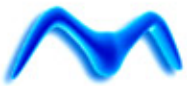
Diseño Gráfico 3D con Software Libre

(Una Introducción Práctica a Blender)


Blender 2.33


Yafray 0.0.6


Wings3D 0.98


GIMP 1.2.4

NICODIGITAL
http://www.nicodigital.com

Carlos González Morcillo
Escuela Superior de Informática de Ciudad Real
Universidad de Castilla-La Mancha
Carlos.Gonzalez@uclm.es

Introducción

¿Qué es este documento? ¿Por qué Software Libre? Algo sobre Blender, Yafray, Gimp, Wings...

Este manual que tienes en tus manos (o en tu pantalla) es una recopilación de los guiones de prácticas de la asignatura "Animación para la Comunicación", impartida en la Escuela Superior de Informática de Ciudad Real (Universidad de Castilla-La Mancha). Por esta razón, encontrarás en el texto referencias a conceptos que fueron explicados en clases de teoría, o se completaron en la misma sesión de prácticas.

Esta nueva versión de los apuntes, del curso 2003-2004, incluyen algunas sesiones totalmente nuevas (como la práctica de Render Realista empleando Yafray), y en general, han sido revisados y ampliados utilizando las últimas versiones de Blender.

Este documento no pretende ser un manual exhaustivo de Blender, ni explicar toda la funcionalidad de la herramienta. Simplemente trata de facilitar la toma de contacto (habitualmente *dolorosa*) con una herramienta de diseño 3D de características profesionales. El planteamiento es tratar de obtener, en 30 horas de prácticas, una visión general del ciclo de desarrollo en un proyecto de síntesis de imágenes tridimensionales por computador, incluyendo modelado, texturizado, animación, render y composición de vídeo.

¿Y por qué utilizar Software Libre?. Buena pregunta... Responderé con otra: ¿Y por qué utilizar Software Propietario (*o Privativo*)?. No hay, en principio, diferencias notables de calidad en los resultados obtenidos por artistas profesionales utilizando Blender, Gimp, etc... con sus equivalentes en software comercial. El uso de software libre nos da, precisamente, la libertad de utilizar el software en cualquier ámbito, en multitud de sistemas operativos, e incluso de realizar adaptaciones del programa a nuestras necesidades; dejando la puerta abierta a futuras ramas de desarrollo. Por otro lado, podemos tener la seguridad de que el programa siempre va a seguir vivo y actualizado con las necesidades de los usuarios (al menos, mientras exista la comunidad de desarrolladores del software que, difícilmente puede desaparecer). La velocidad de crecimiento de Blender y de los programas y plugins que hay alrededor es impresionante. Casi mensualmente aparece una nueva versión con una enorme lista de mejoras importantes. Es, sin duda, un fenómeno imparable. Por estas y otras razones estoy cada día más convencido que la apuesta realizada por el Software Libre es la acertada. En este documento se hace referencia a varios programas (la mayoría libres o, en su defecto, totalmente gratuitos -Como ZweiStein o Teddy-).

Será bienvenida cualquier sugerencia sobre estos apuntes, así como notificaciones de errores encontrados en la dirección Carlos.Gonzalez@uclm.es.

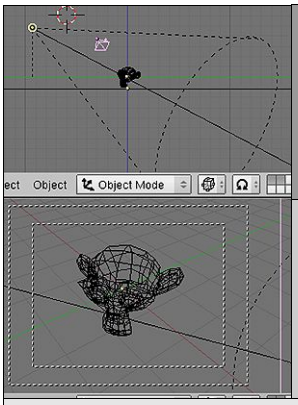
Finalmente, quiero agradecer a todos los compañeros de Nicodigital la ayuda prestada; la lista de nombres sería muy, muy larga. Como no podía ser de otra forma, a los alumnos de Animación de la ESI de Ciudad Real, por el interés mostrado en la asignatura; todo el esfuerzo por mi parte merece la pena. Y por último, y no menos importante, a Nines y a mi familia, por apoyarme en todos los berenjenales en los que me meto :-)

En definitiva: espero que este documento te sirva para comenzar en este mundillo de la síntesis de imágenes tridimensionales por computador. Creo acertado incluir en este punto un refrán que resume a la perfección el ciclo de aprendizaje en diseño 3D:

"Díjole el perro al hueso,
Si tú eres duro, yo tengo tiempo"

Ánimo y... Happy Blending!

Carlos González Morcillo
28 de Mayo de 2004



Práctica 1: Introducción a Blender

Animación para la Comunicación ::: ESI-CR ::: UCLM

Carlos González Morcillo :: Carlos.Gonzalez@uclm.es :: http://www.inf-cr.uclm.es/www/cglez

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.

Software: Blender 2.32. **Fecha:** 04/Marzo/2004

El interfaz que presenta Blender al arrancar es el mostrado en la Figura 1. Todas las ventanas de Blender son personalizables, en el sentido de que podemos dividir la pantalla como nos resulte más cómodo y situar en cada porción el tipo de

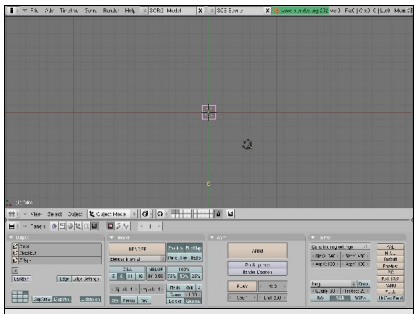


Figura 1

ventana que queramos. Por defecto Blender arranca la pantalla con dos divisiones; la ventana superior 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 siguiente imagen:

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 punto inferior de color amarillo 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 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.

Antes de empezar a trabajar, personalizaremos un poco el entorno de Blender. Dividiremos la ventana 3D con la distribución que se muestra en la Figura 2. 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

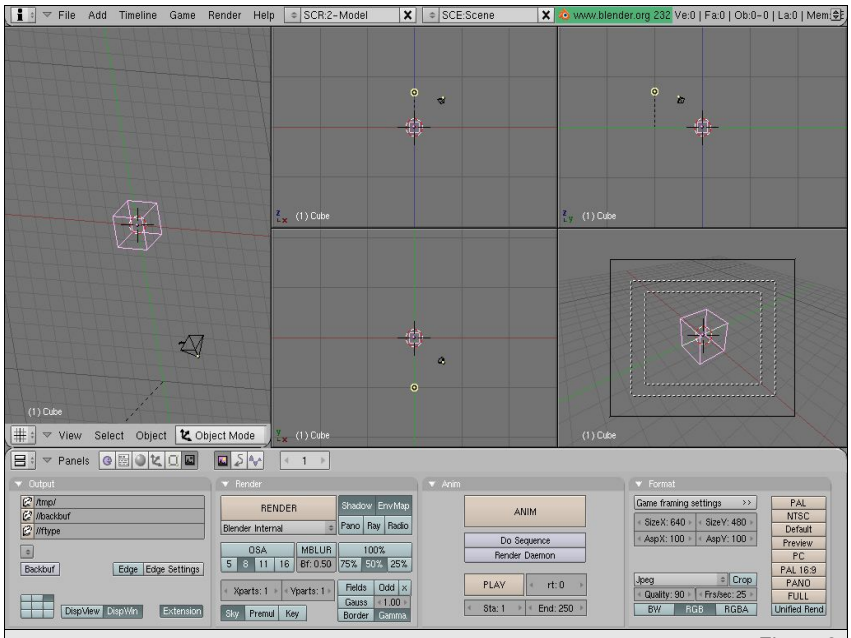


Figura 2

ventana 3D iluminada (seleccionada), haremos y elegiremos *Split Area*. Partiremos la pantalla hasta conseguir una disposición del área de trabajo como se muestra en la Figura 2. Podemos eliminar las cabeceras, así como situarlas en la parte superior de las ventanas haciendo sobre ellas y seleccionando *No Header*.

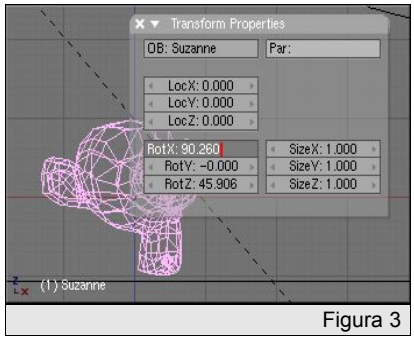


Figura 3

Para rotar el punto de vista de una ventana 3D, arrastraremos con pinchado. En ratones de dos botones, con Alt pulsada, arrastraremos el ratón con . Podemos hacer zoom en cualquier ventana (vista 3D, ventanas de botones, etc...) manteniendo Ctrl pulsado, y arrastrando . En ratones de dos botones, mediante Ctrl Alt . Para desplazar el punto de vista (horizontal y verticalmente), utilizaremos Shift . De igual forma, con ratones de dos botones utilizaremos Alt Shift .

Cambiaremos el modo de vista de cada vista 3D mediante atajos de teclado o los botones destinados a tal efecto. Pulsando 5 en el teclado numérico, cambiaremos entre

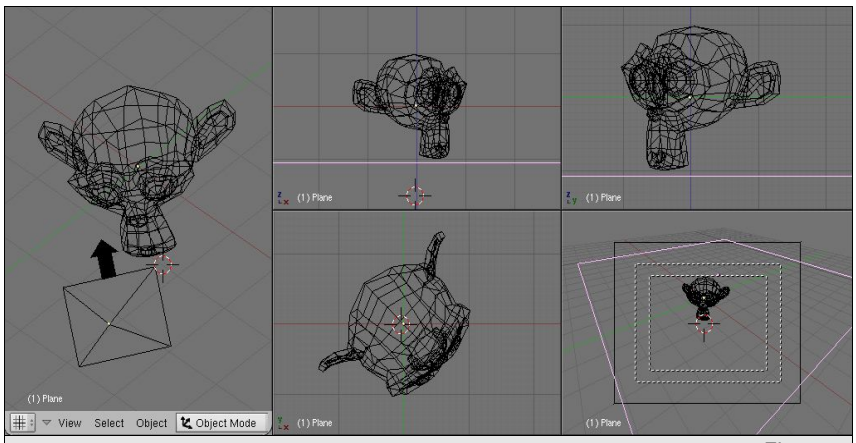


Figura 4

proyección ortográfica y perspectiva. Con 0 tendremos la vista de la cámara. 1, 3, 7 nos darán las vistas de la escena superior, frente, y perfil respectivamente.

Una vez configuradas las vistas como se muestra en la Figura 2, vamos a personalizar nuestra escena. Con el cubo por defecto seleccionado (seleccionamos objetos con 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*. 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.

Algunas operaciones que podemos realizar con el objeto es rotarlo; con él seleccionado pulsamos R y desplazamos el ratón. Si pulsamos ahora X, Y o Z, delimitaremos la rotación a ese eje. Si rotamos con la tecla Shift pulsada, lo haremos en modo de precisión. Si tenemos la tecla Ctrl, lo haremos de 5 en 5 unidades. La información sobre las unidades en que se está realizando la operación aparece en la cabecera de la ventana 3D de trabajo.

Otra operación básica es el desplazamiento (tecla G del inglés *Grabber*). Si desplazamos con la tecla Ctrl pulsada, lo haremos ajustándonos a la rejilla. Al igual que con la rotación, la tecla Shift se corresponde con el modo de precisión.

El escalado se realiza con la tecla S. Para restringir el escalado en alguna dimensión, pulsaremos S, después las letras correspondientes a los ejes cartesianos X, Y, Z. Las restricciones se realizan sobre el sistema de coordenadas local del objeto.

Todas las operaciones anteriores pueden realizarse de forma numérica, si con el objeto seleccionado pulsamos N. Aparecerá una ventana como la que se muestra en la Figura 3. Los valores numéricos pueden introducirse por teclado si hacemos click con sobre una entrada de texto mientras mantenemos pulsada la tecla Shift.

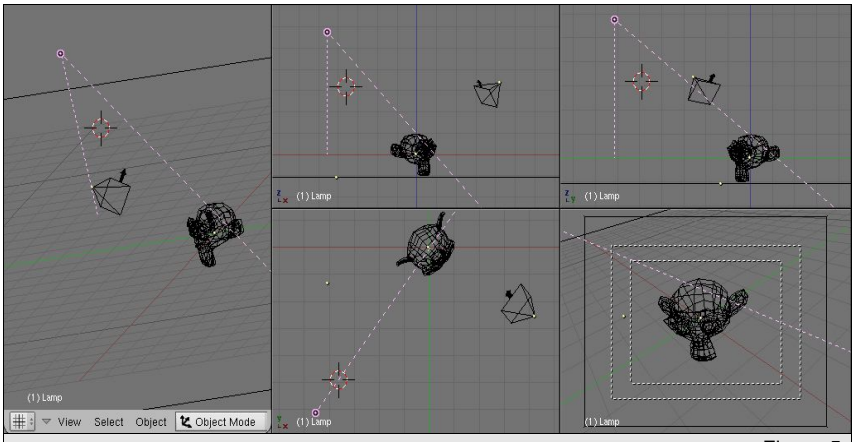


Figura 5

Ayudándonos de las vistas de frente, perfil y superior, situaremos el puntero 3D debajo de la cabeza del mono. Recordemos que la posición del puntero 3D se cambia pinchando con 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. Deberemos obtener una configuración similar a la Figura 4.

El Render se realiza a través del botón del grupo de Escena (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 F11. Añadiremos una luz de tipo **Hemi** en la posición que muestra la Figura5.

Para ello, situaremos el cursor 3D en la posición adecuada. Añadiremos la luz mediante Spacebar **Add**, **Lamp**. Cambiaremos el tipo de lámpara con el botón de gestión de lámparas del grupo de sombreado (o pulsando F8). Seleccionamos el tipo de foco **Hemi**. Habrá que orientar el foco para que apunte al objeto (utilizando la rotación R).

Realizaremos un render y comprobaremos que el resultado es similar al mostrado en la Figura 6. Podemos salvar el render, pulsando F3. El formato de la imagen que salvaremos se selecciona en el apartado del botón de render, 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 implícita en la imagen que guardemos desde Blender. De esta forma, en nuestra primera práctica debemos indicar que el fichero resultado se va a llamar "*resultado.jpg*".

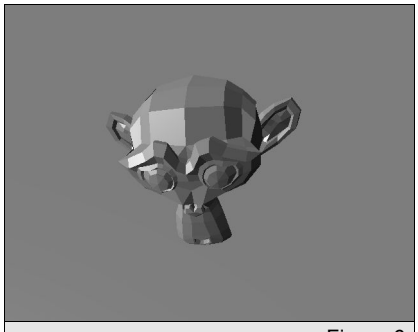
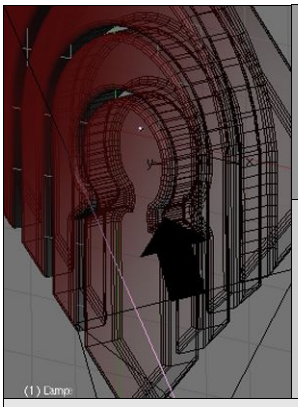


Figura 6



Práctica 2: Modelado y Composición

Animación para la Comunicación ::: ESI-CR ::: UCLM

Carlos González Morcillo :: Carlos.Gonzalez@uclm.es :: http://www.inf-cr.uclm.es/www/cglez

Como continuación de la práctica anterior, aprenderemos a crear en Blender objetos más complejos, partiendo de un diseño 2D. Veremos cómo trabajar con curvas y, finalmente, compondremos el resultado con GIMP.

Software: Blender 2.32. + GIMP 1.2.4. **Fecha:** 11/Marzo/2004

Arrancamos 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*.

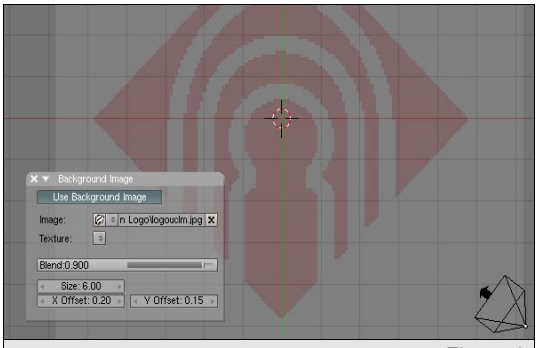


Figura 1

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

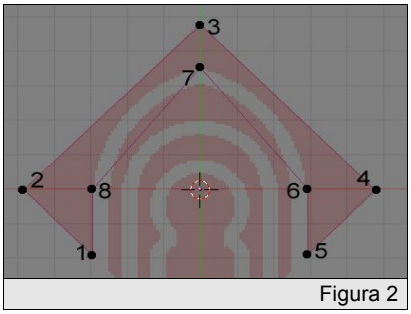


Figura 2

Por defecto, tenemos todos los vértices seleccionados (todos en color amarillo). Pulsamos la tecla A y los deseleccionamos todos (aparecen en rosa). Ahora desplazaremos los vértices al contorno exterior del logotipo (seleccionamos cada vértice de forma individual con , y desplazamos pinchando una vez la tecla G). 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 +) para tener una mayor precisión a la hora de situar los vértices.

Para añadir vértices: manteniendo pulsado Ctrl, haremos . 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). Podemos alinear los manejadores de tensión de la curva pulsando H si en algún momento es necesario. tura en cada sección de la misma.

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. Volvemos a la representación con líneas pulsando de nuevo la tecla Z.

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...*

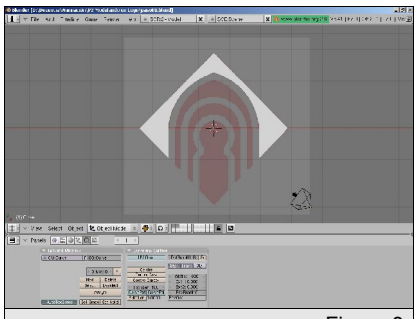


Figura 3

Cada curva creada es un objeto independiente; podemos unirlas creando una única entidad si las seleccionamos todas con (manteniendo Shift pulsada), y después pulsando Ctrl J *Join Selected Curves*.

Pasaremos a extrusionar 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 se-

lección de objeto, cambiamos los valores de extrusión y redondeado tal y como se muestra en la figura 4. Podemos cambiar los valores en los botones de inserción, pinchando y arrastrando sobre ellos, o bien manteniendo pulsada la tecla Shift y haciendo sobre ellos.

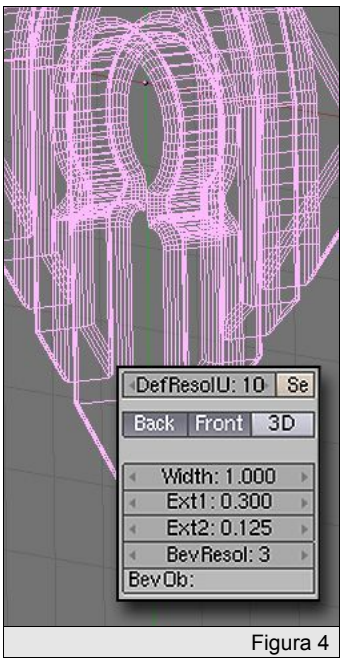


Figura 4

Vamos a aplicarle un material sencillo al modelo. Seleccionamos una de las partes y pinchamos en el botón de materiales, tal como muestra la figura 5 (o bien pulsamos F5). 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*. 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.

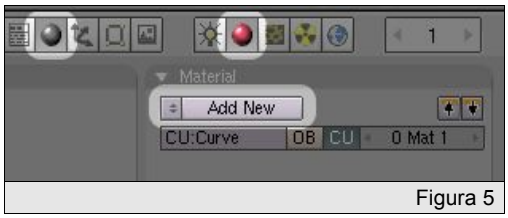


Figura 5

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 y tecleamos un nombre para el material ,por ejemplo, *UCLMRojo*. Aplicaremos el material que hemos creado a las piezas del logotipo. Para ello, lo seleccionamos y, elegimos el material en la lista desplegable de materiales. Podemos activar el modo de visualización sólida pulsando Z.

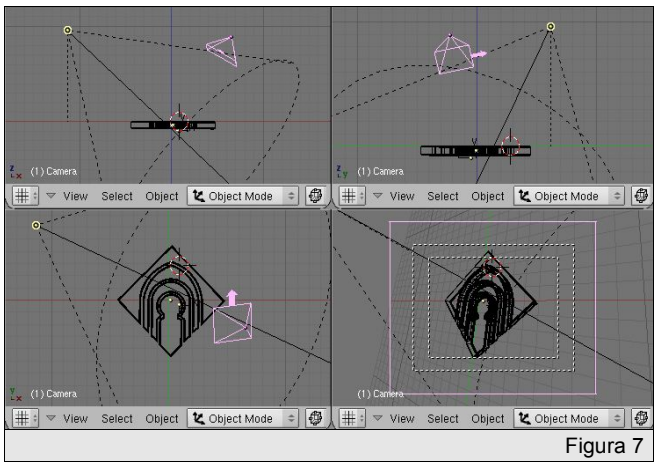


Figura 7

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, insertamos un objeto de tipo *Lamp* a la escena y lo cambiamos a tipo *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 F12. Si el objeto sale muy oscuro, podemos ajustar la intensidad del foco en los botones de foco F4, en el campo *Energy* a un valor superior. Ajustaremos las opciones de renderizado como se indica en la figura 8.

Activamos el botón de *Shadows* para que el motor de renderizado 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 es 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. Podemos cambiar el color de fondo accediendo a las propiedades del mundo y cambiando el color *zenith* (figura 11).

Figura 6

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:UCLMRojo*, cambiamos el color del material en las tres componentes (R,G,B) a 0.300. Además,

Figura 8

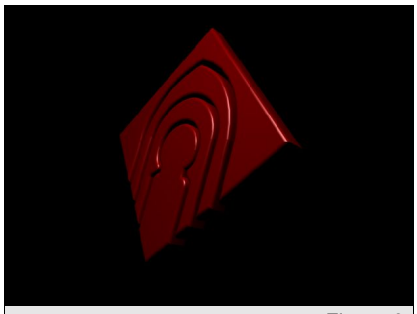


Figura 9

Figura 10

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, guardamos la imagen resultante como *alambre.tga*.

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.

Recordemos que las capas actúan como "acetatos". 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 , y le pondremos como

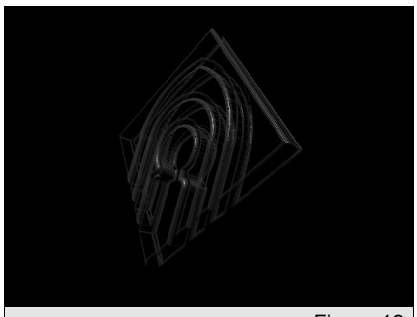


Figura 12

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 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 . Renombramos la capa como "*alambre*".

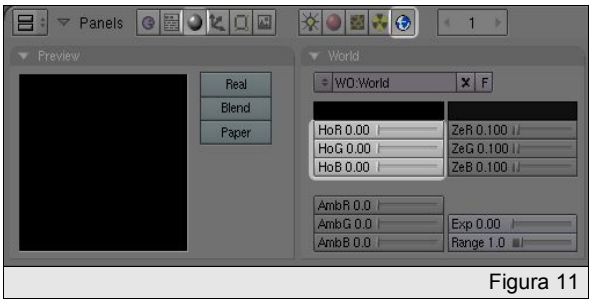


Figura 11

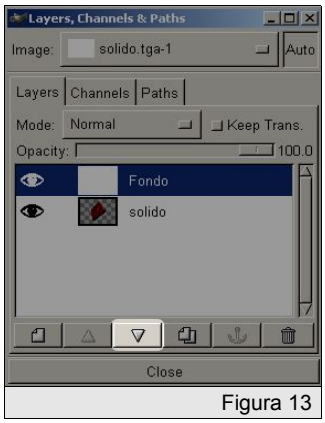


Figura 13

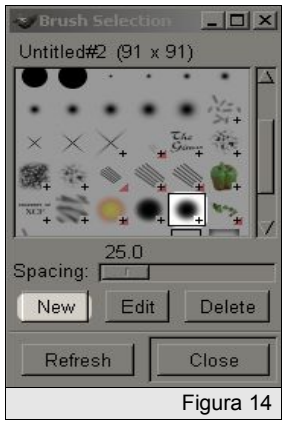


Figura 14

Situamos la capa *alambre* entre la capa *solido* y la capa *fondo*. En la ventana de Selección de brocha, seleccionamos una redonda y pinchamos en *Nuevo*. Con esto, haremos una brocha personalizada partiendo de la que hemos seleccionado.

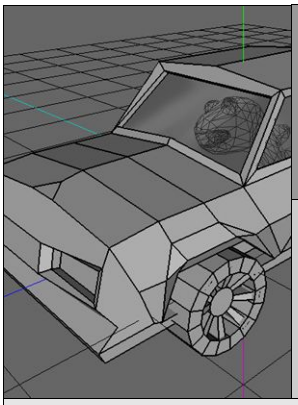
Ajustamos los parámetros a: *radio = 44.8*, *dureza = 0.2* aprox.. *Ángulo* y *razón de aspecto* los dejamos como están. Damos el nombre que queramos a la brocha. Seleccionamos la brocha, la herramienta de borrado , y nos vamos a la capa *solido*. Ajustamos la opacidad de la goma (en la ventana de opciones de herramienta) a 25 aprox. Hecho esto, borramos una parte de la capa sólido (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 Arrojada*. Podemos dejar los parámetros tal y como están y aceptamos.

Vemos que, debajo de la capa alambre se ha creado una capa llamada *Drop-Shadow*. 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 sobre 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. Un formato ampliamente usado es PSD de Adobe Photoshop.



Práctica 3: Modelado con Wings3D

Animación para la Comunicación ::: ESI-CR ::: UCLM

Carlos González Morcillo :: Carlos.Gonzalez@uclm.es :: http://www.inf-cr.uclm.es/www/cglez

El objetivo de esta sesión es aprender a utilizar dos herramientas de modelado que destacan por su facilidad de uso y potencia: Wings 3D y Teddy. Estudiaremos Wings3D con más detalle por ser mucho más versátil.

Software: Wings3D 0.98 + Teddy 1.0 + Blender 2.32 + IOSuite 0.6 **Fecha:** 18/Marzo/2004

En esta práctica no debemos centrarnos en obtener un resultado exactamente igual al mostrado en las imágenes, sino en descubrir las posibilidades que nos brindan Wings3D y Teddy. En próximas sesiones veremos algunas de las capacidades de modelado de Blender (como superficies de subdivisión).

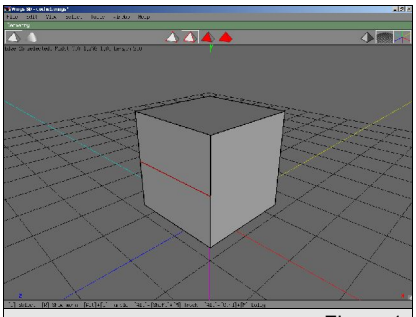


Figura 1

Comenzaremos modelando un coche con Wings 3D. En esta herramienta, partimos de una forma básica (primitiva 3D) y aplicando una serie de operaciones a nivel de vértice, arista, cara o al objeto global, iremos modificándola hasta obtener el modelo final. Antes de empezar a trabajar, personalizaremos el modo de desplazar la cámara por la escena como Blender (para rotar, Ctrl para hacer Zoom...). En el menu superior; *Edit, Camera Mode, Camera Mode [Blender]*.

Añadimos un cubo a la escena con *Cube*. En la parte superior de la pantalla aparece una barra de herramientas. Las dos pirámides de la izquierda nos permiten seleccionar la forma en que se mostrará el objeto (caras o sombreado suave). La parte central nos permite seleccionar en qué nivel queremos trabajar (vértice, arista, cara u objeto). Cuando un elemento está seleccionado (vértice, arista, cara u objeto), aparece en color rojo. Podemos seleccionar de forma incremental elementos de la escena. En cualquier modo de selección que nos encontremos, podemos utilizar las teclas + y - para extender o contraer la selección realizada a los elementos vecinos [realizar la prueba seleccionando un vértice del cubo y pulsar + dos veces y luego -]. Podemos deseleccionar todos los elementos pulsando Spacebar. Wings 3D implementa múltiples niveles de "undo/redo", que podemos recuperar con Ctrl Alt Z y Ctrl Shift Z. Con obtenemos el conjunto de operaciones que podemos aplicar a los elementos seleccionados. Por ejemplo, partiendo del cubo que hemos insertado y seleccionando las dos aristas verticales de la cara frontal, haremos *Connect*. Con esto, obtenemos una nueva arista que conecta las dos anteriores, tal y como se muestra en la figura 1.

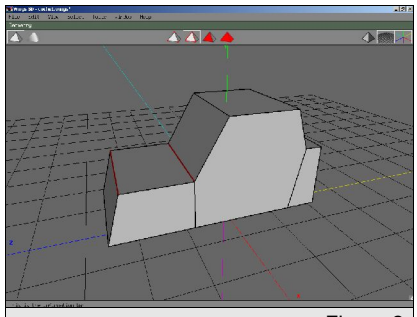


Figura 2

Moveremos la arista recién creada ligeramente hacia abajo (en el eje y), haciendo *Move, Y*. En la cara posterior del cubo, añadiremos otra aristas y la moveremos ligeramente arriba. Seleccionando las caras inferiores, extrusionaremos en la dirección del vector normal de cada cara: *Extrude, Normal*. Ajustaremos las aristas de las caras extrusionadas (desplazamientos en el eje Z), hasta obtener algo similar a la figura 2.

Seleccionaremos las dos aristas longitudinales que forman el capó del coche, y las conectaremos (obteniendo una arista central). Esta nueva arista, la levantaremos ligeramente

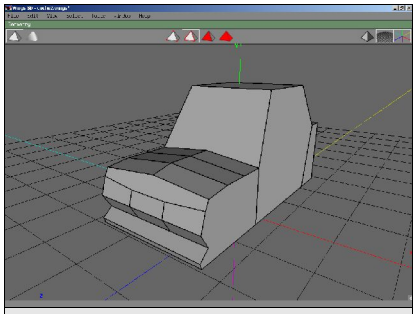


Figura 3

para comenzar a darle forma. Seleccionaremos las tres aristas, y aplicaremos *Cut, 3*. Con esto, obtenemos dos nuevos vértices por arista (las hemos cortado en 3 partes). Utilizaremos ahora (trabajando a nivel de vértice) la orden de *Connect*. Con esto, obtenemos cuatro nuevas aristas que unen los seis vértices creados con la orden anterior. Procederemos de forma similar para modelar el hueco de los faros y el parachoques. Desplazando las aristas obtenidas, deberemos obtener un resultado similar a la figura 3.

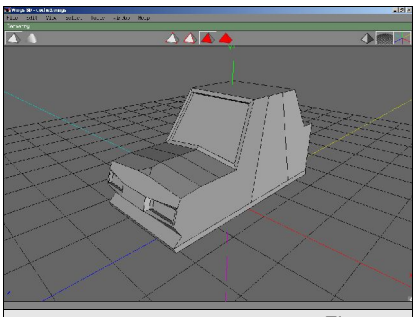


Figura 4

Seleccionaremos las caras que forman la luna delantera y los faros. Aplicaremos *Inset* y a continuación *Extrude, Normal* (hacia el interior). Realizaremos el triángulo de la zona central del capó seleccionando el vértice *Bevel*. El resultado de estas operaciones es el mostrado en la figura 4. Continuaremos dividiendo las caras laterales del coche, ajustan-

do las aristas y vértices resultantes tal y como se muestra en las figuras 5 y 6. En este caso, además, se ha ajustado la altura del techo.

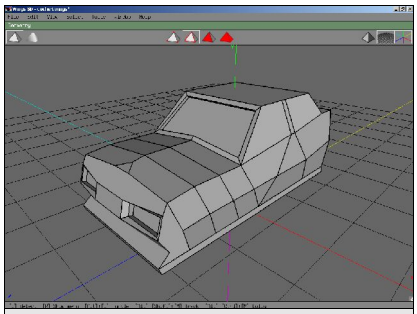


Figura 5

Resulta muy útil la herramienta *Tweak* (*Menú, Tools, Tweak*). Nos permite desplazar los vértices del objeto libremente (cambiando el punto de vista y pinchando-arrastrando con). Saldremos de este modo de edición cuando hagamos . Podemos utilizar el modo "*magnet*" (pulsando), en el que desplazaremos grupos de vértices según el radio de magnetización (se puede variar pulsando y). Debe quedarnos claro que el desplazamiento en este modo es dependiente del punto de vista de la escena. Por tanto, para obtener el resultado esperado, tendremos que realizar multitud de rotaciones de cámara. La figura 7 muestra una captura de edición en este modo.

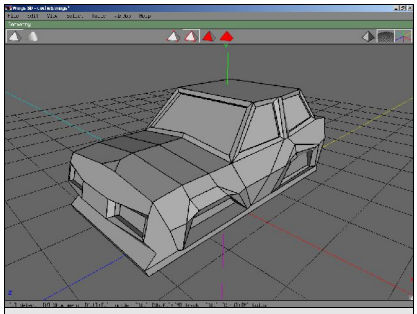


Figura 6

Para modelar las ruedas, primero añadiremos un cilindro a la escena, escalándolo respecto del eje Y. Seleccionaremos las caras superior e inferior, y después *Intrude*. Con esto, obtendremos una especie de "canuto" (ver figura 8). Insertamos otro cilindro, que escalamos para que quede en el interior del anterior, y con la misma altura. Seleccionamos alternativamente las caras laterales y aplicamos extrusión (sin salirnos del cilindro exterior). Para terminar, seleccionamos las caras superior e inferior del cilindro interior, y las desplazamos en dirección a la normal, hacia el centro. El resultado final de la rueda es el mostrado en la figura 8.

Seleccionaremos las dos partes que forman la rueda y *Combine*. De ahora en adelante, Wings 3D tomará la rueda como un único objeto. Hacemos 3 copias más de la rueda (*Duplicate*) y la situamos en el modelo del coche. El modelo del coche finalizado es el que se muestra en la figura 9.

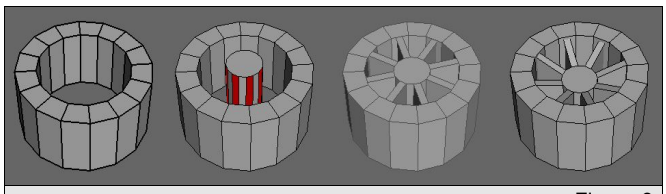


Figura 8

Podemos guardar el modelo para cargarlo desde Blender, en formato Wings3D. Utilizaremos el conjunto de scripts IOSuite para importar modelos en formato *.wings*. Para ello, descomprimimos el archivo *IOSuite2.zip*, dentro del directorio de *scripts* de blender (este directorio se encuentra dentro del directorio *.blender* en el directorio donde se instaló Blender). En sistemas windows, con la instalación por defecto, podemos encontrar el directorio en *"C:\Archivos de programa\ Blender Foundation\ Blender\ .blender\ scripts"*.

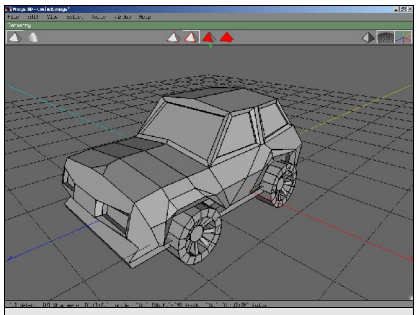


Figura 9

Hecho esto, abrimos blender, dividimos la pantalla y cargamos una ventana de tipo *Text Editor* . Como queremos importar el modelo realizado, cargaremos el fichero *Import.py* (que se encuentra comprimido en *IOSuite1.zip*), y lo ejecutamos con .

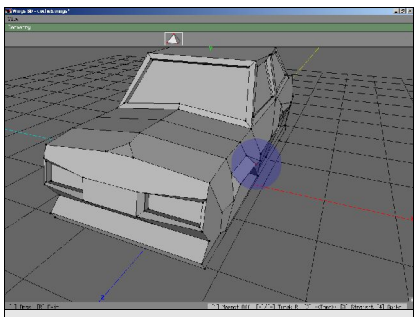


Figura 7

En el navegador de archivos que aparece, seleccionaremos el fichero de Wings3D. El Script identifica el tipo de archivo mediante la extensión. Por tanto, es necesario que la extensión del archivo sea .wings. Entre otros, el Script permite leer y escribir ficheros en formato *.mesh* (de LightFlow), *.cob* (de Caligari Truespace), *.lwo* (de LightWave), *.raw* (malla de triángulos en formato crudo)...

Vamos a modelar el dinosaurio. Abrimos Teddy. Después de leernos el tutorial que acompaña la distribución, haremos un personaje que importaremos en Wings 3D (en formato obj), para salvarlo como .wings (que cargaremos en Blender). Podemos obtener un dinosaurio como el que se muestra en la figura 10. Aplicaremos materiales simples a los objetos y, añadiendo un plano como suelo de la escena, renderizaremos la escena en Blender obteniendo un resultado similar al que se muestra en la figura 11.

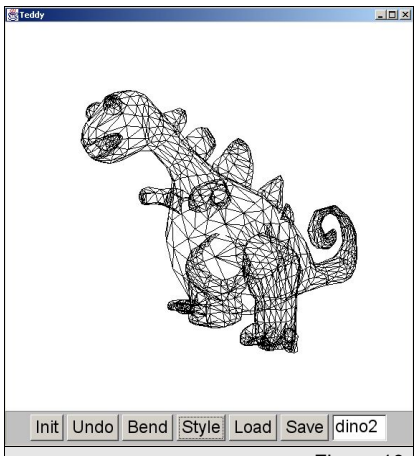


Figura 10

Podemos mejorar el aspecto final del dinosaurio asignando un sombreado suavizado (y no plano), pinchando en el botón *set smooth* del apartado de edición . Además, podemos variar las propiedades de la lámpara para que arroje sombra (cambiando el tipo del foco por defecto a uno de tipo Spot. En la siguiente sesión aprenderemos a asignar texturas mediante UVMapping en Blender.

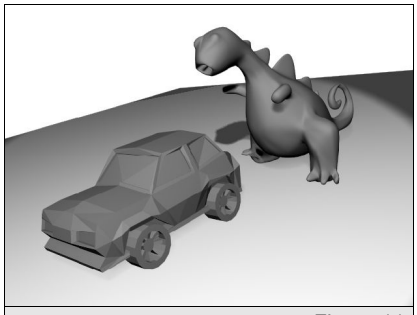
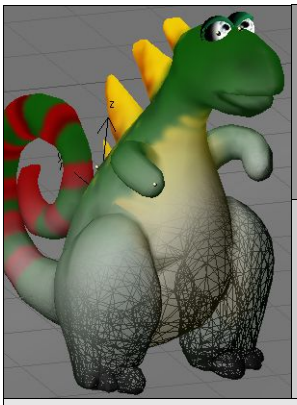


Figura 11



Práctica 4: UV-Mapping en Blender

Animación para la Comunicación ::: ESI-CR ::: UCLM

Carlos González Morcillo :: Carlos.Gonzalez@uclm.es :: http://www.inf-cr.uclm.es/www/cglez

En esta sesión aprenderemos a texturizar objetos mediante UV-Mapping en Blender, para poder crear personajes completos y utilizarlos en nuestros proyectos de animación.

Software: Wings3D 0.98 + Teddy 1.0 + Blender 2.32 + IOSuite 0.6 **Fecha:** 25/Marzo/2004

El mapeado mediante coordenadas UV permite un posicionamiento de "la piel del objeto" muy preciso. El objetivo es texturizar un modelo complejo creado con cualquier técnica de modelado.

Partimos del modelo creado en Teddy mostrado en la figura 1. Sólo nos hemos ocupado de construir la mitad del personaje. La otra mitad la conseguiremos por "espejo" del este modelo en Wings 3D. Podéis encontrar el modelo en el fichero *dino.obj*, no obstante es aconsejable que intentéis construir vuestro propio personaje similar al mostrado en la figura 1.

Importamos el modelo en Wings. Activamos la vista de los ejes y el modo ortogonal en la parte superior derecha de la ventana. El modo ortogonal nos elimina el efecto de perspectiva. Esto nos permite seleccionar de forma más precisa conjuntos de caras y trabajar sin distorsiones en la vista 3D.

Pulsando las teclas X, Y, Z veremos el modelo desde estos ejes. Rotaremos y desplazaremos el modelo para que se quede perfectamente centrado respecto del plano YZ, tal y como se muestra en las figuras 2 y 3. Es importante situar correctamente el modelo en la operación anterior porque nos facilitará la siguiente selección de caras.

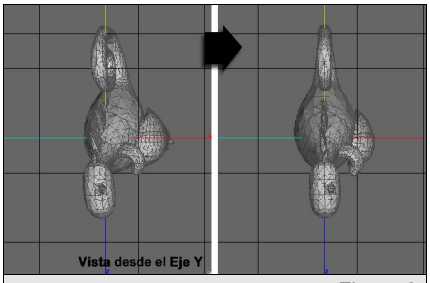


Figura 2

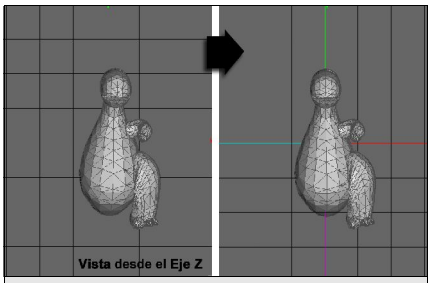


Figura 3

Con la vista desde el eje Z, y el modo de selección a nivel de cara, seleccionaremos la mitad izquierda del modelo pinchando y arrastrando desde la posición *A* de la figura 4, a la posición *B*. Obtendremos una selección de caras bastante precisa, pero habrá que rotar la cámara y ajustar la selección manualmente, sobre todo en la zona de las escamas de la espalda y en la cola. Eliminaremos las caras sobrantes y borraremos (tecla Del) la selección. El modelo debe quedarnos como muestra la figura 5. Para evitarnos efectos indeseables con la herramienta de espejo, aplanaremos primero la cara resultante (aún seleccionada en color rojo) mediante *Flatten, Normal* (alisar la cara respecto de su vector normal). Hecho esto, utilizamos la herramienta de espejo con *Mirror* para obtener el modelo del dinosaurio simétrico. Guardamos el modelo como .wings y lo importamos en blender siguiendo los pasos explicados en la práctica anterior.

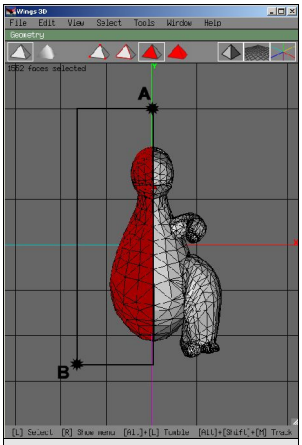


Figura 4

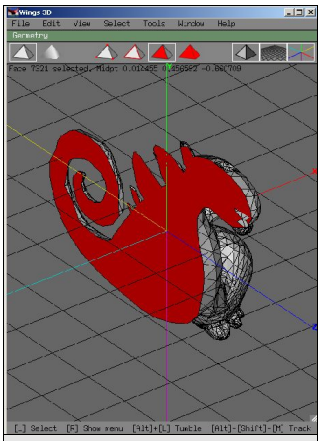


Figura 5

Configuramos Blender para obtener una división de ventanas como se muestra en la figura 6. La ventana de la derecha la cambiamos a tipo ImageWindow (Shift F10, o seleccionando el icono). Antes de continuar, veamos algunas opciones del entorno que nos van a ser útiles en esta práctica. En la cabecera de la ventana 3D, distinguimos los iconos que se muestran en la figura 7 y algunas teclas útiles. Pulsando la tecla 5 del teclado numérico, podemos seleccionar el modo de visualización, en perspectiva o vista ortográfica. La vista or-

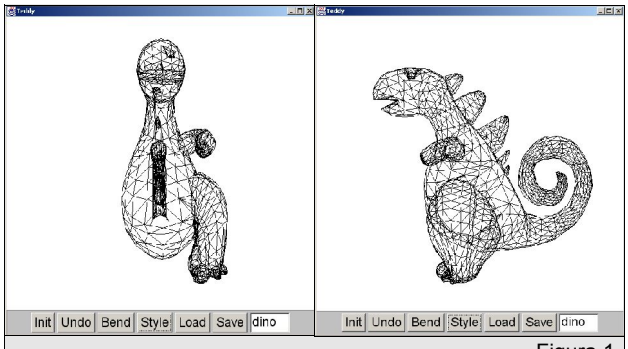


Figura 1

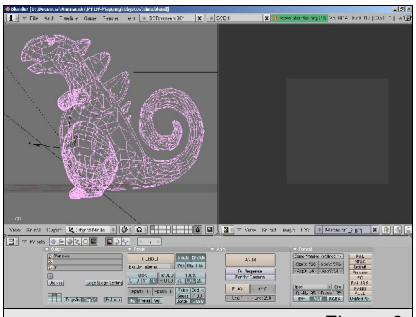


Figura 6

tográfica no agranda los objetos cuanto más cerca están de la cámara. La lista desplegable de iconos de la derecha de la figura 7, nos permite seleccionar el modo de sombreado del objeto mientras trabajamos con él. Nos interesará el modo *Textured*, que nos muestra la textura aplicada al modelo, y el modo *Wireframe* (la representación de alambre que hemos utilizado en prácticas anteriores). El conjunto de iconos de la izquierda nos permite especificar

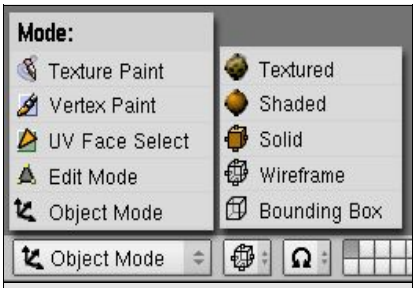


Figura 7

el modo de trabajo con el modelo. El modo de trabajo con caras (*UV Face Select*) se puede alcanzar también pulsando la tecla F. El modo de edición de vértices, como hemos visto en prácticas anteriores, se consigue pulsando la tecla Tab. Con *Vertex Paint* pasamos al modo de pintado de vértices y *Texture Paint* nos permite pintar la textura directamente sobre el modelo. Por último, el botón situado en la cabecera de la vista 3D (no confundir con un botón similar de la ventana de botones) sirve para renderizar la vista actual. Con R podremos salvarla en el formato que hayamos seleccionado en el menú de visualización F9.

Pasamos al modo de selección de caras (pulsando la tecla F). Vemos que el modelo se ha puesto de color blanco, y el icono correspondiente de la figura 7 aparece seleccionado. Seleccionaremos las caras que forman el cuerpo del dinosaurio, dejando las extremidades para más adelante. Para realizar la selección de una forma más sencilla, y sin salir del modo de selección de caras, pulsaremos Tab. Todos los vértices aparecerán seleccionados (en color amarillo), y de igual forma las caras asociadas.
Pulsando dos veces la tecla B, activaremos el modo de selección de vértices mediante un "pincel". El cursor aparecerá delimitado por una circunferencia. Podremos añadir caras a la

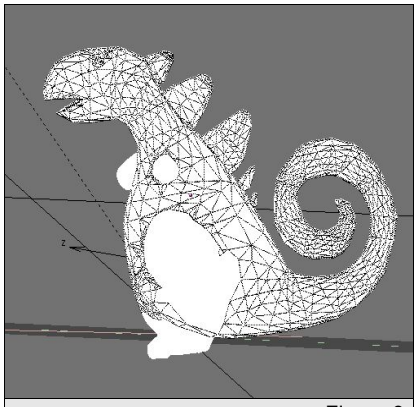


Figura 8

selección pinchando y arrastrando . Para quitar caras a la selección, pinchar y arrastrar con y Alt pulsado. Podemos aumentar el radio del pincel con la tecla +, y disminuirlo con la tecla -. Abandonaremos el modo de selección con esta técnica haciendo . Teniendo en cuenta que la selección de caras "atraviesa" el modelo, seleccionaremos las vistas que nos hagan falta para dejar inactivas las caras de las extremidades del dinosaurio. Podemos volver al modo de edición de caras, pulsando Tab. El resultado de la selección, debe ser el mostrado en la figura 8.

Las extremidades se quedan totalmente blancas porque no hay selección de caras. No pasa nada si no hemos seleccionado perfectamente el recorte de las extremidades, ya que las podremos seleccionar cuando

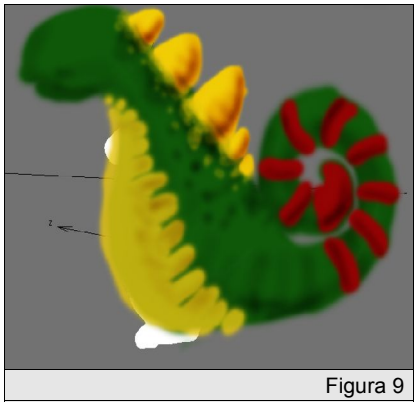


Figura 9

"pintemos" las extremidades.
Rotamos el modelo en la vista 3D hasta obtener una posición totalmente lateral, de forma que coincidan las aristas de forma simétrica (tal y como muestra la figura 8). Para ello, tendremos que activar la vista ortogonal. Hecho esto, renderizamos la vista con el botón de la ventana 3D y salvamos con el nombre de *lateral.jpg*. Cargamos el fichero en GIMP y creamos una nueva capa. Pintaremos sobre esta capa, con un color verde la piel del dinosaurio.

Nos "saldremos" a propósito fuera del área a colorear, ya que esto nos facilitará el situar la textura más adelante. Crearemos una nueva capa donde dibujaremos de color amarillo los huesos que salen de la espalda. Añadiremos detalles más oscuros jugando con la transparencia del pincel. Puede resultar útil variar la transparencia de la capa para tener una visión general de la zona del dinosaurio sobre la que estamos pintando. Cuando hayamos obtenido algo similar a la figura 9, salvaremos el fichero como *lateral_pintado.jpg*.

Cargamos este fichero en la ventana de imágenes de Blender. Pinchamos en el menú *Image, Open* de la ventana de imagen. La imagen aparecerá como fondo de la ventana. Podemos maximizar esta vista (y volver a su estado original) mediante *View, Maximize Window* y *View, Tile Window,* o pulsando Ctrl↓.

Importante: Cuando trabajemos con ficheros añadidos a Blender (como imágenes, sonidos, etc...), hay que indicar a Blender que guarde estos datos dentro del fichero .blend. De otra forma, cuando abramos el fichero en otro ordenador, las texturas no estarán. Esto se realiza en el menú *File, Pack data*. Hecho esto, aparecerá un icono en la barra de menús superior (al lado de la opción *Help*). Esta operación sólo hay que realizarla una vez. De ahora en adelante, cuando guardemos el fichero .blend, se guardarán también las imágenes asociadas.

En la ventana 3D, con las caras del cuerpo seleccionadas, y la vista que se indicó en la figura 8, pulsamos la tecla U, y seleccionamos *UV Calculation, From Window*. Vemos que el conjunto de caras seleccionadas se han "copiado" en la ventana de la imagen. Tendremos que ajustar los vértices para que se cuadren bien a la textura. Pero antes, activaremos la vista detallada (*Textured* pinchando en el icono correspondiente visto en la figura 7*)*. El modelo muestra una representación con la textura aplicada, como se muestra en la figura 10.

Maximizamos la ventana derecha, y ajustamos los vértices. Para ello, los seleccionamos todos A. Podemos escalar el modelo con S. Recordemos que para limitar el escalado en algún eje, pulsaremos Alt cuando proceda-

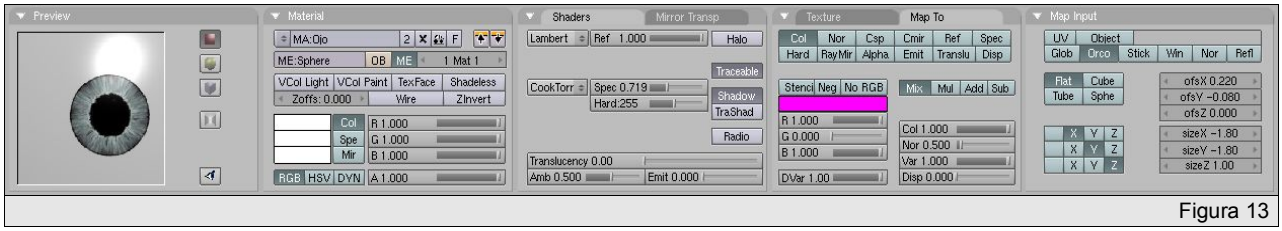


Figura 13

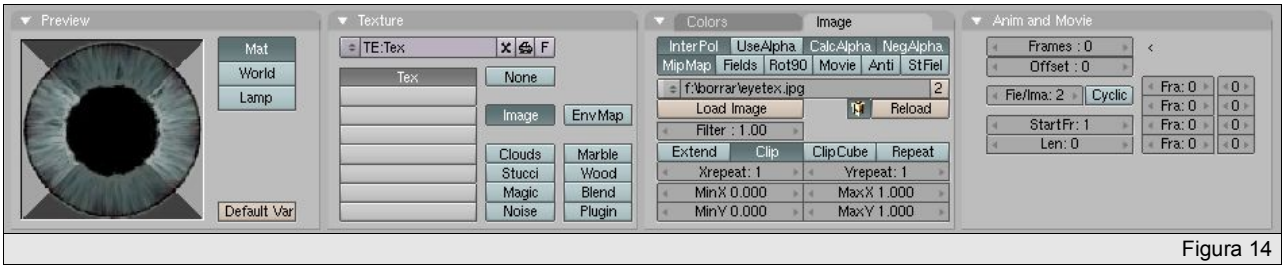


Figura 14

mos al escalado. También podemos rotar los vértices con R. Cuando los vértices se ajusten perfectamente a la textura, minimizaremos de nuevo la ventana de imagen y deberíamos obtener una representación similar a la figura 11.

Procedemos de forma similar con las extremidades. Nos aseguraremos de que, al seleccionar las caras, no nos dejemos ninguna que no seleccionáramos en el paso anterior. Situamos el modelo frontalmente y renderizamos la vista. Coloreamos en GIMP obteniendo una textura similar a la figura 12. Aplicamos la textura frontalmente.

Añadimos una esfera para crear un ojo. Ajustamos el tamaño, la situamos y le aplicamos un material F5. El ajuste de materiales en Blender será visto en detalle en próximas sesiones. De cualquier forma, y para terminar este modelo, daremos el recetario para texturizar el ojo. A este material le asociaremos una textura (F6 *Add New, Image,* Pestaña *Image, Load Image*). Activamos las opciones de Texturas y Material, tal y como se muestra en las figuras 13 y 14.

Para situar con más exactitud el ojo con la textura aplicada, activaremos la vista detallada y lo rotaremos hasta conseguir la posición deseada.

Para terminar, añadiremos un material genérico al modelo del dinosaurio, activando el botón de *TexFace*. También activaremos el botón de *VcolPaint* y *VcolLight*. Si entramos en el modo de pintado de vértices, en el menú "Paint Buttons" , podremos añadir sombras adicionales al modelo (como en los dedos del modelo, y en las caras de unión entre las extremidades y el cuerpo...). Para ello, deberemos activar el icono (g) y el modo de sombreado detallado. En el menú de Vertex Paint, dentro de los botones de edición F9, podremos elegir el color con el que pintaremos los vértices del modelo (dentro de la pestaña *Paint*). En zonas donde queremos que el sombreado sea suave, es aconsejable utilizar el pincel de pintado de vértices con el botón *Soft* activado. Mediante el parámetro *Size* especificamos el grosor del pincel, y con *Opacity* la cantidad de color a aplicar en cada pasada.

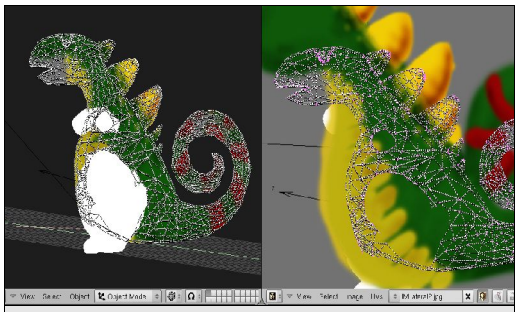


Figura 10

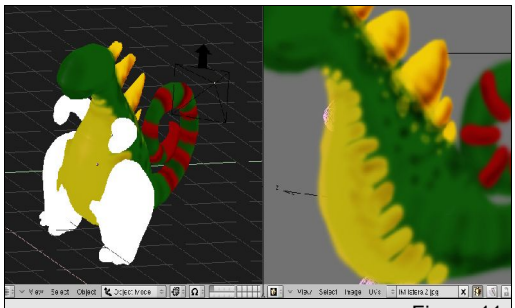


Figura 11

Concluiremos la práctica realizando un render sencillo del modelo. Podéis encontrar un entorno donde probar el modelo en el fichero "*escenariorender.blend*". En este escenario está definido un plano (que servirá de suelo) y un par de focos con una configuración básica.

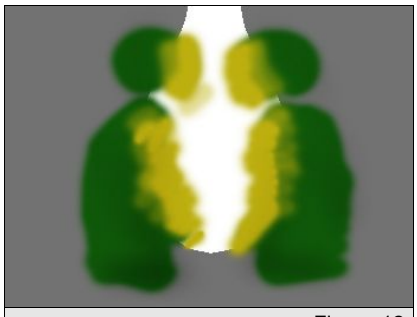
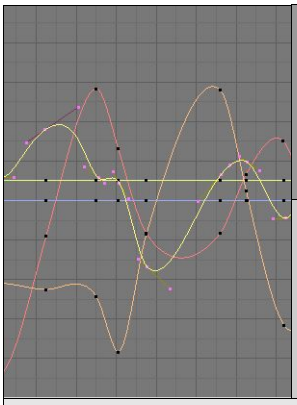


Figura 12



Práctica 5: Animación Básica

Animación para la Comunicación ::: ESI-CR ::: UCLM

Carlos González Morcillo :: Carlos.Gonzalez@uclm.es :: http://www.inf-cr.uclm.es/www/cglez

En esta sesión utilizaremos las curvas IPO en Blender. Este tipo de curvas, basadas en representación de Bezier es la empleada por Blender para el manejo de trayectorias en animación.

Software: Blender 2.32. **Fecha:** 01/Abril/2004

Vamos a comenzar realizando una animación muy sencilla, manipulando directamente las curvas de animación. Cargamos en Blender el fichero "*dino.blend*" . Seleccionamos el modelo del dinosaurio y vamos a intentar rotarlo respecto del eje Z. Como se puede apreciar, hay un problema con los ojos y los párpados, que no siguen el movimiento del cuerpo. Hay que indicar a Blender la jerarquía existente entre estos objetos; el cuerpo del dinosaurio será el "*objeto padre*" de los ojos y los párpados. Para indicar esto, seleccionamos primero un elemento hijo (por ejemplo, un párpado), y con Shift pulsado, seleccionamos después el cuerpo. Hecho esto, hacemos Ctrl P, **Make Parent**. Aparece una línea punteada indicando la jerarquía, tal y como se ve en la figura 1. Procedemos de igual forma con el otro párpado y las esferas de los ojos.

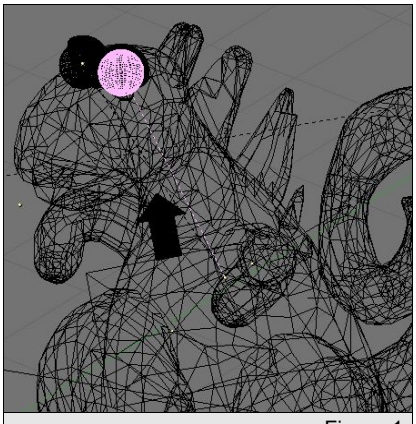


Figura 1

Ahora, si rotamos el cuerpo del dinosaurio, se rotarán con él los ojos y los párpados, pero no al contrario (los padres no heredan posición de los hijos). Vamos a realizar la animación que se puede ver en el fichero "*anim_dino.mpg*".

Rotaremos el dinosaurio respecto del eje Z 360º, intentando que el vídeo pueda repetirse indefinidamente (sin cambios de velocidad al empezar y al finalizar la animación).

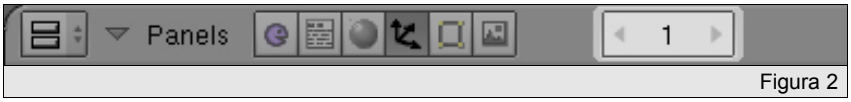


Figura 2

Las curvas de movimiento en Blender nos permiten representar la posición, rotación y tamaño de cada componente (X,Y,Z) de un objeto a lo largo del tiempo. Recordemos que, en animación, el tiempo lo medimos en frames por segundo. El número de frame actual en Blender se representa con un botón, en la parte central de la cabecera principal, como se muestra en la figura 2. Podemos avanzar el número de frame de 1 en 1 con , y de los cursores y con y , de 10 en 10 frames.

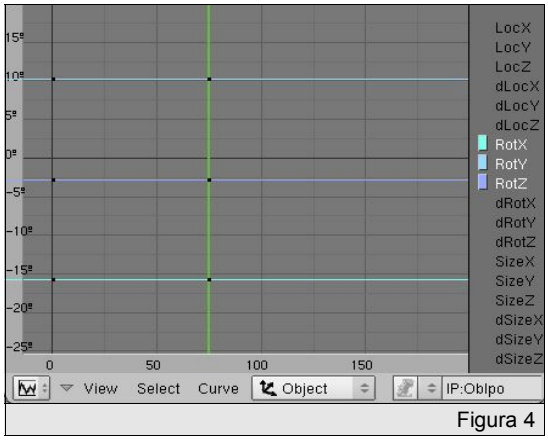


Figura 4

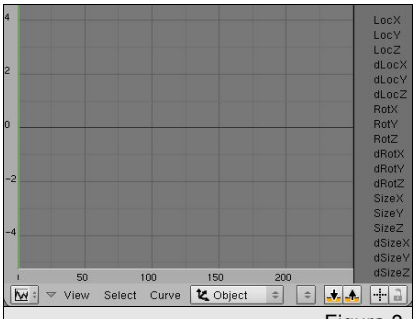


Figura 3

Nos situamos en el primer frame de la animación. Cambiamos una ventana al tipo "*Editor de curvas IPO*" . Aparece una imagen como la mostrada en la figura 3. En el eje de abcisas 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 (Localización, Rotaciones...). La línea verde vertical (que podemos arrastrar para ver la evolución de nuestra animación), representa el frame actual. Volveremos a trabajar sobre las curvas IPO en próximas sesiones.

Con el cuerpo del dinosaurio seleccionado, definiremos el frame 1

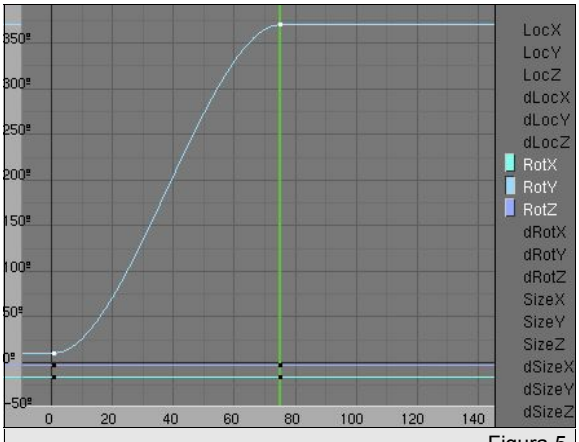


Figura 5

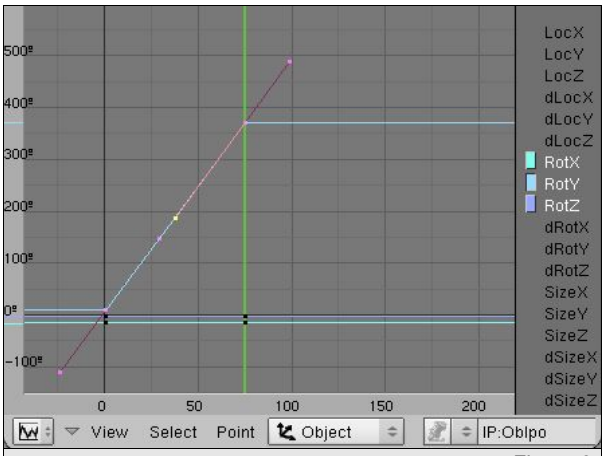


Figura 6

como clave (pulsando la tecla I). Aparece un menú preguntándonos qué tipo de información vamos a manejar en nuestro frame clave. Indicaremos que sólo vamos a variar la rotación (**Rot**).

Como queremos que la animación dure 3 segundos (a 25 fps), definiremos otro frame clave en el 75. Al igual que antes, también de tipo Rot. Obtendremos una imagen como se muestra en la figura 4. En este tipo de ventana podemos hacer zoom y desplazarla de igual forma que en una ventana 3D. Si al hacer zoom o desplazar nos "*perdemos*", podemos dejar que Blender ajuste la vista pulsando la tecla Home.
Seleccionamos la curva IPO llamada RotY, pasamos al modo de edición de objeto Tab y ajustamos los puntos de control para obtener una curva similar a la mostrada en la figura 5. Naturalmente, en el eje Y tenemos que llegar al valor 360 (giro completo en el eje Y). Nos ayudaremos del Zoom para situar el valor exactamente en 360.
Podemos ver una previsualización de la animación, si nos situamos en el frame 1 y pulsamos Alt A en cualquier vista 3D. Paramos la animación con Esc. Al principio y al final de la animación la rotación es más lenta que en el medio. Como queremos que la animación se pueda repetir en un bucle sin que se noten los cortes, ajustaremos la pendiente de la curva IPO en los puntos extremos, desplazando los puntos de tangente. Debemos obtener un resultado como el que se muestra en la figura 6.

Si previsualizamos la animación, comprobamos que ahora el resultado es el que esperamos. Vamos a generar el vídeo AVI. Disminuimos la resolución de salida a **320x240**. Elegimos como formato de salida **AVI Jpeg**. Ajustamos el rango de frames que queremos renderizar con los botones **Sta: 1** y **End: 75**. El directorio donde se va a generar el vídeo por defecto es **/render** (indicado en el apartado **Output**, arriba a la izquierda, ver figura 7). Hecho esto, pinchamos en **ANIM** para iniciar el proceso y esperamos un rato :-).

En la segunda parte de la práctica, también emplearemos curvas IPO, pero no generando el movimiento directamente sobre ellas, sino utilizándolas únicamente para dar los ajustes finales. Cargamos el fichero "*oreto.blend*". Moveremos la cámara por la escena, insertando frames clave en los puntos donde haya que realizar un cambio de dirección. Blender se encargará de interpolar las posiciones intermedias. El vídeo que debemos obtener como resultado (*anim_oreto.mpg*) muestra el proceso final (con post-producción, que veremos en prácticas sucesivas). Debemos ocuparnos de obtener un movimiento de cámara similar. Nos centraremos en colocar la cámara en las posiciones clave (respecto de la localización y la rotación). Para facilitar que la cámara apunte siempre al logotipo, añadiremos un objeto de tipo **Empty** en el centro del logotipo (Spacebar, **Add**, **Empty**). Seleccionamos la cámara primero y luego con Shift pulsado, el Empty recién creado. Pulsamos Ctrl T **Make Track**, **Old Track**. Corregimos el cambio de orientación con Alt R **Clear Rotation**.

En la figura 8 tenemos las curvas IPO asociadas a la cámara. Podemos ver las posiciones donde se han insertado frames clave (en el 1, 25, 50, 61, 75, ..., 165). Al igual que en la primera parte, obtenemos una previsualización de la animación con Alt A.Si los resultados son correctos, renderizamos con los parámetros vistos anteriormente.

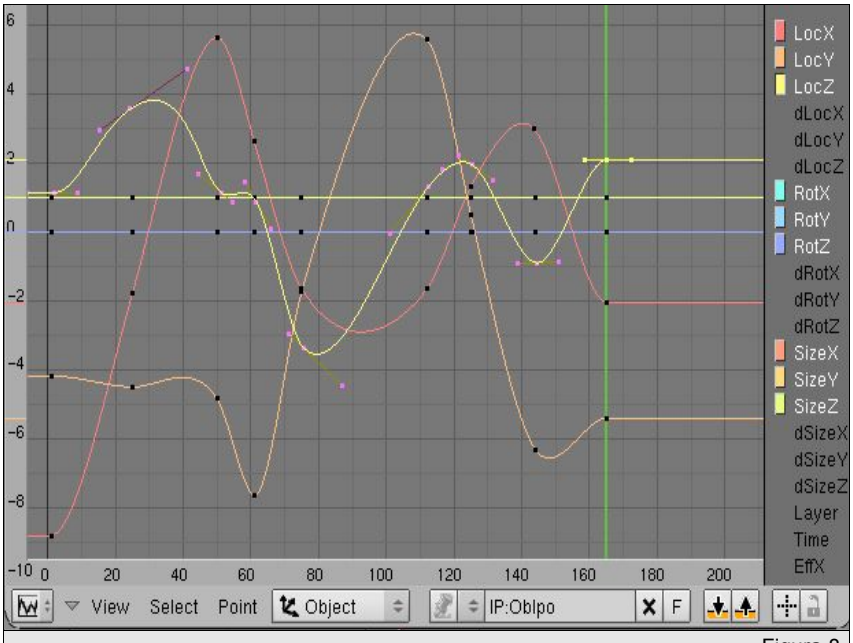
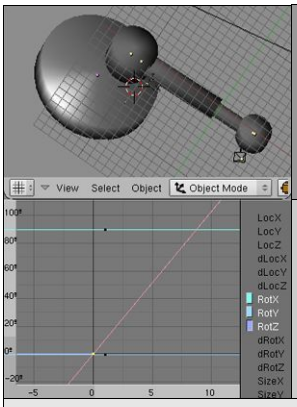


Figura 8

Figura 7



Práctica 6: Animación Jerárquica

Animación para la Comunicación ::: ESI-CR ::: UCLM

Carlos González Morcillo :: Carlos.Gonzalez@uclm.es :: http://www.inf-cr.uclm.es/www/cglez

En esta sesión aprenderemos los conceptos básicos de animación jerárquica en Blender, y realizaremos una composición sencilla de video. Para esto, utilizaremos el compositor de vídeo interno de Blender.

Software: Blender 2.32. **Fecha:** 15/Abril/2004

Comenzaremos modelando el objeto que se muestra en la figura 1. Todos los componentes del modelo deben ser independientes (añadiremos a la escena 5 cilindros, rotando y escalando hasta conseguir la configuración mostrada).

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 teclearemos el nombre en la sección **OB:** de los botones de edición accesibles también con F7.

Podemos indicar a Blender que muestre el nombre de los objetos si activamos, con el objeto seleccionado, el botón **Name** del grupo de propiedades del objeto F7. También se puede ser útil en ciertas ocasiones mostrar el sistema de coordenadas local de cada objeto, accesible mediante el botón **Axis** del mismo grupo de botones (ver figura 3).

Figura 3

Establecemos las relaciones de jerarquía entre PistonA1 - PistonA2 y PistonB1 - PistonB2.

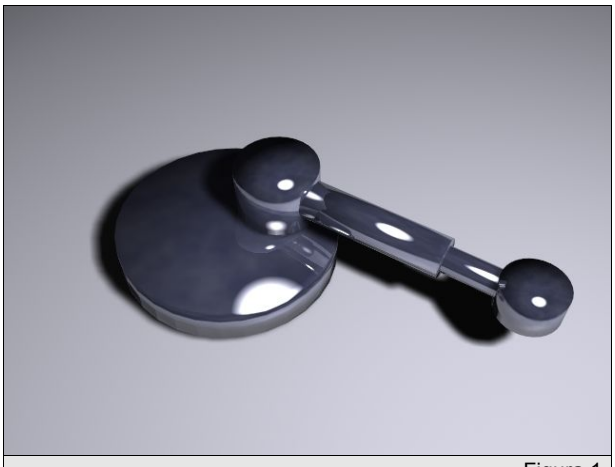


Figura 1

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 pulsado, seleccionamos el padre (PistonA1). Pulsamos Ctrl P y seleccionamos **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.

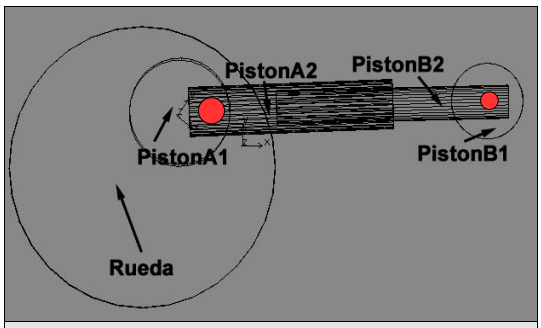


Figura 2

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 (círculo de color rojo que movemos con), a la posición 3D donde queramos situar el nuevo centro y en los botones de edición (F9) 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. Asociaremos el movimiento de los dos pistones, haciendo que se apunten mutuamente. Esta técnica ya la hemos utilizado en sesiones anteriores, cuando incluíamos un objeto *Empty* para situar la cámara. El primer objeto que seleccionamos es el "*dependiente*" (que seguirá el movimiento del segundo; el "*principal*"). Seleccionamos primero PistonA1 y luego (con Shift pulsado) PistonB1 (en este orden) y creamos el track (Ctrl T) y seleccionamos **Old Track**. Repetimos la operación, pero ahora seleccionando primero PistonB1 y luego PistonA1. En caso de que se roten los objetos de forma errónea, pulsamos Alt R y ajustamos de nuevo la rotación de las bases: PistonA1 y PistonB1.

Probamos a mover PistonA1. Si tenemos que el movimiento lo siguen perfectamente el resto de piezas, hemos terminado este paso. En caso contrario, tendremos que ajustar en los botones de propiedades del objeto animación (F8) el eje con el que vamos a apuntar al otro objeto (al otro piston). Según la configuración de la escena, es posible que tenga-

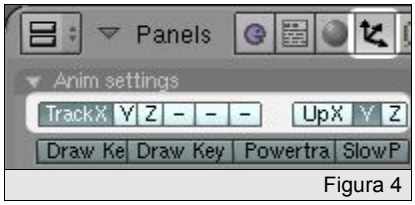


Figura 4

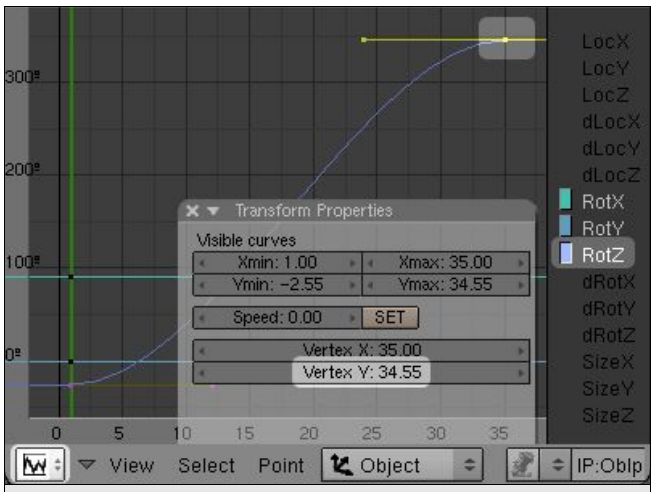


Figura 5

mos que cambiar de **Track Y** a **Track X**. Podría hacer falta cambiar también el eje que apunta "arriba" (vector **Up**), en un conjunto de botones situados a la derecha del grupo de *Track*. Es muy útil activar la visualización de los ejes locales de cada objeto como hemos visto anteriormente.

Activaremos el botón de **PowerTrack** (en las propiedades de animación F7) al pistón A1, para quitarle la rotación (cuando el objeto "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 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 360º de la rueda. Procederemos de forma similar a como se explicó en la práctica anterior con el dinosaurio. 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.

Seleccionamos el último clave y añadimos 36 al valor de **VertexY** de la gráfica **RotZ**, tendremos un giro de 360º. Podemos especificar un valor numérico si pulsamos la tecla N (tal y como muestra la figura 5). En el ejemplo de la figura, el objeto *rueda* partía de una rotación de -2.55 grados; así, el valor de rotación final es de 34.55.

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.

Pasamos a renderizar 2 animaciones con distintos materiales (resolución de 320 x 240). Añadimos un plano debajo de la escena, para que sirva como "suelo" y recoja las sombras arrojadas. Por ejemplo, podemos hacer un render utilizando la técnica "*Toon Shading*"; con el aspecto de los dibujos animados. Para ello, en el material F5 activamos la opción **shadeless**, para que no arroje sombras. En la ventana de render F10, activamos el botón **Edge**, y en Edge Settings al parámetro **Eint** (Edge Intensity; *Intensidad de la arista*), le damos un valor alto (100).

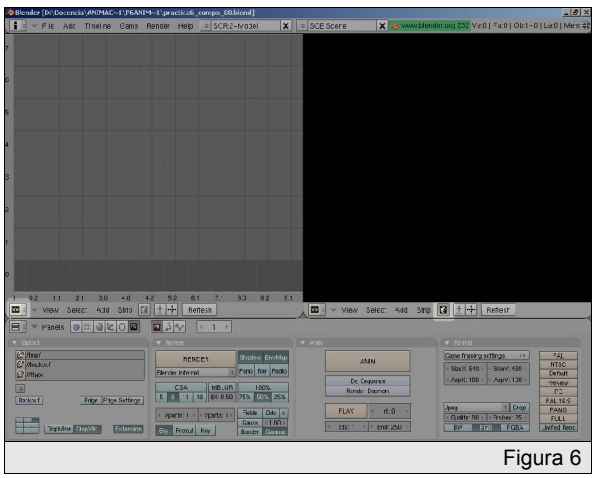


Figura 6

En definitiva, hemos obtenido distintos vídeos con la animación de nuestro pistón. Pasamos al siguiente paso; componerla en Blender. Para ello, abrimos una nueva sesión de Blender y dividimos la pantalla en 2 partes. Cambiamos las dos ventanas a tipo Editor de Secuencia de Vídeo. En una de ellas, activamos el botón para mostrar la previsualización de la composición que estamos realizando. Debemos obtener una representación como muestra la figura 6.

En la parte izquierda de la pantalla iremos añadiendo los vídeos, imágenes, transiciones y efectos. Para añadir un vídeo o imagen (en el vídeo de ejemplo se ha utilizado una para el texto), pulsamos Shift A. Podemos duplicar cualquier elemento que aparezca en esta ventana pulsando Alt D. Pinchando en los triángulos izquierdo y derecho de un vídeo, se puede "escalar". Añadimos los 2 vídeos que hemos realizado, los colocamos uno encima del otro, tal y como muestra la figura 7.

Con los dos videos seleccionados (pinchamos con sobre uno y luego con Shift seleccionamos el otro), podemos añadir un efecto (de nuevo, haciendo Shift A). Podemos experimentar con todas las transiciones que incorpora blender, e incluso descargar plugins para añadir nuevas funcionalidades a la aplicación.

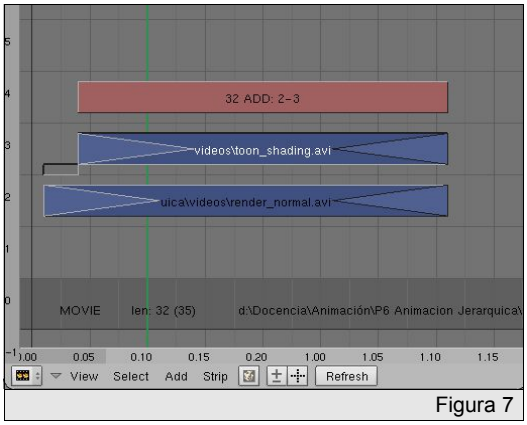
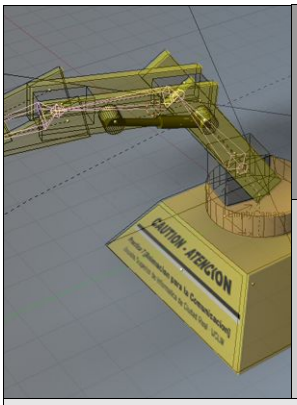


Figura 7

Con un efecto seleccionado, si pulsamos C podemos cambiar el tipo de efecto, o el sentido de aplicación del efecto (A→B o B→A). En algunos efectos, como las transiciones Cross, GammaCross, etc... el orden importa. Con N podemos renombrar los elementos de nuestra composición. Una vez que tenemos un bloque de nuestra animación definido, si seleccionamos todos los elementos que intervienen (vídeos, imágenes, efectos, etc...) y pulsamos la tecla M, podemos agruparlos en una única Metatransición. Con Alt M podemos deshacer este paso.

Una vez tenemos la secuencia creada, la generaremos en un AVI, activando el el botón **do sequence** de los botones de renderizado accesibles pulsando F10 y pinchando luego en **Anim**. Utilizando las herramientas descritas en esta práctica, intentar conseguir un resultado similar al que se puede ver en el fichero *resultado_p6.mpg*.



Práctica 7: Esqueletos - IKs

Animación para la Comunicación ::: ESI-CR ::: UCLM

Carlos González Morcillo :: Carlos.Gonzalez@uclm.es :: http://www.inf-cr.uclm.es/www/cglez

Con esta práctica nos introduciremos en el uso de esqueletos (armatures) en Blender. Partiendo de un modelo sencillo (un brazo robótico), aprenderemos a asociarle, a cada sección, un hueso del esqueleto.

Software: Blender 2.32. **Fecha:** 22/Abril/2004

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.

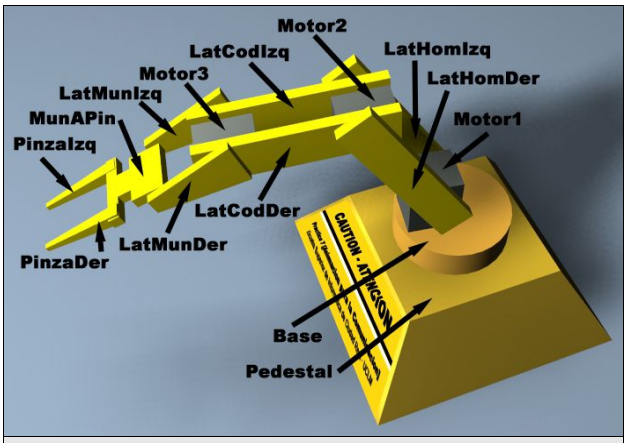


Figura 1

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.

Una vez modelado el robot, y renombradas las piezas, añadiremos el esqueleto. Para ello, pulsaremos Spacebar, **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.

Una vez creado el esqueleto, entramos en el modo de edición del objeto (del esqueleto) con Tab. El color rosa del esqueleto seleccionado será más intenso. Seleccionamos todos los huesos del modelo con A, cambiando su color a amarillo. Vamos a los botones de edición 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.

Todo el proceso que vamos a seguir a continuación, está resumido en el esquema de la figura 5. Las relaciones de parentesco ("*es padre de*"), están representadas por una flecha que va del padre al hijo. Los huesos del esqueleto están representados en color azul. Los nodos hijo de la jerarquía, están representados por un rectángulo con las esquinas sin redondear.

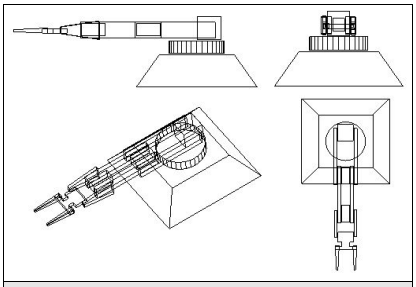


Figura 2

Los elementos auxiliares (*empty*) que se han utilizado, aparecen en color naranja. Por último, las restricciones entre elementos se representan con una flecha punteada, con una circunferencia negra en el origen.

Importante: Recordemos que para establecer una relación de parentesco entre elementos, seleccionaremos siempre **primero** el elemento **hijo**, y **después**, con Shift pulsado, el **padre**. Pulsaremos Ctrl+P **Make Parent**. En caso de equivocarnos, podremos eliminar el parentesco, seleccionando de nuevo los elementos que intervienen en la relación, con Alt+P.

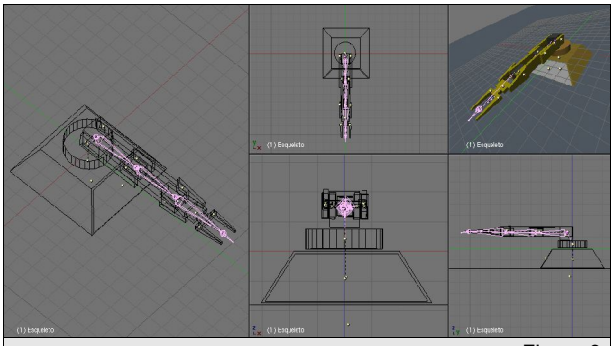


Figura 3

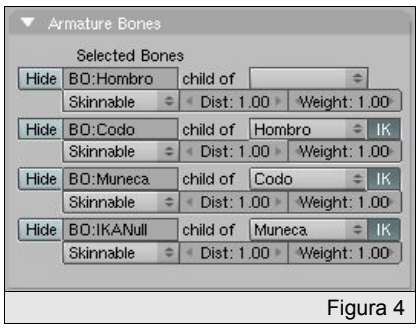


Figura 4

De esta forma, seleccionaremos, por ejemplo, primero el elemento *Base*, y después *Pedestal* y realizaremos la jerarquía. Procederemos de igual modo con *Motor1* y *Base*, el *Esqueleto* (completo) y la *Base*, y *MunAPin* (Muñeca a Pinza) con las dos partes de la pinza.

Las relaciones de parentesco con los huesos se realizan de forma similar. Primero seleccionamos el hijo, y después el esqueleto (completo). Pulsamos Ctrl P **Use Bone** y seleccionamos el hueso que corresponda en cada caso.

Añadiremos un objeto Empty a la escena que nombraremos *EmptyIk*. Este objeto nos servirá para situar el extremo del robot y, por cinemática inversa, Blender calculará la rotación necesaria para cada articulación del robot. Situaremos el puntero 3D cerca del hueso *IkaNull* y situaremos ahí el nuevo objeto *Empty*. Entraremos en el **modo de pose** del esqueleto (que veremos con más profundidad en próximas sesiones), que entre otras cosas nos permite seleccionar individualmente cada hueso. Para cambiar a modo de pose, con el esqueleto seleccionado (en color rosa claro), pulsaremos Ctrl Tab. También podemos acceder a este modo pinchando en la lista desplegable de la cabecera de la ventana 3D (figura 6). El esqueleto pasará a color azul claro.

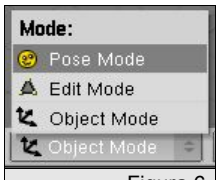


Figura 6

En este modo, seleccionaremos el último hueso (*IkaNull*), que pasará a color verde claro. Vamos al menú de restricciones, pinchando en el botón de edición de objeto. Añadimos una restricción nueva pinchando en **Add Constraint**. El tipo de restricción será **IK Solver**, sobre el nuevo *Empty* (tecleamos en el campo **OB: EmptyIk**). Ver figura 7. Es importante que se indique el nombre de forma precisa, ya que Blender distingue entre minúsculas y mayúsculas en los identificadores de objeto. Cambiamos la influencia que tendrá esa restricción sobre la solución final, poniendo **Influence** a su valor máximo. Así, el *Esqueleto* seguirá perfectamente al *EmptyIk*.

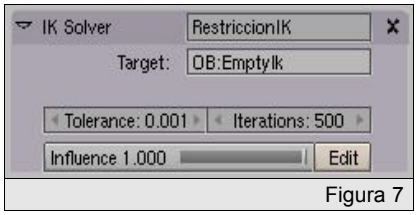


Figura 7

Hecho esto, podemos mover el *EmptyIk* por la escena, y el brazo seguirá el movimiento del mismo. Sin embargo, vemos que el brazo no se comporta correctamente. Las articulaciones no realizan los giros de forma realista, debido a que no tienen restricciones de giro aplicadas.

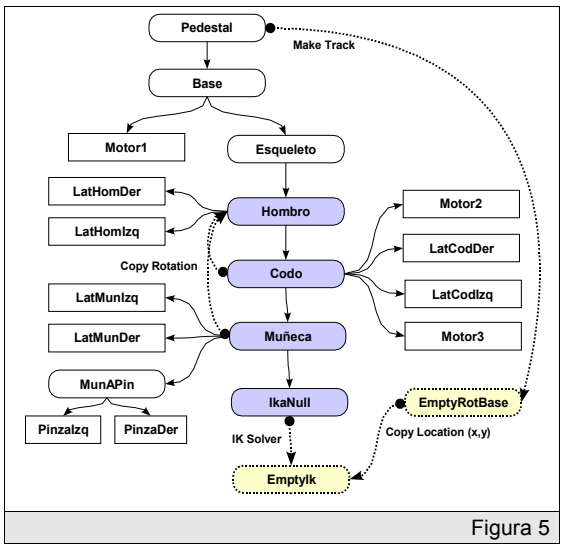


Figura 5

Seleccionamos, en modo pose, el hueso *Codo* y añadimos una restricción para copiar la rotación del hueso *Hombro*. Con esto, conseguiremos parcialmente que los tres huesos giren a la vez. Como se muestra en la figura 8, tendremos que definir el objeto que contiene al hueso llamado *Hombro* (en nuestro caso, *Esqueleto*). Repetiremos la misma operación con el hueso *Muñeca*.

Figura 8

Crearemos 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*, y no tendremos ninguna rotación extraña en las articulaciones.

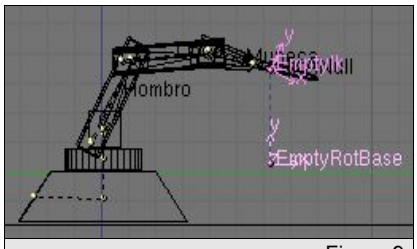


Figura 9

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.

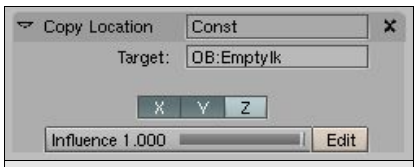
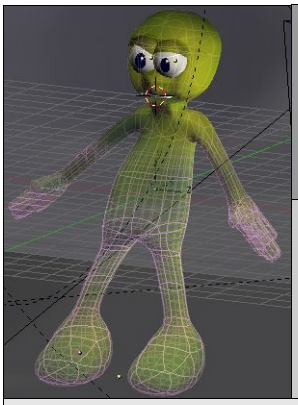


Figura 10

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 T, **Constraint**. Dejamos los parámetros insertados por Blender. 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 I. La interpolación del movimiento de este objeto nos producirá (por cinemática inversa) el movimiento en el resto de articulaciones. Se puede ver el modo de trabajar de la cinemática inversa en el fichero *esqueleto_p7.mpg*.

Añadiremos luces y texturas, e intentaremos conseguir un movimiento similar al mostrado en el vídeo *resultado_p7.mpg*. Es muy importante finalizar correctamente esta práctica para concluir con éxito las próximas sesiones relacionadas con animación de personajes.



Práctica 8: Modelado por Rotoscopia

Animación para la Comunicación ::: ESI-CR ::: UCLM

Carlos González Morcillo :: Carlos.Gonzalez@uclm.es :: http://www.inf-cr.uclm.es/www/cglez

En esta sesión modelaremos un personaje utilizando superficies de subdivisión mediante la técnica de rotoscopia, muy utilizada en el ámbito de la animación de personajes.

Software: Blender 2.32. **Fecha:** 29/Abril/2004

La técnica de modelado mediante rotoscopia se basa en moldear el personaje 3D utilizando, al menos, dos vistas realizadas a mano. Cargaremos los bocetos del personaje que pueden verse en la figura 1. Dividimos la pantalla en 3 ventanas. A la izquierda, cargaremos la imagen de fondo "*frente.jpg*", y en el centro "*lateral.jpg*". La parte de la derecha la dejaremos para ir rotando el modelo, seleccionar vértices, etc... (ver figura 2). La imagen de frente se cargará en la vista frontal (tecla 1 del teclado numérico) y la vista lateral con la tecla 3 del teclado numérico. Recordemos que, para incluir una imagen de fondo en una ventana 3D seleccionaremos en el menú de la cabecera, **View**, **Background Image**.

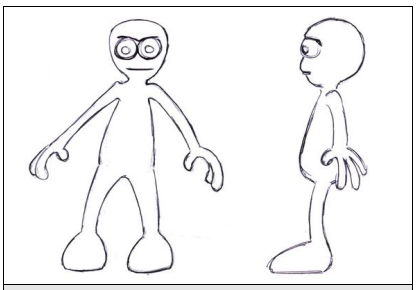


Figura 1

Añadimos un cubo a la escena y lo ajustamos para comenzar a modelar un pie del personaje, como se muestra en la figura 2. Activaremos, como se muestra en la figura 3, las opciones de subdivisión de la malla (botones de edición , F9).

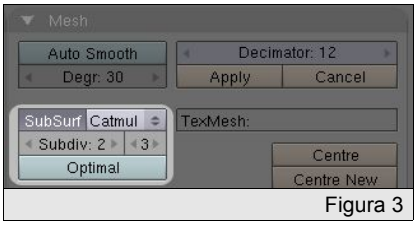


Figura 3

El valor de la izquierda (2 en nuestro caso), es el número de subdivisiones que hace Blender para mostrar la malla en modo de trabajo. Sirve para hacernos una idea de cómo va a quedar. El valor de la derecha, es el número de subdivisiones que se harán a la hora de renderizar. De esta forma, ajustaremos el valor de la derecha para que sea mayor que el de la izquierda (y éste, según la velocidad de nuestro procesador). Si activamos el botón **Optimal**, tendremos una representación de la malla mínima que se crea mediante la superficie de subdivisión.

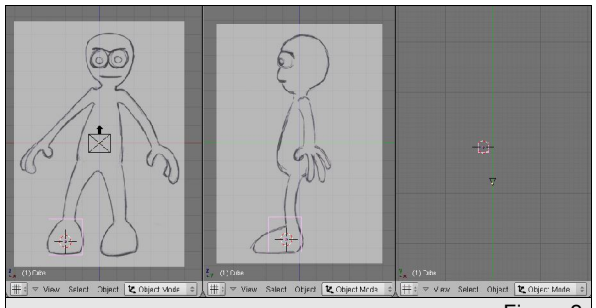


Figura 2

Puede ser útil activar el modo de sombreado Z en alguna vista, para tener una visión más precisa del estado del modelo. La selección de vértices puede realizarse de forma individual, o con la tecla B (o BB). Las operaciones básicas sobre los vértices serán Extrusionar E, con al menos 3 vértices seleccionados, mover G, rotar R y escalar S. La idea básica es ir "calcando" los bocetos, ajustando los vértices de la geometría de la red de control.

Importante: Es crítico conservar el menor número de vértices posible en el modelo a subdividir. Cuanto más aumentemos la geometría del modelo, más difícil resultará desplazar los grupos de vértices.

Es muy útil activar, en modo de edición de vértices, el botón **Draw Faces** de los botones de edición. Con esto, nos aseguramos que hemos seleccionado la cara correcta a la hora de aplicar transformaciones.

Con esta idea en mente, comenzamos a generar un pie del modelo, como se muestra en la figura 4. Para extrusionar una cara (sacar la pierna de la parte superior del pie), seleccionamos los 4 vértices que formarían el "*talón*" y pulsamos E.

Al llegar a la parte de unión entre las piernas, realizaremos un espejo de la pierna izquierda. Para ello, seleccionamos la malla, pulsamos Shift D,. Con esto, obtenemos un duplicado de la pierna, pero tenemos que realizar el espejo para que la curvatura de la misma sea correcta. Para ello, en modo de selección de vértices con todos los vértices seleccionados, pulsamos M y nos aparecerá una ventana preguntado el eje sobre el que vamos a realizar la operación. En el caso del ejemplo sería **X Global**. Si la transformación no ha sido la correcta, podemos realizar "*Undo*", pulsando U.

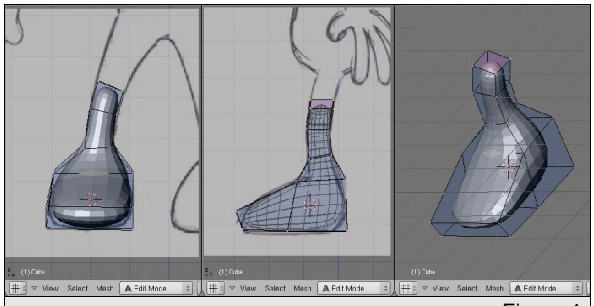


Figura 4

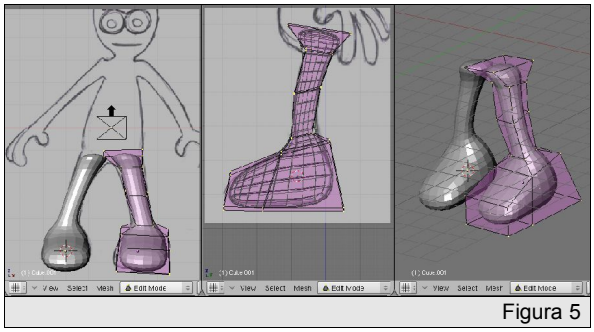


Figura 5

Obtenemos un resultado similar a la figura 5. Pasamos a unir los objetos generados para continuar modelando el torso del personaje.

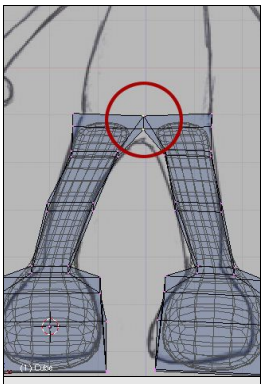


Figura 6

Con las dos piernas seleccionadas (en modo de selección de objeto), unimos las mallas Ctrl J. Entramos en modo de edición de vértices y situamos cerca los vértices que deben ser comunes en ambas piernas (los dos vértices superiores de cada pierna, figura 6). Seleccionamos los 4 vértices que queremos "fusionar", y ajustamos en los botones de edición el valor **Limit** que nos

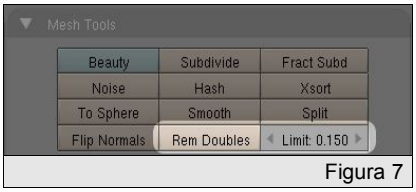


Figura 7

gradúa la distancia que debe haber entre vértices continuos a eliminar. Hecho esto, pinchamos en **Rem Doubles** (ver figura 7) y Blender debe informarnos que ha eliminado 2 vértices.

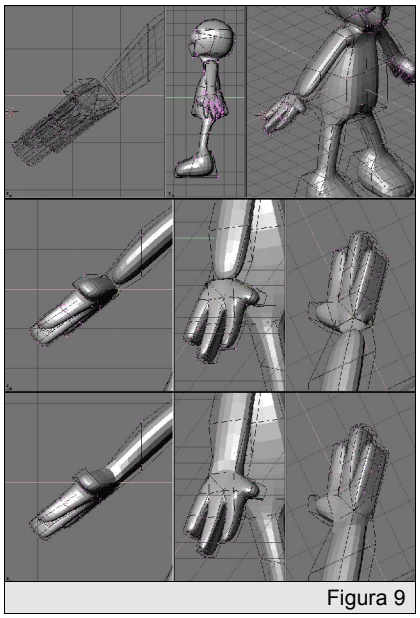


Figura 9

Continuamos modelando el cuerpo del personaje, siempre teniendo en mente que debemos utilizar el menor número de polígonos posible. Hay que ver también las partes donde "interesa" generar nuevas caras. Por ejemplo, para modelar el torso (figura 8 arriba), hemos realizado tres extrusiones cuando habría bastado con dos. De esta forma, hemos obtenido unos vértices donde, en el paso siguiente, generamos los brazos.

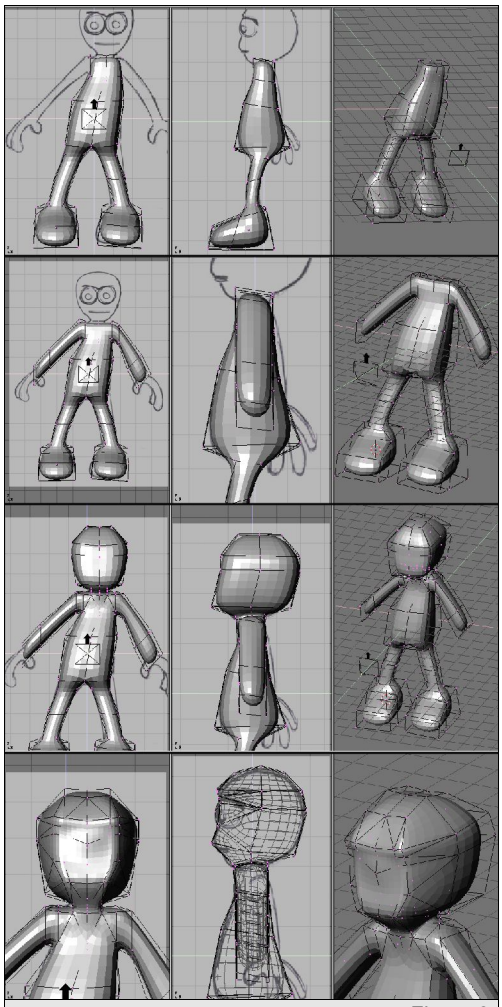


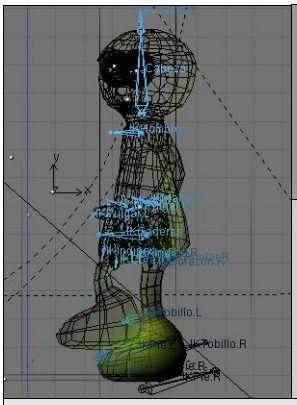
Figura 8

Necesitamos generar nuevos vértices para sacar el cuello y, de ahí, la cabeza. Para ello, resulta útil la herramienta **Subdivide** (botones de edición W). Es importante utilizar esta herramienta sólo cuando sea necesario. El número de vértices crece muy rápido y el modelo podría hacerse inmanejable. También resulta útil la opción de **Smooth**, del mismo bloque de botones. Con este botón suavizamos una superficie, eliminando la apariencia de estar generada de forma simétrica. Esta opción la utilizaremos únicamente cuando la geometría base esté totalmente definida, para añadir naturalidad.

Generaremos una mano del modelo y realizaremos la otra por "*espejo*" (de forma similar a como trabajamos con la pierna). En la figura 9, se muestra el proceso de ajustar los vértices para que "coincidan" con la muñeca. De forma similar, pinchando en **Rem-Doubles** juntaremos las superficies en una. Blender realizará la interpolación de la malla resultante.

Por último, añadimos los ojos, párpados y dientes del modelo. Para las texturas se ha utilizado un color verde de material y después se han pintado los vértices (**Vertex Paint Mode**), como se explicó en la sesión 4 (Modelado del Dinosaurio).

Con esta técnica podemos construir modelos de personajes complejos. Partiendo de unos bocetos 2D sencillos generamos la malla 3D mediante rotoscopia. Al emplear superficies de subdivisión podemos ajustar el nivel de suavizado de la superficie final. Habitualmente un valor de subdivisión de 3-4 en el renderizado, da mallas muy suavizadas. Animaremos este personaje en la siguiente sesión de prácticas con NLA.



Práctica 9: Esqueletos Avanzados

Animación para la Comunicación ::: ESI-CR ::: UCLM

Carlos González Morcillo :: Carlos.Gonzalez@uclm.es :: http://www.inf-cr.uclm.es/www/cglez

La animación de personajes digitales requiere definir un esqueleto interno al que aplicaremos Cinemática Inversa para calcular la posición de todas las articulaciones.

Software: Blender 2.32. **Fecha:** 06/Mayo/2004

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 junto al 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; 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. Utilizaremos esta segunda alternativa, mucho más cómoda para construir animaciones complejas.

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.

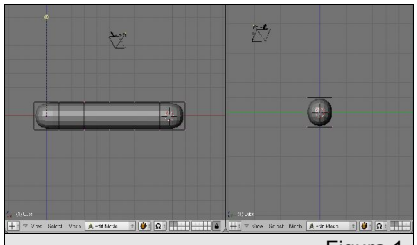


Figura 1

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**).

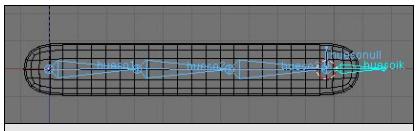


Figura 2

Con el esqueleto todavía en modo de edición de huesos (color rosa oscuro, pulsando Tab), añadiremos un quinto hueso, 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 Names** 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.

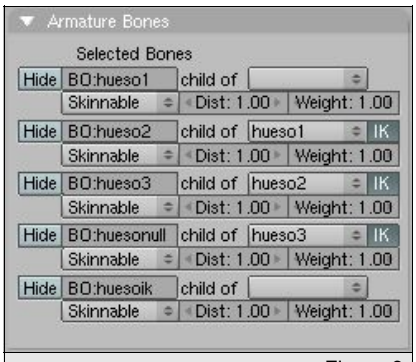


Figura 3

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**. Ahora podemos desplazar (en modo pose) **huesoik**, y el resto del esqueleto se calculará de forma automática mediante cinemática inversa.

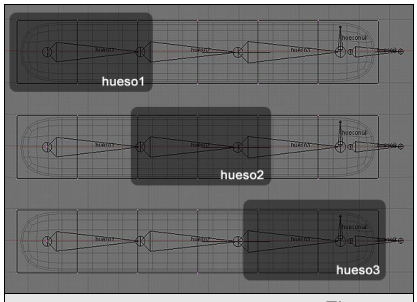


Figura 4

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).

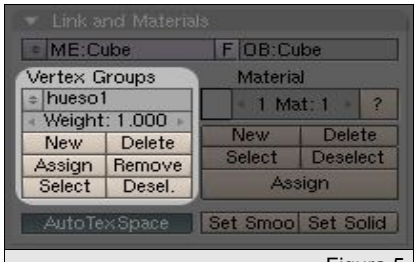


Figura 5

En los botones de edición (F), 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**. De esta forma he-

mos 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, haremos la malla hija del esqueleto Ctrl P, **Use 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.

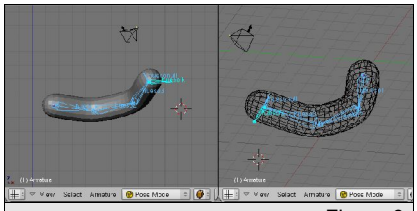


Figura 6

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). Aplicaremos este método al archivo **p9modelo.blend**.

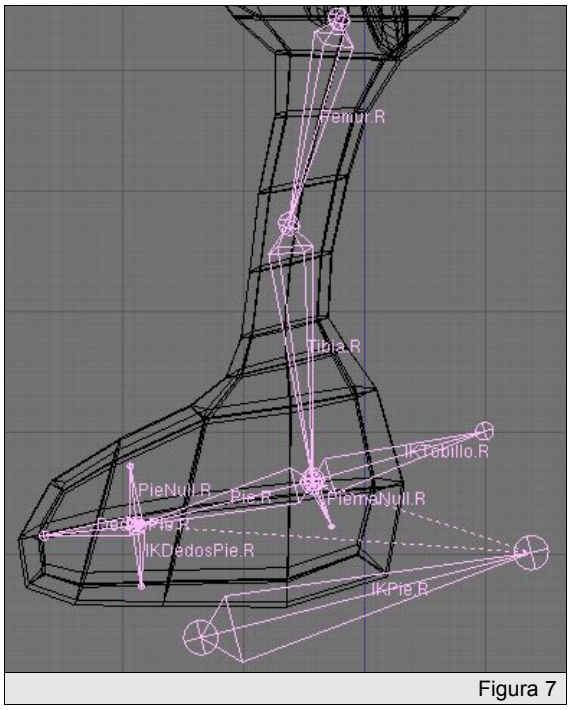


Figura 7

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.

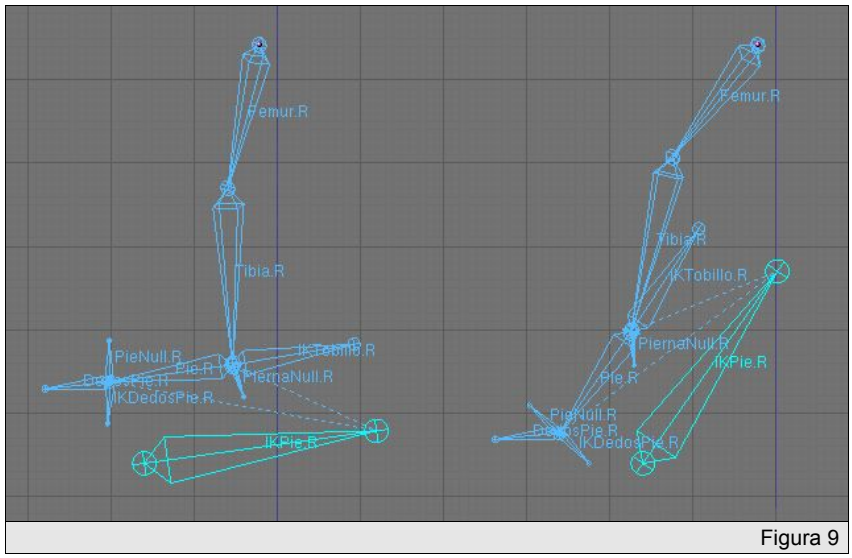


Figura 9

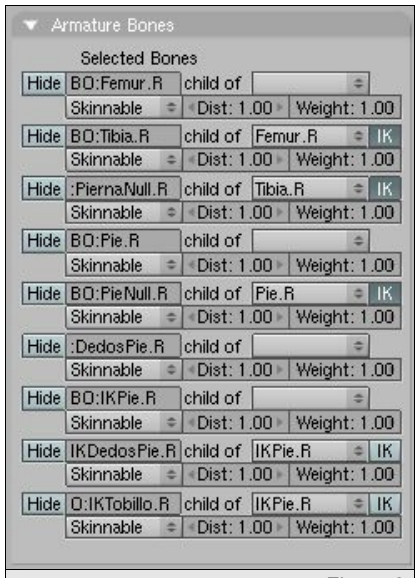


Figura 8

Añadimos como primer Armature el formado por **Fémur.R**, **Tibia.R** y **PiernaNull.R** (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.R** y **PieNull.R**. Finalmente, añadimos como cadenas independientes de un único hueso: **DedosPie.R**, **IkDedosPie.R**, **IkTobillo.R** e **IkPie.R**.

Hecho esto, realizaremos los siguientes parentescos entre huesos. Para ello, iremos a los botones de edición F9, y en modo de edición Tab, con todos los huesos seleccionados A, elegimos el nombre del hueso padre en el campo **child of**; ver figura 8) los dos huesos IK pequeños; **IkDedos-**

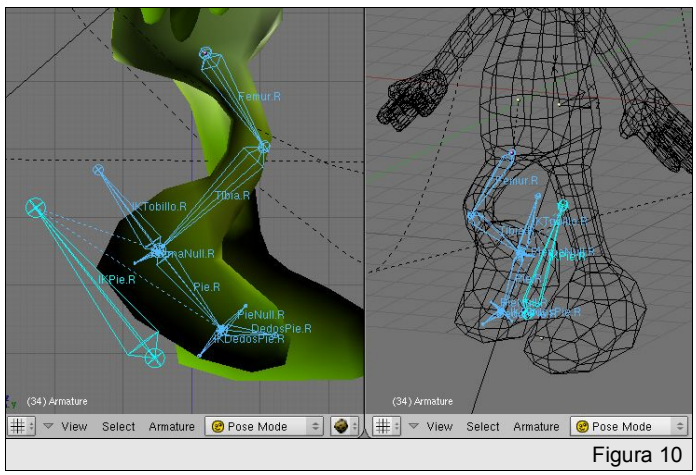


Figura 10

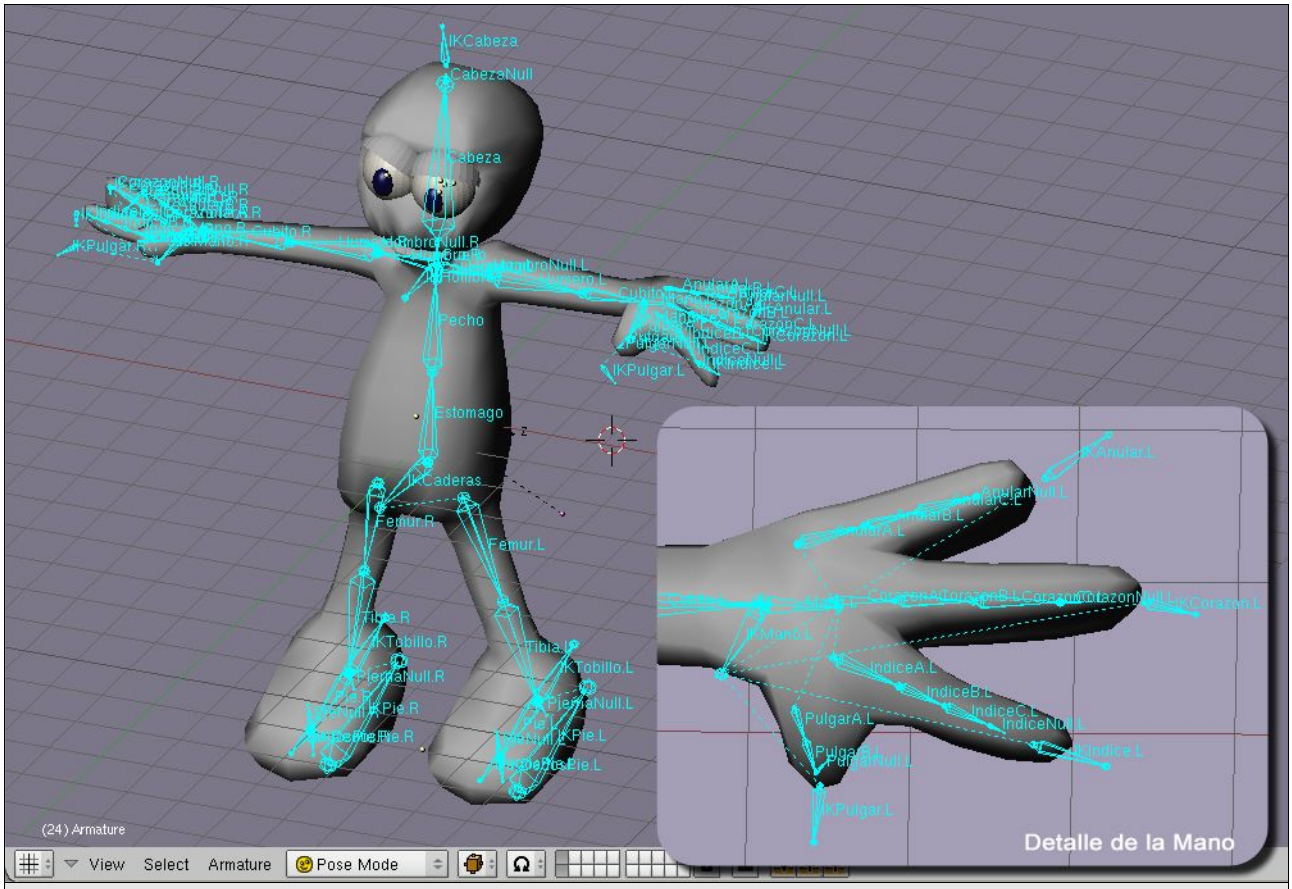


Figura 11

Pie.R e **IkTobillo.R** con el controlador principal del pie: **IkPie.R**. Al hueso **DedosPie.R** añadimos una restricción "*copy location*" hacia el hueso **IkDedosPie.R**. Al hueso **PieNull.R** un *IkSolver* hacia **IkDedosPie.R**. Al hueso **Pie.R** un "*copy location*" hacia **PiernaNull.R**. Finalmente a **PiernaNull.R** un *IkSolver* hacia **IkTobillo.R**. Si todo ha ido bien, tendremos un comportamiento como se ve en la figura 9; rotando **IkPie.R**, se observa un comportamiento general en la pierna, quedando el hueso **DedosPie.R** 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.R**, **Tibia.R**, **Pie.R** y **DedosPie.R**. Al terminar, podemos emparentar la malla 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 10). En la figura 11 puede verse una captura de la jerarquía que tenemos que construir.

La figura 12 muestra los nombres exactos de los huesos y sus relaciones de parentesco. 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**. De ese punto sacaremos dos cadenas de huesos (una para cada lado): **Hombro.R** y **HombroNull.R**. **Hombro.R** tendrá un *CopyLocation* de **IkHombros** y **HombroNull.R** servirá de origen para el *CopyLocation* de **Humero.R**. De forma similar se definirán los huesos y restricciones para el lado izquierdo.

Contruiremos los brazos como una cadena de 3 elementos: **Humero.R**, **Cubito.R** y **Mano.R**. Añadiremos un *IkSolver* (**IkMano**) sobre **Mano.R**. Añadiremos nuevas cadenas de huesos para los dedos. Índice, Corazón y

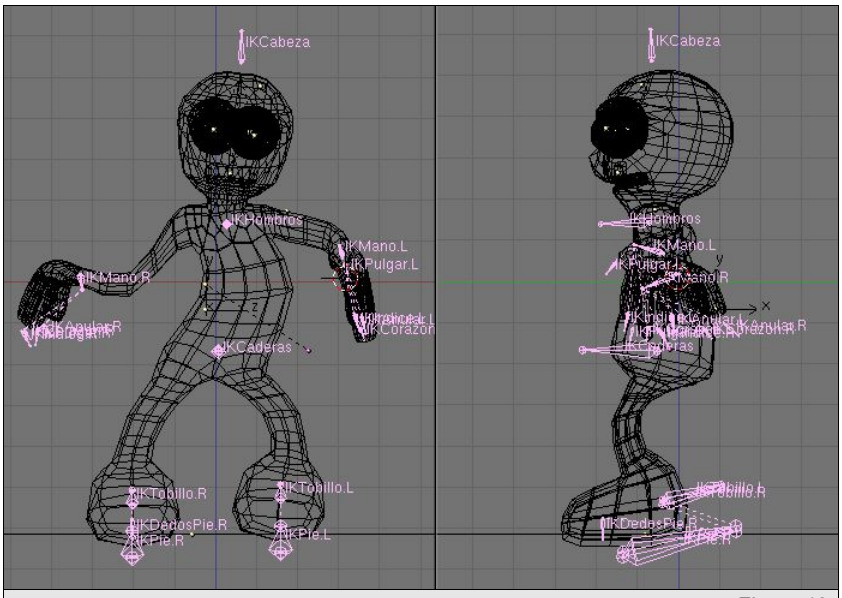


Figura 13

Anular tendrán 3 huesos (el dedo pulgar sólo 2) más un cuarto hueso (*null) que servirá como 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 con **Mano.R**. Una captura del esqueleto de esta parte puede verse en el anexo de la figura 11.

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 al lado del nombre del hueso.

En la figura 12 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**, **IKMano.***, **IKPulgar.***, **IKIndice.***, **IKCorazon.*** e **IKAnular.***

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 la siguiente sesión emplearemos animación no lineal para mezclar acciones individuales aplicadas a ciertas partes del modelo.

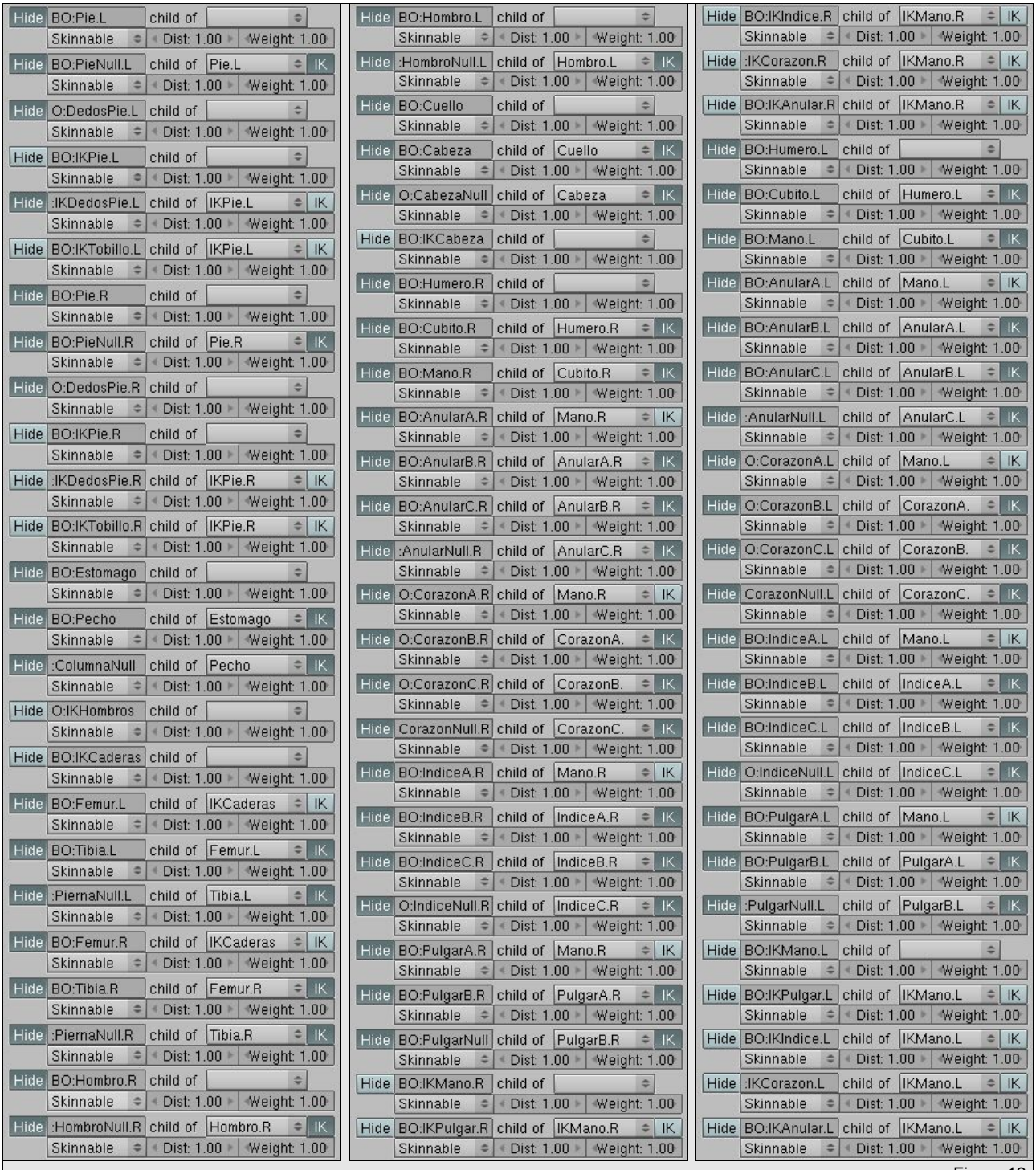
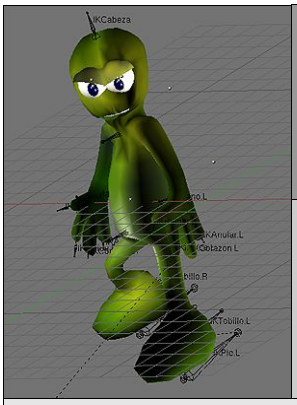


Figura 12



Práctica 10: Animación No Lineal

Animación para la Comunicación ::: ESI-CR ::: UCLM

Carlos González Morcillo :: Carlos.Gonzalez@uclm.es :: http://www.inf-cr.uclm.es/www/cglez

Continuando con la sesión anterior, realizaremos una animación sencilla utilizando el módulo de animación no lineal (ANL) de Blender. Definiremos un conjunto de acciones sencillas que mezclaremos convenientemente.

Software: Blender 2.32. **Fecha:** 13/Mayo/2004

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, en el movimiento de *"andar"*, definiremos la acción de *"mover las piernas"* y la acción de *"mover los brazos"*. De la combinación de ambas acciones en el compositor de NLA obtendremos el movimiento de "*andar*".

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.R**, **IKMano.L**, **IKPie.R** e **IKPie.L**. Con los huesos seleccionados, insertamos un frame clave I 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, pinchamos en el botón o seleccionamos **Armature**, **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 **Armature** visto anteriormente.

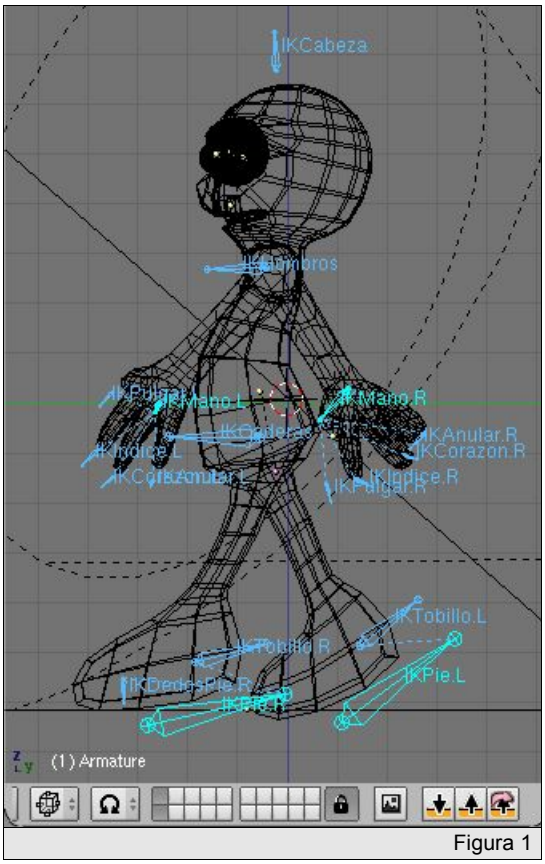


Figura 1

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 I. 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. Finalmente, obtendremos una acción en la ventana de acciones como se muestra en la figura 2. En la ventana NLA aparecerán una serie de cuadritos amarillos indicando los frames clave (figura 3). Podemos ver la animación de la acción pulsado Alt A.

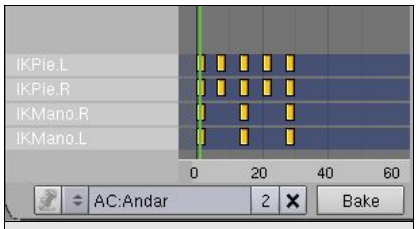


Figura 2

Figura 3

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 recuperarla de nuevo accediendo a la lista desplegable. 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, 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"*). Podemos escalar la duración de la acción de caminar al

Figura 4

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.

Podemos ver cómo quedó la composición de acciones de esta sesión en la figura 4. La parte inferior, que tiene frames clave directamente sobre el objeto Lamp.004, se corresponde con el efecto del foco que ilumina la cara del muñeco al comenzar la animación (insertamos claves respecto de la capa *-layer-*). A continuación analizaremos algunas opciones de composición.

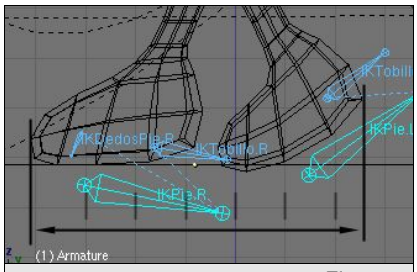


Figura 5

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, 6½ unidades.

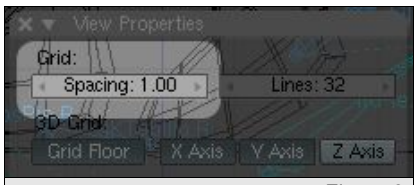


Figura 6

Muy bien, 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 6.5 unidades de rejilla × 1 unidad que mide cada rejilla × 2 "*mitades simétricas*" que tiene cada paso = 13 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.

Figura 7

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).

En la ventana NLA, seleccionamos la acción "*andar*" y pulsamos N. Con esto, accedemos a las propiedes específicas de la acción (ver 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 (13 unidades *reales* en nuestro caso).

Si activamos **UsePath**, 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. Por último, el parámetro **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.

Figura 8

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).

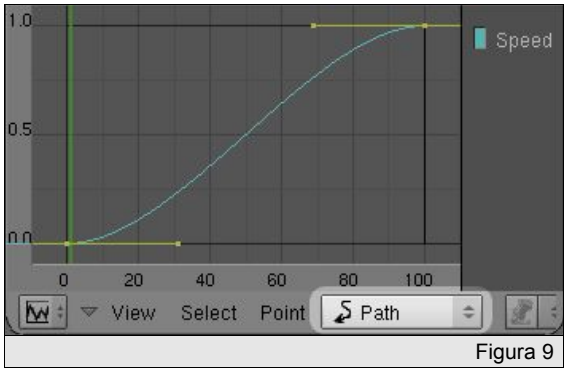
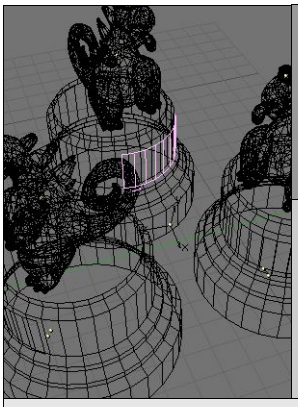


Figura 9



Práctica 11: Render Realista

Animación para la Comunicación ::: ESI-CR ::: UCLM

Carlos González Morcillo :: Carlos.Gonzalez@uclm.es :: http://www.inf-cr.uclm.es/www/cglez

Esta práctica completa los conceptos vistos en teoría sobre trazado de rayos, radiosidad, mapas de entorno (simulación de reflexiones) e iluminación basada en imágenes de alto rango dinámico.

Software: Blender 2.33a. + Yafray 0.0.6 + Gimp 1.2.4 **Fecha:** 20/Mayo/2004

Comenzaremos la primera parte de la práctica cargando el escenario de la figura 1 (en el fichero **P11Radiosidad.blend**). La iluminación de la escena vendrá dada por las áreas de luz de las **superficies** superiores (planos) de la habitación. Recordemos que en Radiosidad, la iluminación vendrá generada por superficies. De esta forma, necesitaremos crear planos a los que asignaremos un material que emita luz.

Es muy importante la dirección del vector normal ya que Blender "*lanzará*" la luz en ese sentido. Por tanto, tendremos que asegurarnos que el vector normal de cada foco apunta "*hacia el suelo*", tal y como muestra la figura 2. Para comprobar que es así, seleccionamos cada "foco", y en modo de edición de vértices Tab, con todos los vértices seleccionados A, accedemos a los botones de edición y activamos el botón **Draw Normals**. Podemos ajustar el tamaño de representación del vector normal en **Nsize**. En caso de que la normal esté invertida, podemos ajustarla pulsando W **Flip Normals** o rotando la cara.

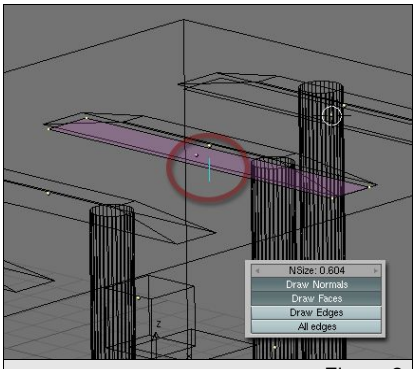


Figura 2

Los elementos que emiten luz tendrán el campo **Emit** del material con un valor mayor que 0. Los focos de la práctica tienen un valor de 0.1 (ver figura 3). A la caja de color rojo se le dio un valor de 0.020.

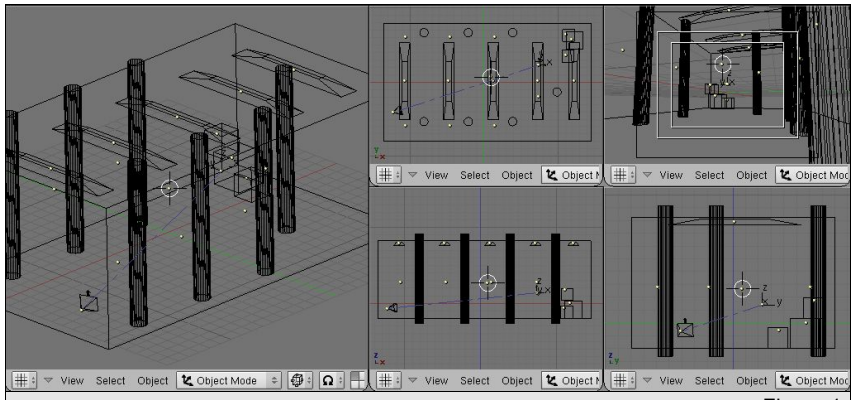


Figura 1

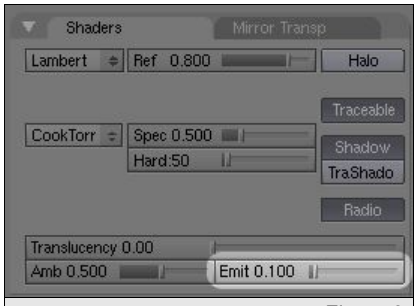


Figura 3

Añadimos una esfera a la escena, situada debajo del segundo foco y le asignaremos un material de color azul (activamos además **Set Smooth** en los botones de edición). De momento, no calcularemos su mapa de entorno.

Accedemos al menú de radiosidad con el botón , que está dentro de las subopciones de sombreado . Seleccionamos todas las mallas que forman nuestra escena A y pinchamos (figura 4) en el botón **Collect Meshes**. Aparecen en la vista de la cámara unos cuadrados blancos y azules. Los azules representan el tamaño mínimo y máximo de los "*Elementos*" (recordemos que cada Parche está formado por un conjunto de Elementos), y los blancos representan los "*Parches*".

Hecho esto, pinchamos en **Limit Subdivide**, seguido de **Subdiv Shoot Element** y **Subdiv Shoot Patch**. Finalmente pinchamos en el botón **Go**. De esta forma comienza el proceso de cálculo de la solución de radiosidad. Podemos parar el proceso en cualquier momento pulsando Esc, quedándonos con la aproximación que se ha conseguido hasta ese instante.

Observamos que hay un error en el cálculo de nuestra escena. Las paredes de la habitación permanecen de color negro. Esto es porque el vector

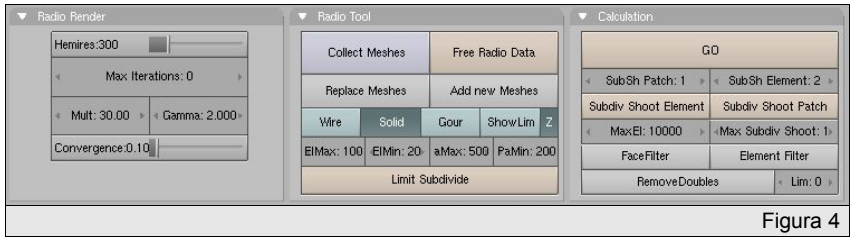


Figura 4

normal de las caras de las paredes están apuntando hacia el exterior. Liberamos la solución de radiosidad anteriormente calculada (pinchando en **Free Radio Data**). Con la habitación seleccionada y en modo de edición de vértices [Tab], con todos seleccionados [A], corregimos las normales [W] **Flip Normals**.

Figura 5

De igual modo, es importante que el material de todo objeto que intervenga en la escena tenga el botón **Radio** activado (ver figrua 5). De otra forma, aparecerá de color negro en la escena, aunque tenga los vectores normales apuntando correctamente.

En la figura 4, el tamaño de los parches (**PaMax** - **PaMin**) determina el detalle final (cuanto más pequeño sea, más detallado será el resultado final), pero incrementamos el tiempo de cálculo. De forma similar ocurre con el tamaño de los elementos (**ElMax** - **ElMin**). En **Max Iterations** indicamos el número de pasadas que Blender hará en el bucle de Radiosidad. Un valor 0 indica que haga las que estime necesarias. **MaxEl** indica el número máximo de elementos para la escena. **Hemires** es el tamaño del hemicubo (técnica utilizada para calcular el parámetro *H* visto en teoría, de cálculo de visiblidad).

Con las normales corregidas, y con los valores por defecto de Blender, seleccionamos de nuevo todos los objetos de la escena A y lanzamos igual que antes el cálculo de la radiosidad.

Una vez terminado el proceso de cálculo (podemos pararlo cuando la calidad sea aceptable con [Esc]), podemos añadir la nueva malla calculada reemplazando las creadas anteriormente (**Replace Meshes**) o añadirla como nueva a la escena (**Add new Meshes**). Elegiremos esta segunda opción. Antes de deseleccionar la malla con la información de radiosidad calculada, la moveremos a otra capa. Las capas en Blender se utilizan de forma similar a como se hace en cualquier otro programa de diseño. Pueden resultar muy útiles cuando la complejidad de la escena crece. Incluso son imprescindibles para algunas operaciones (como en los mapas de entorno). Las capas activas (visibles) en cada momento pueden verse en la cabecera de cualquier ventana 3D (figura 6).

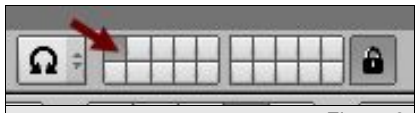


Figura 6

En el caso de la figura 6, sólo la primera capa está activa. Podemos seleccionar múltiples capas como activas pinchando sobre ellas con la tecla [Shift] pulsada.

Movemos los objetos seleccionados (en nuestro caso, la nueva malla) a una nueva capa pulsando la tecla [M]. Nos aparecerá una ventana emergente donde nos pedirá la capa destino del movimiento (figura 7).

Figura 7

Movemos la malla de radiosidad a la segunda capa y liberamos la memoria ocupada por estos datos pinchando en **Free Radio Data**.

Activamos únicamente la segunda capa y renderizamos. Tenemos que obtener una imagen similar a la figura 8.

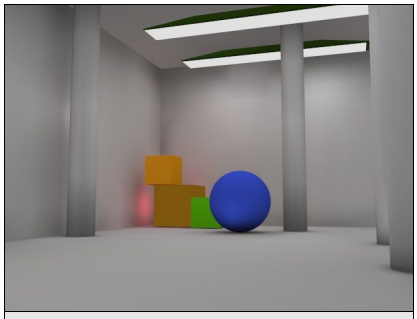


Figura 8

Vamos a añadir el mapa de entorno a la esfera, para que refleje el resto de objetos de la habitación. Para ello, primero separaremos la malla del resto, moviéndola a la capa 3. En modo de edición de vértices [Tab], seleccionamos uno de los que forman la esfera. Pulsamos la tecla [L] y seleccionamos, de esta forma, todos los que están enlazados con ese directamente (los que forman la esfera). Separamos este conjunto de vértices como un objeto nuevo pulsando [P]. Movemos la esfera a la tercera capa (tecla [M] con la esfera seleccionada). Hacemos las dos capas visibles a la vez. Añadimos un objeto Empty en la capa 2, situado en el centro de la esfera, al que llamaremos "EmptyEntorno" (figura 9). Nos servirá para calcular el mapa de entorno cúbico, como si estuviéramos situados en ese punto del espacio.

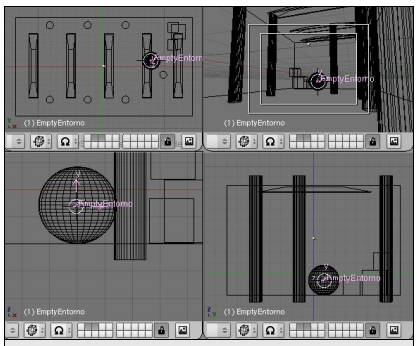


Figura 9

Al material azul de la esfera le asociamos una textura (también mediante el atajo de teclado [H]). Añadimos una textura nueva **Add New**, y en el tipo de textura elegimos **EnvMap**. Aparece una nueva pestaña con las opciones del mapa de entorno. Los botones azules (**Static**, **Anim** y **Load**) indican si el mapa de entorno se va a calcular sólo una vez (en caso de que los objetos del mundo no se muevan), **Anim** (en caso de que haya desplazamiento de los objetos) o **Load** (para cargar el mapa de entorno precalculado en una imagen). **Free data** sirve para forzar a que se recalcule el mapa. En **Ob:** indicaremos el objeto que nos servirá de origen para calcular el mapa. En nuestro caso será **EmptyEntorno**. Con **CubeRes** indicamos la resolución, en píxeles, de cada pared del cubo (hemos aumentado la resolución a 300 píxeles). Al lado, tal y como muestra la figura 10, nos indica qué capa(s) no queremos renderizar.

Figura 10

Es muy importante no renderizar las capas donde se encuentren los objetos a los que vamos a aplicar el mapa de entorno. De otra forma, el mapa obtenido renderizaría el interior del objeto (en nuestro caso la esfera). Así, indicamos que la capa 3 no la vamos a tener en cuenta. Para terminar, en los botones de sombreado [F8], habrá que indicar que queremos usar el vector de reflexión

como coordenadas de textura (**Refl**), como se muestra en la figura 11.

Figura 11

Hecho esto, si renderizamos la escena (con el botón EnvMap de las opciones de render activado), debemos obtener un resultado similar al de la figura 12.

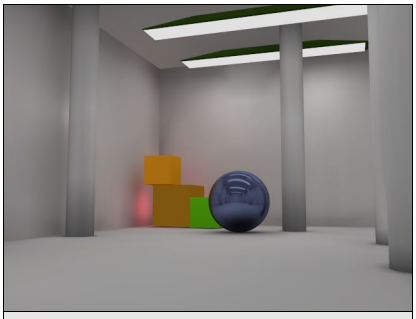


Figura 12

⇨ 2ª Parte ::: Raytracing

En esta segunda parte de la práctica utilizaremos las características de trazado de rayos de blender. Para comenzar, abriremos el fichero **P11-RayTracing.blend**.

En este archivo, encontramos la escena de la figura 13. Únicamente se han asignado texturas procedurales a los pedestales de las estatuas y al suelo. Vamos a asignar materiales a los dinosaurios. Todos los elementos que forman cada dinosaurio (ojos, párpados y cuerpo) comparten el mismo material. Hay tres materiales diferentes, nombrados como **oro**, **plata** y **cristal**. Seleccionamos cada uno de los dinosaurios y, en los botones de sombreado , asignamos las propiedades de material correspondientes a las figuras 14 (oro), 15 (cristal) y 16 (plata).

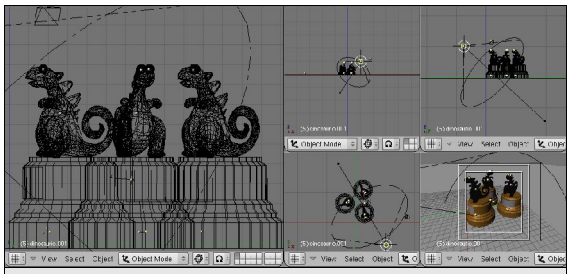


Figura 13

Los materiales de oro y plata tienen activado el botón de **RayMirror**. Con esto, indicamos a blender que debe tomar estos materiales como espejos (en el grado indicado por **RayMir**), calculando el número de niveles recursivos indicado por **Depth**. Mediante el parámetro **Fresnel** ajustamos el nivel de reflexión de este material, suavizado con el parámetro **Fac**.

El material de cristal, además debe ser translúcido. Por esta razón, se ha activado el botón **RayTransp**, con un bajo índice de refracción (parámetro **IOR**). A mayor IOR, mayor refracción del material. Es importante ajustar el parámetro de transparencia **Alpha** (notado con la letra **A**, debajo de la definición de color) en el material. En nuestro caso, se ha definido el material como muy transparente (A = 0.110).

Finalmente, se ha elegido en todos los materiales el modelo de brillo especular **Toon**, por concentrar los brillos, y se han ajustado los parámetros de reflexión y brillo como muestra cada figura.

Figura 17

Si queremos renderizar la escena, hay que activar el render mediante raytracing de Blender. Para ello, activamos el botón **Ray** (ver figura 17).

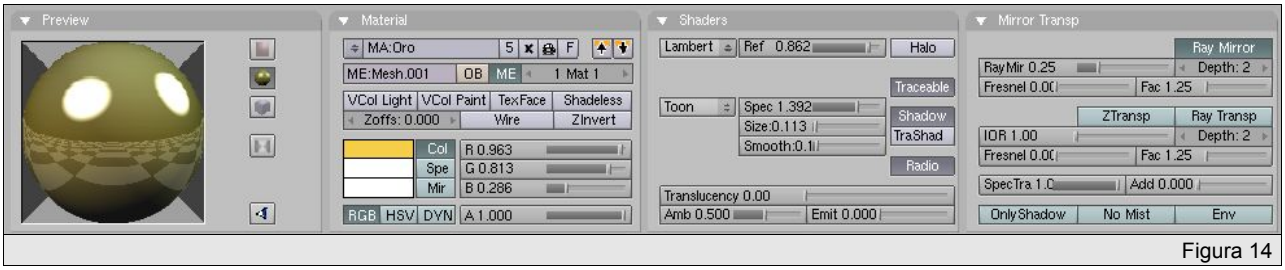


Figura 14

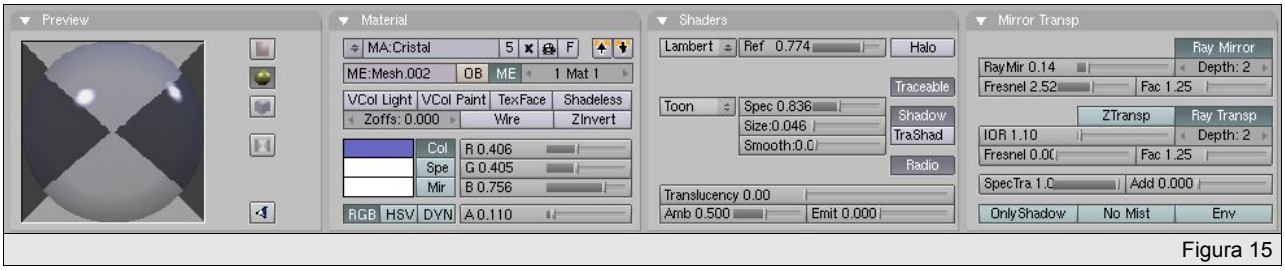


Figura 15

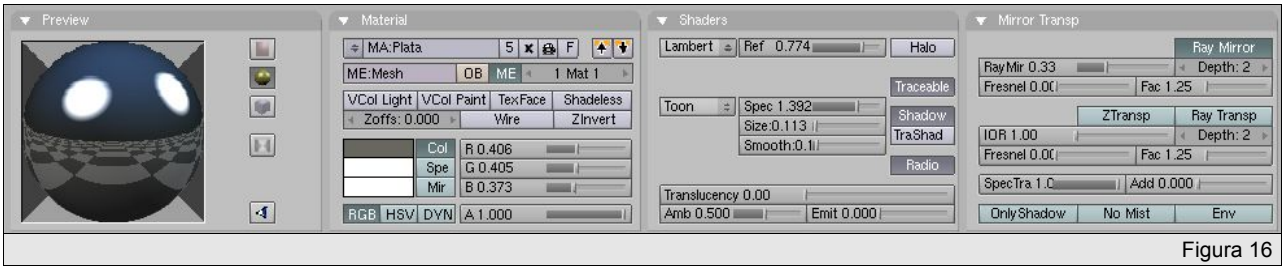


Figura 16

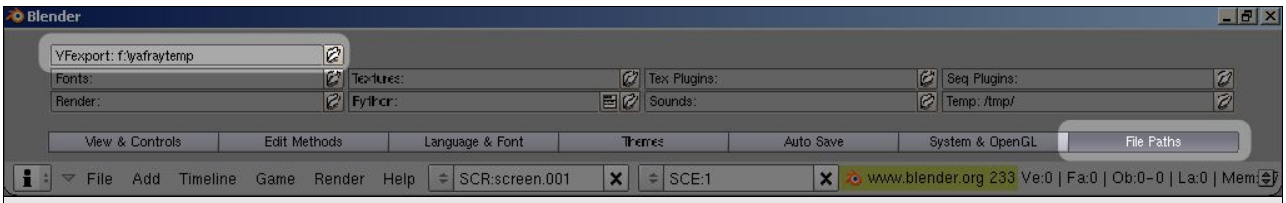


Figura 20

El nivel de Oversampling influye de forma muy directa sobre el tiempo de render. Lo activaremos únicamente cuando queramos obtener la imagen final. Además, es muy aconsejable realizar las pruebas de renderizado a baja resolución.

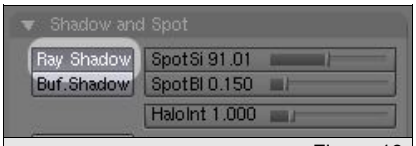


Figura 18

El parámetro inferior (en la figura 17 está marcado con el valor 256) indica la resolución de árbol octal que emplea blender para construir la escena. Con valores más altos, el tiempo de procesamiento previo es mayor pero ahorraremos tiempo en la generación de la imagen. Cambiamos el tipo del foco, activando el botón **RayShadow** en las propiedades del foco (figura 18).

Si renderizamos la escena, obtendremos un resultado similar al de la figura 19. En este caso, se ha utilizado una textura para obtener un cambio en el vector normal (**Nor**) y aplicar así *Bump Mapping* en el letrero de cada estatua.

Vamos a renderizar la misma escena en Yafray. Para ello, primero debemos tener instalado yafray correctamente en nuestro sistema y, desde una consola tecleando "**yafray**", deberá mostrar las opciones del programa. Con yafray correctamente instalado, tendremos que indicar a blender el directorio donde debe guardar los ficheros que generará yafray. Esta opción de configuración se indica desplegando la zona de la cabecera superior de blender (nos situamos en la zona de unión de la ventana 3D con la ventana de información y arrastramos). El directorio temporal de yafray (ver figura 20) se especifica en el campo **YFexport** del conjunto de opciones **File Paths**. Será en este directorio donde se almacene el fichero de descripción de escena **YBtest.xml** que utilizaremos más adelante.

Nos situamos en los botones de renderizado, y seleccionamos en la lista desplegable a Yafray como motor de render (ver figura 21, a la izquierda debajo del botón **Render**). Aparecerán dos pestañas con las opciones de configuración de Yafray (figura 21).

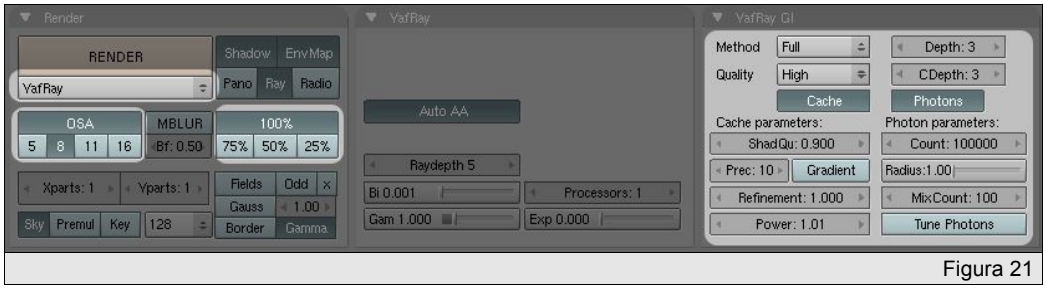


Figura 21

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

Figura 19

Figura 22

En el panel titulado **Yafray** (figura 21), tenemos opciones generales de configuración; como el botón **AutoAA**, que permite a Yafray seleccionar automáticamente el nivel de AntiAliasing (Oversampling), la profundidad máxima del trazado de rayos **Raydepth** y ajustes particulares de exposición de la luz, corrección de gama, etc...

En panel 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 fondo, o **Full**, que realiza un cálculo de iluminación global completa (con iteración de la luz entre objetos). Elegiremos este segundo método. La lista de calidad **Quality** ajusta, según el parámetro seleccionado, las variables que necesitará yafray para renderizar la escena. Estos valores predefinidos según la calidad elegida, podrán cambiarse editarndo el fichero XML (**YBtest.xml**) que exportará Blender. El parámetro **Power** 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.

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 **Caché**, aceleraremos los cálculos y suavizamos el ruido. 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). **Gradient** es un esquema de suavizado de sombras, que no vamos a utilizar en nuestra escena. **Refinement** es un valor umbral, donde especificamos el cambio de intensidad que vamos a permitir entre muestras. Valores más pequeños dan resultados más precisos.

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 restulado más suave. **Radius** indica el nivel de difuminado entre fotones. Es importante encontrar un valor adecuado para que el resultado sea suave. **Mix-Count** indica el número de fotones que van a situarse dentro del radio anterior. Finalmente, el botón **Tune Photons** sirve para obtener una respresentación preliminar del mapeado de fotones. Deberá estar desactivado para generar la imagen resultado.

Con estos parámetros, podemos renderizar la escena F12 y obtenemos la imagen de la figura 22. El progreso de render puede verse en la consola en modo texto de blender (el terminal DOS en windows o la consola desde donde hayamos ejecutado blender). Antes de obtener los parámetros definitivos, hemos realizado pruebas a baja resolución. El proceso de generación de la imagen puede resultar bastante costoso y, dependerá del procesador el tiempo de generación. Podemos abortar el render pulsando Ctrl C en la consola.

Vamos a utilizar ahora iluminación basada en imágenes HDRI. Para ello, eliminamos el foco de la escena y, con los mismos parámetros del ejemplo anterior, lanzamos el render. Esperamos a que yafray cargue los objetos de la escena, analice la gramática y llegue al punto de generación de la imagen (cuando muestra la barra de progreso; ver listado 1). En este

```
[Loader]: Added globalphotonlight light gpm
[Loader]: Added pathlight light path_LT
[Loader]: Added constant background world_background
[Loader]: Added camera MAINCAM
Using a world resolution of 0.00152381 per unit
Rendering with 5 raydepth
2 anti-alias passes and 4 minimum samples per pass, 8
samples total.
Building the bounding tree ... OK
Setting up lights ...
Finished setting up lights
Launching 1 threads for rendering ...
Rendering ...
0%                 50%                 100%
[####.................................]
```

Listado 1

```
<background type="constant" name="world_background" >
       <color r="1.009920" g="1.009920" b="1.009920" />
</background>

<camera name="MAINCAM" resx="550" resy="600" focal="1.193182" >
       <from x="15.503942" y="1.697953" z="17.577042" />
       <to x="14.842168" y="1.940074" z="16.867512" />
       <up x="14.837610" y="1.941742" z="18.281717" />
</camera>

<render camera_name="MAINCAM"
       raydepth="5" gamma="1.000000" exposure="0.000000"
       AA_passes="2" AA_minsamples="4"
       background_name="world_background"
       AA_pixelwidth="2" AA_threshold="0.05" bias="0.001000" >
       <outfile value="f:\yafrayTemp\YBtest.tga" />
</render>
```

Listado 2 (Original)

```
<background type="HDRI" name="entHDRI" exposure_adjust="0.3">
      <filename value="car.hdr"/>
</background>

<camera name="MAINCAM" resx="550" resy="600" focal="1.193182" >
       <from x="15.503942" y="1.697953" z="17.577042" />
       <to x="14.842168" y="1.940074" z="16.867512" />
       <up x="14.837610" y="1.941742" z="18.281717" />
</camera>

<render camera_name="MAINCAM"
       raydepth="5" gamma="1.000000" exposure="0.700000"
       AA_passes="2" AA_minsamples="4"
       background_name="entHDRI"
       AA_pixelwidth="2" AA_threshold="0.05" bias="0.001000" >
       <outfile value="f:\yafrayTemp\YBtest.tga" />
</render>
```

Listado 3 (Modificado)

momento podemos abortar el render Ctrl C (el fichero XML ha sido guardado en el directorio temporal de yafray).

Nos vamos al directorio de exportación de yafray y editamos el fichero **YBtest.xml**. Al final del fichero, realizaremos una serie de cambios relacionados con el modo de iluminación. Del fichero original (listado 2), cambiamos al tipo de fondo **background** (listado 3) y el nombre del fondo en la etiqueta render. En nuestro ejemplo, **car.hdri** es el nombre de la imagen HDRI que hemos utilizado. Existen visores gratuitos de este tipo de imágenes, como *HDRView*.
Puede ser interesante modificar los atributos **AA_passes** del campo Camera (número de pasadas de AntiAliasing) y la resolución de la imagen (campos **resx** y **resy**) a la hora de realizar pruebas. Si la iluminación es correcta, ajustaremos la resolución a su valor definitivo con las pasadas de AntiAliasing convenientes. Es importante ajustar el parámetro **exposure_adjust** para cada mapa HDRI. No hay una receta mágica. Deberemos probar para cada mapa HDRI qué valor debemos ajustar en este parámetro. De igual forma, podemos ajustar la exposición de la cámara (parámetro **exposure**).

Figura 23

Con el fichero modificado, desde un terminal lanzaremos yafray pasándole como parámetro el nombre del fichero XML. Tras un rato de espera, tendremos la bonita imagen renderizada donde apunte el parámetro **outfile** (ver listado 3). Con la imagen HDRI de la figura 23 (*car.hdri*), hemos obtenido el resultado de la figura 24. Podemos encontrar imágenes HDRI en varios sitios de internet. Uno de ellos, en la página del investigador principal de este tipo de técnicas Paul Debevec (http://www.debevec.org/).

En la figura 25 se pueden ver varias pruebas de iluminación empleando diferentes imágenes HDRI. Los resultados dependen, de igual forma, del ajuste del parámetro **exposure_adjust** y la iluminación particular de cada imagen.


Figura 25

Vamos a mejorar la imagen añadiendo iluminación desde un foco. Para ello, volvemos al fichero de la escena con la que obtuvimos la imagen de la figura 22 (cuando aún no habíamos eliminado el foco). Asignaremos a toda la escena el mismo material, que no tendrá nada de brillo especular (ver figura 26). De igual forma, quitaremos las propiedades de reflexión y transparencia en raytracing. Definiremos el material de color blanco.


Figura 24

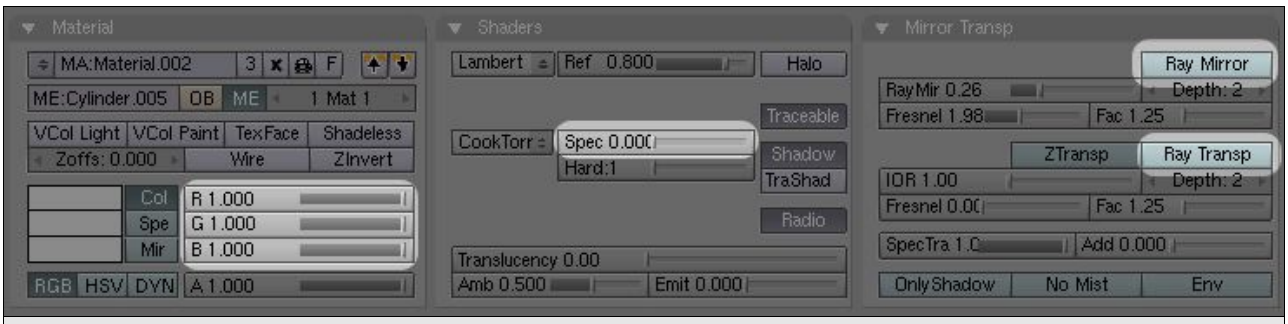


Figura 26

Figura 27

Figura 28

Sin cambiar los parámetros de render (y sin emplear imágenes HDRI), lanzamos la generación de la imagen con Yafray. Mediante este proceso obtendremos una imagen que guardará la información de las sombras en la escena. Como no hay diferencias en los materiales, podemos emplear esta imagen como mapa de sombras y componerla con la escena renderizada con el mapa HDRI (de la figura 24). Abrimos la imagen generada en este paso (correspondiente con la figura 27) con Gimp, y ajustamos el brillo y el contraste para obtener una imagen con la información de las sombras). Para ello, hacemos sobre la imagen, **Image**, **Colors**, **Brightness-Constrast**. Ajustamos el brillo a 70 y el contraste a 110.

Esta imagen resultado (figura 28), la compondremos con el resultado del render mediante HDRI, para obtener la imagen final de esta sesión (figura 30). Para ello, abrimos la imagen resultado del paso anterior en GIMP, creamos una capa nueva y pegamos en esta capa la imagen de la figura 28. Este paso puede verse en la figura 29.

Queremos componer la nueva capa como capa de sombras. Tenemos que indicar a GIMP que componga esta nueva capa, empleando el modo Multiply, que oscurece los colores de las capas inferiores (en este caso, la capa con el render HDRI, con la nueva capa de sombras. Además, ajustaremos la opacidad de la capa de sombras, para que el resultado sea más suave. Con esta técnica podemos ajustar el resultado final de nuestras imágenes dando matices particulares de iluminación, creando y componiendo varias capas.

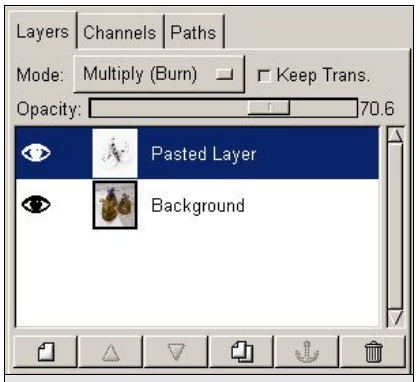
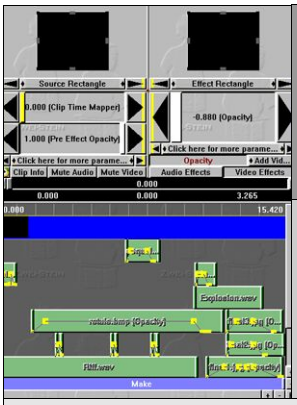


Figura 29

Figura 30



Práctica 12: Composición de Vídeo

Animación para la Comunicación ::: ESI-CR ::: UCLM

Carlos González Morcillo :: Carlos.Gonzalez@uclm.es :: http://www.inf-cr.uclm.es/www/cglez

En esta última sesión aprenderemos a manejar un programa de edición de vídeo: Zwei-stein. Sincronizaremos un conjunto de vídeos, con efectos de transciones y rótulos mediante imágenes.

Software: Zwei-Stein 3.01 **Fecha:** 27/Mayo/2004

Al ejecutar el programa nos encontramos con el interfaz que se muestra en la figura 1.

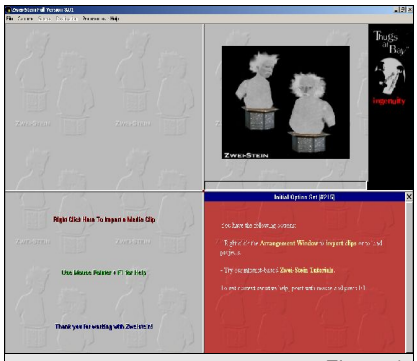


Figura 1

La ventana aparece dividida en 4 partes. Arriba a la derecha está la zona de previsualización. En esta parte podemos reescalar los vídeos que componemos, desplazarlos, etc... Además, ofrece una visión del resultado que obtendremos. Abajo a la derecha aparece la ventana de ayuda. Arriba a la izquierda está la zona de filtros y propiedades. Por último, abajo a la izquierda aparece la típica línea de tiempo de los programas de composición de vídeo.

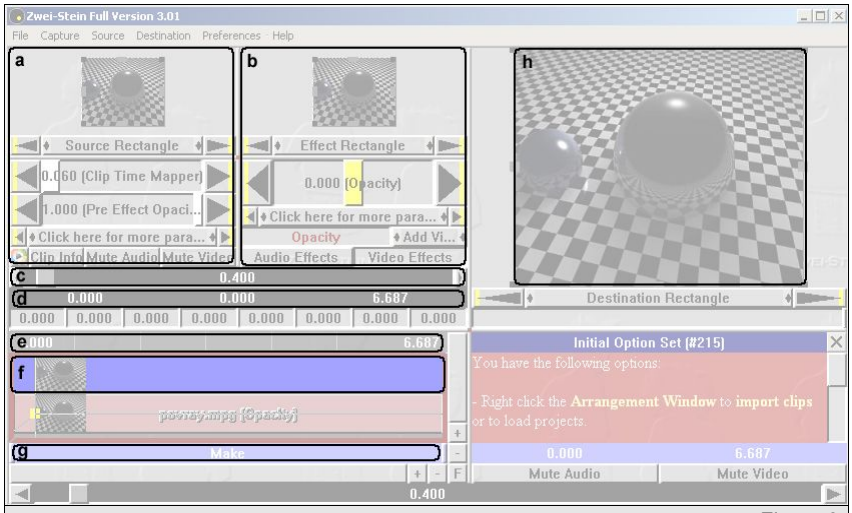


Figura 3

Para comenzar, cargaremos un vídeo haciendo sobre la zona de la línea de tiempo, **Import Video Clip**. Cargaremos "*povray.mpg*" (ver figura 2). Las imágenes y sonidos se importan con esta misma opción. La aplicación soporta AVI, BMP, JPG y WAV.

Antes de continuar cargando los vídeos, configuraremos el proyecto. En **Destination**, **Video Format** elegiremos **352 / 288 (CIF PAL)** , y en **Destination**, **Frames Per Second 25**. Guardamos el proyecto.

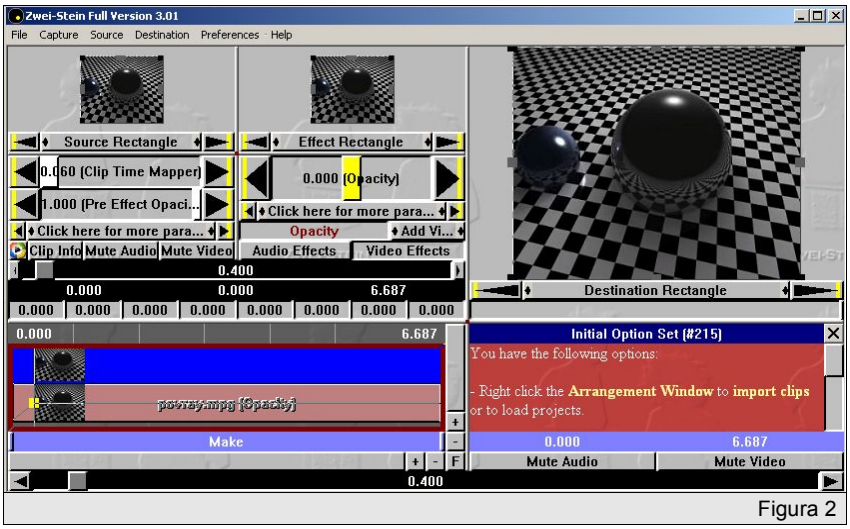


Figura 2

Podemos hacer "*zoom*" sobre la línea de tiempo pinchando en los botones **+** y **-** de su esquina inferior derecha. Pinchando en **F** el programa realiza el ajuste óptimo. Distinguimos nuevas partes del interfaz (ver figura 3). En **(a)** se definen los parámetros de entrada del vídeo (imagen o sonido). Entre otras cosas, podemos "recortar" únicamente la parte de un vídeo que nos interese utilizar. En **(b)** gestionaremos los filtros. En el vídeo de ejemplo, se han utilizado, principalmente, el de opacidad (**Basic effects**, **Opacity**) y el **RGB Key**, que nos permite definir qué color tomar como transparente (muy útil en rotulación). **(c)** nos define la posición, dentro del vídeo seleccionado, en que nos encontramos. Es un valor local. El valor global del vídeo se encuentra en la barra inferior de la ventana. La aplicación mide el tiempo en segundos (no en frames, como Blender).

Podemos acceder a un valor exacto

de tiempo haciendo doble clic con sobre la barra de tiempo. Podemos especificar un valor numérico exacto prácticamente sobre cualquier parámetro (barras deslizantes, etc...) con este método. **(d)** está formado por tres valores: punto de comienzo (en el tiempo general de la composición), comienzo local del vídeo (si lo hemos recortado por el principio, será distinto de cero) y punto final. Dando valores exactos a estos parámetros, podemos ajustar con total precisión las uniones entre partes. **(e)** permite desplazarnos por el tiempo general del vídeo.

Encima de **(e)**, podemos definir "*atajos*" para llegar a ciertas partes del vídeo. **(f)** ofrece una previsualización del frame actual. La barra **(g)** define el intervalo que vamos a generar de vídeo. Cuando realicemos pruebas, sólo generaremos el intervalo sobre el que estemos probando. Cuando queramos generar el vídeo completo, haremos sobre **(g)** **Auto Adjust Make Strip**. Por último, (h) ofrece una previsualización del resultado final y un mecanismo para reescalar y desplazar los elementos que intervienen en la composición.

¿Cómo gestiona Zwei-Stein los frames clave y las interpolaciones? Para esta aplicación, al igual que la mayoría de editores de vídeo, las interpolaciones entre frames clave son lineales. Por tanto, los parámetros cambiarán a velocidad constante entre claves definidas.

Para introducir un frame clave sobre cualquier parámetro, bastará con cambiar el parámetro en el frame actual. Por ejemplo, si quisiéramos introducir un degradado en la transparencia del primer vídeo, nos iríamos al segundo 0.0 y cambiaríamos el valor del filtro **Opacity** a -1. Después, avanzamos un poco (por ejemplo, segundo 0.5) y cambiamos el valor a 0. Observamos que aparece una línea de representación sobre el objeto vídeo, que indica la variación de ese parámetro. Los puntos amarillos indican frames clave.

En la ventana de previsualización también aparecen cuadros amarillos cuando tenemos un frame clave. Podemos realizar operaciones básicas sobre los frames clave en **(a)** y **(b)**, accediendo con . Generamos el vídeo resultado (delimitado por la franja **Make**) accediendo a **Destination**, **Export**, **Video for Windows** *.AVI. Un codec que da muy buen resultado es el de Intel (IYUV) o el Xvid con un buen ratio entre calidad y velocidad de compresión. Es importante conocer el destino del vídeo generado, ya que algunos códecs se ajustan mejor a su publicación online, y otros mejor a publicación multimedia.

Para terminar, en la figura 4 se muestra el esquema final resultante del vídeo generado en esta práctica.

⇨ Más editores de vídeo

Zwei-stein es un programa freeware pero no libre. Otra alternativa algo menos potente es el compositor de vídeo de Blender que se utilizó en prácticas anteriores. Para realizar operaciones sencillas con flujos de vídeo y audio, se puede utilizar **VirtualDub** http://www.virtualdub.org/, una aplicación para windows que permite cambiar el codec de una secuencia, cortar y pegar trozos, etc...

Los usuarios de **Linux** tienen varias alternativas libres. Sin duda, una de las más versátil es **Cinelerra**, http://heroinewarrior.com/cinelerra.php3, desarrollado por heroine warrior. Otra alternativa en linux, mucho más intuitiva de utilizar (aunque con menor funcionalidad) es **Kino**, descargable en http://kino.schirmacher.de/.

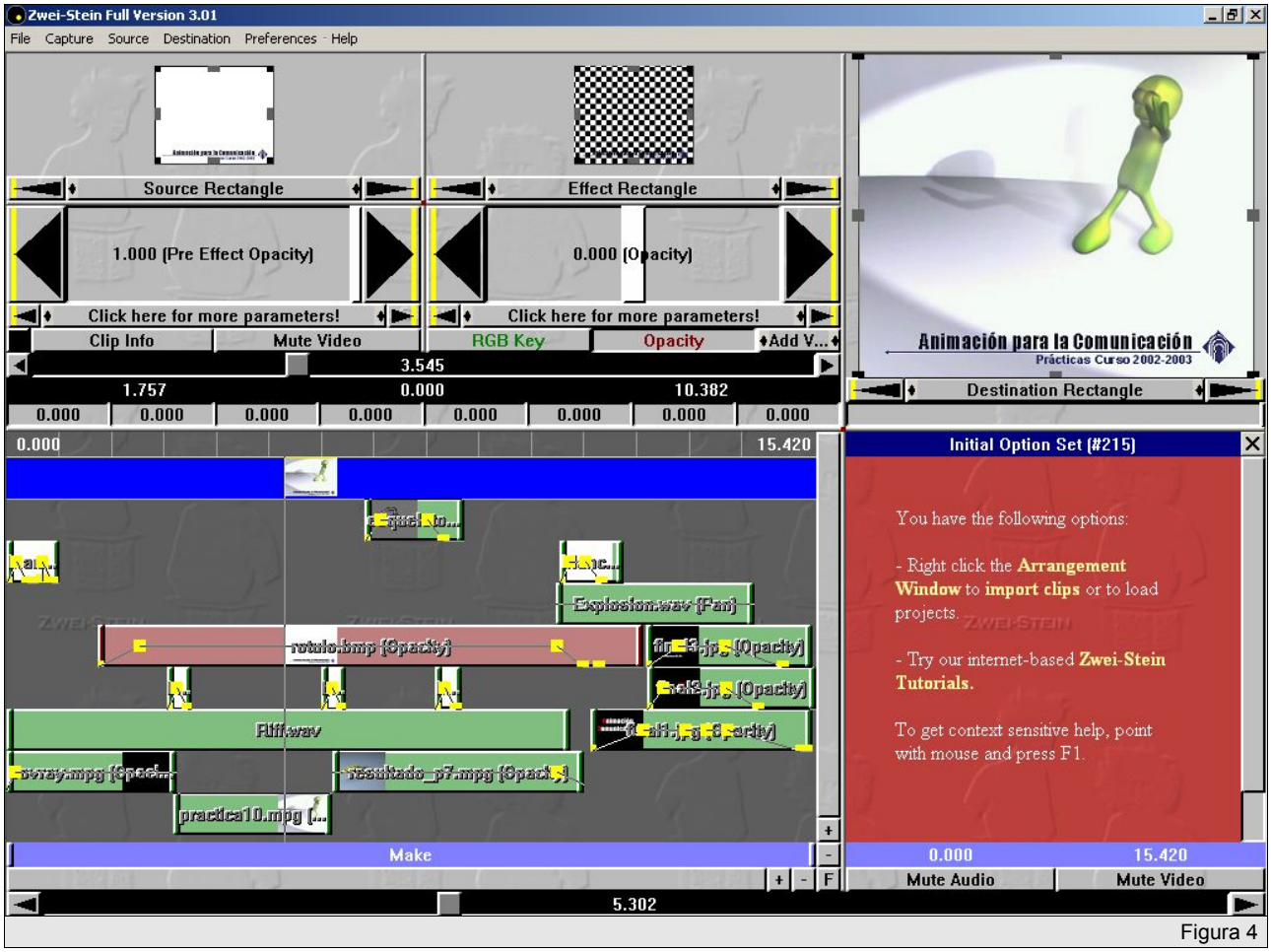


Figura 4