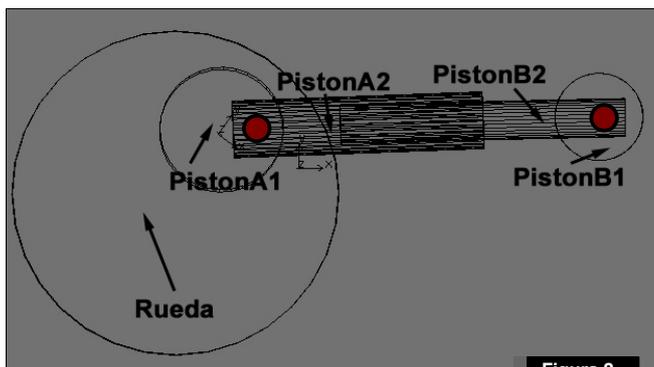


Figura 2

Podemos indicar a Blender que muestre el nombre de los objetos si activamos, con el objeto seleccionado, el botón **Name** del grupo de propiedades **Draw** del objeto (F7). Como vimos en la práctica anterior, también podemos activar la visualización de el sistema de coordenadas local de cada objeto, mediante el botón **Axis**.

Antes de animar los objetos de la escena, vamos a posicionar la cámara de forma cómoda, mediante restricciones de tipo "track".

Posicionamiento cómodo de la cámara

Añadimos un objeto **Empty** en el punto a donde queremos apuntar con la cámara (en nuestro caso, el centro de la escena formada por los objetos anteriores). Hecho esto, seleccionamos primero el objeto hijo (C), y después con (Shift+C) pulsado, seleccionamos el objeto padre (P). En este caso, queremos que la cámara (objeto hijo) apunte hacia donde está el Empty (objeto padre). Así, seleccionamos la cámara (C) (Shift+C) y seleccionamos el empty (P). Pulsamos (Ctrl+T) para añadir el track, de entre los tipos de track elegimos **TrackTo Constraint**. Si ahora movemos la cámara por la escena, siempre apuntará a la posición del Empty.

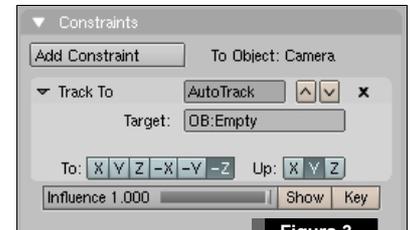


Figura 3

Para conseguir esto Blender ha creado una restricción de animación para el objeto cámara. Si vamos a los botones de Objeto (I), en la pestaña **Constraints** encontramos la restricción que se muestra en la figura 3. En el **Target** se ha indicado (automáticamente), el valor del objeto al que se quiere apuntar; en este caso el objeto cuyo nombre es Empty.

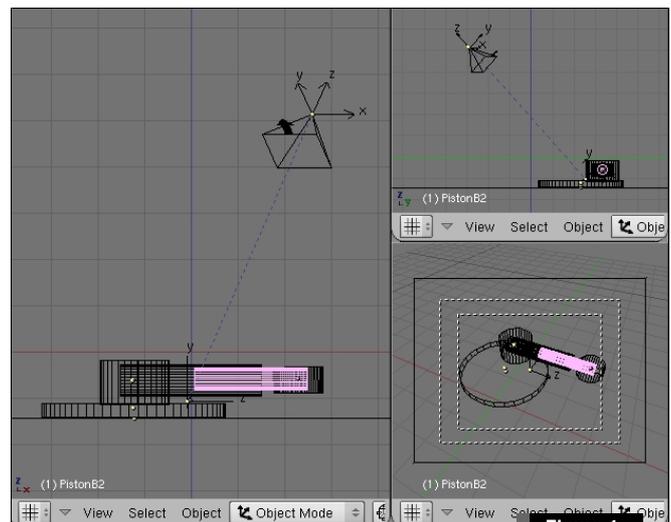


Figura 4

En **To:** se especifica la dirección, en el eje de coordenadas local del objeto al que aplicamos la restricción (la cámara), que vamos a utilizar para apuntar al objeto **Empty**. **Up:** por su parte indica el vector perpendicular a la dirección **To**, que apunta hacia arriba. Estos valores pueden verse en la figura 4. Si se cambia cualquiera de los vectores **To** y **Up**, podemos ver el efecto sobre la cámara de la escena. Finalmente, **Influence** controla la influencia de esta restricción sobre el objeto, desde 1 total a 0 ninguna.

Jerarquías y Restricciones

Establecemos las relaciones de jerarquía entre PistonA1 - PistonA2 y PistonB1 - PistonB2. Queremos que, cuando movamos las bases de los pistones (A1 y B1), los extremos tengan el mismo movimiento. Por tanto, en la jerarquía tendremos que PistonA2 es hijo de PistonA1. Para indicar esta jerarquía a Blender, seleccionamos primero el elemento hijo (PistonA2), y con **Shift + P** pulsado, seleccionamos el padre (PistonA1). Pulsamos **Ctrl + P** y elegimos **Make Parent**. Ahora, si rotamos PistonA1, el giro también lo sufrirá PistonA2, pero no al revés. Realizamos la misma operación con la otra parte del pistón (B1, B2). Si nos equivocamos, podemos eliminar la relación de parentesco seleccionando los elementos emparentados y pulsando **Alt + P**.

Cambiaremos los centros de los objetos PistonA2 y PistonB2 para que las rotaciones se hagan respecto del extremo (en vez de estar situado en el centro geométrico del objeto). Para ello, situaremos el puntero 3D, a la posición 3D donde queramos situar el nuevo centro y en los botones de edición **Ctrl + B** pincharemos en **Centre Cursor**. Colocaremos los centros en los puntos rojos que se muestran en la figura 2.

Pasamos a establecer una relación de dependencia entre objetos. Al igual que hemos hecho con la cámara, asociaremos el movimiento de los dos pistones, haciendo que se apunten mutuamente. Vamos a crear la restricción manualmente. Para ello, seleccionamos el **PistonA1**, y en los botones de Objeto añadimos una restricción pinchando en el botón **Add Constraint** de la pestaña **Constraints**. Añadimos la restricción de tipo **Track To**. En el campo Target tecleamos el nombre del pistón al que queremos que apunte (**PistonB1**). Activamos la representación del sistema de coordenadas del objeto **PistonA1** (botón **Axis** de la pestaña **Draw**). Con esta representación del sistema de coordenadas, decidimos qué eje tiene que ser el que

apunte al otro objeto, y cual será el vector Up. En el caso de nuestra escena, la configuración es la mostrada en la figura 5. Puede ser necesario ajustar la rotación del objeto PistonA2 después de haber añadido esta restricción, para que apunte correctamente y quede alineado.

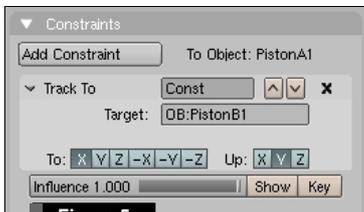


Figura 5



Figura 6

Realizamos la misma operación con PistonB1 respecto de PistonA1, para que ambos elementos se apunten mutuamente. Los valores necesarios para esta operación en la escena de ejemplo se muestran en la figura 6.

Probamos a mover PistonA1. Si tenemos que el movimiento lo siguen perfectamente el resto de piezas, hemos terminado este paso. En caso contrario, debemos revisar que las dos restricciones están creadas correctamente, con los valores de los ejes correctamente definidos. Activamos



Figura 7

"Rueda" gire, no se tendrá en cuenta esta rotación, sólo la posición). Añadimos un **Empty** en el centro geométrico de PistonA1, y hacemos padre al **Empty** de PistonA1. Hecho esto, hacemos hijo al **Empty** de Rueda. Si rotamos Rueda respecto del eje Z, el mecanismo general debería funcionar.

Pasamos a hacer una rotación de 360° de la rueda. Procederemos de forma similar a como se explicó en la práctica anterior con el ventilador. Insertamos dos frames clave respecto de la rotación en el frame 1, y en el frame 35. Seleccionamos la gráfica **RotZ** (queremos rotar el objeto sobre el eje Z del sistema de coordenadas global en nuestro ejemplo), y entramos en el modo de edición de vértices pulsando **Tab**.

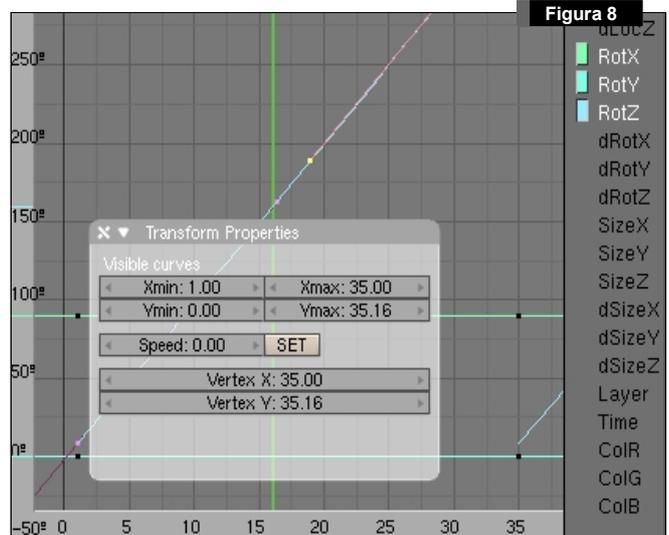


Figura 8

Seleccionamos el último clave y añadimos 36 al valor de **VertexY** de la gráfica **RotZ**, tendremos un giro de 360°. Podemos especificar un valor numérico si pulsamos la tecla **N** (tal y como muestra la figura 8). En el ejemplo de la figura, el objeto *rueda* partía de una rotación de -0.84 grados; así, el valor de rotación final es de 35.16.

Tal y como se explicó en la práctica anterior, ajustaremos la tensión de la curva IPO en los extremos para evitar la aceleración inicial y el frenado al final del intervalo, e indicamos que la repetición de la animación va a ser cíclica; en la cabecera 3D: **Curve/ Extend Mode/ Cyclic**.

Sistemas de Partículas

Añadimos un pequeño cilindro en la parte superior de PistonA1, y un plano encima de él, de un tamaño ligeramente inferior, tal y como muestra la figura 9. Hacemos hijo el plano y el nuevo cilindro de la rueda.

Vamos a utilizar el plano como un emisor de partículas. Seleccionamos el plano, y en los botones de Objeto **Object Mode**, dentro de los botones de simulaciones físicas **Physics**, vamos a la pestaña de **Particles** y pinchamos en **New**. Aparecerá una nueva pestaña titulada **Particle Motion**. Ajustamos los valores en ambas pestañas como se muestra en la figura 10. Veamos qué indican algunos de los parámetros de la figura anterior:

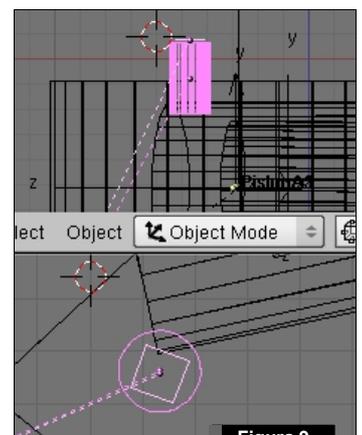


Figura 9

- **Amount:** Indica el número total de partículas que se van a lanzar. Estas partículas serán las que se crean uniformemente a lo largo del intervalo de frames.
- **Sta:** Inicio del intervalo de lanzamiento de partículas. En nuestro caso, a partir del frame número 40.

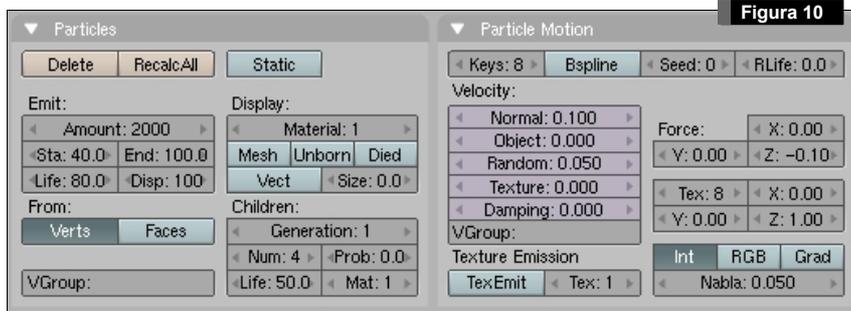


Figura 10

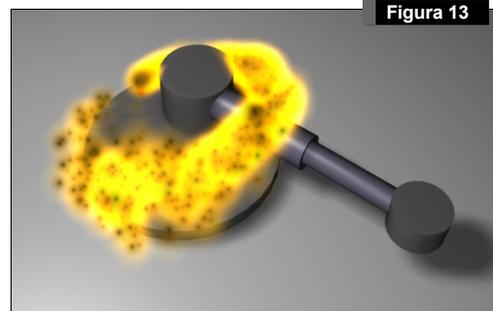


Figura 13

- **End:** Fin del intervalo. Después del frame 100 no lanzaremos más partículas.
- **Life:** Tiempo de vida de las partículas (medido en frames). Desde su lanzamiento, las partículas permanecerán activas 80 frames.
- **Disp:** Porcentaje de partículas que se mostrarán en la ventana 3D interactiva (aunque se renderizarán el 100%).
- **Normal:** Velocidad inicial con la que se lanzan las partículas. En nuestro caso, inicialmente 0.1.
- **Random:** La velocidad inicial de cada partícula será el resultado de sumar el campo **Norm**, con un valor aleatorio entre 0 y el valor indicado en **Random** (en nuestro caso, 0.05). Esto añade un poco de aleatoriedad a las partículas.
- **Force:** Simulación de fuerzas constantes (como viento, o gravedad). Indicando el valor -0.1 en el eje Z hacemos que las partículas caigan al suelo.

Puede ser necesario, si realizamos cambios sobre algunos parámetros, o la posición del plano que genera partículas, que se recalculen los valores de las partículas. Para esto, pulsaremos el botón **RecalcAll**.

Hecho esto, vamos a añadir un material a las partículas. Con el plano emisor de partículas seleccionado, añadimos un nuevo material y activamos las opciones que se muestran en la figura 12.

Elegimos un color naranja para las partículas. En la pestaña de Shaders, activamos el botón **Halo**, y aparecerán un conjunto de nuevas opciones que comentamos a continuación:

- **Halo Size:** El tamaño del halo.
- **Hard:** La dureza del halo, un valor alto da un resultado concentrado.
- **Add:** Si es 0, el halo es "sólido"; un valor mayor hará que el halo tenga un poder de emisión de luz.

Nota: Los sistemas de partículas pueden emitir cualquier tipo de elemento. Por ejemplo, podríamos cambiar el tipo de objetos que se emiten por cubos, metabolas o cabezas de mono!. Basta con hacer hijo de la cara emisora el elemento que queremos lanzar en el sistema de partículas y activar el botón **DupliVerts** de la pestaña **Anim settings** de los botones de **Objeto**. Cuidado, ya que en ese caso, estamos creando gran cantidad de objetos (uno por cada partícula). Será aconsejable disminuir el número de partículas a lanzar.

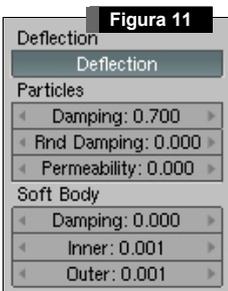


Figura 11

Como queremos que las partículas reboten sobre el plano del suelo, tenemos que definir cómo se va a realizar esta simulación. Para ello, seleccionamos el plano, y en los botones de simulaciones físicas, vamos a la pestaña **Fields and Deflection**, pinchamos en **Deflection** (para que se calcule el rebote de las partículas, basándose en colisiones físicas) configurando los valores como muestra la figura 11.

Si el botón **Flare** está activo (como es nuestro caso), aparecen nuevos valores numéricos:

- **FlareSize:** El factor por el que el efecto flare es mayor que el halo. Con valores grandes, aumenta el efecto de destello.
- **Boost:** Para dar una fuerza adicional al flare.

Es posible activar diferentes formas y propiedades al halo, que se especifican en la columna de botones de la derecha:

- **Rings:** Determina la forma en aros del halo.
- **Lines:** Dibuja líneas sobre el halo básico.
- **Star:** El halo básico se dibuja como una estrella.
- **HaloTex:** Si está activo, permite incorporar una textura al halo; por ejemplo, una imagen.
- **HaloPuno:** Si está activo, el tamaño del halo se determina por el vector normal de la superficie emisora.

Veamos la explicación de estos parámetros, dentro del grupo **Particles**:

- **Damping:** Dureza del rebote en la colisión. Varía entre 0 y 1. Cuanto más cercano a 1, menos rebote habrá.
- **RndDamping:** Factor aleatorio de rebote.
- **Permeability:** Si es mayor que cero, las partículas pueden atravesar el objeto.

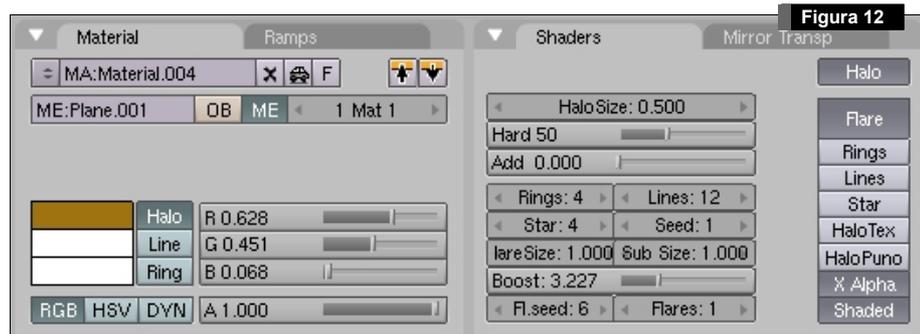


Figura 12

- **X Alpha:** Extreme Alpha; con esta opción se fuerza una progresión más fuerte en la transparencia y en el color del halo.
- **Shaded:** Si está activo, el halo puede recibir luz, dando una sensación de elemento con cuerpo 3D.

Finalmente renderizamos el proyecto, desde el frame 1 al 210. Añadimos dos focos, y algunos materiales con propiedades de color a los elementos que forman la escena.



Animación basada en Esqueletos

Animación para la Comunicación · Escuela Superior de Informática · Universidad de Castilla-La Mancha

➤ El uso de cinemática inversa resulta imprescindible cuando aumenta el grado de libertad del modelo a animar. En esta sesión veremos el uso de "armatures" y su asociación con las partes que componen el modelo.

Comenzaremos modelando las diferentes piezas que formarán nuestro brazo. Debemos obtener una geometría similar a la mostrada en la figura 1. Antes de ponernos manos a la obra, debemos tener en mente que al modelo se le deberá asociar un esqueleto. Resultará mucho más sencillo realizar esta operación (y el propio modelado), si en la construcción del objeto empleamos una postura "cómoda". Un ejemplo de construcción del modelo se muestra en la figura 2. Así, el esqueleto se situará de forma sencilla a lo largo del eje X, siguiendo la geometría del brazo.

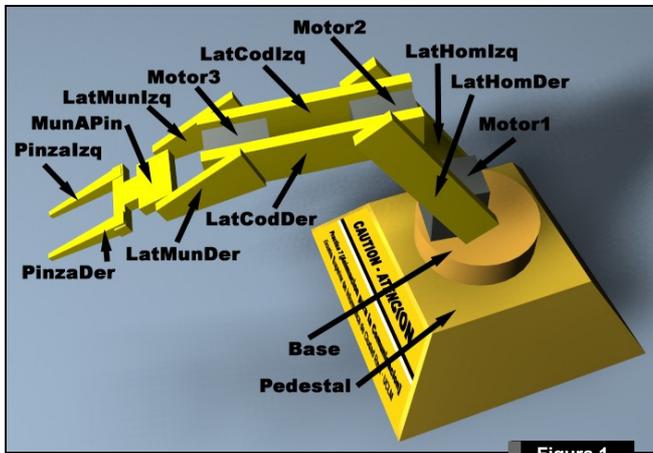


Figura 1

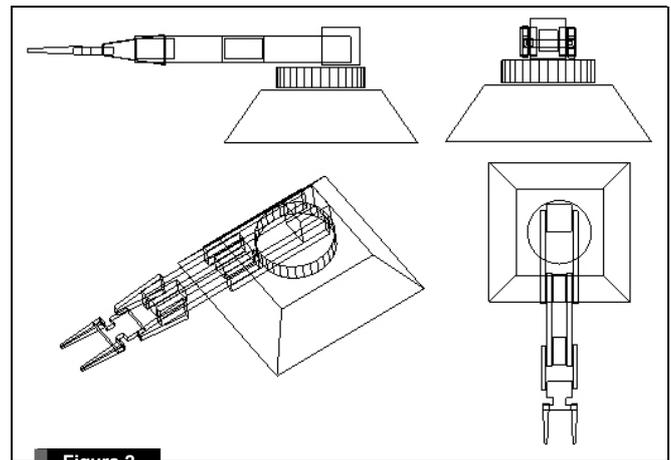


Figura 2

Creación del esqueleto

Una vez modelado el robot, y renombradas las piezas, añadiremos el esqueleto. Para ello, pulsaremos **ADD/Armature**. Construiremos un esqueleto formado por 4 huesos, centrado en el interior del brazo robótico, tal y como se muestra en la figura 3. Los tres primeros están ajustados a las articulaciones del brazo del robot (hombro, codo y muñeca). La última, muy pequeña, se utilizará para gestionar la cinemática inversa del modelo.

Recordemos que se pueden duplicar objetos con **Shift + D**. Será útil definir todos los componentes del robot como mallas independientes. Para ello, deberemos salirnos del modo de edición de vértices cada vez que añadamos una parte del robot a la escena, pulsando **Tab**. Nombraremos cada componente del robot como se muestra en la figura 1. Será importante tener claros los nombres asignados a cada parte a la hora de gestionar la jerarquía y las asignaciones de parentesco a cada hueso del esqueleto. Por esto, se recomienda seguir el mismo convenio de nombrado que se utiliza en este documento.

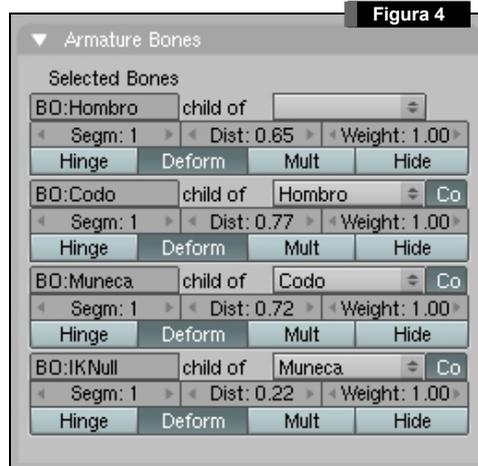


Figura 4

Para añadir más huesos al modelo usaremos la tecla de extrusión **E**. Una vez creado el esqueleto, entramos en el modo de edición del objeto (del esqueleto) con **Tab**. Seleccionamos todos los huesos del modelo con **A**, cambiando su color a amarillo. Vamos a los botones de edición **E** y cambiamos el nombre de los huesos como se muestra en la figura 4.

Al igual que antes, se recomienda seguir el mismo convenio empleado en este documento en el nombrado de los huesos. Pasamos a definir las jerarquías entre elementos, y a conectar las diferentes partes del robot con el esqueleto que hemos creado.

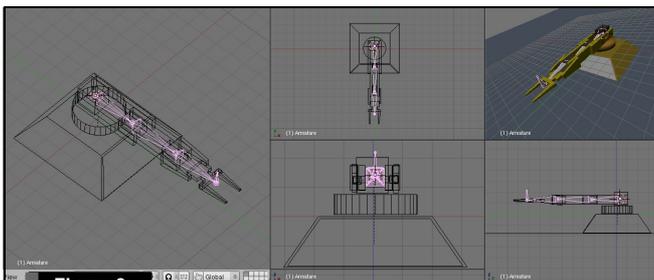


Figura 3

El botón **Hide** de la parte inferior del grupo de botones de cada hueso permite ocultar la representación del hueso en el esqueleto. Esto es útil cuando la escena es compleja (animación de personajes). Como puede verse en la figura 4, se establece la relación de parentesco entre huesos del esqueleto de forma automática **Child of**. Podemos editar esta jerarquía de forma manual. Esta posibilidad la utilizaremos en próximas sesiones.

los diferentes modos de representación del esqueleto: **Octahedron** (por defecto), **Stick**, **B-bone** y **Envelope** (que muestra el radio de acción de cada hueso).

En el caso de nuestro modelo (aunque en la realización particular de cada práctica puede variar), se tiene que limitar la rotación en el eje X y en el eje Y. Para ello, pulsamos los botones **Lock X Rot** y **Lock Y Rot** de la pestaña **Armature Bones** para los huesos **Muneca**, **Codo** y **Hombro** (figura 8).

Continuamos creando un nuevo objeto vacío, al que llamaremos *EmptyRotBase*.

Nos servirá para definir la rotación que se tiene que aplicar al objeto *Base* (recordemos que es padre de casi toda nuestra jerarquía), cuando movamos *EmptyIk*. De esta forma, el robot siempre apuntará al objeto *EmptyIk*. Situaremos el nuevo Empty, como se muestra en la figura 9, a la misma altura que la base (para que al hacer el **Track** entre ellos, la Base no se "incline" hacia ningún lado; únicamente rote respecto del eje Z). La idea es hacer que este nuevo Empty sirva de "sombra" al *EmptyIk* con el que controlamos el movimiento del brazo. Para conseguir este efecto de "sombra", añadiremos una restricción de copiar la localización del *EmptyIk*, como se muestra en la figura 10. La localización se copiará respecto del eje X e Y. La altura en el eje Z la mantendremos fija.

Hecho esto, añadiremos un seguimiento (*Track*) de la base respecto del Empty. Para ello, recordemos que habrá que seleccionar primero el objeto dependiente, y después el principal, pulsando **Ctrl U**, **Old Track** (creamos un track con el antiguo método de blender, más estable para animaciones basadas en esqueletos). Si hemos seguido los pasos correctamente, podemos mover el *EmptyIk* y el resto de la geometría del robot seguirá el movimiento.

Para animar el modelo, bastará con insertar frames clave que guarden la posición (**Loc**) del objeto *EmptyIk* con **Ctrl U**. La interpolación del movimiento de este objeto nos producirá (por cinemática inversa) el movimiento en el resto de articulaciones.

Si el brazo robótico dobla el codo de forma incorrecta (hacia abajo), se puede ayudar al motor de cinemática inversa rotando en modo pose levemente el hombro hacia arriba.

Restricciones y curvas IPO

Para terminar esta práctica, vamos a crear una restricción que va a estar activa sólo en un intervalo de frames. Con este comportamiento simularemos que el robot cambia un objeto de posición.

Según vemos en el vídeo resultado, el cilindro de color rojo tiene que tener una restricción de **Copy Location** activa sólo en el intervalo de frames donde la pinza sujeta al objeto. Crearemos un Empty auxiliar, situado entre las dos pinzas, que se llamará **EmptyObjeto** y que será hijo del objeto **MunAPin**. La restricción añadida al cilindro rojo de tipo **CopyLocation** (ver figura 11) estará activa desde el frame 40 hasta el 120. El EmptyObjeto será el que servirá como punto de ancla con la pinza.

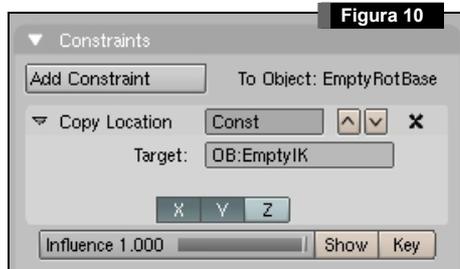


Figura 10



Figura 11

Para activar la restricción sólo en un intervalo de frames, pincharemos en el botón **Show** que está situado a la derecha de Influence (figura 11). Con esto, podemos editar la Influencia como una curva IPO. Abrimos una ventana IPO, y ponemos el tipo a **Constraint**. Seleccionamos **Inf** en la parte superior derecha de la ventana y podemos añadir puntos de control a la curva pinchando en el botón **Key** (Figura 11). Establecemos los valores para que el valor Inf de la curva IPO cambien entre 0 y 1 en el intervalo de frames elegido (ver figura 12). Recordemos que con la tecla **N**, en modo de edición de la curva, podemos dar un valor numérico a cada punto de la curva IPO.

Mientras el objeto está controlado con la restricción **CopyLocation**, cambiará su posición a la que tenga el objeto **EmptyObjeto**. Sin embargo, cuando pasemos del frame 120, el objeto volverá a su posición original. Para evitar esto, tendremos que cambiar la posición del objeto mientras estamos en ese intervalo de frames, de tal forma que, cuando acabe la restricción **CopyLocation**, el objeto esté en la posición nueva. Esto lo conseguimos añadiendo frames clave con la nueva posición, como se muestra en la figura 13.

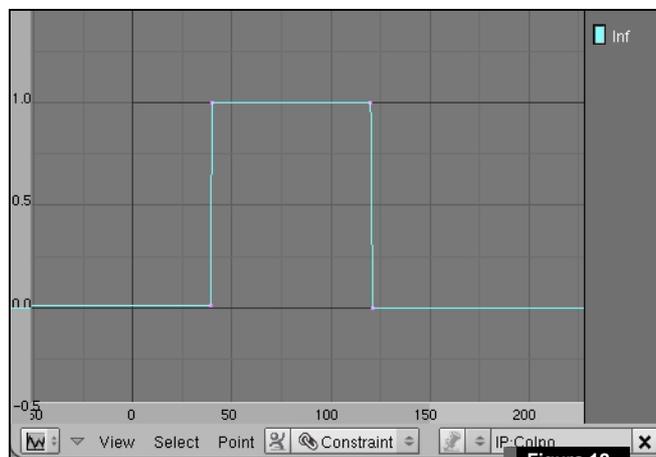


Figura 12

Recordemos que es muy útil para conseguir posiciones exactas utilizar el cursor 3D para alinear objetos; con **Shift S** podemos situar el cursor 3D en el centro de un objeto, y un objeto en la posición del puntero 3D.

Para terminar, generamos la animación añadiendo materiales con propiedad de RayMirror y renderizamos empleando el Trazador de Rayos integrado de Blender.

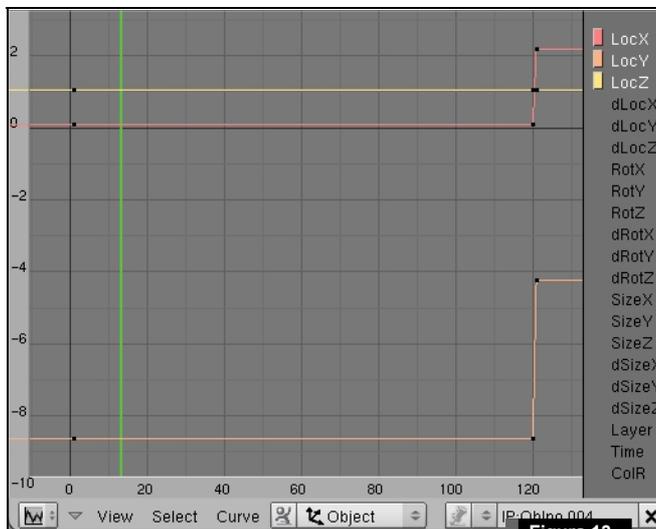


Figura 13



➤ Al personaje que modelamos en la práctica 4 vamos a añadirle un esqueleto interno para poder realizar poses y animarlo en la siguiente sesión empleando cinemática inversa y animación no lineal (NLA).

Las articulaciones del esqueleto que vamos a definir generarán unas deformaciones que habrá que trasladar a la malla del modelo 3D. Por último se aplicará un movimiento al esqueleto, construyendo una secuencia de animación.

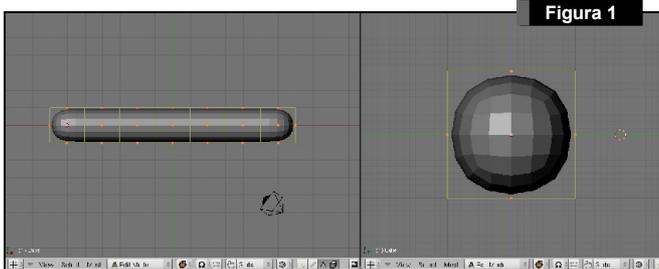
Así pues, en esta sesión identificamos un primer objetivo; construir un esqueleto para nuestro personaje, que asociaremos a una malla. Indicaremos a Blender qué vértices de la malla tendrá que deformar con cada hueso del esqueleto.

En la próxima sesión, construiremos una secuencia de animación para el esqueleto. Podremos realizar esta secuencia mediante la técnica de animación lineal empleada en sesiones anteriores (brazo robótico); definiendo la posición de todos los puntos finales (IK Solver) en cada frame clave, o bien definiendo acciones en el módulo de Animación No Lineal. Dependiendo del nivel de exactitud que necesitemos en el resultado utilizaremos una técnica u otra.

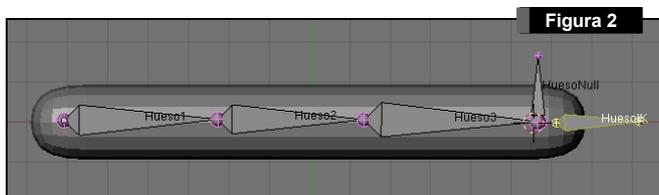
Un pequeño ejemplo antes de comenzar...

Antes de definir el esqueleto del modelo creado en la práctica anterior, veamos cómo asociar un Armature a una malla de polígonos, y el convenio de nombrado que utiliza Blender.

Empecemos con un ejemplo sencillo; insertamos un cubo a la escena. Extruimos la cara lateral y activando las superficies de subdivisión, construimos una malla como la mostrada en la figura 1.



Añadimos un Armature formado por 3 huesos principales (hueso1, hueso2, hueso3) y un cuarto hueso pequeño que utilizaremos para resolver la cinemática inversa (huesonull). Con el esqueleto todavía en modo de edición de huesos, añadiremos un quinto hueso (pulsando la barra espaciadora, Add / Bone), independiente de los anteriores,



aunque forma parte del esqueleto, que utilizaremos para calcular la cinemática inversa (huesoik). Obtendremos una configuración similar a la figura 2.

Importante: Todos los huesos deben formar parte de un único objeto Armature. Por tanto, cuando añadamos nuevos huesos al esqueleto, lo haremos siempre desde el modo de edición del objeto.

Podemos activar la representación del nombre de cada hueso pinchando en el botón **Draw Name** de los Botones de Edición. También es útil, cuando el modelo está representado en sombreado (tecla **Z**), activar en el mismo menú **X-Ray** para ver el esqueleto a través de la malla.

Nombraremos los huesos (ver figura 3) como se explicó en la sesión 8, y añadiremos una restricción (en modo pose del esqueleto; **Ctrl+Tab**) de tipo **Ik-Solver** al **huesonull** con objetivo en **huesoik** (Objeto "Armature" y Hueso "Hueso-ik"). Ahora podemos desplazar (en modo pose) **huesoik**, y el resto del esqueleto se calculará de forma automática mediante cinemática inversa. Quitaremos el botón **UseTip** que aparece activo por defecto.

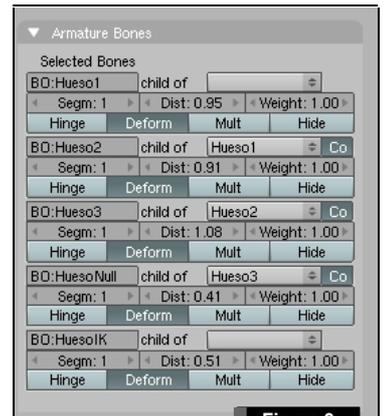


Figura 3

Falta por indicar a blender cómo deformar la malla. Para ello, crearemos grupos de vértices con el mismo nombre que los huesos del esqueleto. Asociaremos a cada conjunto de vértices un hueso de los anteriores. Entramos en modo de edición de vértices del objeto y seleccionamos el grupo de vértices que están situados sobre el **hueso1** (ver figura 4). Seleccionamos todos los vértices sobre los que tendrá influencia ese hueso.

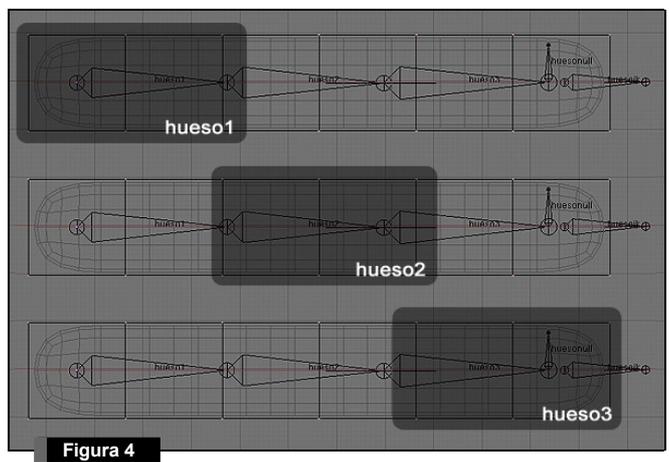


Figura 4

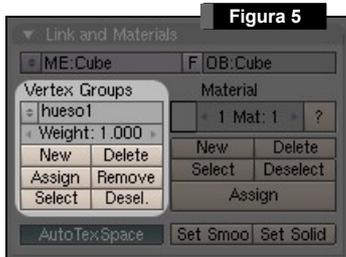


Figura 5

En los botones de edición  , crearemos un nuevo grupo de vértices (pinchamos en **new**). Tecleamos el nombre del primer hueso "**hueso1**" y, con los vértices asociados al primer hueso seleccionados, pinchamos en **Assign** (ver figura 5). De esta forma hemos creado el primer grupo de

vértices, asociados a **hueso1**. Si nos equivocamos al asignar vértices, podemos quitar los erróneos con **Remove**. El botón **Select** sirve para mostrar los vértices seleccionados de un grupo y **Deselect** para deseleccionar los vértices del grupo. Creamos un grupo para cada hueso y asignaremos los vértices que se muestran en la figura 4.

Si nos fijamos, hay vértices que pertenecen a dos grupos. Es buena práctica que los vértices situados en la zona de unión de dos huesos, pertenezcan a los dos grupos de vértices asociados. Blender calculará cómo deformarlos para que el resultado final en la malla sea suave.

Hecho esto, con el esqueleto en modo de edición de objeto, haremos la malla hija del esqueleto , **Make Parent to/ Armature/ Don't create groups**. Con esta última opción indicamos a blender que no genere automáticamente los grupos de vértices (ya que hay huesos que no deben utilizarse en las deformaciones de vértices). Por lo general, es mejor asignar manualmente los grupos de vértices.

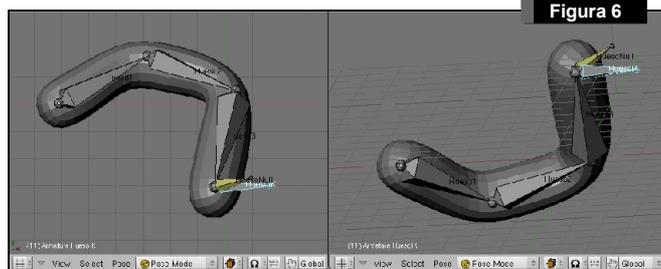


Figura 6

Entramos en modo de pose para el esqueleto y movemos el hueso **huesoik**. Si todo ha ido bien, se deberá aplicar la deformación en el modelo (ver figura 6).

Huesos para las piernas

Aplicaremos este método al archivo **P9modelo.blend**. Comenzaremos por el esqueleto de las piernas. Los huesos que tengan simétricos en el otro lado del cuerpo, como por ejemplo las piernas y los brazos, los nombraremos acabando en **.R** (los huesos de la derecha) y en **.L** (los de la izquierda). Esto nos servirá para "*copiar*" y "*pegar*" posiciones simétricas en la práctica de la siguiente sesión. De momento nos ocuparemos de mantener la notación exacta y consistente.

Añadimos como primer Armature el formado por **Fémur.L**, **Tibia.L** y **PiernaNull.L** (ver figura 7). Después, y recordemos que tenemos como objetivo tener un único Armature con todos los huesos, sin salir del modo de edición, añadimos una nueva cadena de huesos: **Pie.L** y **PieNull.L**. Finalmente, añadimos como cadenas independientes de un único hueso: **DedosPie.L**, **IkDedosPie.L**, **IkTobillo.L** e **IkPie.L**.

Hecho esto, realizaremos los siguientes parentescos entre huesos. Para ello, iremos a los botones de edición , y en modo de edición , elegimos el nombre del hueso padre en el campo **child of**; ver figura 8) los dos huesos IK pequeños; **IkDedosPie.L** e **IkTobillo.L** con el controlador principal del pie: **IkPie.L**. Al hueso **DedosPie.L** añadimos una restricción "*copy location*" hacia el hueso

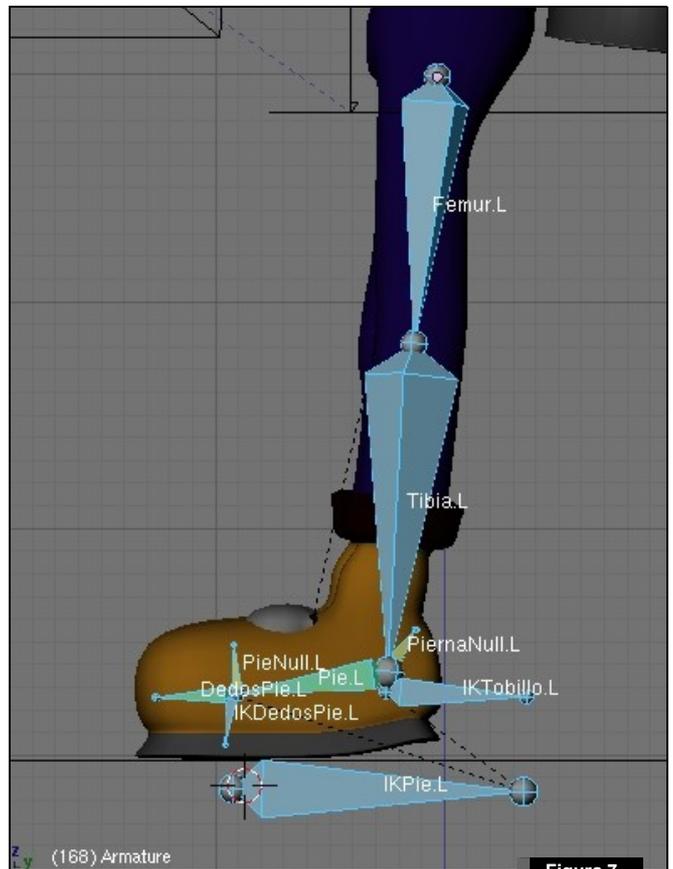


Figura 7

IkDedosPie.L. Al hueso **PieNull.L** un **IkSolver** hacia **IkDedosPie.L**. Al hueso **Pie.L** un "*copy location*" hacia **PiernaNull.L**. Finalmente a **PiernaNull.L** un **IkSolver** hacia **IkTobillo.L**. Si todo ha ido bien, tendremos un comportamiento como se ve en la figura 9; rotando **IkPie.L**, obtenemos una rotación general en la pierna, quedando el hueso **DedosPie.L** paralelo siempre al suelo.

Crearemos grupos de vértices como se explicó con el primer ejemplo de la práctica, nombrando los grupos igual que los huesos que los deformarán: **Fémur.L**, **Tibia.L**, **Pie.L** y **DedosPie.L**. Como hay dos objetos que tienen que usar el esqueleto (la bota y el cuerpo del personaje), habrá que crear grupos de vértices para cada objeto. En la figura 10 podemos ver los grupos de vértices creados para la bota. Además, habrá que emparentar la malla que sirve de "cordones" de la bota con el hueso **Pie.L** para que copie su rotación () **Make parent to/ Bone/ Pie.L**.

Al terminar, podemos emparentar las dos mallas (bota y cuerpo) con el esqueleto de la pierna. La deformación debe verse en este punto. A partir de ahora, todos los huesos que añadamos al esqueleto y grupos de vértices en la malla tendrán resultado directo en el modo de animación "pose" (ver figura 11). La figura 12 muestra la jerarquía final que tendremos que construir.

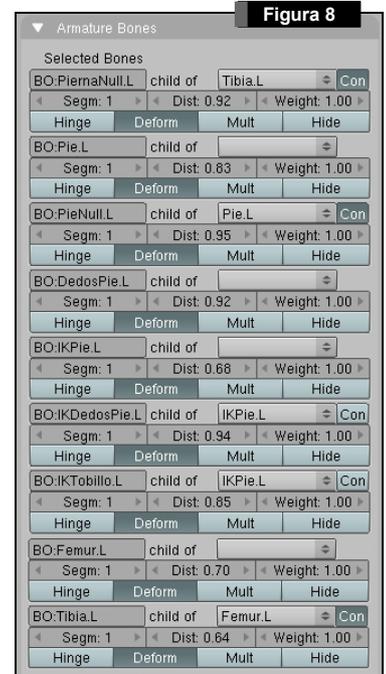
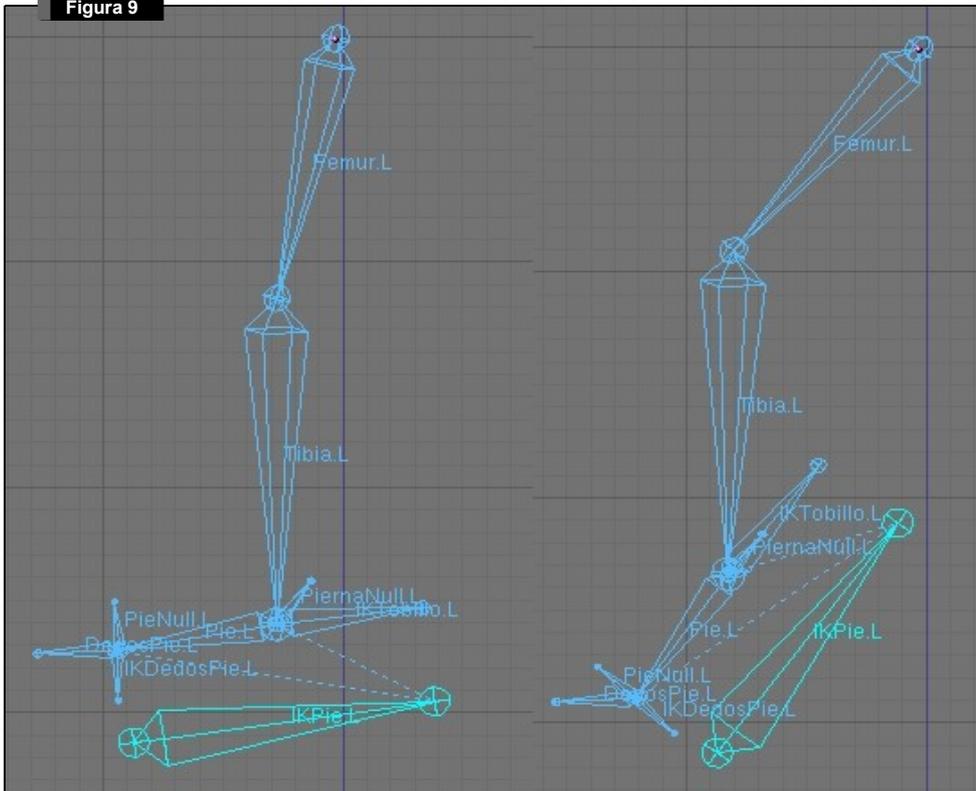


Figura 8

Figura 9



Como se muestra en la figura 12, añadiremos una cadena de huesos para formar la columna vertebral (**Estomago**, **Pecho** y **ColumnaNull**). Para limitar el inicio y el fin de la columna, insertamos 2 nuevos huesos (**IkCaderas** e **IkHombros**). Añadimos una restricción de tipo *CopyLocation* a **Estomago**, respecto de **IkCaderas**. A **ColumnaNull** un *IkSolver* sobre **IkHombros**.

Contruiremos los brazos como una cadena de 3 elementos: **Humero.R**, **Cubito.R** y **Muneca.R**. Añadiremos un *IkSolver* (**IkMuneca.R**) sobre **Muneca.R**. Haremos el hueso **Humero.R** hijo del hueso **Pecho** (eligiendo este último en la lista desplegable "child of" asociada a ese hueso, dentro de la pestaña **Armature Bones**). A continuación añadiremos un nuevo hueso llamado **Mano.R** que será hijo de **Cubito.R**, con el que podremos orientar la mano completa.

Añadiremos nuevas cadenas de huesos para los dedos. Índice, Anular y Menique tendrán 3 huesos (el dedo pulgar sólo 2) más un cuarto hueso (*null) que servirá como

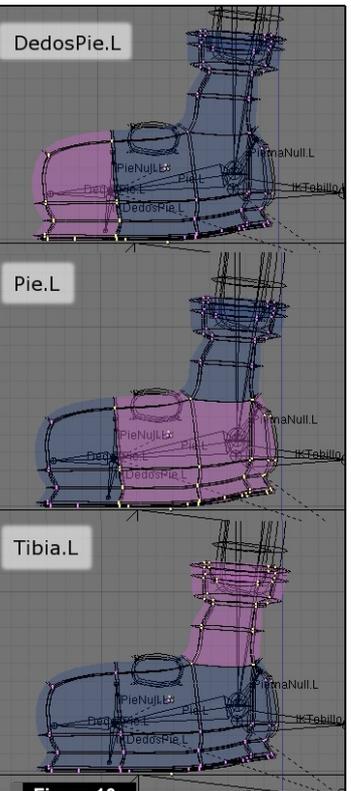


Figura 10

origen del *IkSolver*. Añadiremos un hueso independiente para cada dedo, que servirá como destino del *IkSolver*. Además, emparentaremos el origen de cada cadena de huesos y el hueso **IK** de cada dedo con **Mano.R** (de esta forma, cuando rotemos el hueso **Mano.R**, rotarán también todos los dedos). Una captura del esqueleto de esta parte puede verse en el lateral de la figura 12.

La cabeza estará formada por un único hueso llamado **Cabeza**, que tendrá dos hijos (**MandibulaSup** y **MandibulaInf**) que nos permitirán abrir y cerrar la boca. Los objetos que representan los dientes y la lengua serán hijos de estos huesos; la dentadura superior de **MandibulaSup** y la dentadura inferior junto con la lengua de **MandibulaInf**. El hueso **Cabeza** a su vez será hijo del hueso **Pecho**.

Finalmente podemos ocultar algunos huesos para que, cuando estemos en modo de edición de pose, sólo nos aparezcan los que vayamos a utilizar. Esto se consigue activando el botón **Hide** que aparece en color azul en el grupo de botones inferior al nombre del hueso.

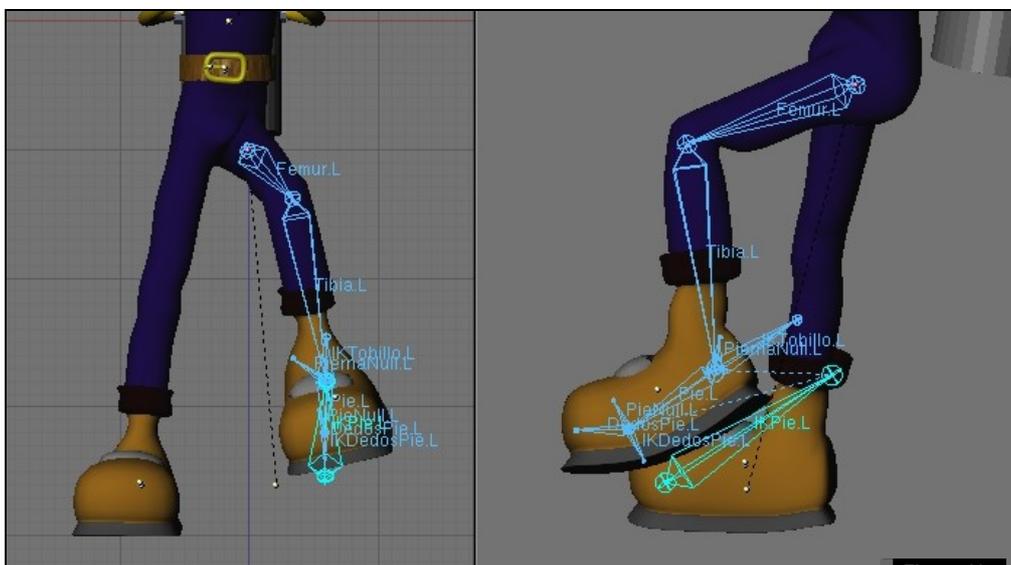
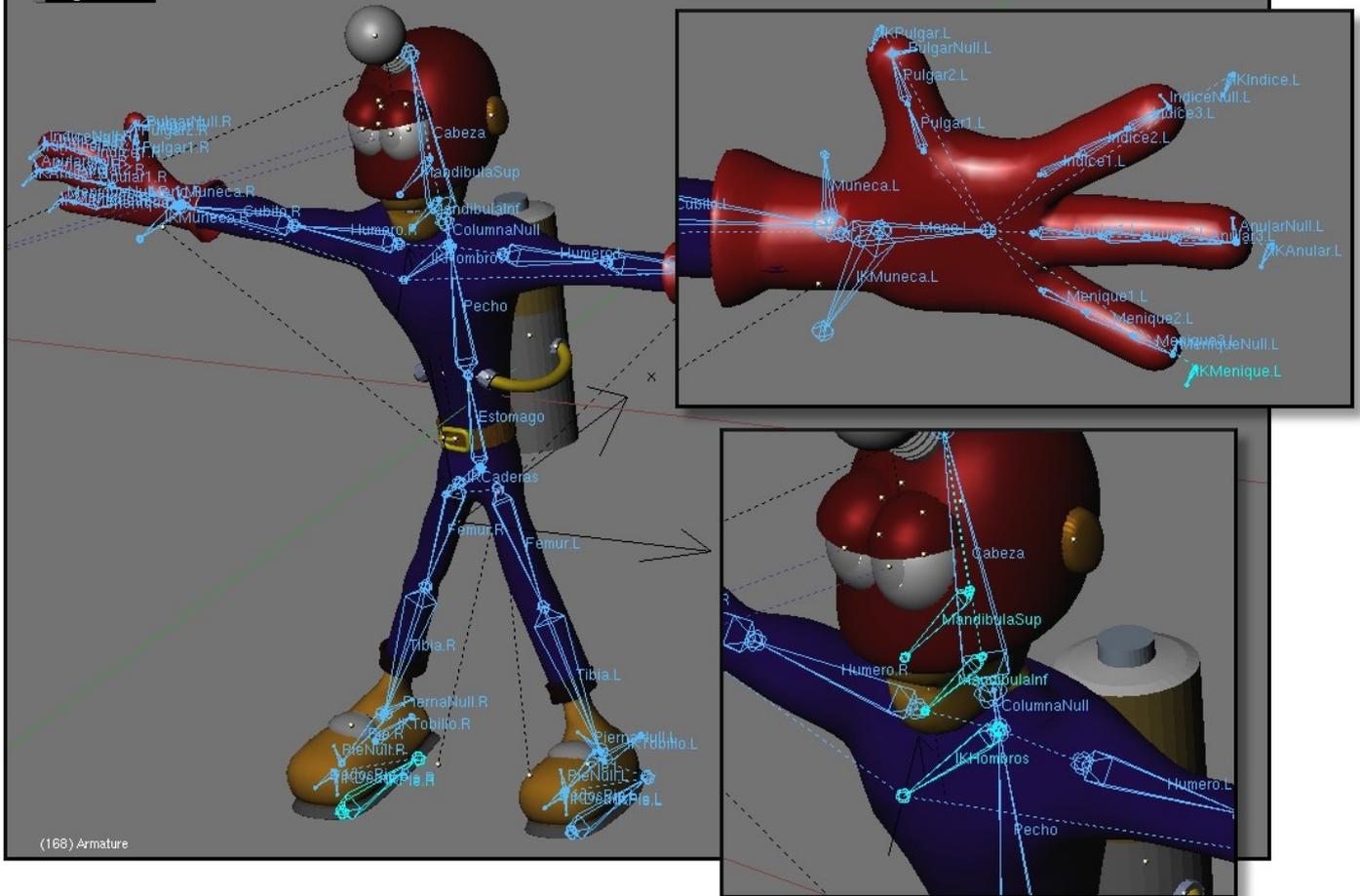


Figura 12

En la figura 13 aparece el conjunto de huesos que serían de utilidad a la hora de animar el personaje (básicamente los *IKSolver* de todas las cadenas de huesos, que serán los huesos a desplazar):

- IKPie.***,
 - IKDedosPie.***,
 - IKTobillo.***,
 - IkCaderas**,
 - IkHombros**,
 - IkCabeza**,
 - Mano.***,
 - IKPulgar.***,
 - IKIndice.***,
 - IKAnular.*** e **IKMenique.***,
- junto con algunos que nos permitirán realizar rotaciones de las extremidades (como los huesos **Femur** y **Humero**), además del hueso **Cabeza** y las mandíbulas.

Figura 12



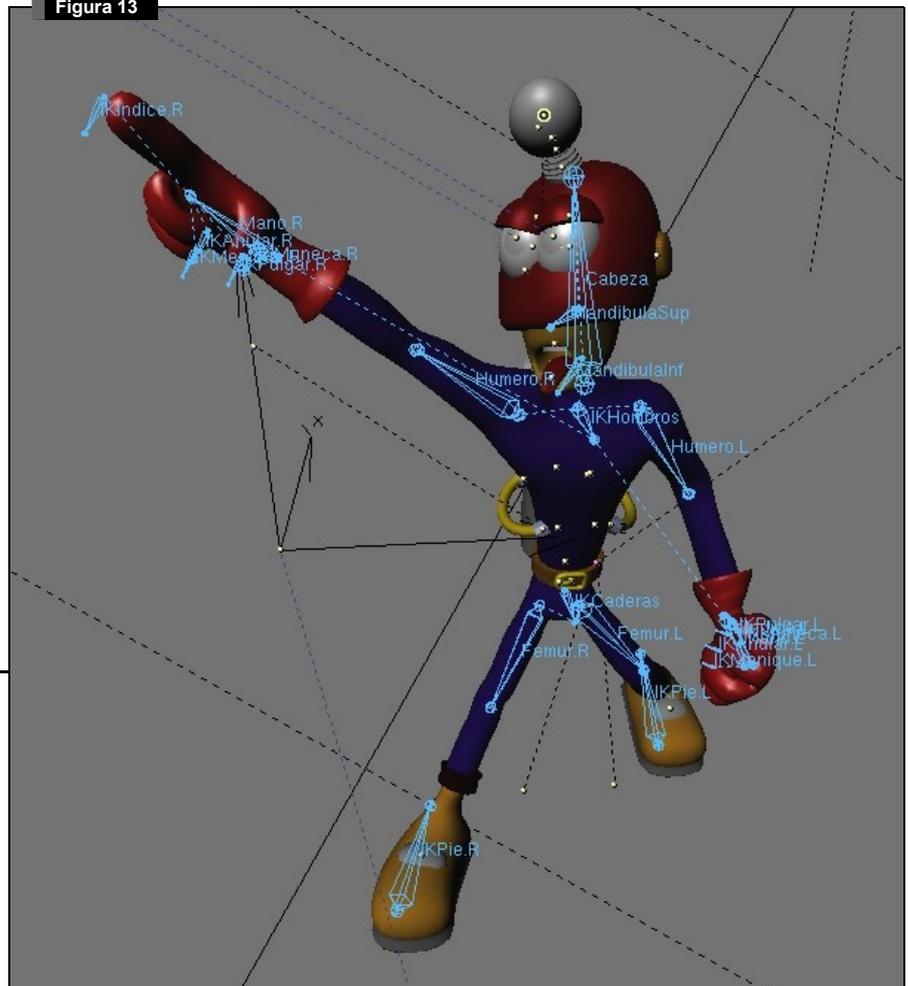
Puede resultar de utilidad tener el esqueleto en una capa distinta del resto de elementos. Esto nos permitirá ocultarlo cuando no estemos trabajando directamente con él. Después de ocultar un subconjunto de los huesos del esqueleto, obtenemos una configuración como la mostrada en la figura 13.

En el listado de la página 5 se representan, a modo de referencia, todos los huesos utilizados en la práctica con la jerarquía asociada entre ellos. Esta figura no representa las restricciones añadidas, aunque en general todos los huesos tipo ***Null** tendrán una restricción de tipo **IKSolver** sobre un hueso **IK*** del mismo nombre. El campo de la tabla "Conectar con padre" se activa en el interfaz de blender mediante el botón azul **Con** situado a la derecha de la lista desplegable **Child Of**.

Últimos detalles

Finalmente quedan por emparentar los elementos que forman el casco (todos hijos del hueso **Cabeza**), el cinturón (hijo del hueso **Estomago**), y los elementos que forman la pila (hijos de **Pecho**). Los elementos que forman la bombilla serán hijos del casco (al igual que la fuente de luz puntual situada en el interior).

Figura 13



Para facilitar la orientación de los ojos se ha añadido un empty situado suficientemente lejos de la cabeza, que será hijo del casco. Cada ojo tendrá una restricción de tipo TrackTo sobre este empty. Habrá que ajustar los ejes de To y Up para que el track sea correcto. De esta forma, cuando desplazemos el empty por la escena, los ojos apuntarán a esa posición.

En la siguiente sesión veremos cómo utilizar animación no lineal aplicada a este esqueleto, y poder reutilizar acciones previamente definidas.

| Visible? | Hueso | Hijo De | Conectar con Padre? |
|----------|---------------|------------|---------------------|
| | Pie.L | | |
| | PieNull.L | Pie.L | X |
| | DedosPie.L | | |
| X | IKPie.L | | |
| | IKDedosPie.L | IKPie.L | |
| | IKTobillo.L | IKPie.L | |
| | Pie.R | | |
| | PieNull.R | Pie.R | X |
| | DedosPie.R | | |
| X | IKPie.R | | |
| | IKDedosPie.R | IKPie.R | |
| | IKTobillo.R | IKPie.R | |
| | Estomago | | |
| | Pecho | Estomago | X |
| | ColumnaNull | Pecho | X |
| X | Humero.L | Pecho | |
| | Cubito.L | Humero.L | X |
| | Muneca.L | Cubito.L | X |
| X | Mano.L | Cubito.L | |
| | Pulgar1.L | Mano.L | |
| | Pulgar2.L | Pulgar1.L | X |
| | PulgarNull.L | Pulgar2.L | X |
| | Indice1.L | Mano.L | |
| | Indice2.L | Indice1.L | X |
| | Indice3.L | Indice2.L | X |
| | IndiceNull.L | Indice3.L | X |
| | Anular1.L | Mano.L | |
| | Anular2.L | Anular1.L | X |
| | Anular3.L | Anular2.L | X |
| | AnularNull.L | Anular3.L | X |
| | Menique1.L | Mano.L | |
| | Menique2.L | Menique1.L | X |
| | Menique3.L | Menique2.L | X |
| | MeniqueNull.L | Menique3.L | X |
| X | IKPulgar.L | Mano.L | |

| Visible? | Hueso | Hijo De | Conectar con Padre? |
|----------|---------------|------------|---------------------|
| X | IKIndice.L | Mano.L | |
| X | IKAnular.L | Mano.L | |
| X | IKMenique.L | Mano.L | |
| X | Humero.R | Pecho | |
| | Cubito.R | Humero.R | X |
| | Muneca.R | Cubito.R | X |
| X | Mano.R | Cubito.R | |
| | Pulgar1.R | Mano.R | |
| | Pulgar2.R | Pulgar1.R | X |
| | PulgarNull.R | Pulgar2.R | X |
| | Indice1.R | Mano.R | |
| | Indice2.R | Indice1.R | X |
| | Indice3.R | Indice2.R | X |
| | IndiceNull.R | Indice3.R | X |
| | Anular1.R | Mano.R | |
| | Anular2.R | Anular1.R | X |
| | Anular3.R | Anular2.R | X |
| | AnularNull.R | Anular3.R | X |
| | Menique1.R | Mano.R | |
| | Menique2.R | Menique1.R | X |
| | Menique3.R | Menique2.R | X |
| | MeniqueNull.R | Menique3.R | X |
| X | IKPulgar.R | Mano.R | |
| X | IKIndice.R | Mano.R | |
| X | IKAnular.R | Mano.R | |
| X | IKMenique.R | Mano.R | |
| X | Cabeza | Pecho | |
| X | MandibulaSup | Cabeza | |
| X | MandibulaInf | Cabeza | |
| X | IKCaderas | | |
| X | Femur.L | IKCaderas | |
| | Tibia.L | Femur.L | X |
| | PiernaNull.L | Tibia.L | X |
| X | Femur.R | IKCaderas | |
| | Tibia.R | Femur.R | X |
| | PiernaNull.R | Tibia.R | X |
| X | IKHombros | | |
| X | IKMuneca.L | IKHombros | |
| X | IKMuneca.R | IKHombros | |



Esqueletos Avanzados

Animación para la Comunicación · Escuela Superior de Informática · Universidad de Castilla-La Mancha

➤ Al personaje que modelamos en la práctica 4 vamos a añadirle un esqueleto interno para poder realizar poses y animarlo en la siguiente sesión empleando cinemática inversa y animación no lineal (NLA).



El motor de cálculo de cinemática inversa de blender en sus últimas versiones requiere que se le indique el número de huesos que forman cada cadena de IK, para que el cálculo sea correcto. Por ejemplo, en la figura 1 se muestra que la cadena donde *Muneca.L* es el hueso que tiene la restricción de **IK Solver** está formada por 2 huesos (*Humero.L* y *Cubito.L*). Por esa razón, el campo **ChainLen** (figura 2) tiene que indicar 2. De igual forma (figura 3), el hueso *PulgarNull.L* es el extremo de una cadena de 2

huesos (*Pulgar1.L* y *Pulgar2.L*). El hueso *IndiceNull.L* por su parte (figuras 4 y 5) es el extremo de una cadena de 3 huesos. El hueso *ColumnaNull* será el extremo de una cadena de 2 unidades, etc... En general, todos los huesos que tengan restricciones de tipo **IK Solver** tendrán que indicar el número de huesos de los que consta la cadena (sin contarse a sí mismos). Es imprescindible establecer el parámetro **ChainLen** de los huesos que lo requieran antes de comenzar ninguna animación de la práctica 11.

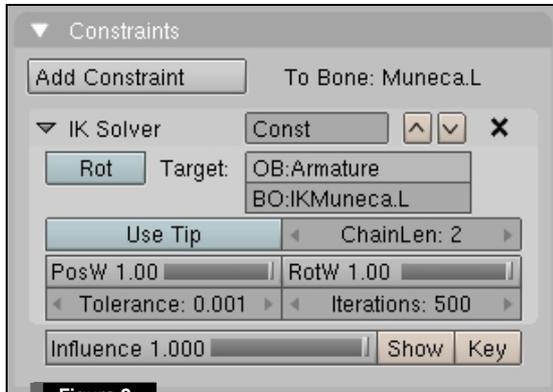


Figura 2

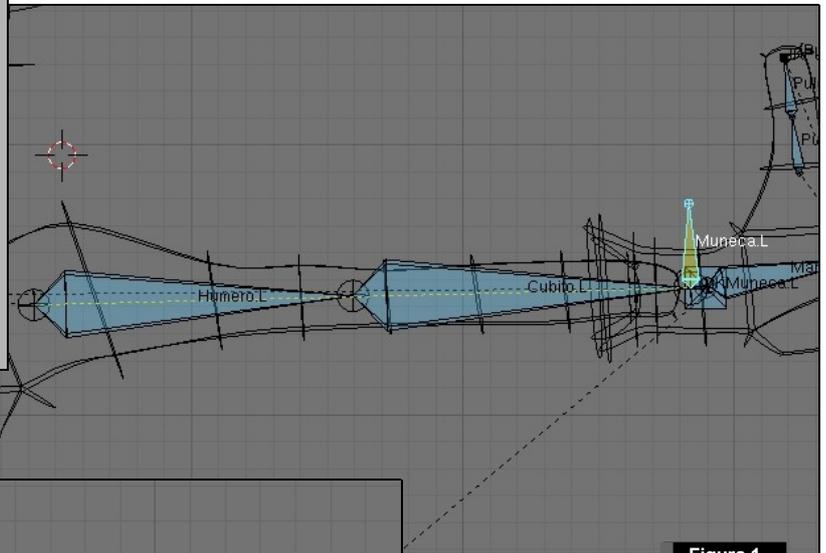


Figura 1

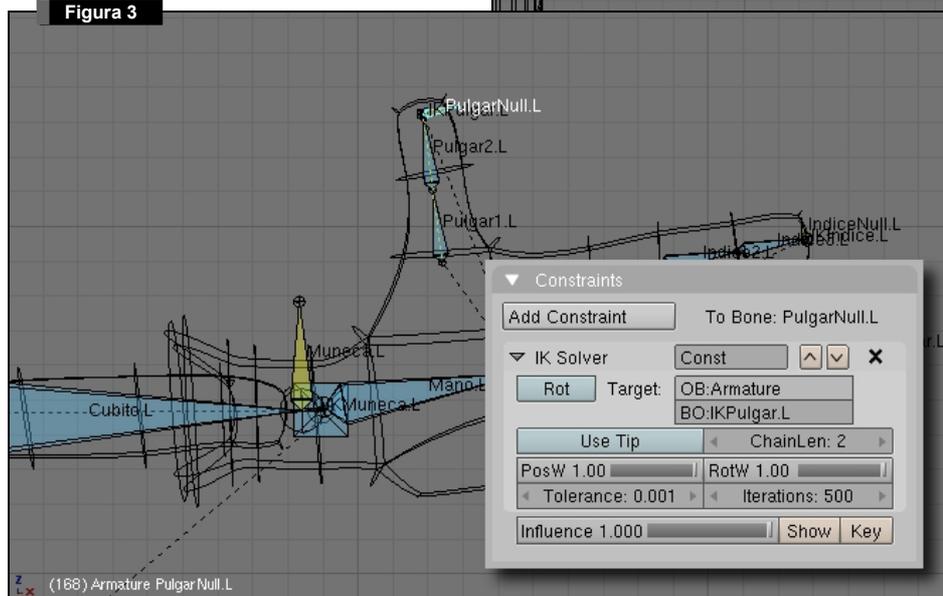


Figura 3

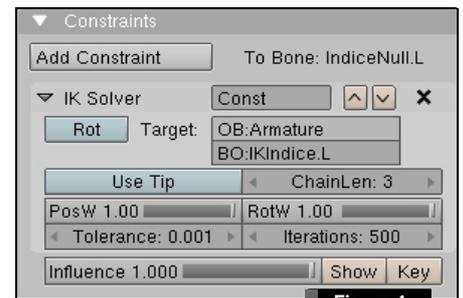


Figura 4

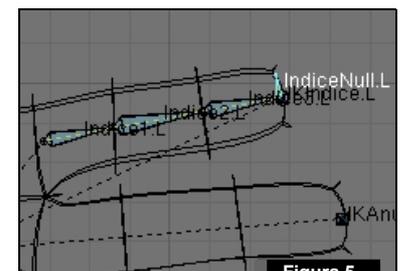


Figura 5



➤ En esta sesión utilizaremos diferentes técnicas de renderizado fotorrealista (como Iluminación Global de Yafray, Mapas HDRI y Raytracing con Ambient Occlusion integrado en Blender).

Antes de comenzar, prepararemos una escena sencilla con la que trabajaremos en las tres primeras partes de esta sesión. Vamos a utilizar un modelo escaneado en 3D del repositorio de modelos 3D de Stanford¹. Existen varios proyectos de escaneado de figuras 3D, como el Digital Michelangelo Project² que cuenta con multitud de estatuas en muy alta resolución.

El formato en el que se encuentran los modelos escaneados ha sido desarrollado en la propia universidad de Stanford; se utilizó un script para importar estos modelos en blender. Pese a utilizar un modelo con un alto número de polígonos del dragón (más de 45.000), en realidad es un modelo a baja resolución. Podemos encontrar versiones de mayor calidad en la web mencionada anteriormente. Cargamos el modelo con **File / Append**. Activamos el modo de sombreado suave en el modelo (**Set Smooth**) en los botones de edición.

En el modelado de la copa se utilizó una superficie de revolución (como vimos en sesiones anteriores). Empleamos una curva de bezier para hacer el contorno, que convertimos a polígono y generamos la superficie mediante el botón **Spin**. Es importante que el contorno sobre el que vamos a realizar la revolución modele el grosor del cristal y así la simulación de refracción será mucho más realista (ver figura 1).

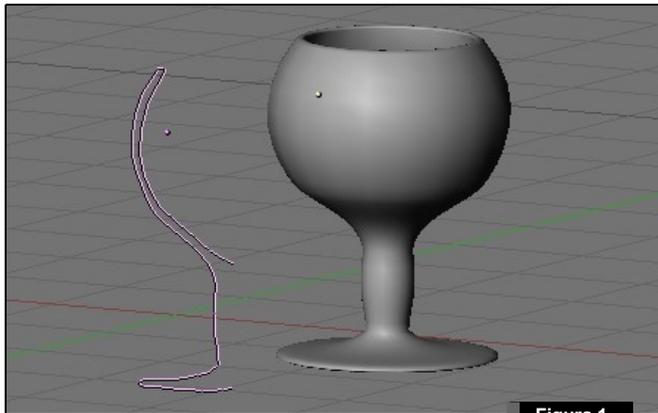


Figura 1

Para modelar la zona del licor, vamos a duplicar parte de la malla de la copa, y crearemos después el conjunto de caras superior para cerrar la malla. Primero seleccionamos (en modo de edición de vértices) el conjunto de vértices central del interior de la copa (ver figura 2).

Podemos aumentar o disminuir la selección a elementos vecinos con **Ctrl**+**+** y **Ctrl**+**-** (teclado numérico). De esta forma, con los vértices anteriores seleccionados, pulsaremos repetidas veces **Ctrl**+**+** hasta conseguir una selección similar a la mostrada en la figura 3.

1 <http://graphics.stanford.edu/data/3Dscanrep/>
2 <http://graphics.stanford.edu/projects/mich/>

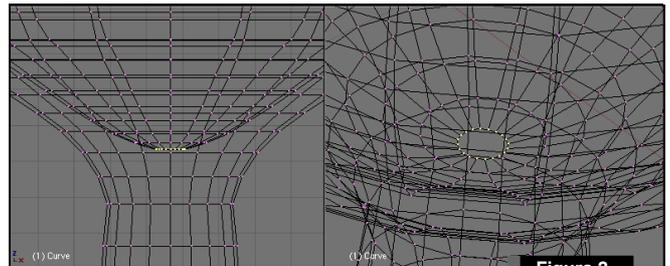


Figura 2

Duplicamos la malla **Shift**+**D** y sin moverla de la posición que ocupa, la separamos de la copa **P**. Con esto, debemos tener un objeto independiente, al que le falta la "tapa" superior (ver figura 4).

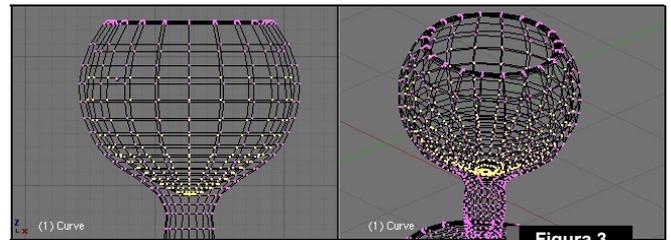


Figura 3

Para terminar el modelo, seleccionamos los vértices de la parte superior del nuevo objeto (desde la vista lateral, pulsamos la tecla **B** -una vez- y marcamos mediante una caja los vértices superiores). Hecho esto (sin salir del modo de edición de vértices), pulsamos **Spacebar** **Edit/ Face/ Fill**. Se habrán creado un conjunto de caras para cerrar la parte superior del nuevo objeto. Pasemos a dar materiales y texturas a los objetos de la escena.

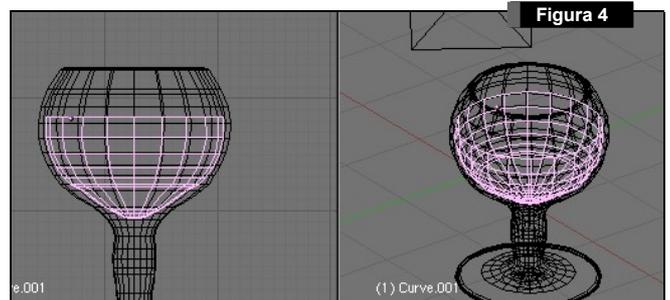


Figura 4

En la figura 5 se recogen las propiedades del material aplicado al dragón. Se ha activado el botón **RayMirror** de la pestaña **Mirror Transp**, para que refleje un poco el entorno (el nivel de la variable **RayMir** se ha dejado bajo). El parámetro **Depth** se ha establecido en 2, para que sea el máximo nivel de recursión que podrá alcanzar una reflexión en el motor de trazado de rayos. También podemos destacar que el color de brillo especular (**Spe**) y de Espejo (**Mir**) se han cambiado por distintos tonos azules, que simularán mejor el comportamiento de la porcelana. Este nivel de detalle nos permite crear materiales complejos.



Figura 5



Figura 6

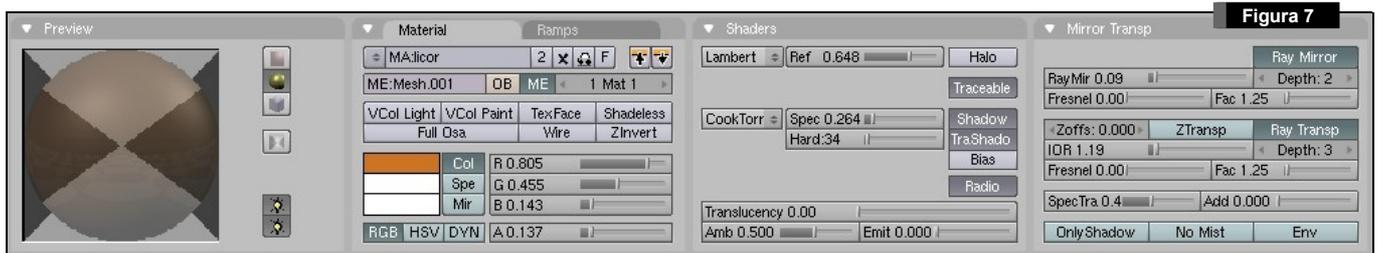


Figura 7

Recordemos que para cambiar estos colores, basta con pinchar sobre el cuadro de color (no es necesario hacerlo cambiando directamente las componentes RGB).

Respecto del material creado para el cristal y el licor, son básicamente iguales. Ambos materiales tienen transparencia, que viene dada por el parámetro **Alpha** (debajo de las componentes RGB). Además, tienen un poco de nivel de reflexión (**RayMirror** bajo). El campo **IOR** mide el Índice de Refracción (*Index Of Refraction*). Este índice es particular de cada tipo de material. Por ejemplo, el aire tiene un IOR de 1, el agua de 1.3 y el cristal de 1.5. Podemos comprobar cómo afecta esta propiedad al material en la pestaña **Preview**. Finalmente, es muy importante activar el botón **TraShadow** para que el material reciba sombras transparentes (y la refracción sea realista).

Pasemos a ver algunos métodos de render de la escena que hemos creado. Las diferencias de calidad vienen normalmente asociadas a diferencias en tiempo de render. Es un aspecto muy importante a tener en cuenta, sobre todo si queremos generar animaciones (en cuyo caso, tendríamos que estimar el tiempo de render que nos llevará procesar la secuencia). Una importante decisión será la elección del método de renderizado para que el tiempo de generación sea manejable. Comenzaremos trabajando con el motor de raytracing integrado en Blender, y el uso de Ambient Occlusion.

Parte 1: Ambient Occlusion

Si activamos el botón **Ray** de la pestaña de Render, indicamos a Blender que queremos utilizar raytracing para el cálculo de reflexiones y refracciones. Esto sólo tiene sentido si vamos a utilizar el motor de render integrado de Blender. Desde la versión 2.36, Blender tiene integrados dos motores de render; el interno de Blender y Yafaray. Podemos especificar el motor de render a utilizar en la lista desplegable situada debajo del botón **RENDER**.

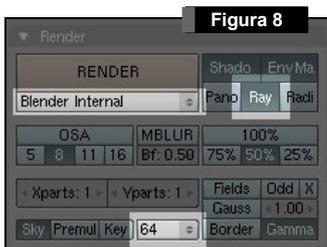


Figura 8

Cuando pinchamos en el botón **Ray** indicado anteriormente, aparece una nueva lista desplegable en la zona inferior de la pestaña de Render (ver figura 8). Esta lista permite indicar la resolución del árbol octal que utiliza el motor de raytracing para acelerar el cálculo de las intersecciones rayo/objeto. Una buena elección del tamaño de este parámetro ahorrará mucho tiempo de render. Un mayor tamaño requerirá más memoria, pero acelerará los cálculos en escenas complejas (con un alto número de polígonos). Un valor de 64 funcionará bien para escenas poco complejas. En nuestro caso hemos elegido 512 debido al alto número de polígonos de la escena.

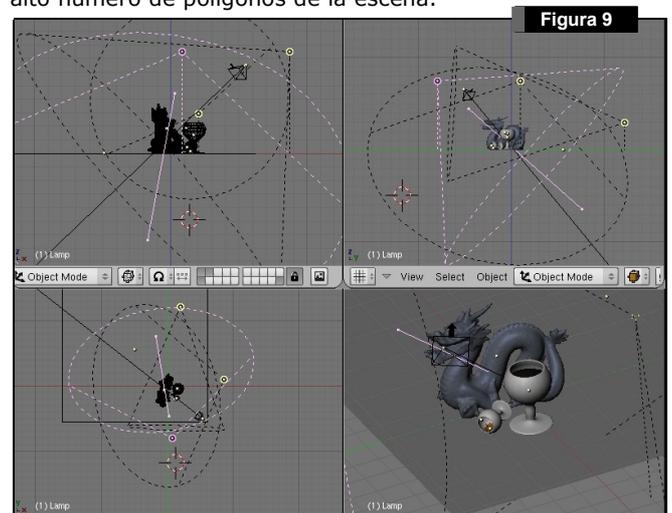


Figura 9

Si utilizamos un esquema de iluminación de 3 puntos (como el que vimos en teoría), añadiremos un único foco que arrojará sombra, y otros dos en las posiciones clásicas de iluminación (ver figura 9). El resultado del render será el obtenido en la figura 10.

El resultado es bastante bueno, aunque hay varios puntos que se podrían mejorar. Uno de ellos es la calidad de las sombras; pese a que hemos utilizado un esquema de iluminación de 3 puntos, no hay sombras difusas. Además, la transparencia del cristal dista mucho de la realidad.



Figura 10

Podemos mejorar el aspecto de las sombras utilizando **Ambient Occlusion**. Este "truco" de iluminación se basa en asignar a cada punto de la escena un valor de sombra que es proporcional al porcentaje de cielo que se puede ver desde esa posición. Es decir, trazamos un conjunto de rayos hacia el cielo, y se comprueba el número que llega al cielo. De esta forma se asigna un valor de iluminación a ese punto de la escena. Naturalmente no es un modelo totalmente realista, pero da resultados bastante aceptables.

Para activar el AO, nos vamos a las propiedades del mundo (figura 11). Definimos el color del horizonte, que será utilizado como color de la luz en AO (y en los métodos de iluminación de Yafray). En la pestaña **Amb Occ**, activamos el botón superior **Ambient Occlusion**. El parámetro Samples indica el número de rayos que se trazan para alcanzar el cielo (en realidad, es el cuadrado del número que indiquemos). A mayor número, mejor calidad de las sombras, menos ruido (pero mayor tiempo de render). Un valor de 10 suele ser aceptable la mayoría de las veces.

Con los valores de **Dist** y **Use Distances** podemos tener más control sobre las sombras calculadas. Los botones de **Add**, **Sub** y **Both** permiten controlar cómo se comporta la oclusión:

- **Add**: El punto recibe luz según los rayos que no se han chocado con ningún objeto. La escena es más luminosa que la original sin AO.
- **Sub**: El punto recibe sombra según los rayos que han chocado con algún objeto. La escena es más oscura que la original sin AO.
- **Both**: Utiliza ambos métodos a la vez.

Mediante el siguiente grupo de botones podemos controlar el color de la luz empleada en iluminación AO; **Plain** emplea luz de color blanca, **Sky Color** utiliza el color



definido en el horizonte (en nuestro caso, es igualmente blanco), o **Sky Texture** si queremos utilizar un mapa de entorno (en este caso, el color de la luz se corresponderá con el color de pixel con el que choque cada rayo).

Energy indica la intensidad que tendrán asignados los rayos de AO. Finalmente, **Bias** permite especificar el parámetro de mezclado para que el resultado sea suave (no se noten las caras poligonales). A mayor valor de **Bias**, menor es el efecto de poligonalización del render.



Figura 12

El resultado del render con AO (empleando los parámetros de la figura 11) puede verse en la figura 12.

Parte 2: Iluminación Global con Yafray

Yafray es un motor de render externo a Blender, aunque permite lanzarlo y ver el resultado obtenido desde el interfaz de Blender. Tendremos que instalarlo independientemente (hay un enlace a la página web oficial del programa desde la sección de descargas de Blender).

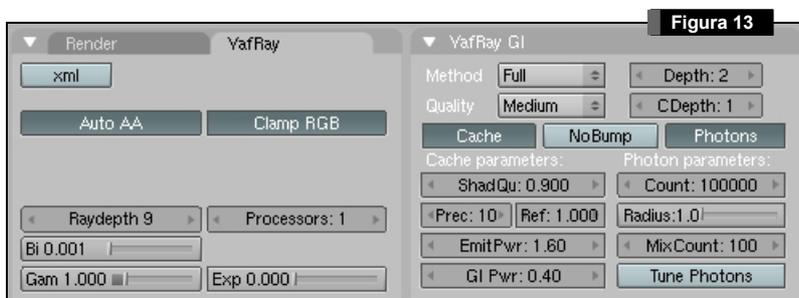


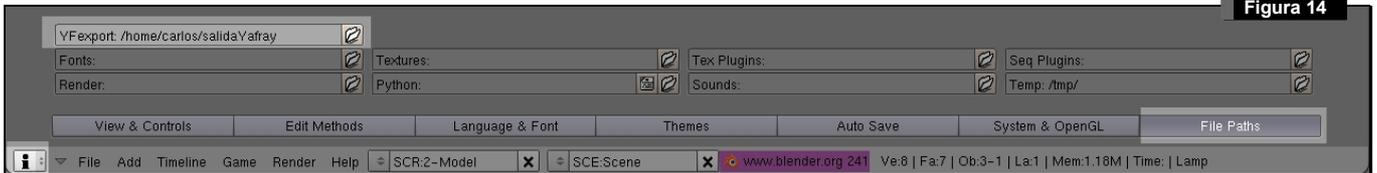
Figura 13

Una vez instalado, podemos elegirlo como motor de renderizado en la lista de motores de render que vimos en la figura 8. Cuando lo seleccionemos, aparecerán dos pestañas con las opciones de configuración de Yafray (figura 13). En la pestaña **YafRay** (figura 13), tenemos opciones generales de configuración; como el botón **AutoAA**, que permite a Yafray seleccionar automáticamente el nivel de AntiAliasing (Oversampling), **Clamp RGB** para realizar AntiAliasing en zonas con alto contraste, la profundidad máxima del trazado de rayos **Raydepth** y ajustes particulares de exposición de la luz, corrección de gama, etc... Si disponemos de un computador con más de 1 procesador, podemos indicar aquí el número de procesadores que queremos que utilice yafray (a nivel de hilos).

El botón superior de color azul **xml** permite exportar el fichero que yafray toma de

Figura 11

Figura 14



entrada. El directorio de exportación se indica en las opciones de configuración generales de blender (accesibles si desplazamos hacia abajo la ventana de información superior, ver figura 14). Este directorio temporal se especifica en **YFexport**. Podemos ejecutar Yafray desde línea de comandos indicándole el fichero XML que se ha generado, para realizar el render de forma totalmente independiente de Blender. Esta forma de trabajar es la típica en cualquier proyecto grande, enviando la última fase de renderizado a un cluster de ordenadores dedicados (*granja de render*).

En la pestaña de iluminación global (**Yafray GI**), en principio presenta dos listas de selección. En la lista de método de iluminación **Method**, podemos elegir entre **Skydome**, en el que la luz se calcula proveniente únicamente desde el cielo (un cielo infinito del color que hayamos definido el valor **Sky** en las propiedades del mundo -figura 11-), o **Full**, que realiza un cálculo de iluminación global completa (con iteración de la luz entre objetos).

Figura 15



Si elegimos el método Sky, Yafray únicamente nos pide el valor de emisión **EmitPwr**. Esta es la intensidad con que se realizará la iluminación de la escena. Es habitual que una escena bien iluminada con el motor integrado de blender se vea algo oscura en Yafray. Mediante este parámetro podemos ajustar el nivel total de iluminación. En la escena de la figura 15 después de algunas pruebas a baja resolución, se estableció en 1.9. Las primeras pruebas se realizaron con el valor **Quality** a "low".



Muy Importante: Las primeras pruebas de render las realicemos a baja resolución (25% de la resolución definitiva o incluso menor), a baja calidad y desactivando el *Oversampling* (OSA). Cuando los valores de iluminación sean correctos, activaremos el render a máxima calidad con un nivel de *Oversampling* aceptable (8 suele ser suficiente).

Una vez lanzado el render, veremos cómo la escena va apareciendo poco a poco en la ventana de render. Podemos parar el render pulsando **Esc**.

El resultado utilizando este método de render es el mostrado en la figura 15. Un detalle **importante** a tener en cuenta cuando trabajemos con **texturas de imagen** es

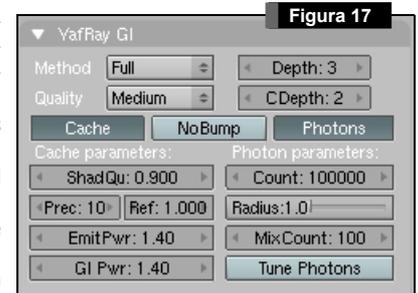
que no pueden ir empaquetadas en el fichero de blender en el momento de hacer el render. Como Yafray es un motor externo, requiere que los ficheros de textura estén disponibles en alguna ruta del sistema de archivos. Si tenemos una textura empaquetada (aparece el icono del paquete pulsado), podemos desempaquetarla automáticamente pulsando sobre el botón con un icono de un paquete (figura 16). A nivel global de todas las texturas de una escena que estén previamente empaquetadas, mediante la orden **File / Unpack Data**.

Figura 16



El otro método que proporciona yafray (seleccionando en la lista desplegable **Method** la opción **Full**) es mucho más realista. La lista de calidad **Quality** ajusta, según el parámetro seleccionado, las variables que necesitará yafray para renderizar la escena. En definitiva es el número de muestras de luz que calculará. El parámetro **EmitPower** situado en el inferior del panel permite ajustar la intensidad de las fuentes de luz de la escena. En algunas escenas habrá que ajustar este parámetro (normalmente incrementarlo) para obtener un valor de iluminación correcto.

Figura 17



Con el método de renderizado completo, tendremos que especificar dos nuevos parámetros; **Depth** (rebotes de la luz entre objetos) y **CDepth** (rebotes de la luz dentro de objetos translúcidos para obtener caústicas).

Si activamos el modo **Cache**, aceleraremos los cálculos empleando *Irradiance Cache*. Esta cache nos evita tener que calcular la iluminación global en cada punto; selecciona algunos píxeles para realizar los cálculos y después interpola el valor de iluminación. En **ShadQu** indicamos la calidad de las sombras; con valores más altos obtenemos mejores resultados (naturalmente, con mayor tiempo de cálculo). **Prec** indica la precisión del píxel (un valor más bajo implica mayor calidad). **Ref** es un valor umbral, donde especificamos el cambio de intensidad que vamos a permitir entre muestras. Valores más pequeños dan resultados más precisos.

El botón **NoBump** permite evitar el cálculo de cambio del vector normal en la superficie si se utiliza Bump Mapping. Esto es mucho más rápido si se está utilizando *Irradiance Cache*. Aunque es menos realista, debemos evitar el uso de Bump Mapping porque necesita muchos más cálculos en Pathtracing con Cache.

Los fotones **Photons** únicamente ayudan al cálculo de la iluminación. Funcionan especialmente bien en escenas de interiores donde no hay cielos visibles, donde la luz entra por una zona claramente definida. Aunque en nuestra escena no serían necesarios, vamos a utilizarlos y a estudiar sus parámetros. **Count** indica el número de fotones a lanzar, un número mayor implica un resultado más suave. **Radius** indica el nivel de difuminado entre



Figura 18

fotones. Es importante encontrar un valor adecuado para que el resultado sea suave. **MixCount** indica el número de fotones que van a situarse dentro del radio anterior. Finalmente, el botón **Tune Photons** sirve para obtener una representación preliminar del mapeado de fotones y ver si la imagen acabará correctamente iluminada. Para más información sobre el uso de fotones y consejos de iluminación de escenas interiores, consultar la sección de documentación de la web oficial de yafray¹.

Los parámetros de renderizado de la escena que se muestra en la figura 18 empleando el método completo de iluminación son los mostrados en la figura 17.

Parte 3: HDRI, Caústicas y DOF

Vamos a mejorar el resultado anterior empleando algunas características avanzadas de yafray. Una de ellas es la posibilidad de emplear mapas de iluminación HDRI. Estos mapas almacenan información sobre iluminación (además de las componentes de color).



Figura 20

En nuestro caso, vamos a emplear el mapa HDRI que se muestra en la figura 19. Se puede consultar más información sobre este tipo de mapas (además de algunos ejemplos para descargar) en la web de Paul Debevec².

➔ Iluminación basada en Imágenes HDR

Para cargar el mapa HDRI, en las propiedades del mundo (World Buttons) añadimos una textura de tipo imagen y cargamos la imagen HDR (en nuestro caso, **car.hdr**). Como es una mapa HDR esférico, tendremos que activar el botón **AngMap** en el tipo de mapeado de la textura (ver figura 20).

En la pestaña **Map To**, tendremos que activar el botón **Hori** para que el mapa HDR se proyecte en el horizonte.

Ahora tenemos que ajustar los parámetros de iluminación de nuevo, ya que el mapa HDR almacena información sobre iluminación. No hay ninguna fórmula mágica para conseguir el resultado esperado, ya que no sabemos el rango de iluminación de

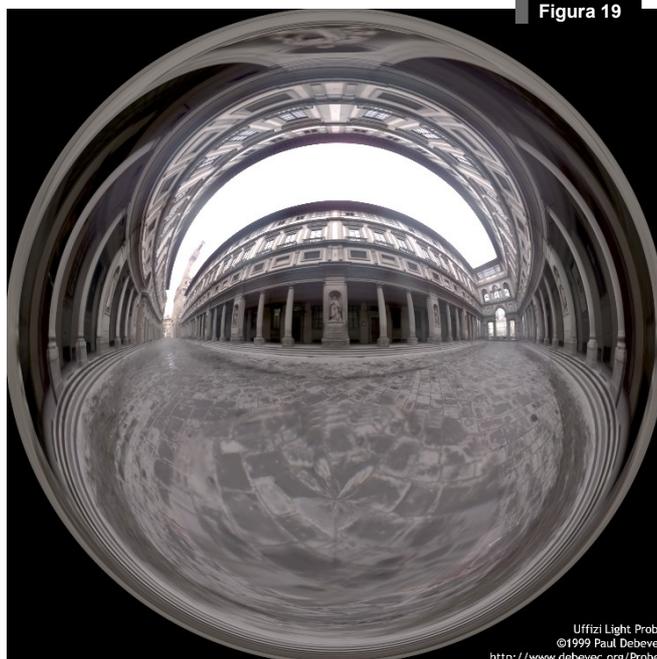


Figura 19

la escena hasta que no lanzamos el render. En nuestro caso, los parámetros que mejor se ajustaban a la escena eran **EmitPower** y **GIPower** en 0.4.

➔ Profundidad de Campo (DOF)

Otro efecto muy interesante y fácil de configurar en Yafray es la profundidad de campo (los objetos que estén en la zona de enfoque aparecerán nítidos, y según nos alejamos de la zona de enfoque, los objetos aparecen desenfocados).

Con la cámara seleccionada, nos vamos a los botones de edición y pinchamos en ShowLimits. Esto nos mostrará el volumen de visualización de la cámara y una equis de color amarillo. Esta equis representa la distancia de enfoque perfecto de la cámara (debemos colocarlo donde queramos que la visualización sea nítida, variando la distancia **DoFDist**). Mediante el parámetro **Aperture** indicamos cómo de borrosos queremos que aparezcan los objetos alejados del punto indicado en **DoFDist**; cuanto más grande es **Aperture**, más desenfoco habrá. Para la escena de esta sesión, los valores utilizados son los mostrados en la figura 21.

➔ Caústicas

Las caústicas se refieren a los patrones de refracción que presentan algunos materiales, como el cristal, que concentran los fotones en un punto y modifican el color de las superficies sobre las que se proyectan.

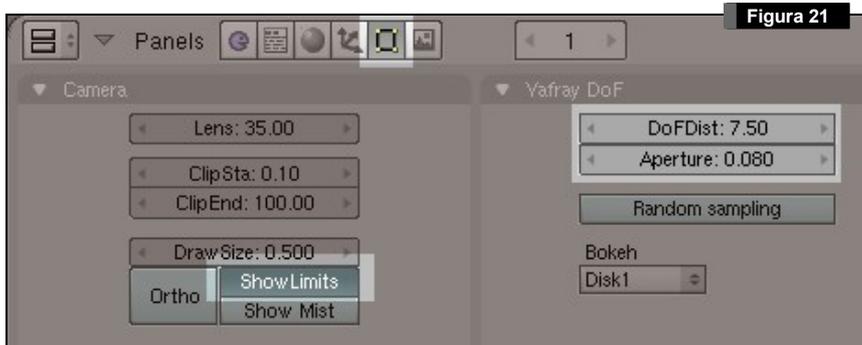
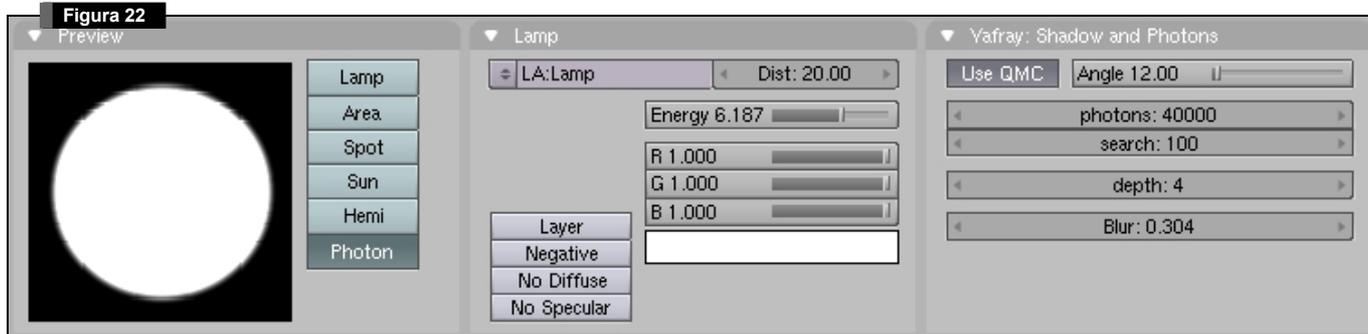


Figura 21

1 <http://www.yafray.org>
2 <http://www.debevec.org/>



Algunos elementos de iluminación (como luces de área) arrojan caústicas sobre los objetos directamente. Los mapas HDR no; sin embargo, existe un tipo de fuente de luz específica para añadir este efecto, la fuente de luz **Photon**. Esta fuente no ilumina la escena realmente; únicamente añade fotones que logran este efecto de caústicas.

Añadimos una fuente de luz de este tipo y ajustamos los parámetros como se muestra en la figura 22. Si activamos **QMC**, utilizamos el método de **Quasi Monte Carlo** para distribuir los fotones de la fuente. En **Angle** especificamos el ángulo de salida de los fotones (cuanto más concentrados estén, más intensas serán las caústicas). En **Photons** indicamos el número de fotones a lanzar, en **Depth** especificamos el número de rebotes que pueden dar los fotones en el interior de los objetos. Los parámetros de **search** y **blur** miden el mezclado entre los fotones (con valores mayores tenemos más sensación de uniformidad en la caústica). Al igual que antes, estos parámetros son muy particulares de cada escena y no hay fórmulas mágicas

para encontrar el punto correcto. El número de fotones no debemos subirlo mucho en ninguna circunstancia; suele ser mejor aumentar la energía de los fotones hasta que veamos algunas manchas en la escena y luego variar los parámetros de mezclado.

La figura 23 es el resultado de aplicar el mapa HDR, el efecto de profundidad de campo y caústicas.

En muchas ocasiones la variación en el número de samples de la PathLight (especificado en el interfaz de Blender mediante la lista desplegable **Quality** - figura 13 -), no se corresponde con un aumento en la calidad de la imagen final. Es muy importante elegir el nivel de samples adecuado, ya que el tiempo de render de una misma escena puede verse afectado incluso en idos órdenes de magnitud!. Si el resultado del proyecto es una imagen estática, podemos repartir el render de la imagen en varios computadores y componer el resultado dejando una zona reservada entre fragmentos para poder realizar la composición de la imagen final.



Figura 23



Animación No Lineal (NLA)

Animación para la Comunicación · Escuela Superior de Informática · Universidad de Castilla-La Mancha

➤ En esta sesión utilizaremos el esqueleto al que realizamos el "rigging" en la práctica 10 para generar una sencilla animación empleando acciones básicas combinadas en el módulo de ANL.

Para comenzar a trabajar, abrimos dos ventanas nuevas en blender. Una que contendrá el editor de Acciones y otra de tipo editor NLA (Non Linear Animation). En animación no lineal, definiremos las acciones como conjuntos de posiciones clave de un subconjunto de huesos. Así, crearemos el movimiento de "andar", el de "girar la cabeza" y el de "señalar", y los compondremos para obtener la animación final.

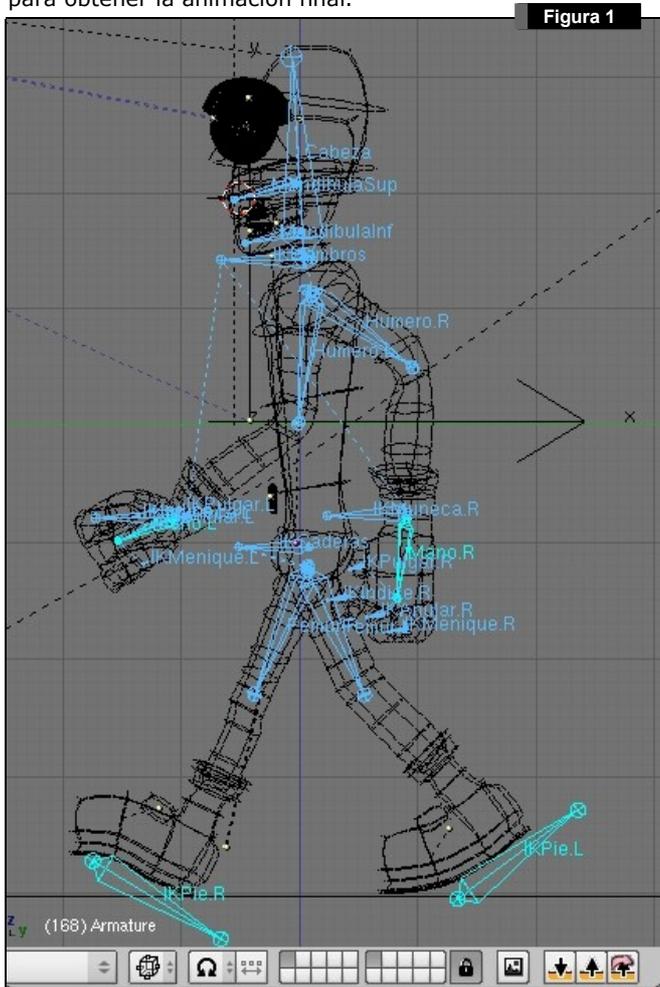


Figura 1

pinchamos en el botón o seleccionamos **Pose / Copy current Pose**, ambas opciones en la cabecera de la ventana 3D. Al lado de este icono hay dos que nos permiten **pegar la posición** copiada y **pegar la posición invertida** . Podemos realizar estas operaciones con sus correspondientes opciones dentro del menú **Pose** visto anteriormente.

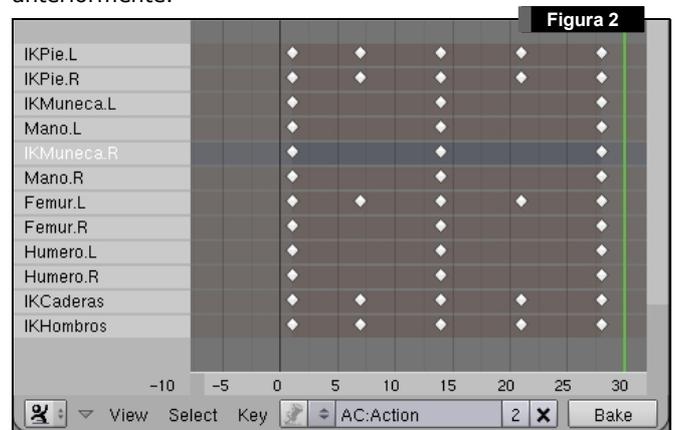


Figura 2

Mediante el pegado invertido, siempre que hayamos utilizado la notación especial empleada por blender de nombrar mediante **.R** y **.L** los huesos simétricos en cada lado del cuerpo, podremos pegar la pose empleada invertida.

Así, pinchamos en copiar la posición actual y nos desplazamos al frame 14. Pegamos la posición invertida e insertamos frame clave . Nos vamos al frame 28 y pegamos la posición normal . Insertamos de nuevo un frame clave. Si reproducimos la acción, tendremos un movimiento parecido a andar, pero sin levantar los pies del suelo (estilo *Moonwalker*). Bastará con que nos situemos en los frames intermedios (7 y 21) y añadamos nuevos frames clave con la pierna que corresponda levantada. Debemos conseguir posicionar las articulaciones como se muestra en la figura 10.

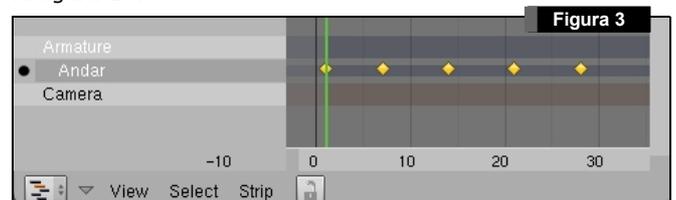


Figura 3

Vamos a añadir el movimiento de mover las piernas al caminar. En la ventana 3D, situamos el esqueleto en la posición que muestra la figura 1. Para ello, desplazaremos y rotaremos en modo Pose **Ctrl+Tab** los huesos de definición de la cinemática inversa finales; **IKMano.***, **IKMuneca.***, **Femur.***, **Húmero.*** e **IKPie.***. Con los huesos seleccionados, insertamos un frame clave de Localización y Rotación (**LocRot**). En la ventana de Acciones deberán aparecer unos cuadros de color amarillo en el frame 1. Con los cuatro huesos seleccionados, vamos a copiar la posición actual al *buffer* de blender. Para ello,

Hecho esto, obtendremos una acción en la ventana de acciones como se muestra en la figura 2. En la ventana NLA aparecerán una serie de cuadros amarillos indicando los frames clave (figura 3). Podemos ver la animación de la acción pulsado **Alt+A**. Como la acción está terminada, la renombraremos como "Andar", tal y como muestra la figura 2 y la quitaremos de la ventana de NLA. Para esto, pinchamos en el botón de la cabecera de la ventana de acciones. Esto no elimina la acción realmente; podemos

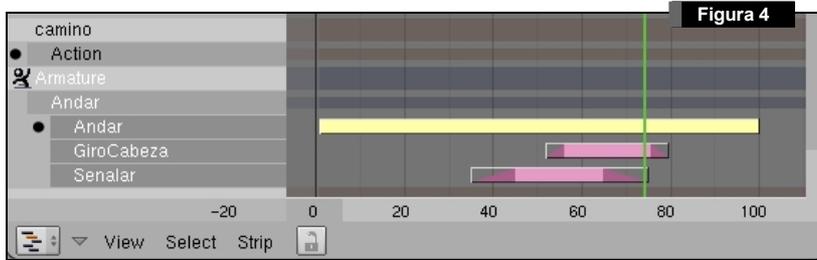


Figura 4

recuperarla de nuevo accediendo a la lista desplegable [v]. Con la acción creada, podemos utilizarla como franja de Animación No Lineal; que se representa mediante barras amarillas que podemos escalar, suavizar y componer.

En la ventana de NLA [v], y con el esqueleto seleccionado (la banda "Armature" tiene que aparecer con el rótulo en color blanco en la ventana de NLA), pulsamos [Shift+A], **Add Action** y seleccionamos la acción recién creada ("Andar"). Si la acción se ha quedado insertada en la ventana de NLA como un conjunto de frames clave (como aparecía en la ventana de Acciones), podemos convertirla a barra de animación con la tecla [C]. Una vez que tengamos la barra de animación NLA, podemos escalar la duración de la acción de caminar al número de frames que queramos (como cualquier objeto, con [S]), o desplazarlo en la línea de tiempo con [G]. Crearemos nuevas acciones (como taparse la cara, girar la cabeza...) de forma similar. Podemos añadir acciones ya creadas a la ventana NLA pulsando de nuevo [Shift+A].

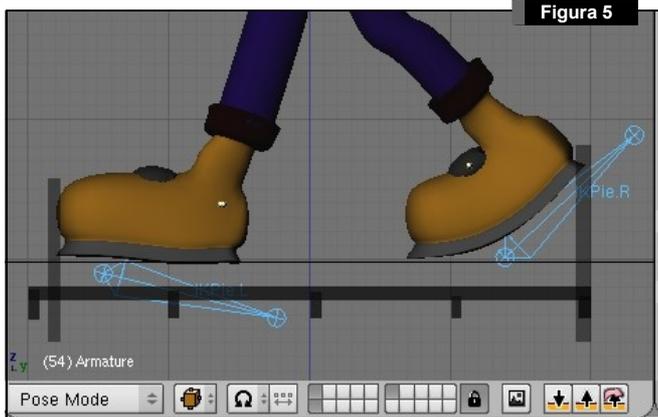


Figura 5

Podemos ver cómo quedó la composición de acciones de esta sesión en la figura 4. A continuación analizaremos algunas opciones de composición.

Antes de hacer que el personaje siga un camino, vamos a medir la longitud de paso de nuestro personaje. Para ello, desde la vista lateral de la acción de andar, y con el paso totalmente extendido (figura 5), anotamos el número de unidades de la rejilla que abarca el paso (en el ejemplo de la figura, 3.75 unidades).



Figura 6

Con esto ya sabemos el número de unidades de la rejilla, pero... ¿Cuánto mide la rejilla? Para ver esto, accedemos a **View/ View Properties** en la cabecera de la ventana 3D, y como muestra la figura 6, tenemos el valor en el campo Grid: Spacing. En el caso de nuestro ejemplo, cada celda de la rejilla mide 1 unidad real.

Finalmente, ¿cuánto avanza el personaje en cada paso? Pues, avanza 3.75 unidades de rejilla x 1 unidad que mide cada rejilla x 2 "mitades simétricas" que tiene cada paso = 7.5 unidades reales por cada paso. Anotamos este valor, porque lo utilizaremos más adelante.

¿Cómo hacer que el personaje siga un camino? Necesitamos añadir un elemento de tipo **Curve, Path**. Ajustamos los puntos de control y emparentamos el esqueleto (hijo) a la curva (padre), [Ctrl+P] **Normal Parent**. Hecho esto, si reproducimos la animación [Alt+A] comprobamos que el personaje sigue el camino, pero no rota adecuadamente.

Para conseguir que el esqueleto gire según el vector normal al camino que hemos añadido, insertamos una restricción al esqueleto de tipo **Follow Path** (figura 7). Indicamos en el campo de Objeto (**OB**), el nombre del path insertado (en el caso del ejemplo, "Curve"). Activamos el botón **Curve Follow**. Es importante indicar el eje correcto que apunta hacia el techo **Up**, y el vector que apunta hacia delante **Fw** (en el ejemplo de la figura 7 sería el eje negativo de las Y).

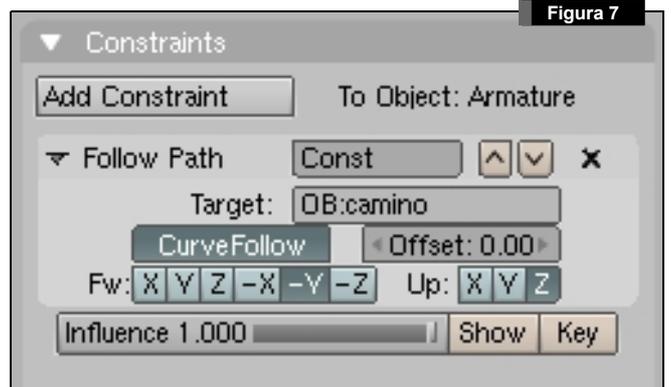


Figura 7

En la ventana NLA, seleccionamos la acción "andar" y pulsamos [N]. Con esto, accedemos a las propiedades específicas de la acción (ver figura 8).

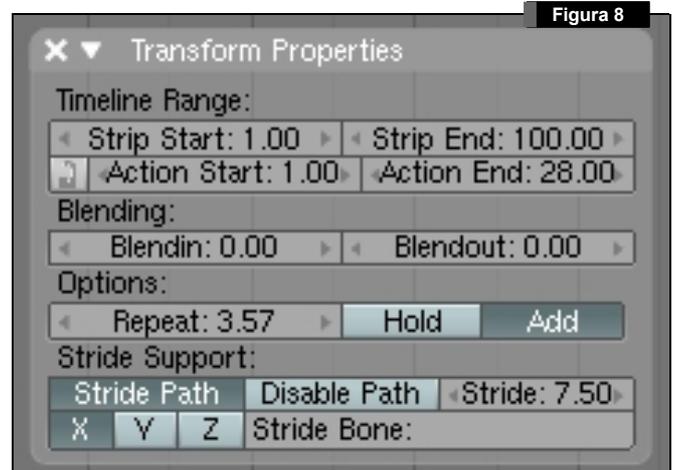
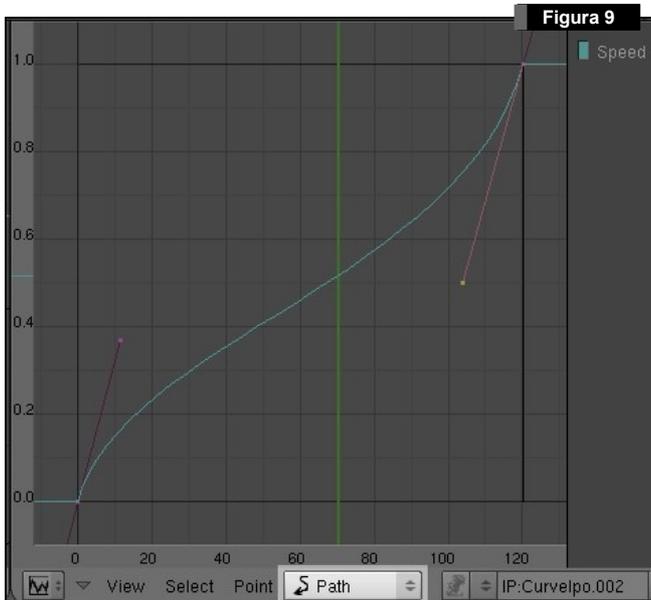


Figura 8

- Las propiedades de **Strip Start** y **Strip End** indican el momento en el que la acción comienza y termina en la planificación NLA actual.
- **Action Range** hace mención a la duración (en frames) de la acción (independiente de dónde empiece).
- **Blendin** y **Blendout** son el número de frames que damos para mezclar la acción con las siguientes. Es un factor de suavizado que aplica Blender en la interpolación entre acciones.
- **Repeat** indica el número de veces que vamos a repetir la acción.
- **Stride** es un parámetro muy importante si estamos utilizando caminos (*paths*), como en nuestro caso.

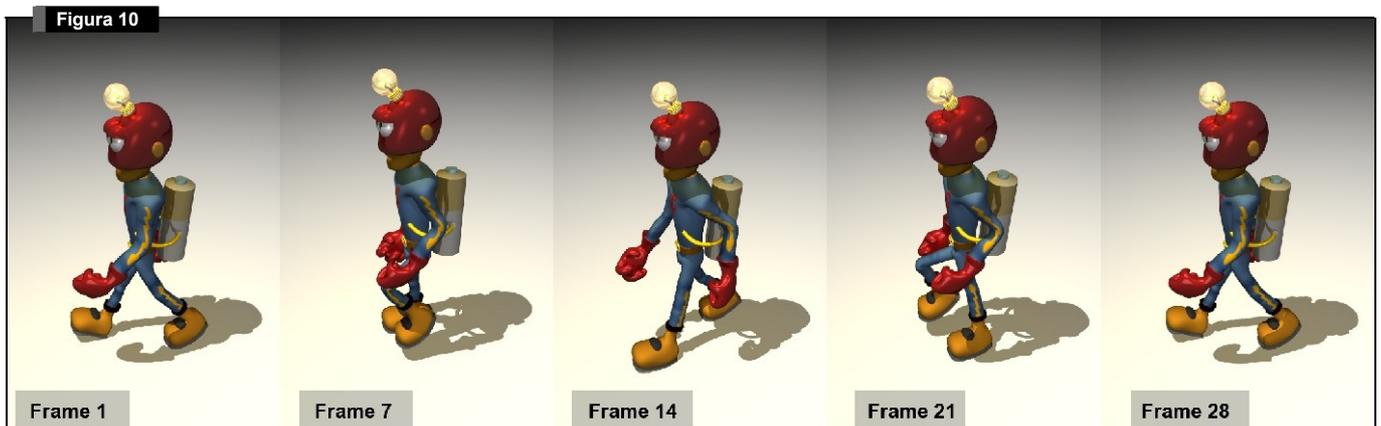


Indica a Blender el número de unidades que avanza el modelo en cada ciclo. Debemos ajustar este parámetro para evitar que el modelo "patine" sobre el suelo. Aquí debemos indicar el parámetro calculado anteriormente (7.5 unidades *reales* en nuestro caso).

- Si activamos **Stride Path**, la acción se va a sincronizar con el avance del path. En nuestro ejemplo, activaremos este botón únicamente en las acciones que tengan que ver directamente con el acto de caminar; es decir, únicamente "andar".
- El botón de **Hold**, si está activo, nos conserva la última posición alcanzada por una acción.
- El botón **Add** indica que el movimiento final resultará de la composición de esta acción con el resto (sus efectos se suman). Por lo general, este parámetro tendrá que estar siempre activo para todas las acciones (salvo en la acción que sirva de base del movimiento; en nuestro caso, *andar*).

Por último, recordaremos que Blender permite trabajar hasta el mínimo nivel de detalle mediante curvas IPO. Podemos, por ejemplo, variar la velocidad del personaje que sigue sobre el path accediendo a la curva IPO asociada al camino (ver figura 9). Para esto, tendremos que elegir de la lista desplegable la opción Path. Podemos variar la duración del camino desplazando hacia la izquierda el punto final de esta curva (en el ejemplo está ajustado a 100 frames).

La animación no lineal puede combinarse perfectamente con animación basada en frames clave que hemos utilizado



en sesiones anteriores. En el video de ejemplo de esta sesión, el movimiento de los ojos ha sido animado insertando frames clave entre las acciones de ANL.



¿Y ahora qué?

Ahora te queda continuar "investigando" por tu cuenta; estudiando algunos módulos muy interesantes que no hemos podido ver en sesiones de prácticas por falta de tiempo, pero que podrás desarrollar sin problemas como:

- **Simulaciones físicas:** Empleando el motor de simulaciones físicas *Bullet* (se selecciona dentro de las opciones del mundo), dando propiedades físicas en el game engine y activando la opción **Game / Record Game Physics to IPO**, del menú principal de *Blender*.
- **Softbodies:** Incorporados en la versión 2.40, permiten la simulación física de interacciones con objetos que no son totalmente rígidos.
- **Simulación de fluidos:** Basta con indicar qué objeto será el fluido y cual el contenedor. La simulación conseguida es muy realista.
- **Composición de video:** Blender incorpora un compositor de video y audio bastante completo. Se puede acceder a él cambiando una ventana a tipo Video Sequence Editor. En esta parte se pueden importar secuencias de video y audio, imágenes y realizar transiciones entre ellas. Para generar el resultado habrá que activar el botón Do Sequence de las propiedades de Render.
- **Otras características** como la generación de pelo, animación facial (drivers), métodos de render no realista (tipo Toon Shading) efectos de profundidad de campo, etc... Suerte y happy blending! :-)