



# Introducción al Modelado Virtual 3D con Blender

**Carlos González Morcillo** (Carlos.Gonzalez@uclm.es)  
<http://www.inf-cr.uclm.es/www/cglez/cepcordoba/>

Grupo de Investigación ORETO


Escuela Superior de Informática  
Universidad de Castilla-La Mancha

---


## Práctica 05. Texturas y Foto-Realismo

---

**P**ara obtener resultados realistas es necesario establecer correctamente los parámetros de materiales y texturas. En esta sesión trabajaremos con texturas procedurales e imágenes para dotar del máximo realismo a una escena previamente modelada.

En esta sesión utilizaremos el modelo llamado “**escena.blend**”, que se encuentra junto con las texturas que se emplearán en el fichero comprimido “**ficheros\_practica05.zip**”. La escena que contiene este fichero se muestra en la figura 1. Al realizar un render  de esta escena comprobamos que ningún objeto tiene asignado ningún material.

Comenzaremos definiendo las propiedades del plano que servirá como mesa. Añadiremos a este plano una imagen como textura empleando mapeado ortogonal plano, definiendo además un pequeño nivel de reflexión (espejo), simulando una madera barnizada.

Con el plano que sirve de mesa seleccionado, en los botones de sombreado  (**Shading**), dentro de la pestaña de **Links and Pipeline** pinchamos en el botón **Add New** (ver Figura 2), para añadir un nuevo material al objeto. Hecho esto, el objeto tendrá un material con unas propiedades por defecto que modificaremos más adelante.

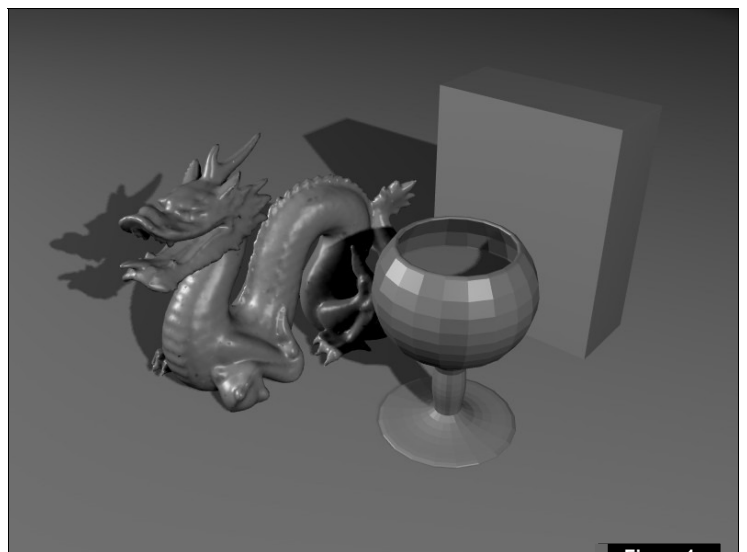


Figura 1


Como queremos utilizar una imagen como textura, en este caso no modificaremos el color del material. Para añadir una textura, pinchamos en el botón **Add New** de la pestaña **Texture** del mismo grupo de botones “**Shading**”. El resultado de pinchar en ese botón se muestra en la figura 3. De esta forma, se ha creado una textura por defecto (nombrada **Tex.001**) asociada al material anterior. Vamos a modificar esa textura indicando a Blender que cargue una imagen. Nos vamos al grupo de botones de textura , y en la pestaña de **Texture** indicaremos que el tipo de Textura es una Imagen (elegimos **Image** en la lista desplegable de **Texture Type**). Hecho esto, Pinchamos en el botón **Load** de la nueva pestaña **Image** y cargamos el fichero “*madera.jpg*” que contiene la fotografía de la textura de madera.



Figura 2



Figura 3

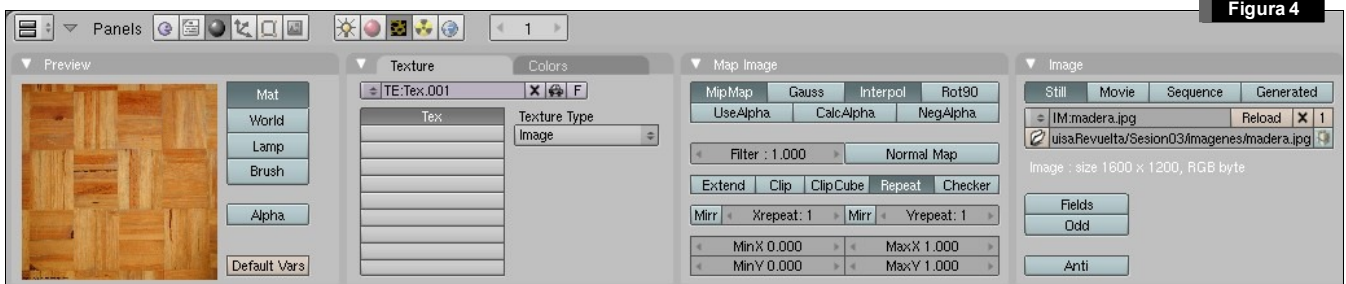


Figura 4


El resultado de la carga de esta imagen puede verse en la Figura 4. En la parte izquierda del interfaz puede verse la previsualización de la textura. Volvemos a las propiedades del material (el botón situado a la izquierda de ) y en la parte de previsualización del material vemos que aparece aplicada la nueva textura (Figura 5). Por defecto el mapeado ortogonal de Blender es de tipo plano, que es el que nos interesa en este ejemplo. Si realizamos un render de la escena, el resultado es el mostrado en la figura 6.



Figura 5



Figura 6

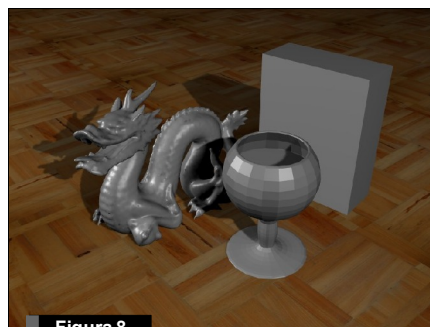


Figura 8

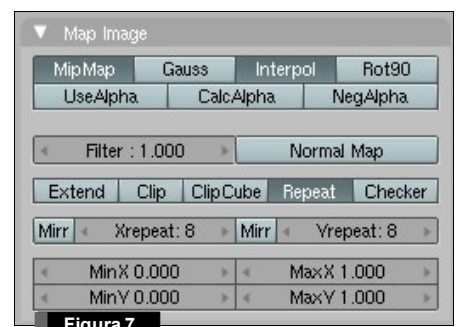



Figura 7

Los tablones de la textura son demasiado grandes, y la escena pierde realismo. Vamos a indicar a blender que repita la textura en el eje X y en el eje Y un número superior de veces. Para ello, volvemos a las propiedades de la textura , y en la pestaña **Map Image** (Figura 7) indicamos mediante el campo **Xrepeat** e **Yrepeat**, que el mapeado se realice mediante 8 repeticiones en cada eje. Si ahora realizamos un render, el resultado se puede ver en la figura 8.



Vamos a añadir al material una propiedad de reflexión de la luz (espejo). Volvemos a las propiedades del material  y en la pestaña de **Mirror Transp** activamos el botón de **Ray Mirror**. Si activamos este botón, el material se comportará como un espejo. El parámetro **RayMirr** indica la cantidad de reflexión de la luz, y el campo **Depth** indica la profundidad de rebotes en el proceso de trazado de rayos. Configuraremos el parámetro de **Depth** en 2 y **RayMirr** en 0.10 aproximadamente (ver Figura 9). Si renderizamos la escena, el resultado será el mostrado en la Figura 10.



Figura 9

Vamos a definir las propiedades del dragón. Seleccionamos este objeto y al igual que hemos hecho con la mesa, dentro de las propiedades de sombreado , pinchamos en el botón **Add New**. Vamos a utilizar una textura procedural para simular que el dragón es de mármol. Además, modificaremos los valores de reflexión especular para darle un mayor realismo.

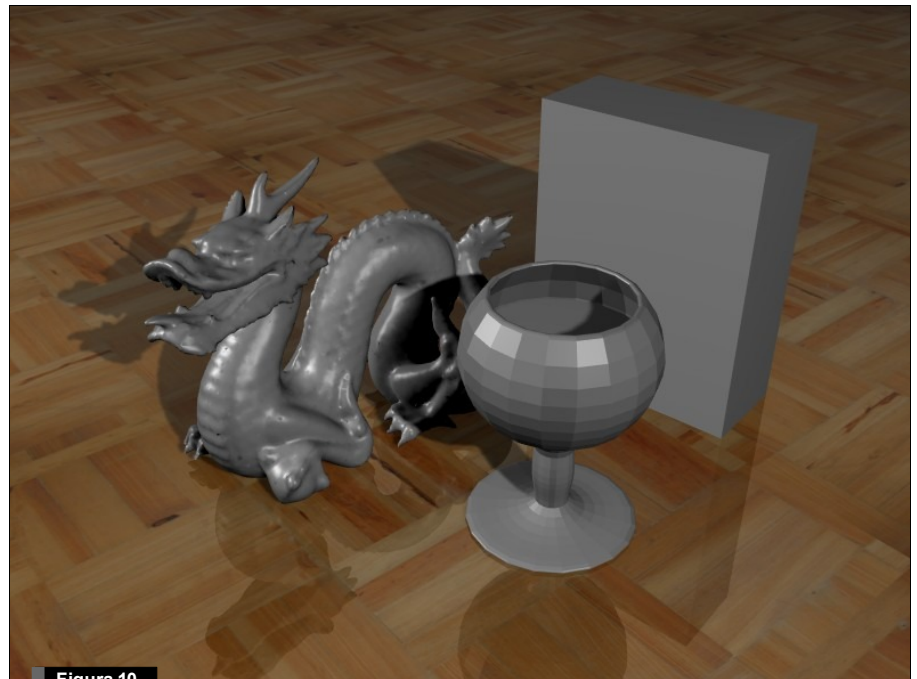


Figura 10

Siguiendo el mismo proceso que el utilizado para la mesa, añadimos una textura al dragón (Pinchamos en el botón **Add New** de la pestaña **Texture**), nos vamos a las propiedades de la textura y en este caso, en la lista desplegable de **Texture Type** seleccionamos **Marble**, y en las propiedades de la pestaña **Marble** pinchamos en el botón **Sharper** (Figura 11). El resultado de estas modificaciones puede verse en la pestaña **Preview**.

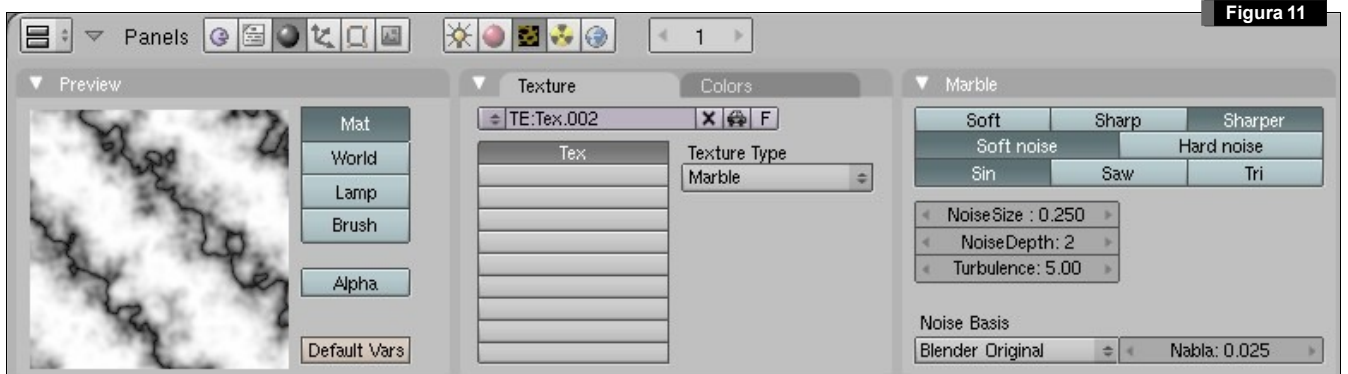


Figura 11

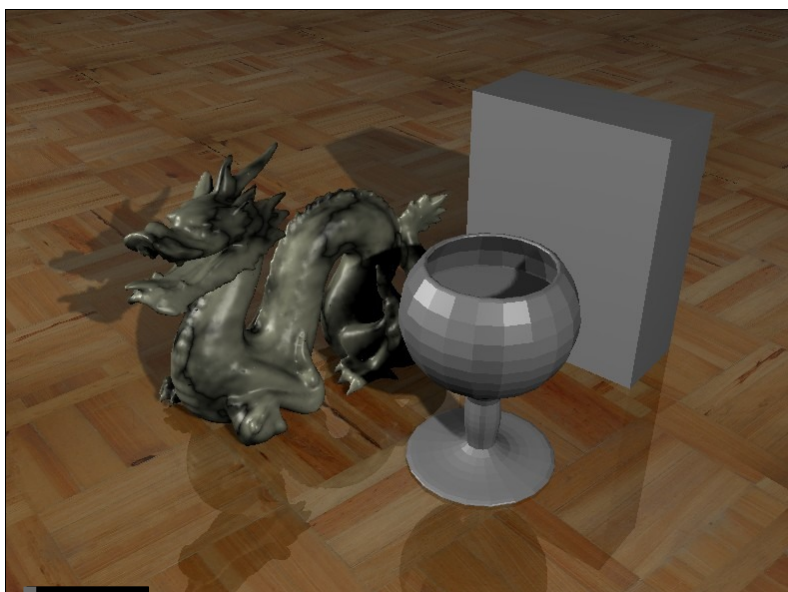


Figura 13



Figura 12

Queremos que esta textura procedural modifique el color del material. Volvemos a las propiedades del material y en la ventana de preview situada a la derecha vemos que, por defecto, el color

modificado es rosa. En la pestaña **Map To** indicaremos que el color de la textura será un amarillo claro, y el modo de mezclado lo cambiaremos a **Multiply** (en la lista desplegable), tal y como se muestra en la figura 12. Si realizamos un render de la escena en este momento, obtendremos un resultado como muestra la figura 13. Para que el material parezca mármol, vamos a modificar la cantidad de luz que refleja y la dureza del brillo especular, para ello, en la pestaña **Shaders**, modificaremos el valor **Spec** a 1.2 y **Hard** a 100 (ver Figura 14). Ahora el resultado del render es el mostrado en la figura 15. Al igual que hicimos con la mesa, podemos añadirle a este material un poco de reflexión (espejo).


A continuación definiremos las propiedades del material de la copa y el licor del interior de la copa, pero antes vamos a eliminar el efecto de “cara poligonal” de la copa, indicando a blender que el modelo de sombreado será suavizado. Con la copa seleccionada, y en los botones de edición , pinchamos en el botón **Set Smooth** de la pestaña **Link and Materials** (figura 16). El botón **Set Solid** sirve para volver al modo de sombreado plano. Hecho esto, el resultado del render sería el mostrado en la figura 17.



Figura 14

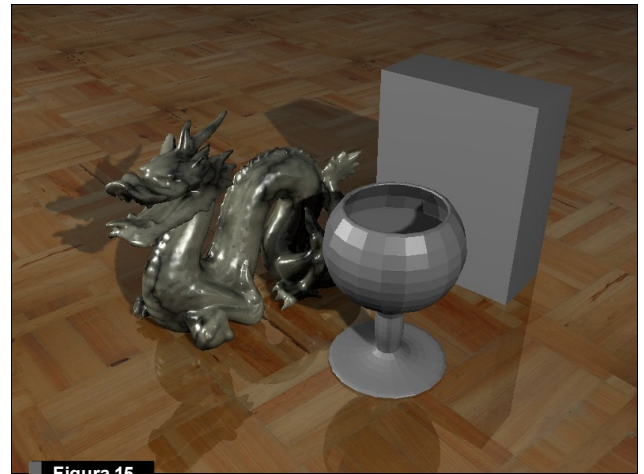


Figura 15

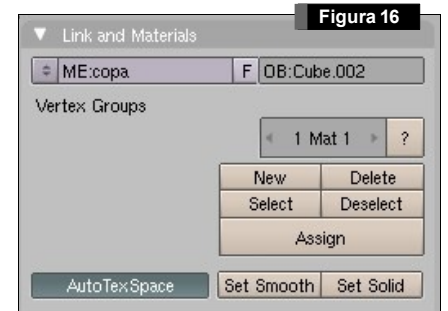


Figura 16

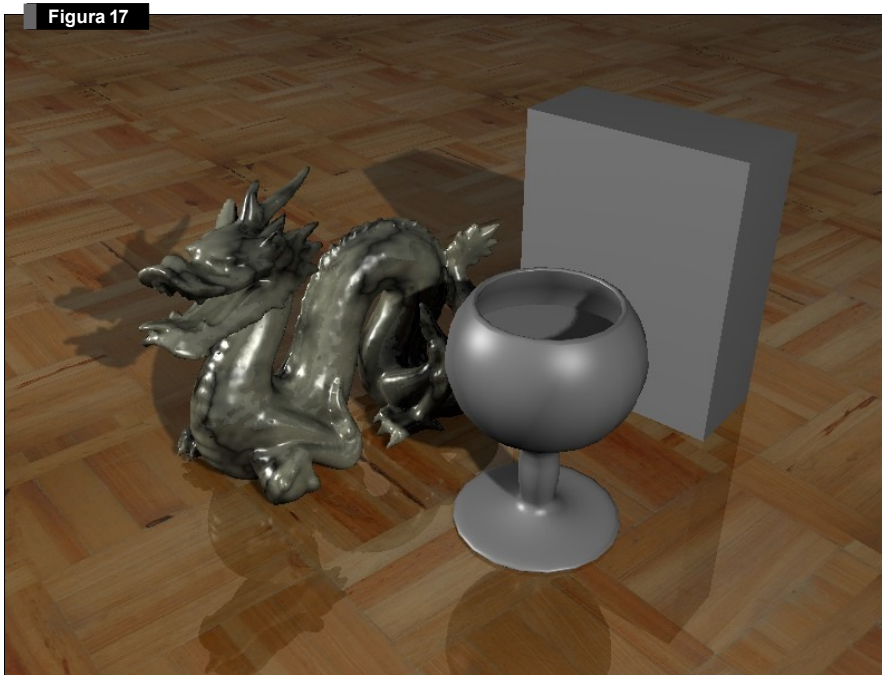


Figura 17

Como hemos realizado con los objetos anteriores, añadimos un material a la copa. En la copa, definiremos propiedades de transparencia y refracción de la luz. La transparencia se indica en el parámetro **Alpha**, situado debajo de las componentes RGB de la pestaña **Material** (ver figura 18). En nuestro caso, indicaremos a blender un valor de 0.2.

Además, activaremos el botón **Ray Transp** de la pestaña **Mirror Transp** (Figura 18), para que el método de render calcule la transparencia por trazado de rayos. El **Índice de Refracción (IOR)** para el vidrio es 1.5. Indicaremos además que la profundidad del trazado (**Depth**) es de 5 niveles de recursión.

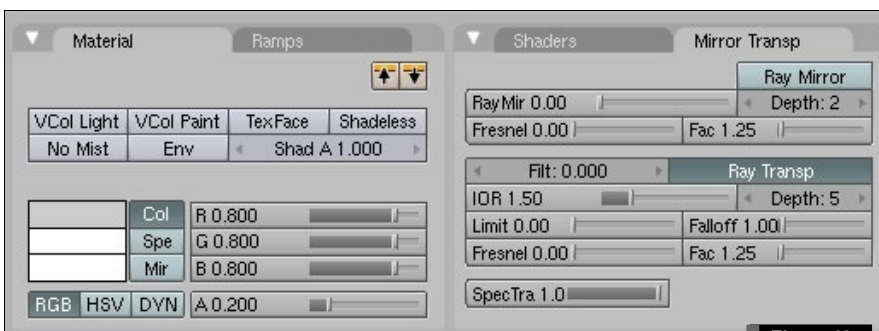


Figura 18

El resultado de renderizar la escena ahora se muestra en la Figura 19. Seleccionando el objeto del licor, activamos igualmente el modo de sombreado suave (**Set Smooth**), definiendo además los mismos pa-

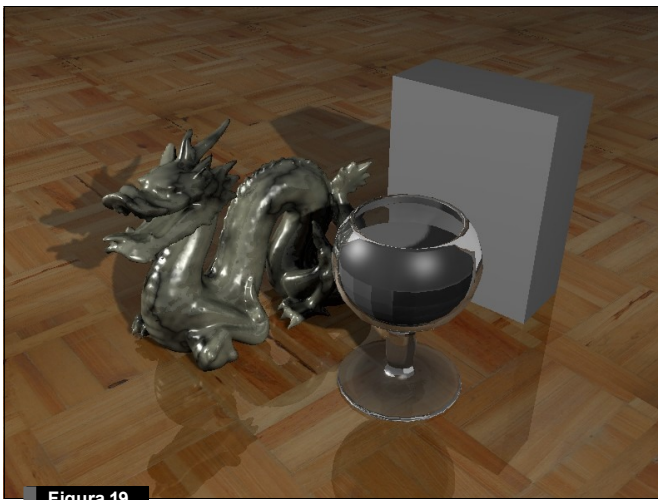

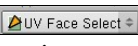


Figura 19



Figura 20

rámetros de **Alpha**, activando **Ray Transp** y con 5 niveles de recursión, pero indicando esta vez que el índice de refracción es 1.3 (el mismo que el agua), y configurando el color del licor como naranja. El resultado del render se muestra en la figura 20.

Para finalizar, añadiremos una textura empleando el mapeo UV a la caja. Seleccionamos la caja y añadimos un nuevo material. Podemos ver el resultado de aplicar la textura empleando este método si seleccionamos **Textured**  como método de representación en la cabecera de una ventana 3D. Con la caja seleccionada, elegimos el modo de trabajo **UV Face Select**  (directamente accesible pulsando la tecla **U**). Seleccionamos la cara frontal con el botón derecho del ratón y cambiamos una ventana 3D a tipo **UV Image Editor** (ver Figura 21).

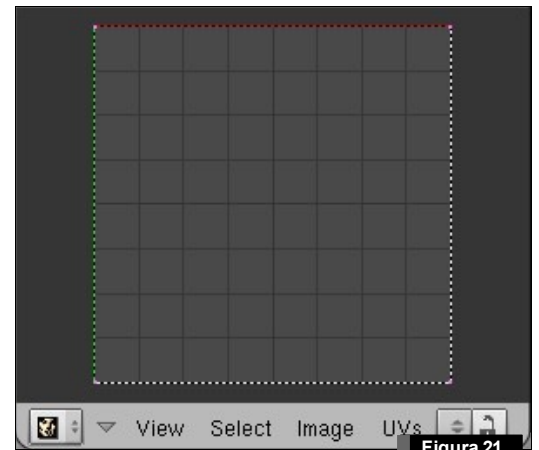


Figura 21


En la cabecera de esta ventana, elegimos en el menú **Image/ Open** y elegimos la imagen "cajacereales.jpg". Seleccionando todos los vértices en la ventana del UV Image Editor usando la tecla **A**, trasladamos **C** y rotamos **R** los vértices de esta ventana para que la representación en la vista 3D sea similar a la mostrada en la figura 22. Hecho esto, seleccionamos la cara superior y lateral de la caja de cereales en la vista 3D y procedemos de forma similar para asignar la textura, pero en esta ocasión no hay que cargar de nuevo la imagen. Se puede acceder a ella directamente pinchando sobre el icono  situado en la cabecera de la ventana **UV Image Editor**, que nos desplegará una lista de selección. El resultado del texturizado total de la caja de cereales se muestra en la figura 23.



Figura 22

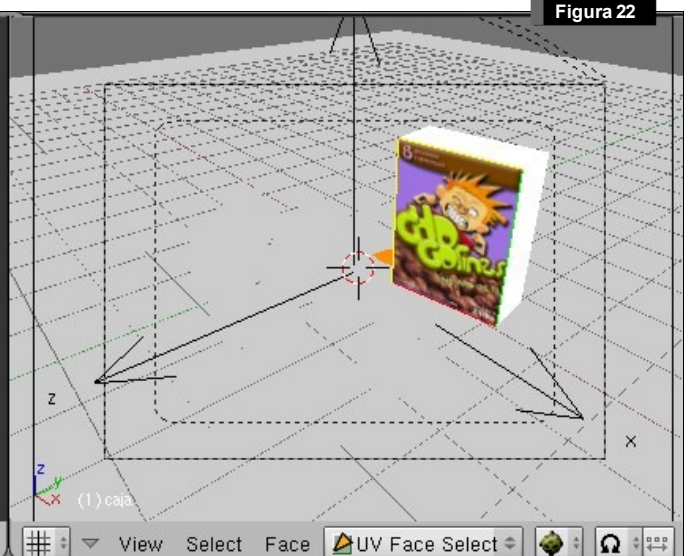
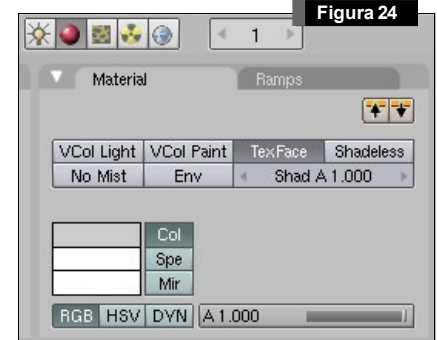


Figura 23



Figura 24




Volvemos al modo de edición de objeto (pulsando la tecla **E** o seleccionándolo en la lista desplegable de una ventana 3D). Para que las texturas aplicadas en este modo UV se muestren en el proceso de render es necesario activar el botón **TexFace** de la pestaña **Material** dentro de los botones de sombreado (**Shading**) (ver Figura 24). Finalmente renderizamos la escena y obtendremos un resultado similar al mostrado en la figura 25. En esta primera parte de la práctica hemos creado nuestros propios materiales; sin embargo podemos utilizar algunas librerías de materiales disponibles en la red.

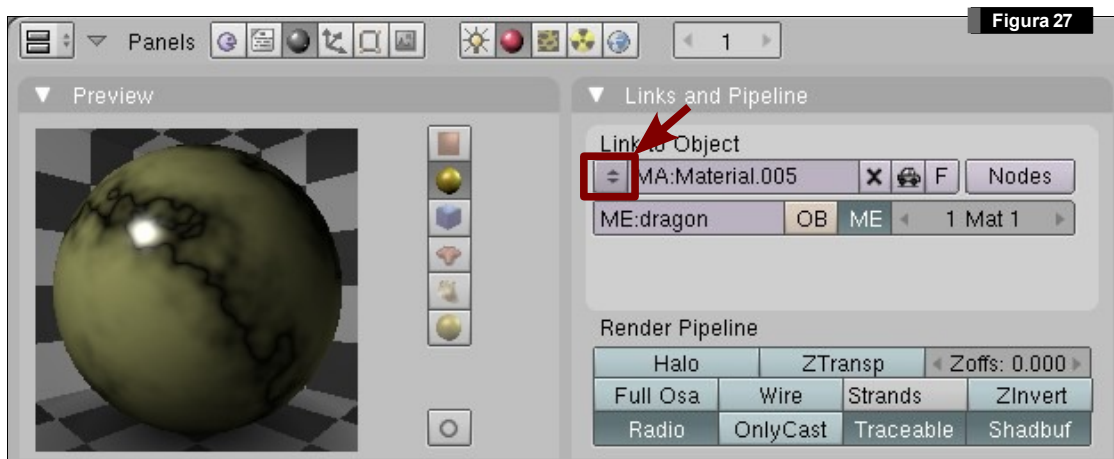


Figura 25

Por ejemplo, en la dirección <http://repository.parastudios.de/> existe un repositorio con multitud de materiales prediseñados, con una representación previa del resultado de su aplicación (ver figura 26). Nos descargamos el material “Gold”, de la categoría “Metal”. Ahora añadiremos el material “gold” cuya definición se encuentra en el fichero gold.blend que acabamos de descargar. Para ello, en nuestra escena pinchamos en el menú principal **File/ Append**. Seleccionamos el fichero gold.blend con el botón izquierdo del ratón, y aparece un listado con los elementos que podemos importar de este fichero. Nos vamos a material con botón izquierdo y seleccionamos (con botón derecho) el material *Gold*. Pinchamos en el botón **Load Library**.



Una vez hecho esto, el material de oro está incluido en nuestra escena. Para aplicarlo a un objeto (por ejemplo, al dragón), bastará con cambiarlo en la lista desplegable de materiales. Con el dragón seleccionado (en modo objeto), nos vamos a los botones de materiales , y pinchamos en la lista desplegable de materiales (ver figura 27). Elegimos el material *Gold*.



Si ahora renderizamos la escena, el resultado deberá ser similar al mostrado en la figura 28. El resultado es bastante realista, pero como vimos en la exposición de la primera sesión, para conseguir mejores resultados necesitamos métodos de representación basados en técnicas de Iluminación Global. Blender no incorpora este tipo de mecanismos en su motor de representación interno. No obstante, existe la posibilidad de utilizar motores de render específicos que calculan la iluminación global empleando modelos más complejos. Uno de estos motores, que además tiene soporte integrado en Blender es Yafray. A continuación renderizaremos nuestra escena empleando una configuración sencilla de los parámetros de este trazador.



Antes de configurar la escena para renderizarse con Yafray, debemos instalar este programa. En la página web del curso se puede descargar la versión de Windows y Linux. También puede obtenerse la última versión de la página web oficial del programa (<http://www.yafray.org>).

Una vez instalado el motor de render, pasamos a ver su uso básico. En la lista de motores de render (debajo del botón marrón de RENDER), seleccionaremos **Yafray** (ver figura 29). Hecho esto, aparecerán dos pestañas nuevas con parámetros de configuración de Yafray (ver figura 30).

En la pestaña **Yafray**, tenemos opciones generales de configuración; como el botón **AutoAA**, que permite a Yafray seleccionar automáticamente el nivel de AntiAliasing, la profundidad máxima del trazado de rayos **Raydepth** y ajustes particulares de exposición de la luz, etc...



Figura 29

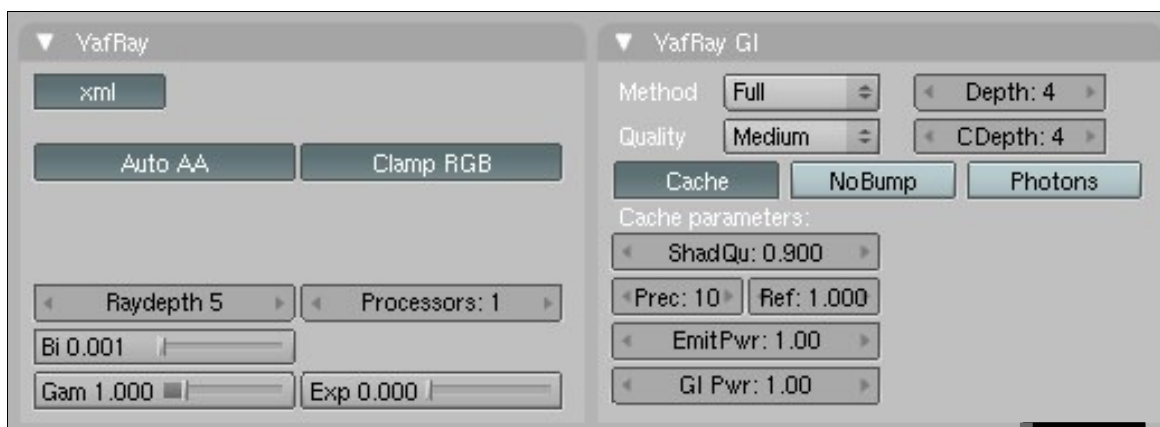


Figura 30

Si disponemos de un computador con más de 1 procesador, podemos indicar aquí el número de procesadores que queremos que utilice yafray (a nivel de hilos). El botón superior de color azul **xml** permite exportar el fichero que yafray toma de entrada. Para ver la evolución del render, este botón tendrá que estar desactivado. Podemos parar el render en cualquier momento pulsando la tecla ESC.

En la pestaña de iluminación global (**Yafray GI**), en principio presenta dos listas de selección. En la lista de método de iluminación **Method**, podemos elegir entre **Skydome**, en el que la luz se calcula proveniente únicamente desde el cielo, o **Full**, que realiza un cálculo de iluminación global completa (con iteración de la luz entre objetos). Emplearemos el método Full por ser más realista. La lista de calidad **Quality** ajusta, según el parámetro seleccionado, las variables que necesitará yafray para renderizar la escena. En definitiva es el número de muestras de luz que calculará. El parámetro **EmitPower** situado en el inferior del panel permite ajustar la intensidad de las fuentes de luz de la escena. En algunas escenas habrá que ajustar este parámetro (normalmente incrementarlo) para obtener un valor de iluminación correcto. Finalmente tendremos que especificar dos nuevos parámetros; **Depth** (rebotes de la luz entre objetos) y **CDepth** (rebotes de la luz dentro de objetos translúcidos para obtener caústicas).

Si activamos el modo **Cache**, aceleraremos los cálculos empleando *Irradiance Cache*. Esta cache nos evita tener que calcular la iluminación global en cada punto; selecciona algunos píxeles para realizar los cálculos y después interpola el valor de iluminación. En **ShadQu** indicamos la calidad de las sombras; con valores más altos obtenemos mejores

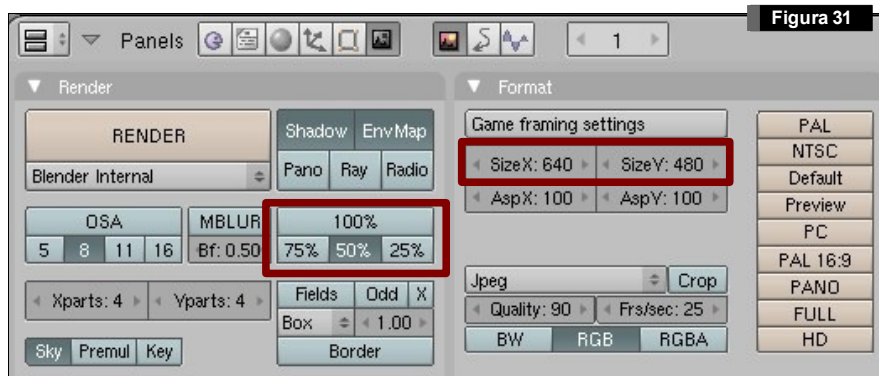


Figura 31



resultados (naturalmente, con mayor tiempo de cálculo). **Prec** indica la precisión del píxel (un valor más bajo implica mayor calidad). **Ref** es un valor umbral, donde especificamos el cambio de intensidad que vamos a permitir entre muestras. Valores más pequeños dan resultados más precisos.

Empleando este motor de render, con los parámetros mostrados en la figura 30 (el botón XML desactivado), el resultado obtenido se muestra en la figura 32. Es interesante realizar pruebas a baja resolución, ya que el tiempo de render suele ser alto. La resolución de la imagen final, como vimos en la sesión 1, se selecciona en los botones de escena en las cajas de texto SizeX y SizeY. La resolución final puede modificarse empleando los botones azules 100%, 75%, 50% y 25% (ver figura 31).

Existe multitud de documentación en la red para profundizar sobre el uso de este fantástico motor de Render. Un buen punto de comienzo es el tutorial en castellano “Yafray como motor de render integrado en Blender”, disponible en la dirección <http://wiki.yafray.org/bin/view.pl/UserDocEs/Gaurav-GuideEs> . Otro buen documento también en castellano es la descripción de los parámetros de Render de Yafray, disponible en <http://wiki.yafray.org/bin/view.pl/UserDocEs/GiTuto> .



Figura 32