

Guía rápida de *blender* 3D

O lo que viene a ser lo mismo: un pliego de instrucciones para perder el miedo a Blender y asimilar los conceptos mínimos en los que se estructura una aplicación 3D. Desconozco si alguna vez ha pasado por vuestras manos una herramienta de modelado y animación 3D. En caso afirmativo, la mayor parte de los procedimientos que expliquemos a continuación no os tienen que suponer un gran esfuerzo. Si, por el contrario, nunca hemos “jugado” con un programa 3D, al principio puede hacerse algo complicado entender el funcionamiento del mismo. En parte, requiere un mínimo de intuición espacial (vamos a trabajar en un entorno tridimensional) y cierto entrenamiento para lograr los primeros resultados.

Esta guía tiene como doble objetivo explicar los conceptos fundamentales de la creación de gráficos 3D y conocer el funcionamiento de una herramienta específica de este área: **Blender**. La elección de Blender parte de una creencia muy sencilla: considero que la curva de aprendizaje de esta herramienta, en comparación a otras, es la más elevada. En poco tiempo se pueden conseguir trabajos notables y, por supuesto, es una herramienta que, con una sintaxis muy básica, ofrece unas enormes posibilidades creativas.

No he pretendido hacer una guía exhaustiva del programa (para eso ya está en desarrollo el manual). Sí, en cambio, una mirada rápida y atenta de sus posibilidades. Blender es una herramienta en desarrollo y, como tal, puede ser la presente guía: una suma de fichas de carácter siempre introductorio y acumulativo.

Espero que disfrutéis de este aprendizaje y que podáis sacar el máximo de rendimiento de un programa tan atractivo como adictivo.

Javier Belanche Alonso,
Barcelona, septiembre de 2001

Introducción

Cualquier persona que abra Blender por primera vez tendrá la misma sensación que todos tuvimos en su momento: una presentación diferente, con un millón de botones de distintos tamaños repartidos por la pantalla y un aspecto nada familiar respecto a la típica aplicación windows. Para colmo, si decidimos “chafardear”, tocando aquí y allá, nada parece respondernos. Incluso la opción de “salir” del programa es casi imposible de encontrar. Reconozco que Blender no tiene una interficie estándar, pero también está comprobado que, una vez conocido su mecanismo de trabajo, ofrece una mayor rapidez de realización. Veréis que, con unas pocas funciones, pueden resolverse problemas de diferente índole.

Blender es una completa herramienta de modelado, editor de materiales y texturas, animación y editor de juegos. Todo un servicio sin coste alguno. Sí, es gratis. Nadie tiene que pagar licencia. Por supuesto, no es una herramienta que está a la altura de otras, pero claro, no pagas 2000\$ y los resultados son de una excelente calidad.

Blender nace con el propósito de ser una herramienta de 3D de trabajo de campo. Su filosofía es tan sencilla como su manejo: ser prácticos y optimizar el máximo rendimiento de trabajo. Se supone que una herramienta te ha de facilitar un entorno lo suficiente flexible para adaptarte en cada momento a la demanda del cliente o a tus necesidades. La importancia de Blender reside en que la herramienta no es el fin del aprendizaje: el objetivo de Blender es que puedas traducir en 3D cualquier idea o fantasía que puedas pensar o imaginar.

En cada ficha podrás comprender, de forma progresiva, la interficie y sintaxis de Blender. No voy a explicar cada botón con detalle (esto lo podéis consultar en el manual) pero sí una gran parte de ellos serán comentados para poder seguir con coherencia las fichas de aprendizaje que componen esta guía.

En la **Ficha I** se estudia la **navegación** por la interficie de Blender.

En la **Ficha II** se enseña cómo podemos aplicar las **transformaciones básicas** (movimiento, escala y rotación) de los objetos en Blender.

En la **Ficha III** se explica la forma de **añadir objetos** en la escena.

En la **Ficha IV** estudiamos el funcionamiento de las **luces** en Blender

En la **Ficha V** conoceremos los **materiales, texturas y el motor de render** de Blender.

En la **Ficha VI** se estudia la animación básica a partir de los **keyframes** y del **path**.

En la **Ficha VII** nos acercaremos al editor **gameBlender** para la construcción de juegos.

En la **Ficha VIII** veremos el funcionamiento elemental de **Python** como script en Blender.

En la **Ficha IX** estudiaremos cómo Blender calcula la **radiosidad**.

Y por último, en la **Ficha X** hablaremos de diferentes técnicas de modelado en Blender, concretamente la técnica de la **extrusión, spin, spin dup, subdivision surfaces y decimator**.

Finalmente, os recuerdo que Blender es una herramienta que permite al usuario crecer con ella. Esta afirmación os puede parecer algo exagerada, pero la mayoría de personas que han trabajado con la herramienta, os dirán decir algo parecido. Por otro lado, Blender está en continuo desarrollo: si visitáis la página oficial (www.blender.nl) podéis encontrar una abundante información de las mejoras de cada actualización. Ahora estamos en la versión 2.2x. y se espera que a finales de este año, **NAN** –la empresa creadora de Blender- haga saltar la serie 2.5x. Para una mayor información, leed el índice A del manual (la evolución de Blender).

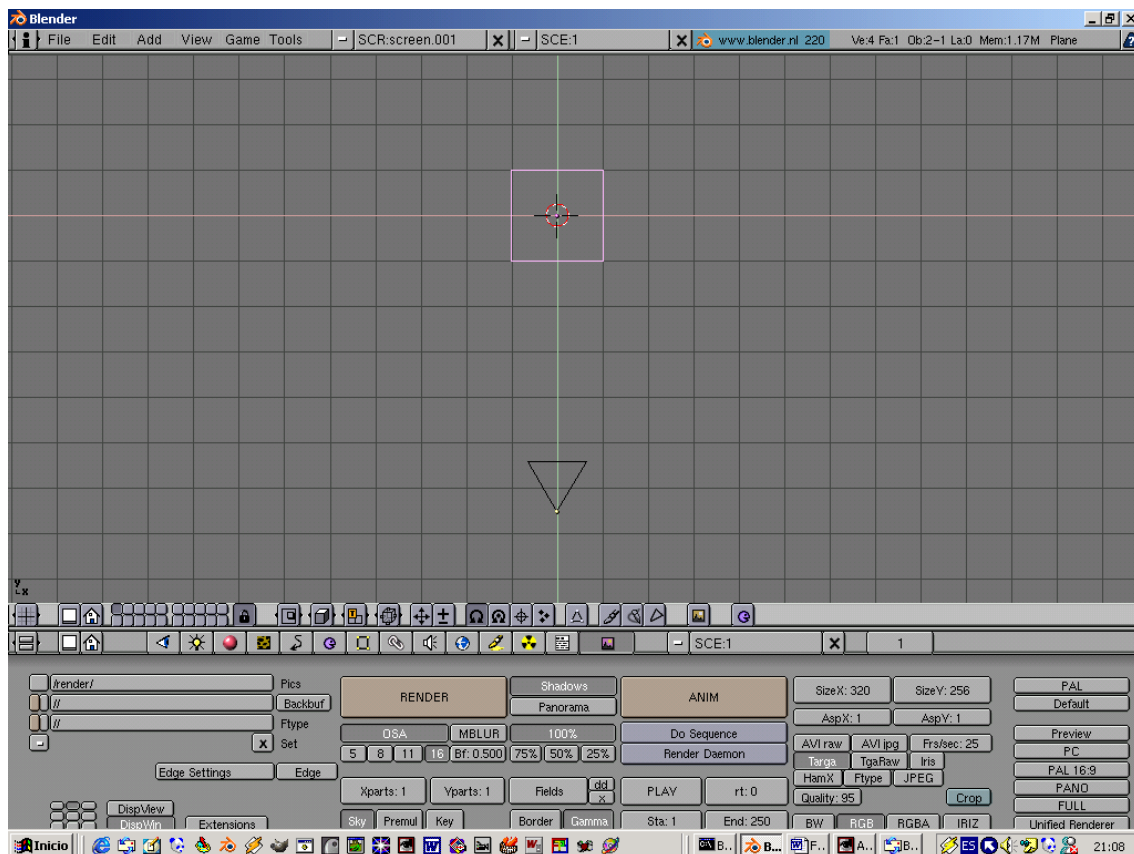
FICHA I. NAVEGAMOS POR LA INTERFICIE DE BLENDER

Antes de ponerme a explicar cómo podemos navegar por la interficie de Blender, es recomendable que conozcáis la regla de oro de Blender (*Golden Rule of Blender*):

“Keep one hand on your keyboard and one on you mouse”

traducido sería algo así como: **pon una mano sobre el teclado y otra en el ratón**. ¿por qué? Blender tiene como principio optimizar el rendimiento de tu trabajo. Cualquiera que sea la acción que activemos en Blender, tiene su correspondencia con una tecla (**hotkey**). Esto no es nada nuevo. Al contrario, cualquiera que se especialize en un programa, acaba por configurar sus **hotkeys** para ir mucho más rápido y evitar tanto movimiento de ratón. Trabajar desde un principio con **hotkeys** puede resultar un tanto pesado y caótico, pero en poco tiempo comprobarás tú mismo la lentitud con la que trabajas con otros programas y la rapidez con la que ejecutas acciones en Blender.

Bueno, menos rollos y vamos a trabajar con Blender: abrimos el programa y, en teoría, deberíamos ver la siguiente imagen en la pantalla del ordenador:



Reconozco que el principio no es nada alentador. Tal como os dije, la parte inferior está repleta de botones de diferentes tamaños y colores. En la ventana gráfica aparece dos símbolos: un cuadrado rosado y un triángulo, y no sabemos muy bien qué puede significar. Finalmente, como cierta dosis de alivio, podemos identificar una barra de menús en la parte superior izquierda.

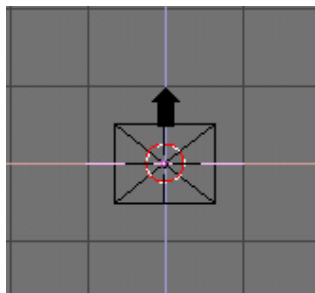
Vamos a ir poco a poco. Empezaremos por la ventana 3D:

- El cuadrado rosa representa al plano estándar de Blender. **Si tiene un color rosa, significa que el plano está seleccionado.**
- El triángulo invertido con un punto amarillo en su vértice inferior representa a la **cámara**.

- c. Las finas líneas negras forman la cuadrícula o rejilla (**grid**). Nos serviremos de ella para poder transformar los objetos de la escena con mayor precisión.
- d. El círculo de color rojo-blanco que está situado justo en el centro del plano es el **puntero 3D**. Con él podremos determinar el lugar donde situaremos nuestros objetos en la escena. Con el **Botón Izquierdo del Ratón (BIR)** podemos cambiar su posición: lleva el cursor a cualquier parte de la ventana 3D y pulsa **BIR**.

Ya que estamos explicando la ventana 3D, vamos a dar unas cuantas instrucciones de cómo podemos navegar por ella. Una penúltima advertencia: la mayoría de nosotros trabajamos bajo entorno Windows Microsoft y tenemos un ratón de dos botones->**BIR (Botón izquierdo del ratón)** y **BDR (Botón derecho del ratón)**. Blender nació bajo entorno Linux, lo que hizo que Blender se optimizara para trabajar con tres botones. Nosotros vamos a trabajar con dos. Al final de la guía podéis encontrar la relación de acciones con los dos tipos de ratón. Vamos a mover el punto de vista o ángulo de visión:

- a. Si miramos la pantalla y sabemos un poco de geometría descriptiva, podemos reconocer una vista ortogonal. Es decir, que los rayos de proyección son perpendiculares a la ventana 3D y, por lo tanto, podemos ver los objetos que estén dispuestos de forma paralela a la ventana 3D en su verdadera magnitud: si observamos el plano, comprobamos que podemos determinar la longitud exacta de cada arista. Las proyecciones ortogonales nos ayudan a poder trabajar de forma muy precisa en el modelado de objetos en la escena.
- b. La vista que tenemos activada corresponde a una vista de **planta**. Nuestro punto de vista es superior.
- c. Vamos a cambiar la vista a otra llamada **frontal**, es decir, desde un punto de vista de frente. Para ello pulsaremos una vez la tecla **Key Pad 1 (corresponde al teclado numérico de la izquierda del teclado. Si no funciona, debemos activarlo pulsando la tecla Bloq Num)** y la imagen que encontraremos en la pantalla será la siguiente:

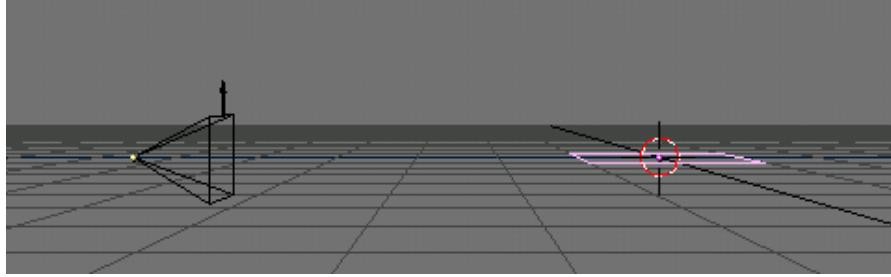


- d. El plano, desde el punto de vista frontal, sólo aparece como una arista y la forma de pirámide con una flecha indicando una dirección superior es la cámara.
- e. ¿Y la **vista lateral**? Muy sencillo: pulsamos una vez **Key Pad 3** y obtendremos la siguiente imagen en pantalla:



- f. En la **vista lateral** seguimos viendo el plano como arista y la cámara como un triángulo tumbado.
- g. Fijaos que en las vistas ortogonales podemos saber con precisión la distancia que hay de la cámara al plano, la altura de cada uno de los dos respecto a los ejes (explicaremos más adelante) o, como hemos dicho más arriba, las verdaderas dimensiones de los objetos.

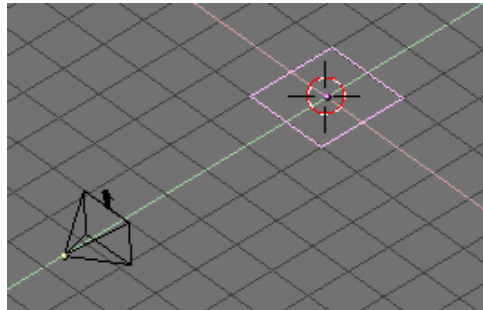
- h. ¿Qué ocurre si ahora pulsamos **Key Pad 5** y movemos el ángulo de visión mediante la tecla **Alt** y el **BIR** apretado ? Tendremos que ver la siguiente imagen:



- i. Acabamos de activar el **modo perspectiva**. ¿Qué significa? A diferencia del modo ortográfico, el modo perspectiva intenta representar la imagen como si fuera vista por el ojo humano. Es un modo de visualización más que de trabajo. Fijaos que, en este modo, las líneas de la cuadrícula, en lugar de ser paralelas, todas ellas convergen en un punto conocido como **punto de fuga**. Las diferentes vistas (lateral, planta y alzado) también trabajan con el modo perspectiva.
- j. Si queremos volver al modo ortográfico del principio y activar la vista superior, pulsaremos **KeyPad 5->KeyPad 7**

Aparte de estas vistas predeterminadas, hay una infinidad de vistas para poder trabajar en Blender, como también podemos aproximarnos o alejarnos (acción **zoom**) o imitar la acción de mover una hoja con la mano (acción **Pan**) para desplazarnos lateralmente por la pantalla.

- a. Si mantenemos de nuevo la tecla **Alt** y movemos el cursor con el **BIR** apretado, **modificamos el ángulo de visión** de la ventana 3D, tal como hemos realizado más arriba. Un posible cambio sería el siguiente:



- b. Esta posible vista también se identifica como **vista axonométrica**. Es una visión muy parecida a la vista en modo perspectiva, con la diferencia que, en la primera, las líneas de la cuadrícula se mantienen paralelas.
- c. Si lo que queremos es conseguir el efecto de **zoom**, sólo tenemos que mantener apretadas las teclas **Ctrl+Alt** y movemos el cursor con el **BIR** apretado hacia arriba, si queremos acercarnos a la imagen- o hacia abajo –si lo que queremos es alejarnos.
- d. Si lo que queremos ahora es **desplazarnos lateralmente por la pantalla**, como si moviéramos la imagen con la mano, pulsaremos mantenidas **Shift+Alt** y moveremos el cursor con el **BIR** apretado hacia cualquier dirección.

Redefiniendo las ventanas.

Blender te permite configurar las dimensiones de tus ventanas a tu gusto como crear otras nuevas. Por ahora, empezaremos cambiando el tamaño de la ventana 3D.

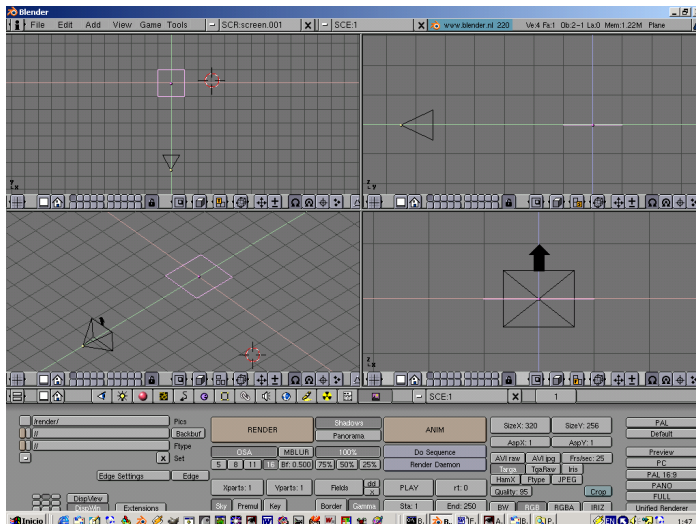
- Si situamos el cursor sobre la barra que sirve de frontera entre las dos ventanas (la ventana 3D y la ventana llamada **display buttons**), comprobamos que el cursor adquiere la forma de flecha de dos direcciones, con lo que nos indica que, si mantenemos pulsado el **BIR** y movemos el nuevo cursor, estaremos cambiando las dimensiones de la ventana.



- Ahora que ya sabes la manera de poder redimensionar las ventanas de trabajo, vamos por la segunda parte: dividir y unir ventanas (**split areas/join areas**).
- Situamos de nuevo el cursor en la frontera entre las dos ventanas de trabajo hasta que aparezca la doble flecha como cursor. Apretamos el **BDR** y nos aparecerá muy cerca del cursor la siguiente ventana flotante:

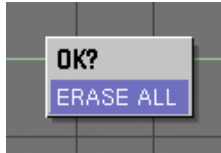


- Tenemos dos posibilidades: **Split Area** significa que dividimos en dos el area seleccionada. ¿Cuál tenemos seleccionada? Fíjate que el gris de la barra de botones de la ventana de **Display buttons** está más iluminada que la barra que pertenece a la ventana 3D (si queréis comprobarlo, mover el cursor de ventana a otra en la parte de las barras de botones y podéis comprobar que, si pasamos el cursor a la barra de botones de la ventana 3D, ésta se iluminará o cogerá un tono de gris más claro y viceversa)
- Si llevamos el cursor a la opción **Split**, aparecera una línea muy fina que se moverá siguiendo el cursor. Es la futura barra divisoria entre las dos ventanas. Una vez situada, hacemos un click con el **BIR** y nos aparecerá la nueva ventana.
- Podemos hacer tantas divisiones como permita el espacio mínimo de la ventana: una clara división de ventanas con la que estamos acostumbrados a trabajar en otras herramientas 3D es la siguiente:



La imagen final representa el primer ejercicio práctico. Es muy sencillo: ir dividiendo las ventanas y moverse dentro de las vistas para conseguir un resultado lo más parecido a la imagen. Por otro lado, se sobreentiende que **JOIN** significa **unir** dos ventanas en una. El segundo ejercicio se basa en volver a conseguir la presentación inicial de Blender uniendo las ventanas divididas.

Si no nos sale a la primera el ejercicio anterior, disponemos de una salida rápida: **CTRL+X**. Esta acción nos permite borrar todo el contenido de la escena y empezar de cero.



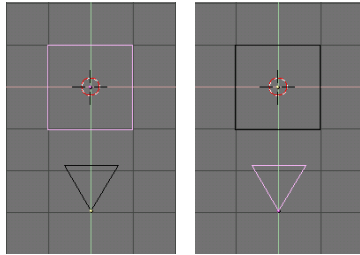
FICHA II. TRANSFORMACIONES BÁSICAS: SELECCIÓN, MOVIMIENTO, ROTACIÓN Y ESCALADO DE OBJETOS.

Para comenzar a trabajar, reiniciaremos Blender (**Ctrl+X**) y miraremos de explicar cómo funciona la selección en Blender y cómo podemos aplicar a los objetos de la escena las transformaciones geométricas básicas, es decir: el movimiento, la rotación y el cambio de escala del objeto.

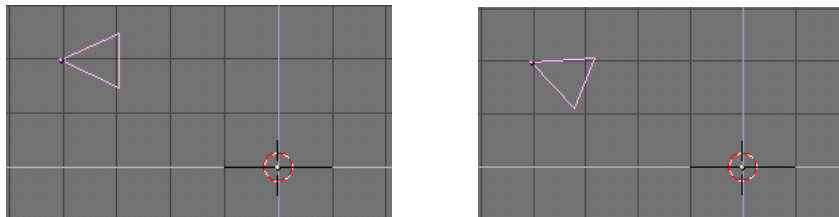
La selección:

Dijimos en la primera ficha que, si el plano aparecía de color rosa, significaba que estaba en modo selección, es decir, que podemos aplicar diferentes acciones al objeto como, por ejemplo, moverlo o rotarlo. ¿Qué ocurre si quiero seleccionar la cámara de la escena, por ejemplo, y moverla? Vamos por partes:

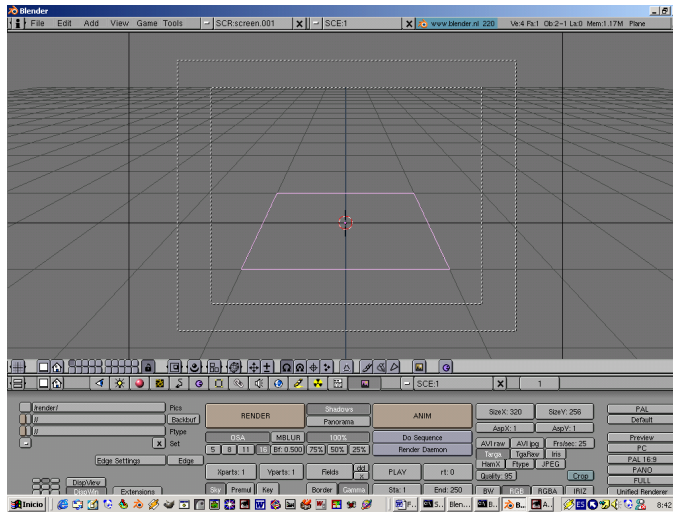
- Un objeto que esté seleccionado en la escena siempre aparecerá de color rosado. El plano está seleccionado, por lo tanto, tendrá un color rosa.
- Para seleccionar otro objeto de la escena debemos llevar el cursor sobre el objeto que queramos seleccionar y haremos un click con el **BDR**.
- Para probar lo dicho, llevamos el cursor a la cámara y pulsamos **BDR**: la cámara quedará seleccionada, cambiando su color a rosa y el plano, al quedar deseleccionado, quedará de color negro.



- En las ilustraciones se muestra cuando el plano está seleccionado (imagen de la izquierda) y cuando la cámara está seleccionada (imagen de la derecha).
- Aprovechemos que tenemos la cámara seleccionada para **moverla** por la vista de **planta**.
- Si tenemos seleccionado un objeto -en este caso la cámara-, y pulsamos una vez la tecla **G** (de **grabber** o arrastre), el color del objeto se volverá blanco, por lo que indica el inicio de una acción. Una vez pulsada la tecla sólo tenemos que desplazar el cursor por la pantalla y comprobamos que el objeto lo sigue. Una vez que queramos acabar la acción de mover, pulsamos el **BIR**. **Si lo que queremos es “romper” la acción, pulsamos la tecla ESC o el BDR.**
- Una indicación y al mismo tiempo una ayuda: si iniciáis una acción, es muy aconsejable que el cursor esté muy cerca del objeto seleccionado en el momento que la iniciéis.
- Vamos con la acción de **rotar el objeto**. Seguimos con la cámara y, para la acción de rotar, previamente activaremos la **vista de perfil** pulsando **Key Pad 3** y moveremos la cámara hacia arriba mediante **G**.
- Para rotar la cámara tal como vemos en la imagen, pulsamos una vez la tecla **R** (de **rotate**), desplazamos el cursor y rotamos el objeto en cualquier dirección.



- j. Una vez que hayamos conseguido rotar la cámara, vamos a descubrir una nueva vista: **la vista de cámara**. Para activar la vista de cámara, pulsa **Key Pad 0**. El resultado será el siguiente:



- k. He aprovechado desde la vista de cámara para seleccionar el plano y así poder aplicarle la acción de escalar.
- l. La acción de **escalar** se activa mediante la tecla **S** si el objeto está seleccionado. Una vez pulsada, para modificar las dimensiones del objeto sólo tenemos que movernos con el cursor.
- m. Con el plano seleccionado, podeís también moverlo y rotarlo en cualquier dirección.

Seguramente algunos se han preguntado: “De acuerdo, esto está muy bien, pero **¿cómo consigo restringir el movimiento o escalado de un objeto en una sola dirección o en la dirección de un eje?** La respuesta viene ahora.

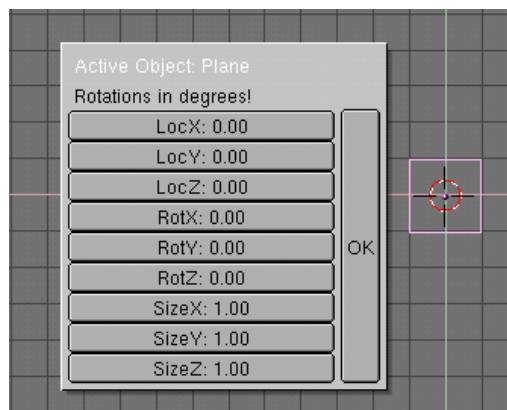
Para una mejor comprensión de cómo funcionan las restricciones siguiendo una dirección de los ejes, vamos a reiniciar la escena (**CTRL+X**). Seleccionamos el plano:

- Estamos en la **vista de planta**. Con el plano seleccionado pulsamos **G** y movemos el cursor: el plano se mueve.
- Si nos movemos de derecha a izquierda o viceversa, mantenemos pulsado la tecla **Alt** mientras movemos el cursor y pulsamos una vez el **BIR**, restringimos el movimiento del plano en la dirección del eje X.
- Si nos movemos de arriba a abajo o viceversa, mantenemos pulsado la tecla **Alt** mientras movemos el cursor y pulsamos una vez el **BIR**, restringimos el movimiento del plano en la dirección del eje Y.
- Fijaos que en los dos casos, una vez queda restringido el movimiento, si movemos en cualquier dirección el cursor, el plano siempre se desplazará siguiendo la dirección del eje.
- Para conseguir dominar esta acción se necesita un poco de práctica. No desesperéis si no os sale a la primera ni a la cuarta.
- Vamos ahora a experimentar con el **cambio de escala con restricción en una dirección**.
- Para conseguirlo, simplemente hemos de seguir el mismo proceso del movimiento en una sola dirección, pero con la diferencia de escalar (**S**) en lugar de mover (**G**) el objeto.

¿Y si queremos una mayor precisión en las transformaciones geométricas del objeto?

Blender no es una herramienta CAD, es decir, una herramienta de dibujo técnico asistido por ordenador. Pero no por ello carece de unas funciones específicas para lograr este objetivo:

- Con el plano seleccionado, moverlo (**G**) y, una vez en movimiento, pulsamos y mantenemos apretado **CTRL**. Veréis que el objeto se mueve por unidades de cuadrícula.
- Soltemos **CTRL** y, sin dejar de mover el plano, pulsamos y mantenemos apretado **SHIFT**. El plano se moverá de forma más precisa a nivel de unidades de rejilla (por lo que se moverá más lentamente).
- Las dos acciones anteriores sirven tanto para **escalar** y **rotar**.
- Si lo que queremos es mover, rotar o escalar introduciendo nosotros las unidades, pulsamos **N** y aparecerá la siguiente ventana:

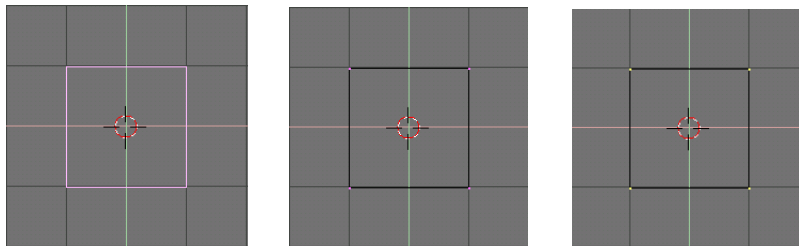


- Para acceder a los botones de **LocX, LocY, LocZ, RotX,...** debemos llevar el cursor a cualquiera de ellos y, con **Shift** apretado, pulsamos **BIR** y ya podemos introducir el valor que queramos.

ANEXO I: **EDIT MODE Y SELECCIÓN DE VÉRTICES**

Antes de seguir con la ficha III, hemos de hacer un alto en el camino y explicar un apartado que resultará crucial para poder continuar con el aprendizaje de Blender. Hablamos del modo **Edit Mode** y la selección de vértices en Blender.

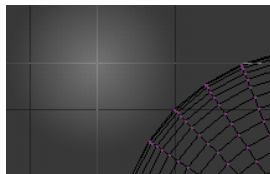
- Blender tiene dos formas de presentar la geometría en la escena: **modo edición objeto y modo de edición de vértices**. A este cambio de edición se le conoce como **EDIT MODE**.
- Por ahora, mientras trabajábamos con el plano, estábamos trabajando en modo edición objeto. **¿Qué ocurre cuando queremos trabajar con los vértices del plano?**
- Tenemos seleccionado el plano (recordad que la selección en modo objeto viene mostrada por el color rosa) y queremos acceder a su geometría básica. Para ello pulsamos una vez **TABULADOR (TAB)** y podemos ver los cuatro vértices que la forman



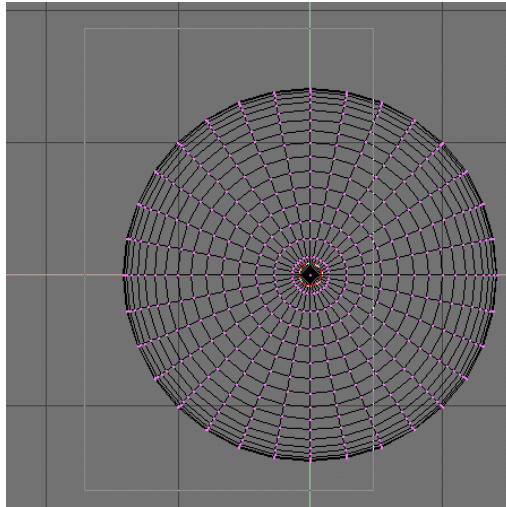
- Fíjate en las tres imágenes: la de la izquierda representa el plano seleccionado en modo edición objeto; la central representa que hemos activado, pulsando **(TAB)**, el modo edición de vértices. Los cuatro vértices quedan dibujados de color de rosa: en este caso, no significa que los cuatro estén seleccionados. Vamos a seleccionarlos todos pulsando la tecla **A**. La tecla **A** **selecciona o deselecciona todos los objetos de la escena**. Una vez pulsada la tecla **A**, obtendremos la imagen de la derecha, donde los cuatro vértices del plano aparecen seleccionados de color amarillo.

Ahora viene una de las partes más bonitas de Blender: **la selección de vértices**, tal cómo lo habíamos anunciado en un principio.

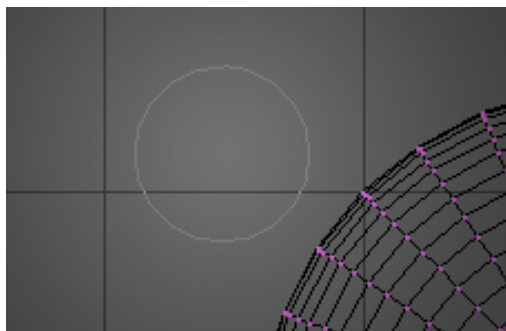
- Cómo ya hemos señalado en el apartado **d**, la tecla **A** nos ayuda a seleccionar o deseleccionar todos los vértices de la escena.
- Si lo que queremos seleccionar es un vértice, llevaremos el cursor al vértice que queramos seleccionar y pulsaremos **BDR**.
- Fíjate que si quieres seleccionar otro vértice, teniendo seleccionado el anterior, en el momento de seleccionar el nuevo vértice, el primero quedará deseleccionado. ¿Cómo remediar este problema? Seleccionamos los vértices, uno a uno, pulsando **BDR**, pero manteniendo apretada la tecla **SHIFT**.
- Pero **¿qué ocurre si queremos seleccionar grandes áreas de vértices?**
- Con la tecla **B** **activamos la selección rectangular**. Nuestro cursor se transformará en un puntero de cruz, tal como vemos en la siguiente imagen.



- f. Llevamos el puntero de cruz a una posición determinada de la ventana y pulsamos **BIR** **mientras movemos el cursor en diagonal para dibujar la ventana de selección.** Una vez dibujada la ventana de selección, soltamos BIR y quedarán seleccionados los vértices comprendidos en ella.



- g. Si lo que queremos es tener un modo de selección más libre, tenemos la posibilidad de “**pintar la selección**”. Si pulsamos dos veces **B**, es decir **BB**, nuestro cursor se convertirá en un puntero circular tal como podemos ver en la siguiente imagen



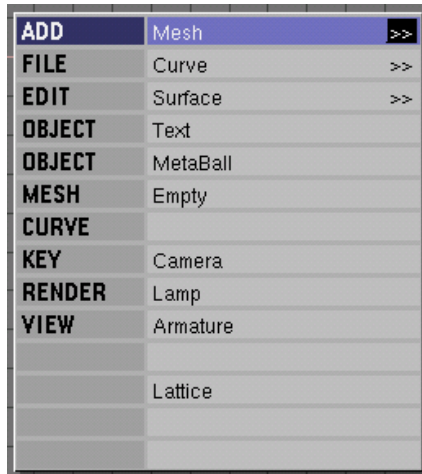
- h. Si queremos cambiar el diámetro del cursor, podemos hacerlo mediante las teclas de **+/- del KeyPAD.**
- i. Para efectuar la selección, sólo tienes que **mantener apretado el BIR mientras te mueves por encima de los vértices y quedarán seleccionados los vértices “pintados”.**
- j. **Procedemos del mismo modo para deseleccionar, pero con la tecla ALT apretada.**

Si tenemos seleccionados los vértices o sencillamente, uno de ellos, podemos moverlos en grupos de selección, escalar partes seleccionadas de los objetos, etc. Esto lo desarrollaremos en la próxima ficha.

FICHA III. LA CAJA DE HERRAMIENTAS (TOOLBOX). AÑADIMOS OBJETOS A LA ESCENA.

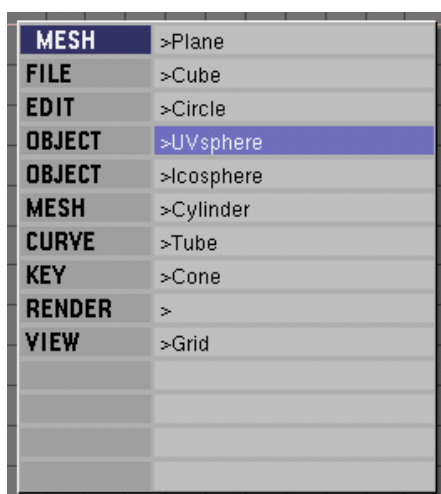


La mayoría de funciones en Blender tiene su correspondiente acceso mediante el teclado o el ratón. La caja de herramientas contiene gran parte de estas funciones. Puedes llamarla siempre que lo desees a partir de la **barra de espacio (SPACE)** o a través del icono con el símbolo de interrogante que se encuentra en la esquina superior derecha de la interficie de Blender.

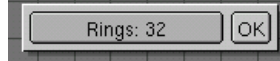


Vamos a explicar un poco cómo funciona la caja de herramientas de Blender.

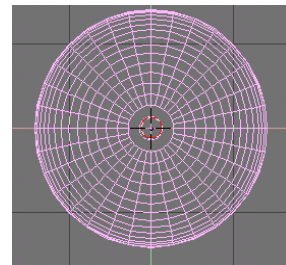
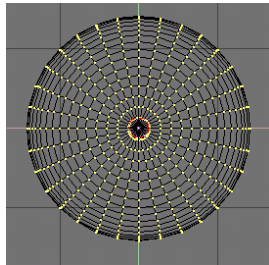
- La columna de un gris más intenso –la primera empezando por la izquierda- pertenece a la categoría de funciones de Blender.
- Si movemos el cursor por encima de cada una de ellas, cambiará las opciones de la columna de un color más claro. La categoría seleccionada a la izquierda estará iluminada por un azul oscuro, mientras la función que seleccionemos con el cursor (al pasar por encima) estará iluminada por un azul más claro.
- Si tomamos como ejemplo la imagen superior de la caja de herramientas, estamos en la categoría **ADD (añadir)** y en la función **Mesh (Malla poligonal)**.
- Si alguna de las funciones, como la del ejemplo anterior, tiene al final de la fila el símbolo **>>**, significa que esa función contiene un submenú al que podemos acceder mediante un click del **BIR**. Si hacemos click sobre **Mesh**, obtenemos las siguientes opciones en la caja de herramientas



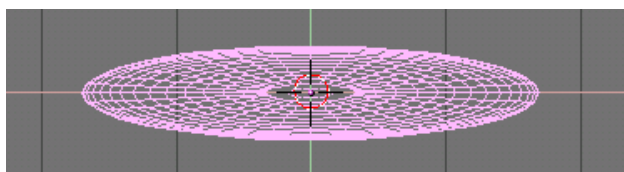
- e. De todas ellas, hemos seleccionado **UVSphere**. ¿Qué significa entonces la operación **ADD->Mesh->UVSphere**? Significa que nosotros queremos añadir (**add**) una esfera poligonal (**mesh->uvsphere**) en la escena actual.
- f. Si hacemos click en **UVSphere** con el **BIR**, desaparecerá la caja de herramientas y Blender nos preguntará la resolución del objeto. Recuerda que a mayor resolución gráfica del objeto, mayor es la memoria que consumimos de nuestro ordenador.



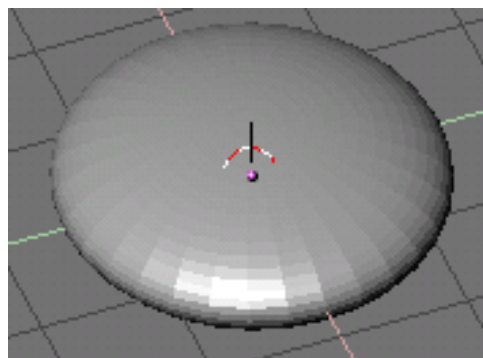
- g. Para modificar los valores de **segments** o de **rings** (segmentos y anillos que define la Uvesfera), bastará con situarse con el cursor encima del botón, pulsar **BIR** y, con el **BIR pulsado**, nos movemos con el cursor hacia un lado u otro (a la izquierda si queremos disminuir el valor o derecha, si lo que queremos es aumentarlo).
- h. Para introducir valores por el teclado, basta con situarse con el cursor sobre el botón y pulsar **BIR**.
- i. Finalmente, pulsamos **OK** para asignar los valores de segmentos y anillos de la Uvesfera y aparecerá en la escena actual la Uvesfera.



- j. De acuerdo con lo que vimos en el **apéndice de la ficha II** dedicado al modo **editmode**, la Uvesfera aparece en pantalla en modo edición de vértices todos seleccionados.
- k. Para salir del modo edición de vértices a modo edición de objeto, pulsa **TAB** y tendremos la esfera seleccionada (de color rosa) como objeto
- l. Ahora es el momento de mover, rotar y escalar la esfera en la escena. No dudes en jugar a escalar el objeto en una sola dirección para conseguir nuevas formas de la esfera. Aquí tienes unos ejemplos. Puedes añadir más objetos a la escena como **tubos** o **pirámides** y deformarlos.



¿Cómo podemos activar el sombreado (**shade**) en la ventana 3D? Muy fácil: con la tecla **Z** podemos activar modo de presentación de alambre o modo **shade** o **sombreado**, visible en la ilustración de la derecha.



ANEXO I:

EL puntero 3D de BLENDER. SNAP!

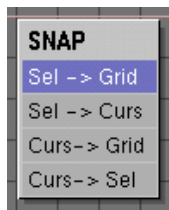


Este anexo tiene como finalidad mostrar la importancia del puntero 3D como una excelente herramienta de trabajo. Vamos a explicar sus propiedades:

- El puntero es un "elemento" de posición: su función primera es indicarnos siempre el lugar donde situaremos los objetos en la escena.
- Para desplazar el cursor por la ventana3D sólo tenemos que hacer click con el **BIR** sobre la posición escogida con el cursor.
- Recuerda que el puntero sólo se desplaza en dos dimensiones. En algunos casos necesitarás mover el puntero en dos vistas diferentes.

SNAP menu:

El menú **SNAP** (podemos traducirlo como menú de "ajuste" o "magnetizar") te permite situar el cursor o el objeto seleccionado en el punto más cercano de la cuadrícula o directamente en la posición del puntero 3D. Para abrir el menú **SNAP**, pulsa **SHIFT+S**.



- Sel->Grid**: nos desplaza la selección al punto de la cuadrícula más próximo.
- Sel->Curs**: nos desplaza la selección a la posición del puntero 3D.
- Curs->Grid**: nos desplaza el puntero 3D al punto de la cuadrícula más próximo.
- Curs->Sel**: nos desplaza el puntero 3D al centro de la selección.

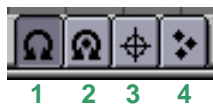
ANEXO II

Centros de gravedad, centros de rotación y más centros:

¿Qué es un centro de gravedad? Es el punto propio de cualquier objeto en la escena por el que se tomará como referencia en el momento de rotar el objeto, escalar o realizar otras funciones.

Se representa como un punto de color rosa cuando el objeto está seleccionado o de color amarillo cuando el objeto no está seleccionado. Cuando añadimos un objeto a la escena, su centro de gravedad se encuentra justo en el centro geométrico del objeto. El centro de gravedad de una esfera será el centro de la misma esfera.

¿Podemos utilizar un centro diferente al centro de gravedad para acciones de rotación o escala? Sí. Sólo tenéis que activar el botón que representa al puntero en el menú de botones (tal como se ve en la ilustración) de la ventana 3D (en su parte inferior).

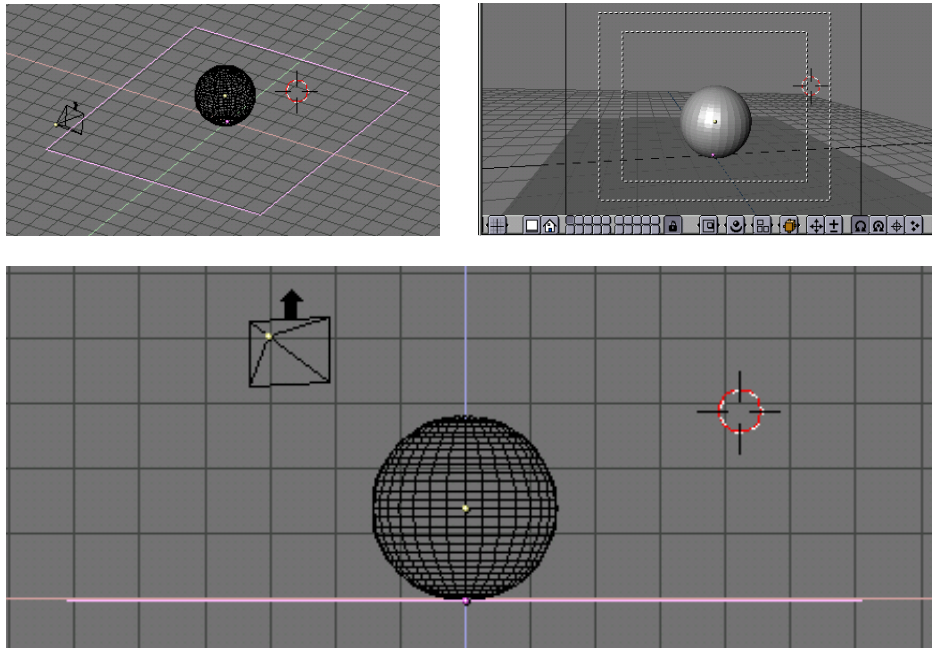


- El giro se realiza sobre el punto medio de la caja que, supuestamente, contiene el objeto.
- Rotación sobre el centro de gravedad.
- Rotación sobre el puntero.
- Si hay varios objetos seleccionados, cada uno gira sobre su centro de gravedad propio

FICHA IV. Luces y cámara.

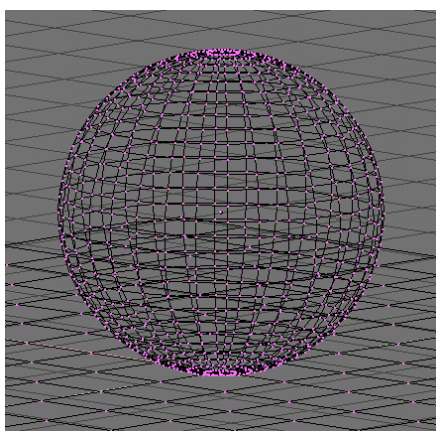
Añadir luces a la escena es un paso básico para poder representar y visualizar nuestros modelos. En esta ficha enseñaremos cómo se añaden una o varias luces, una manipulación básica de las mismas y, finalmente, explicaremos aspectos básicos de la cámara.

Para comenzar, vamos a trabajar a partir de esta escena básica. Aquí tenéis tres vistas: una vista axonométrica, una vista de cámara y, debajo de ellas, la vista frontal.

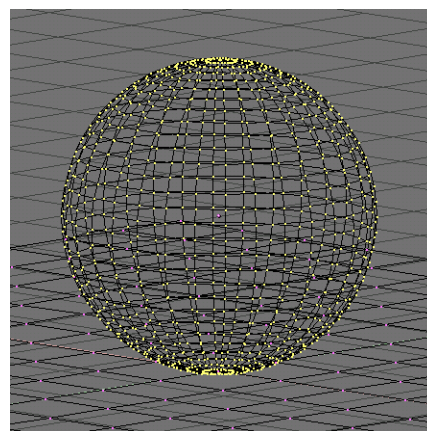


Dejamos al lector el trabajo de “montar” la escena sin ninguna ayuda debido a la sencillez de la misma.

NOTA: Es aconsejable que, si queremos añadir más objetos a una misma escena, siempre trabajemos con **modo edición objetos**. ¿Por qué? Si añadimos el plano, éste aparece en modo edición de vértices y, si bajo este modo añadimos la esfera, los dos objetos estarán “unidos” o “agrupados” (**join**) y no los podremos trabajar por separado. **Si queremos separar un objeto de otro tendremos que seleccionar el objeto a separar en modo edición de vértices, pulsar P (Separate) y confirmar (OK).** Vamos a ver un ejemplo práctico:



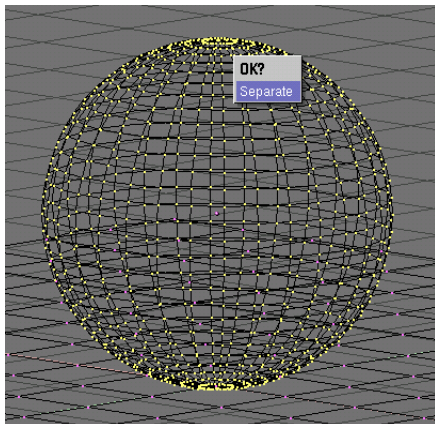
1



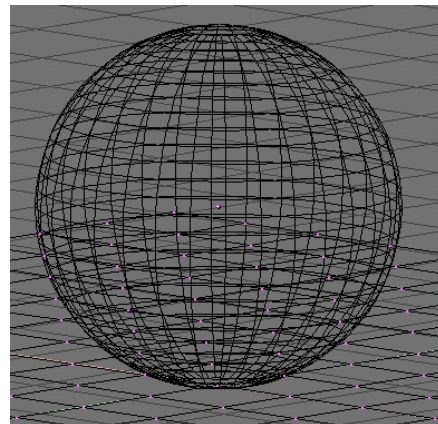
2

En la figura 1 tenemos la esfera y el plano en modo edición de vértices debido a que los dos objetos están unidos o agrupados.

En la figura 2 seleccionamos sólo la esfera. ¿Cómo? Llevamos el cursor sobre cualquier parte de la esfera y pulsamos **L**. Esta función es muy práctica para seleccionar de forma única objetos “agrupados”.



3



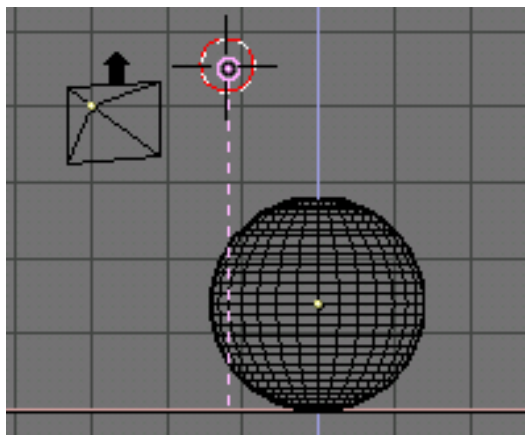
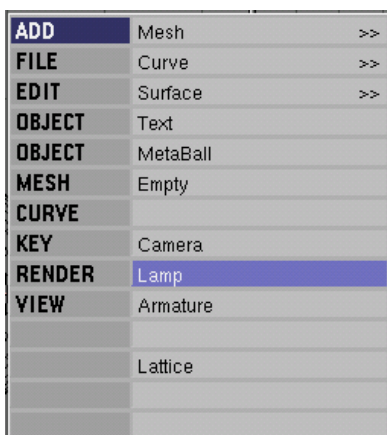
4

La figura 3 muestra cómo activamos la función de separar la esfera seleccionada del plano. Pulsamos **P** de **separate** y confirmamos **(OK)** la acción.

En la figura 4, finalmente, vemos cómo la esfera queda deseleccionada (color negro), mientras el plano continúa seleccionado en modo edición de vértices.

AÑADIMOS UNA LUZ A LA ESCENA

Para añadir una luz a la escena, basta con llamar a la caja de herramientas (**SPACE**) y **ADD->Lamp**. La luz se situará en la posición del puntero 3D.

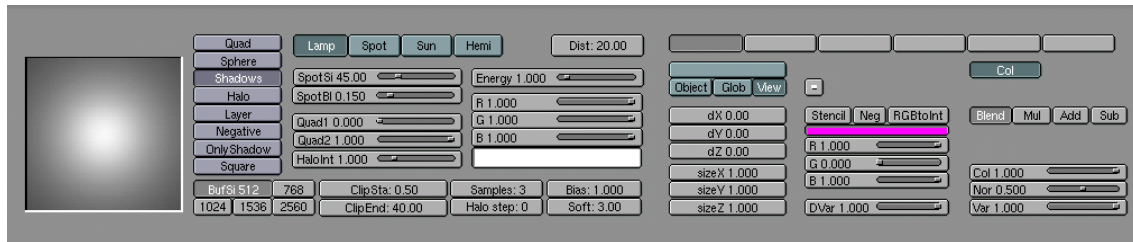


La luz siempre aparece, por defecto, como **omnidireccional**, es decir, emite la luz de manera uniforme en todas las direcciones. La línea discontinua vertical de la luz nos indica la distancia de la luz al plano del suelo o **XY**.

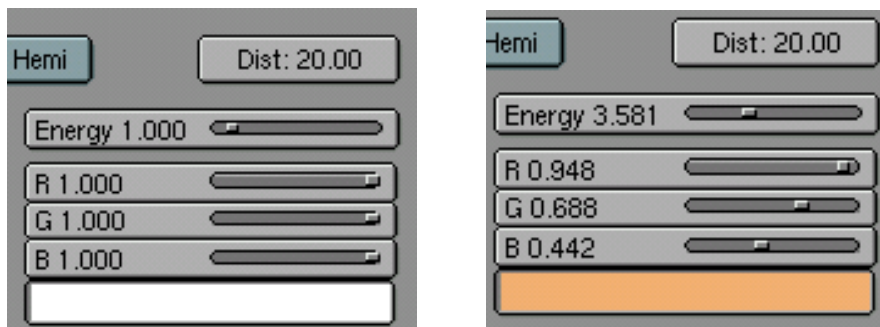
La luz tiene unas propiedades que podemos modificar, incluso el tipo de luz. Para acceder a los parámetros de color, tipo de luz, distancia, etc. Pulsaremos **F4**, o haremos click con el **BIR** en el icono que podemos encontrar en el menú de botones de la ventana inferior.



Si activamos el menú de botones de la luz (**lamp buttons**) con la luz seleccionada, nos aparecerá la siguiente imagen en pantalla:



De momento vamos a conocer la manera de cambiar el color de la luz, la energía que emite y la distancia que cubre su emisión. Vamos a ampliar la zona:

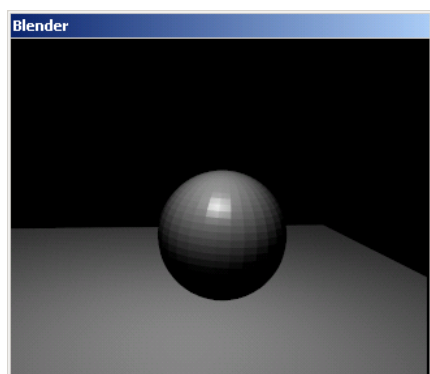


RGB corresponde a **Red, Green y Blue** (Rojo, Verde y Azul) y, si desplazamos los botones deslizantes, tal como hemos realizado en la segunda imagen, podemos obtener toda la gama cromática.

Energy representa el valor de energía que emite la luz. A mayor cantidad, mayor será la energía emitida por la luz.

Dist es la distancia que cubre la emisión de la luz. A mayor distancia, mayor será el área iluminada de la escena.

Por ahora, nosotros vamos a seguir los valores que vienen predeterminados de la luz y lanzaremos nuestro primer **render** de la escena. **Pulsa F12 y te aparecerá la siguiente imagen en una ventana nueva:**



FICHA V. El render en Blender.

En la ficha anterior pudimos ver cómo se añadía una luz en la escena y, de paso, lanzábamos nuestro primer render. No era precisamente un resultado conmovedor, pero por fin vamos familiarizándonos con Blender. Y claro está, el objetivo de la guía es ayudar a tener cierta soltura con el programa y con su sintaxis. Luego viene la parte de investigación y de práctica, que resulta imprescindible para poder elaborar escenas más complejas con resultados más que dignos.

En esta ficha vamos a mejorar el aspecto del render. Pondremos más luces, sombras, materiales y texturas, encuadraremos mejor la toma y, finalmente lanzaremos el render.

¡Más Luces!

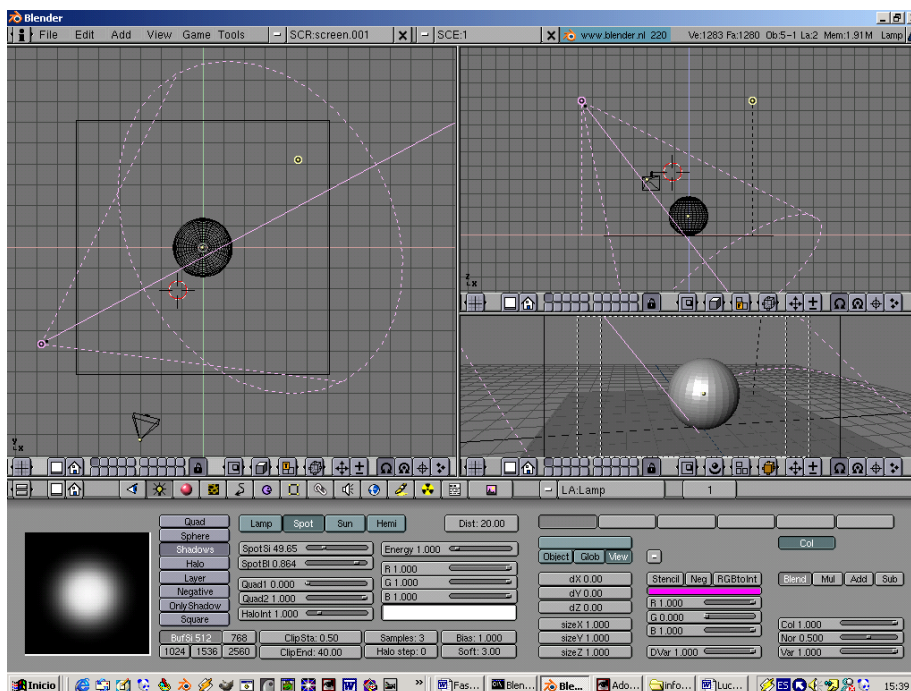
Vamos a convertir nuestra luz **omni** en otra llamada **spot**. En Blender, la luz **spot** es la única que puede proyectar sombras de nuestros objetos de la escena ¿Y eso? Cuestión de economía de memoria y rapidez a la hora de realizar el render. Blender, como otros programas que no soportan **ray tracing**, utiliza un algoritmo llamado **shadow buffer**. Este algoritmo realiza el render desde el punto de vista de la fuente de luz (en este caso, la luz **spot**), pero sólo almacena en memoria la distancia que va del píxel renderizado a la fuente de luz. En el proceso de render, el programa compara las distancias de cada píxel para determinar si el punto debe estar iluminado o en sombra. El **shadow buffer** se guarda en un archivo comprimido, lo que significa que el shadow buffer de un render de dimensiones 1024x1024 requiere una cantidad de 1.5MB de memoria.

Para convertir la luz **omni** en **spot**, tenemos que seleccionar la luz->**F4** y, en el menú de botones de la luz, pulsar **spot**, tal como vemos en la imagen



En pantalla, la luz tomará otro aspecto. La forma de la luz **spot** es la de un cono, donde el vértice superior corresponde al punto emisor. Vamos a mover y a rotar la luz **spot** hasta conseguir que la base del cono caiga sobre el área de la esfera. Finalmente, añadimos otra luz a la escena de tipo **omni**.

La imagen en pantalla debería quedar de la siguiente manera:



NOTA:

Para cambiar el diámetro de la base del cono de la luz **spot** modificaremos el botón deslizante de **SpotSi**.

Con **SpotBi** podremos modificar la intensidad del haz de luz en relación al diámetro del cono de luz. A mayor valor, el haz de luz cubrirá todo el cono de la luz.

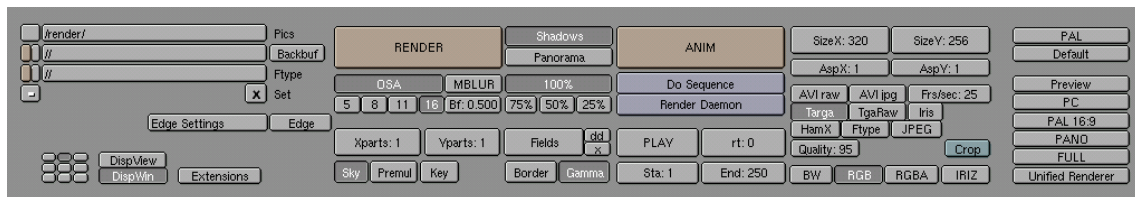


Un poco más sobre el render



Si hiciéramos de nuevo el render, las sombras podrían no estar activadas y, por lo tanto, no aparecer en la imagen.

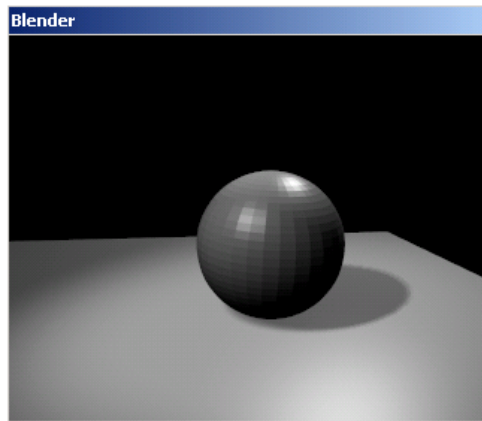
Para activar las sombras, tenemos que dirigirnos al **menú de botones de render (F10)** y veremos lo siguiente en nuestra pantalla.



A la derecha del enorme botón de **render**, tenemos el botón de **shadows**, que podemos activar o dejarlo desactivado si no queremos sombras en nuestra imagen.

Otro punto importante para conseguir un buen resultado sería activar el botón **OSA**. Uno de los problemas de trabajar con el **buffer shadows** es la aparición de perfiles dentados o pixelados en la imagen. **OSA** reproduce el efecto de **antialiasing** y, según el valor que le asignemos (**5,8,11,16**), el resultado será más afinado, pero claro, a coste de mayor tiempo de render.

Vamos a lanzar de nuevo el render. Pulsa **F12** y observa el resultado.

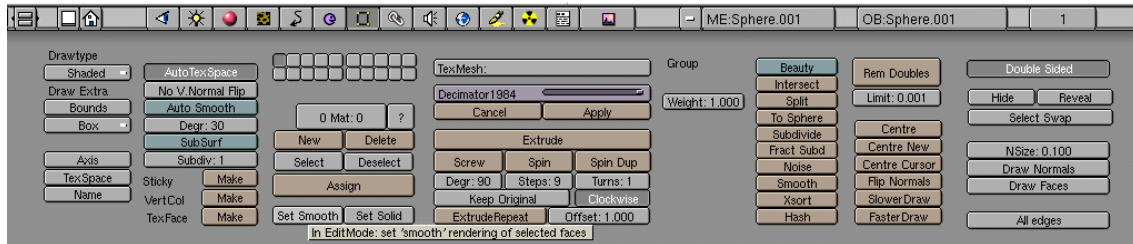


La esfera aparece “facetada” en lugar de tener una superficie impecable. Y falta dar color y texturas para conseguir un mayor grado de “realismo”. Vamos por partes.

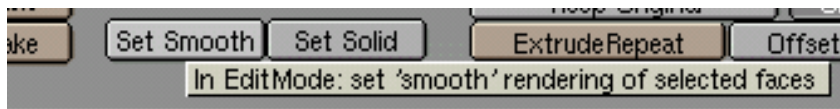
Suavizado Y Materiales

Vamos a empezar por suavizar la esfera:

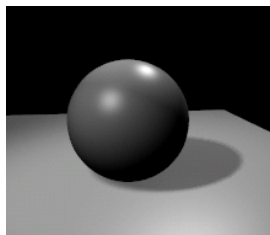
- Seleccionamos la esfera y nos dirigimos al **menú de botones de edición (F9)**. Nos tiene que aparecer la siguiente ventana de botones:



- Por ahora nos vamos a entretener en revisarlos todos. Pero sí miraremos con atención la parte inferior, debajo del botón **Assign**.



- Con la esfera seleccionada, hacemos click sobre el botón **Set Smooth**. En el caso que queramos volver a la antigua apariencia “sólida”, pulsamos **Set Solid**.
- ¿Hacemos de nuevo un render? Pulsa **F12** y fíjate en el resultado.



- Ya tenemos suavizada la esfera. ¿Le ponemos un color? Con la esfera seleccionada, pulsamos **F5** para dirigirnos al **menú de botones de materiales**. Allí podremos asignar un material a la esfera, al plano y a cualquier sólido de la escena.



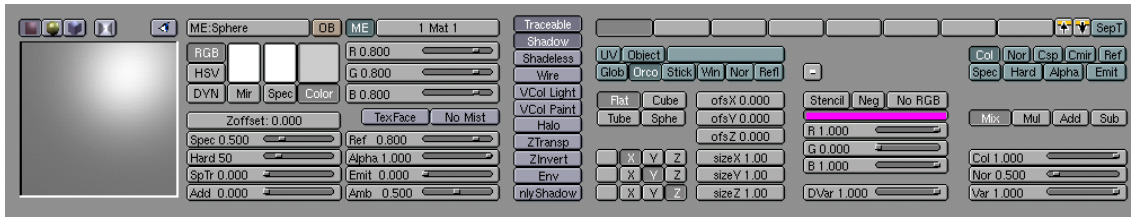
- Si nos aparece esta imagen en la pantalla, significa que todavía no hemos creado ningún material para asignar a la esfera o a cualquier objeto. Crear un material nuevo es sencillo:



- Para crear un material nuevo, nos dirigimos con el cursor al botón con el símbolo blanco de “menos” (en la imagen aparece rodeada por un círculo). Si apretamos con el **BIR** nos aparecerá un mensaje de **ADD NEW**, es decir, añadir uno nuevo.



- h. Una vez añadido el nuevo material, nos aparecerá **el menú de botones de material**.



- i. Por ahora nosotros sólo cambiaremos el color de la esfera (**RGB**); subiremos su valor de luz especular (**Spec**) y, al mismo tiempo, le añadiremos dureza (**Hard**) a su superficie, para conseguir que el área de la luz especular no sea muy grande.
- j. Los nuevos parámetros serán:



- k. De paso, si repetimos de nuevo la misma operación, pero con el plano seleccionado, creamos un nuevo material para el plano con las siguientes propiedades.

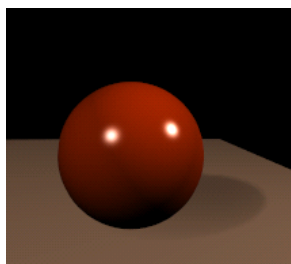
NOTA: Empezar a acostumbraros a bautizar objetos, materiales, etc. En el momento de crear el nuevo material, no os equivocáis seleccionando el material de la esfera! Añadir uno nuevo con **ADD NEW!**



- l. Al plano vamos a asignarle un color de marrón claro, con baja luz especular.



- m. Volvemos a por el render (**F12**). Obtenemos una imagen bien distinta de la primera!



TEXTURAS

¿Y qué ocurre si quiero asignarle una bonita textura a mi esfera? ¿Cómo lo hago?

Asignar texturas es realmente sencillo en Blender. Puedes conseguir resultados como el siguiente:



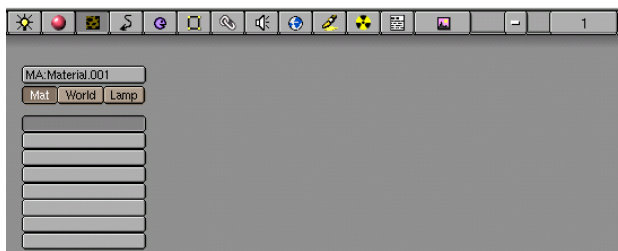
La esfera no tiene asignada ninguna textura. En cambio, el plano sí. De paso, veremos cómo se duplica un objeto (el plano se ha duplicado dos veces para la construcción de las paredes laterales).

En Blender, los materiales y las texturas forman dos bloques de información separados. La relación entre el Material y su Textura recibe el nombre de Mapeado o **mapping**. A través de una información que proporcionamos al editor de Materiales y Texturas, el mapa de textura será bidimensional (proyección de una imagen o **bitmap**) o tridimensional (una función matemática).

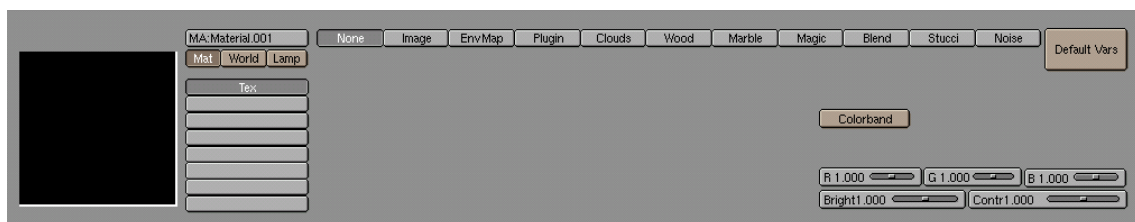
En el caso de la imagen, la textura del plano es una imagen bitmap y la aplicamos mediante un tipo de proyección plana. Vamos a ver cómo se traduce en Blender.

AÑADIMOS UNA TEXTURA

- Seleccionamos el plano y nos dirigimos al editor de texturas (**F6**), donde podremos ver lo siguiente:



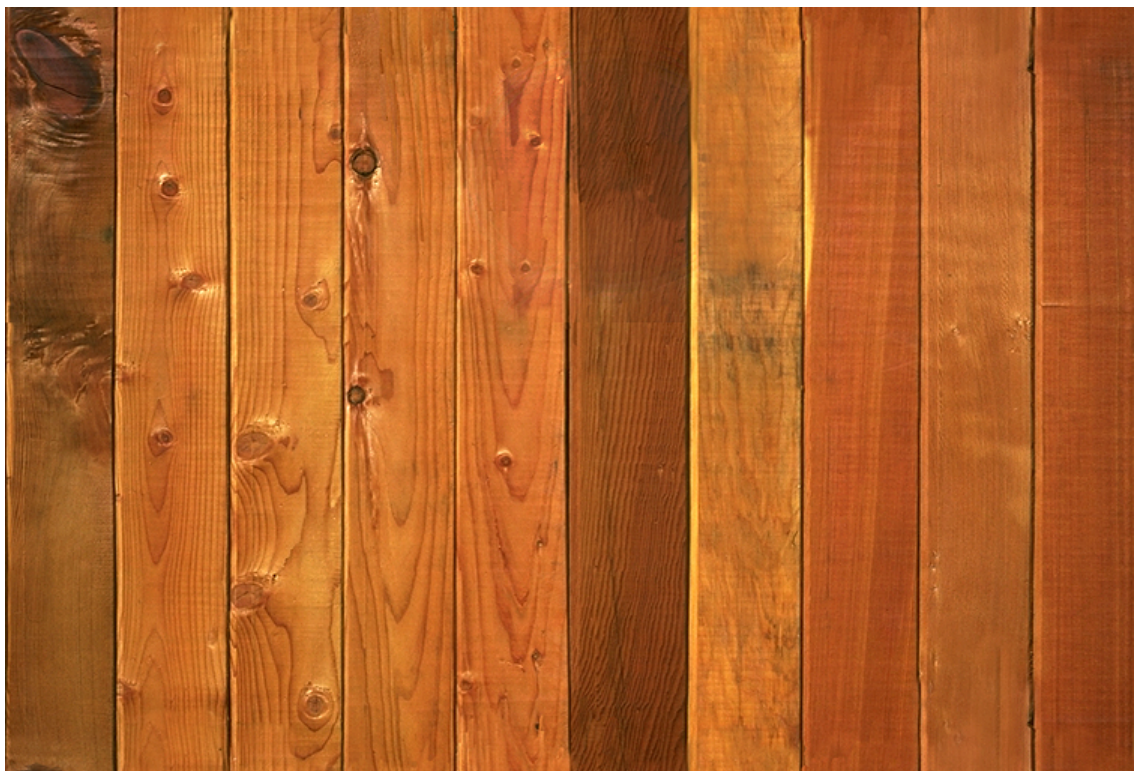
- Por el mismo procedimiento de añadir un material nuevo a cualquier objeto, llevamos el cursor al botón con la etiqueta '-' y añadimos una textura. Aparecerá el menú de texturas



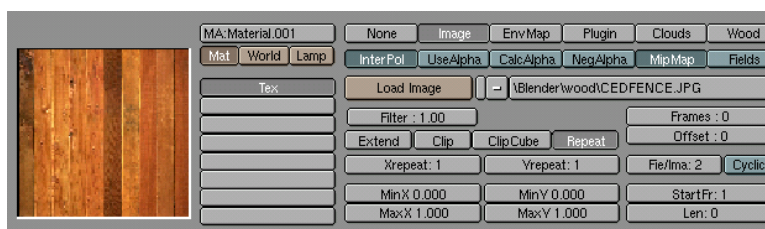
- c. Hemos dicho más arriba que al material del plano le aplicaríamos una proyección plana de una imagen o **bitmap**. De las diferentes posibilidades que nos ofrece el editor de texturas, encontramos el botón **Image**. Haz click sobre él y te aparecerá en pantalla las siguientes opciones.



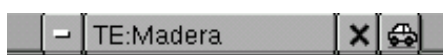
- d. Guarda esta imagen en el directorio de Blender o cualquier otros en el que puedas localizarla luego, ya que es la que utilizaremos como textura.



- e. Y ahora, desde el editor de texturas, cargamos la imagen pulsando el botón **Load Image**. Busca en el directorio la imagen que acabas de guardar en el directorio de Blender. Una vez abierto, deberías ver la siguiente imagen en pantalla:



- f. Es momento de cambiar el nombre de la textura (**Tex**) por el de **Madera** en la casilla encabezada por **TE**: y volver al editor de materiales. Pulsa **F5**.



- g. Vamos a centrarnos en la parte derecha del editor de materiales. Es el momento de indicar a Blender cómo queremos proyectar la textura sobre el material (**mapping**).
- h. Cada material tiene ocho canales diferentes de texturas al que pueden vincularse. Cada canal tiene su forma independiente de “proyectarse” sobre el material. En la imagen podemos ver que nuestra textura llamada **Madera** ha ocupado el primer canal



- i. Siguiendo el manual oficial de Blender, el siguiente paso será indicar qué **tipo de coordenadas 3D** asignamos a la textura. Podemos decidir si la textura será reflectiva o animada. En nuestro caso, dejaremos la que viene por defecto: **Orco (Original Coordinates)** que corresponde a las coordenadas 3D del objeto tipo poligonal o mesh.



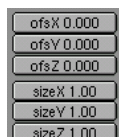
- j. **Tipos de proyección o conversión de 3D a 2D:** Blender presenta cuatro tipos de proyección de la textura: plana, cúbica, esférica y de tubo. En nuestro caso, elegiremos la opción **Flat** (plano).



- k. **¿Activar coordenadas X,Y,Z?** Es una posibilidad que nos permite Blender de cómo y en qué coordenadas proyectamos la textura sobre el objeto. En nuestro caso, sólo debemos activar la X y la Y; la coordenada Z, al no haber profundidad, no es necesaria.



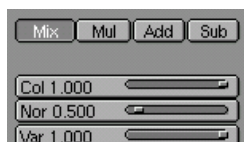
- l. El siguiente paso solucionaría la necesidad de cambiar o modificar las coordenadas de la textura. Vamos a dejar intacto este apartado tal como viene predeterminado.



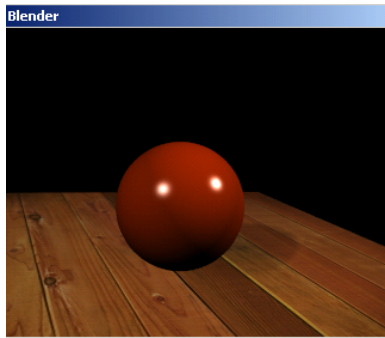
- m. **Mapping.** La salida de la textura en el material. Aparte del color, las texturas pueden afectar a otras características propias del material, como su superficie (**Nor**), la luz especular (**Spec**) o el valor **Alpha**. Nosotros vamos a dejar que la textura sólo modifique el color del material, por lo que sólo activaremos el botón de **Col**.



- n. **Valores de salida.** Aquí podemos indicar la fuerza o efecto de la textura sobre el material. Podemos mezclar, sumar o restar valores de texturas de diferente canal. Si por ejemplo bajáramos a cero el nivel de color de la textura, ésta no aparecería. Vamos a dejarla tal como está.

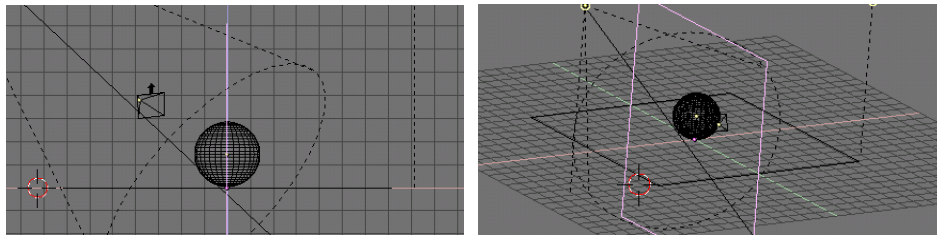


¿Probamos de nuevo a realizar un render? Pulsa **F12** para ver el nuevo resultado.

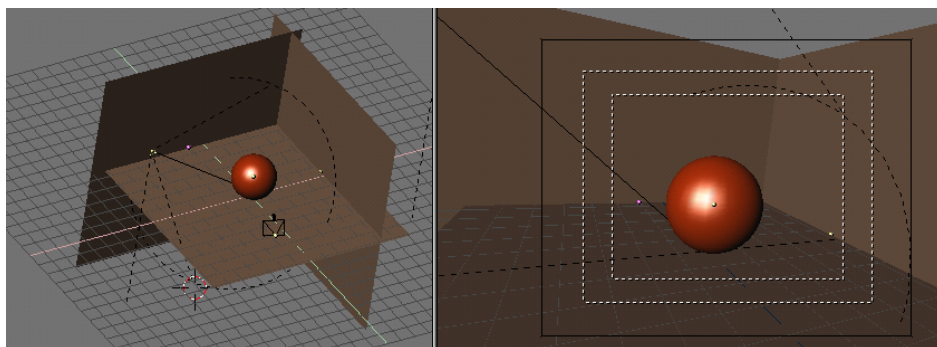


A pesar de que al hacernos un duplicado, no llevamos detrás toda la información de materiales y texturas, podemos cambiar dicha información sin miedo a cambiar la original. En el caso de que hubiésemos hecho una **instancia** en lugar de un **duplicado**, sí cambiaríamos la información original al cambiar la de la instancia.

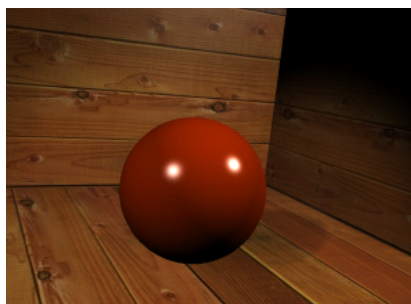
Para **duplicar** el plano, en primer lugar, lo seleccionamos y, desde una vista frontal pulsamos **SHIFT+D** (acción de **duplicar un objeto**). Sin mover el cursor, pulsamos **R** (**rotar**) y **giramos el plano hasta situarlo vertical** (puede servir de ayuda la opción de mantener apretado **CTRL** para ir de cinco en cinco grados)



Moveremos el plano hasta situarlo en uno de los laterales del plano-base. Duplicamos de nuevo y lo desplazamos hasta conseguir la otra pared. Un posible resultado sería el siguiente:



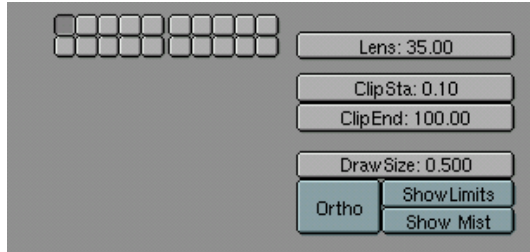
Ahora sólo falta lanzar el render(**F12**) y obtenemos la siguiente imagen:



LA CÁMARA

Ya conocemos la manera de acceder a la vista de cámara (pulsar **KeyPad 0** sobre la ventana 3D). Ahora vamos a saber qué propiedades tiene la cámara en Blender y cómo podemos modificar sus parámetros.

Seleccionamos la cámara y pulsamos **F9**. Observa la ventana:



¿Qué significan estos parámetros?

Lens: indica el valor de la longitud focal en milímetros. Técnicamente, la longitud focal representa la distancia entre la ventana de proyección (o película) y el centro de la lente de la cámara. Una menor distancia (35mm) conocida como gran angular permite abarcar una amplia visión de la imagen, pero sufre una gran distorsión; la distancia de 45-50 mm corresponde a la visión del ojo humano y, una gran distancia (conocida como teleobjetivo) de 100-250 mm permite aproximar la imagen.

DrawSize: Determina el tamaño del símbolo de la cámara en la ventana 3D.

Ortho: Activa una proyección ortogonal en lugar de una proyección cónica.

ShowLimits: Muestra en pantalla el límite de visión de la cámara, definido por el intervalo de **ClipSta/ClipEnd**. Cualquier objeto de la escena, desde el punto de vista de la cámara, quedará representado si se encuentra dentro del intervalo determinado por éste.

¿Qué significan los tres cuadros que aparecen en visión de cámara?

La exterior representa a la cámara, por si queremos seleccionarla desde su propio punto de vista. El siguiente rectángulo determina el área que será renderizada (si activamos la opción **DispView** en el menú **DisplayButtons (F10)**, podemos ver cómo el render se ejecuta en el visor de cámara). Finalmente, el último rectángulo se le conoce como **title safe** y representa el espacio que todas las televisiones pueden mostrar cuando el render se ha ejecutado en modo video.

¿Cómo puedo conseguir que mi cámara siempre fije su punto de atención a un objeto?

Nada más fácil que usar la función **Track**. Para ello nos serviremos de un objeto vacío o **dummy** (es decir, sólo indica posición) que en Blender se lo conoce por **Empty**.

- Añade un **empty (ADD->Empty)** a la escena.
- Selecciona la cámara primero y seguidamente, con **SHIFT** pulsado, selecciona el **empty**.
- Pulsa **CTRL+T (Make Track)**. Veréis que la cámara cambia de orientación. Eso es debido a que la cámara toma la orientación del **empty**. Para corregir el cambio de orientación, selecciona la cámara y pulsa **ALT+R (Clear Rotation)**.
- Ahora ya podéis mover la cámara por la escena: comprobad que el punto de mira siempre está restringido a la posición del **empty**.
- También podéis mover el **empty** desde el punto de vista de la cámara: de esta manera, el encuadre resulta más fácil de controlar.

FICHA VI. Animación básica.

Esta ficha intentará explicar de forma general las posibilidades de Blender como herramienta de animación. Una ventaja de animar en Blender es su rápido motor de render: en un tiempo relativamente corto podemos renderizar la animación de una escena compleja.

¿Cómo podemos animar en Blender? Por ahora vamos con los dos métodos más sencillos que cualquier herramienta de animación utiliza para animar los objetos en la escena: **Key Frames**, **Motion Curves** (nosotros mantendremos la acepción inglesa para no marearnos mucho...en castellano podríamos hablar de **Claves de fotogramas** y de las **Curvas de animación**) y la **Path Animation**. En el manual, este apartado se completa con la animación mediante **lattices**, **vertexkeys** y con el nuevo módulo de **character animation** mediante **IKAS**.

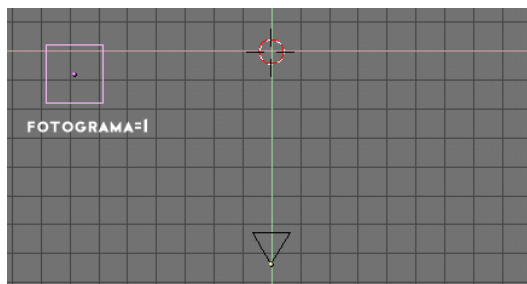
Key frames:

Como hemos dicho más arriba, las **key frames** son claves de fotograma en las que asignas un valor de un atributo "animable" (posición, escala, rotación,...) del objeto en un punto temporal (fotograma) determinado.

Si asignamos dos o más claves en un intervalo de tiempo del mismo atributo del objeto, dibujaremos una curva que describe el valor de las claves en cada uno de los fotogramas que pertenecen al rango determinado por los dos o más fotogramas clave.

Mejor explicarse con un ejemplo:

Vamos a iniciar una sencilla escena donde animaremos el plano que siempre nos viene por defecto...lo movemos a la izquierda tal como aparece en la ilustración.



- He insertado un texto debajo indicando el nº de fotograma al que corresponde la posición del plano.
- Para insertar una **key frame** o clave de fotograma del objeto seleccionado, pulsamos **I (Insert Key)** y aparecerá la siguiente ventana



- De los diferentes atributos de nuestro objeto (plano poligonal), hacemos click sobre **Loc** (posición o localización). **¿Qué acabamos de hacer?** Hemos insertado una clave de posición del objeto **plane** (nuestro plano) en el fotograma 1. **¿Por qué?** Por ahora, sólo nos interesa mover el plano en un intervalo de tiempo determinado por dos fotogramas clave; hemos activado el primero. Vamos a por el segundo:

- d. Insertamos otra clave, pero esta vez en un fotograma distinto, por ejemplo, el fotograma 24 ¿cómo y dónde podemos cambiar el nº de fotogramas? Por ahora decir que la forma más directa es modificar el valor que aparece en este botón:



- e. Este botón aparece siempre, a la derecha, en la barra del menú de botones. Para cambiar su valor, nos serviremos de los cursores del teclado o del ratón.
Cursor superior: suma 10 unidades al valor del fotograma actual.
Cursor inferior: resta 10 unidades al valor del fotograma actual.
Cursor derecha: suma 1 unidad al valor del fotograma actual.
Cursor izquierda: resta 1 unidad al valor del fotograma actual.

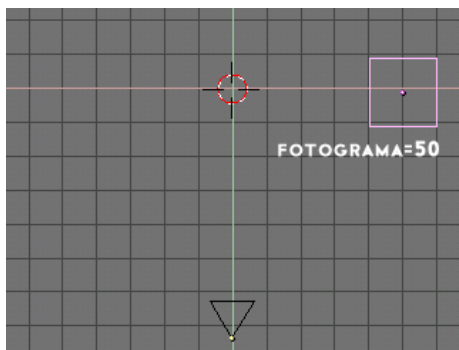
Con el **BIR pulsado** sobre el botón de fotogramas, nos desplazamos con el cursor a la derecha, sumando más fotogramas o, restando en el caso de movernos hacia la izquierda.

Para introducir desde el teclado el valor del fotograma, pulsa **BIR**.

- f. Sabido esto, vamos al fotograma nº50.



- g. Una vez en el fotograma 50, movemos el plano a la posición que indica la imagen:



- h. Pulsamos **I** para insertar otra **key frame** en el fotograma 50, tal como hemos hecho en el apartado **b**. Recuerda que insertamos una clave del atributo **Loc**,
i. Ya tenemos dos fotogramas clave (de valores 1 y 50) que guardan información de posición (**Loc**) de nuestro plano. **Volvamos al fotograma 1** y comprobamos que el plano se sitúa en la primera posición del fotograma clave uno.
j. Pulsa **ALT+A** para ver tu primera animación. Si has seguido correctamente los pasos, verás que el plano se mueve de la primera posición del fotograma uno a la segunda posición de la segunda clave del fotograma nº50.

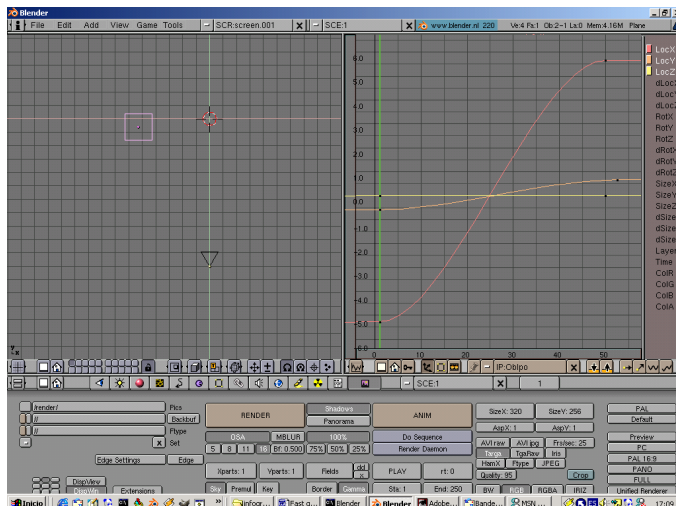
¿IPO?

¿Qué significa **IPO**? **IPO** es el editor de curvas de animación en Blender. En esta guía rápida sólo trataremos de explicar, de forma muy general, el trabajo con las curvas **IPO** a partir del ejemplo anterior.

- Para comenzar, dividiremos verticalmente la ventana3D. Existen diferentes tipos de ventana, tal cómo podéis ver en pantalla si mantenéis pulsado el **BIR** sobre el botón de tipo ventana3D (siempre está situado a la izquierda de la barra de botones de la ventana)



- Si pulsáis sobre el segundo icono, comenzando por abajo –el que representa una gráfica–, entraréis en la ventana-editor **IPO**. Pulsa **INICIO** del teclado sobre la dos ventanas para maximizar el contenido de las mismas. Vuestro Blender debe tener el siguiente aspecto:



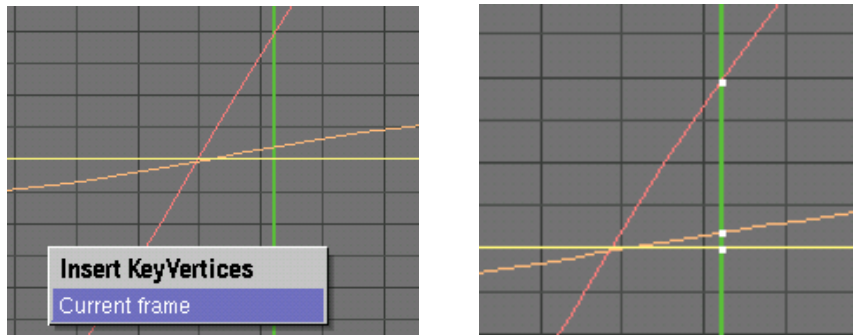
A la derecha tenéis la ventana del editor **IPO**, y a la izquierda la ya familiar ventana3D. Fíjate en las tres curvas que aparecen dibujadas. Las tres representan el valor de la posición X (color rosado), Y (color naranja) y Z (color amarillo) del objeto a través de la barra de fotogramas.

El editor **IPO** está representado por un sencillo sistema de coordenadas donde el eje horizontal corresponde al nº de fotogramas y, el eje vertical, al valor del atributo (en este caso, al valor de posición X,Y y Z del plano)

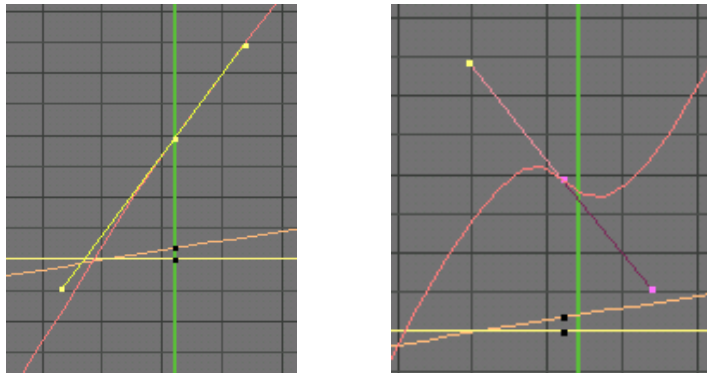
La barra verde vertical representa al nº de fotograma actual de la animación. Si la movemos (mantenemos el **BIR** pulsado y desplazamos de izquierda a derecha el cursor), veremos que nuestra animación se activa.

Los puntos de color negro indican la posición de los **key frames** o claves de fotograma. La selección y edición de curvas es la misma que la edición de objetos (**ver anexo I de la ficha II**).

- c. Vamos a trabajar directamente desde el editor **IPO**. Y lo vamos a hacer insertando una nueva **key frame** en el fotograma **31**. ¿Cómo? Muy fácil: arrastra la barra verde justo al fotograma 31, selecciona las tres curvas (pulsas **A**, los puntos de **key frames** se volverán blancos) y pulsa **I**. Debe aparecerte la siguiente ventana de confirmación.



- d. En la ventana de la derecha aparecen los nuevos **keyframes**. Vamos a editar uno de ellos para modificar la animación original.
- e. Selecciona la **curva Loc X (color rosa)** sólo pulsando con el **BDR** en cualquiera de sus **key frames**. Cambia el **edit mode (TAB)** a **modo vértices** (en este caso, los puntos de control de la curva) y pulsa con el **BID** en el **key frame** del fotograma **31**. Debería aparecerte la siguiente imagen:

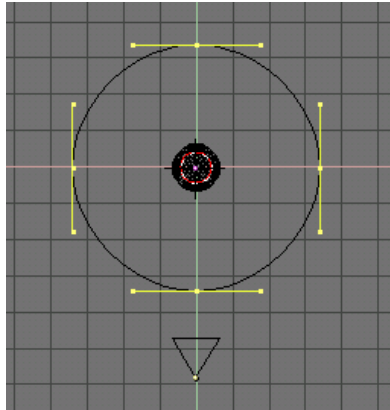


- f. Fíjate que podemos seleccionar los puntos de control que determinan la forma de la curva. Los dos puntos exteriores se los conoce como **anclas** y permiten modificar la curvatura en la zona determinada por su punto de control (punto central) o **key frame**.
- g. Ahora sólo queda que experimentéis un poco vosotros añadiendo nuevos **key frames** a otros atributos del plano (probad con la escala y la rotación) y modificarlos desde el visor IPO.

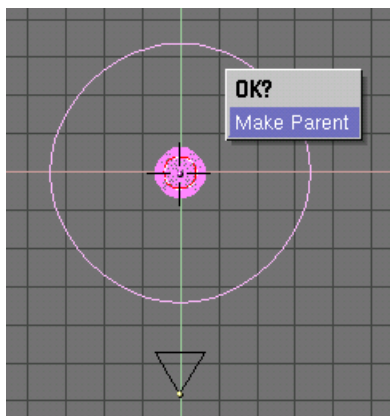
PATH ANIMATION

En algunos casos, necesitaremos que un objeto siga un recorrido definido por una curva, en lugar de ir insertando **key frames** de forma masiva para conseguir una trayectoria similar, pero no idéntica. Es el caso del recorrido de un planeta alrededor de otro, o una vagoneta que sigue el recorrido de las vías. Blender dispone de la posibilidad de convertir las curvas en recorridos (**path**). Todos los objetos que vinculemos a la curva-recorrido se moveran en relación a la forma de la curva: un ejemplo, si vinculamos como “hijo” a un círculo-recorrido, la esfera trazará un movimiento circular. Vamos por el ejemplo:

- Abrimos Blender y añadimos una esfera (**ADD->Mesh->UVSphere**) y un círculo b ezier (**ADD->Curve->Bezier Circle**). Aumentamos las dimensiones del c rculo y disminuimos la esfera.



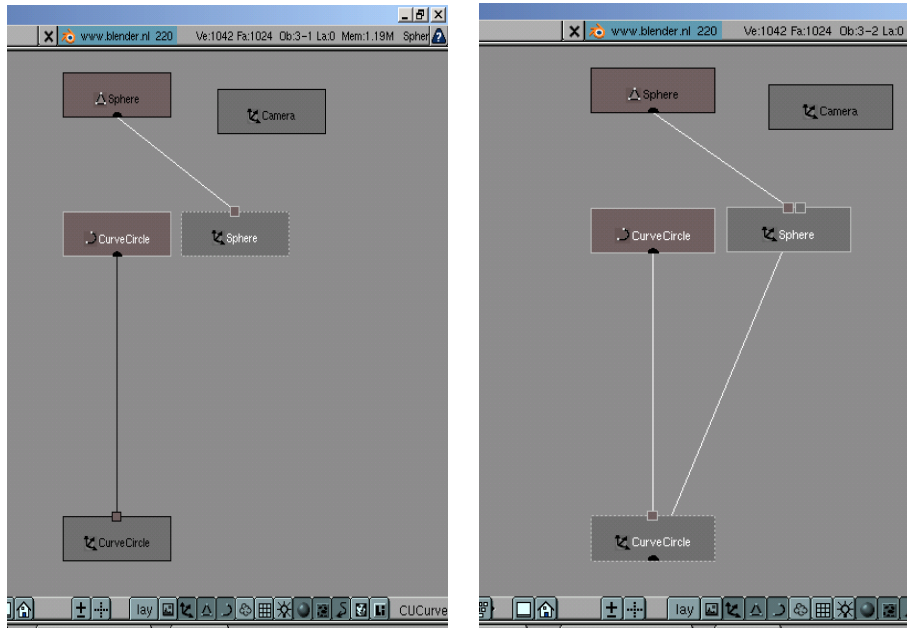
- He representado el c rculo b ezier en modo edici n de puntos para diferenciarlo del c rculo poligonal. Recuerda que una curva b ezier es una ecuaci n que dibuja una curva “real” y que el c rculo poligonal es una suma de segmentos dispuestos de forma circular.
- Vamos a vincular la esfera a la curva b ezier: para ello hemos de salir del modo edici n de puntos (**TAB**) de la curva. Pulsamos **A** para deseleccionar la curva. Seleccionamos la esfera y, con **SHIFT** apretado, pulsamos **BDR** sobre la curva. Recuerda, el primero seleccionado siempre ser  el “hijo” (**child**) del segundo en el momento de “emparentarlos” (**make parent**). Pulsa **CTRL+P** y te aparecer  la ventana de confirmaci n de **Make Parent**.



- Ahora la esfera es “hija” de la curva. ¿tenemos alguna manera de visualizar estas jerarqu as en Blender? S , para ello dividimos la ventana 3D y, activamos el icono tercero empezando por debajo



- e. En la ventana nueva aparece una representación de los objetos de la escena en forma de cajas o bloques de información. Vamos a explicar una diferencia



La imagen de la izquierda corresponde al momento en que la esfera todavía **NO** tiene ningún vínculo con la curva (las dos cajas que corresponden a la información paramétrica de la esfera están desligadas de las cajas correspondientes a la información del plano).

La imagen de la derecha corresponde al momento en el que “emparentamos” la esfera con el plano (Aparece una línea que “vincula” las dos cajas –esfera y plano).

- f. Volvamos a nuestra ventana 3D. Después de “emparentar” la esfera con la curva b ezier, comprobamos que no ha sucedido nada “relevante”, excepto en la ventana de “jerarqu as”, tal como hemos visto m as arriba.
- g. Seleccionamos  nicamente la curva y nos dirigimos a la ventana de botones de animaci n (**F7->Animation Buttons**) donde pulsaremos **CurvePath**.



- h. A partir de este momento, la curva b ezier se comportar  como una curva de recorrido (**path**). Un hecho que lo demuestra es que la esfera se ha situado en un punto de la misma. Y si ejecutamos la animaci n (**ALT+A**), comprobamos que la esfera se mueve siguiendo el recorrido determinado por la curva b ezier.
- i. A la izquierda del bot n **CurvePath**, encontramos **PathLen**, que indica la longitud de la curva en fotogramas, **NO SU VELOCIDAD!**
- j. A la derecha, encontramos **Curve Follow**, en la que la curva-recorrido aplica una rotaci n al objeto “hijo”. Prueba a insertar en la escena un cubo, v nculalo a la curva-recorrido y observa las diferencias.

FICHA VII. GameBLENDER Basis.

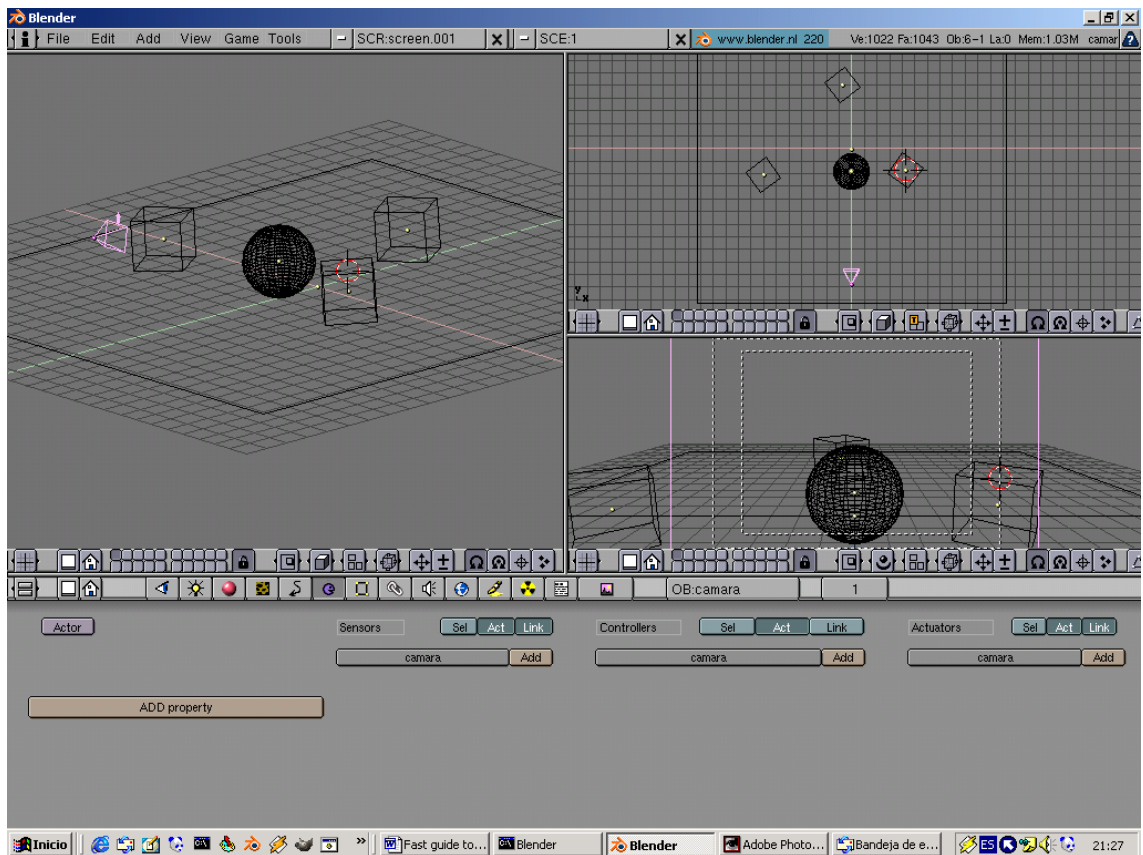
Esta ficha representa una aproximación muy básica al editor **gameBlender**. El editor apareció con la versión 2.x de Blender y cuenta con varios puntos fuertes:

- Un entorno integrado, con modelado, animación y gameplayer.
- Dinámicas de cuerpos rígidos (**rigid body dynamics**) y simulación de colisión.
- Sensores y otros aparatos lógicos predefinidos, de fácil interactividad.
- Python como script para juegos de mayor complejidad.
- Multiplataforma: Windows, Linux, FreeBSD, BeOS, Irix, y próximamente MacOSX.

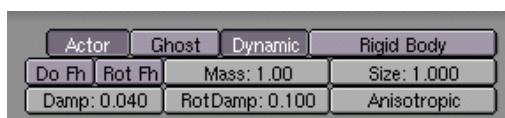
Lo que viene a continuación es un pequeño tutorial donde se muestra el funcionamiento básico del **gameBlender** a partir de la lectura y traducción de "**gameBlender Documentation**"

v0.99b.? Ton Roosendaal, Willem Zwarthoed y Carsten Wartmann.

- Partimos de la siguiente escena: un plano, una esfera, tres cubos y una cámara.



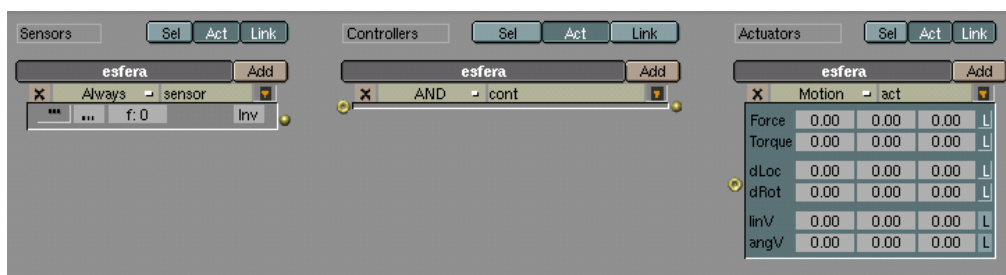
- Rebautizamos los objetos de la escena con nombres tan familiares como **esfera**, **suelo**, **camara**, **cubo1**, **cubo2** y **cubo3**. Recuerda que, para modificar los nombres de los objetos, tienes que seleccionarlos y activar cualquier menú donde aparezca el botón **OB:**.
- Activamos el menú **Realtime buttons (F8)** donde se centra la mayor parte del trabajo. Vamos a comenzar por la esfera. Si no tienes activado el menú **Realtime buttons**, pulsa **F8**. Selecciona la esfera.
- Ahora haz click con el **BIR** sobre el botón llamado **Actor (en la esquina superior derecha)**. Una vez activada la opción **Dynamic** te aparecerá los siguientes botones:



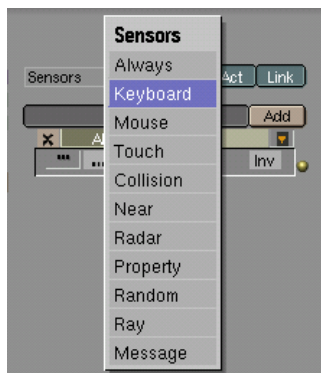
- f. Al activar la opción **Dynamic**, el objeto queda sujeto por las reglas físicas del **gameBlender**. Si ponemos en marcha el **gameEngine** (el juego, por decir algo) pulsando **P** sobre la ventana3D izquierda, verás que la esfera cae sobre el plano (**suelo**) y botará varias veces (bastantes veces) hasta quedar inerte. Para salir pulsa **Esc**.

MOVIMIENTO DE LA ESFERA

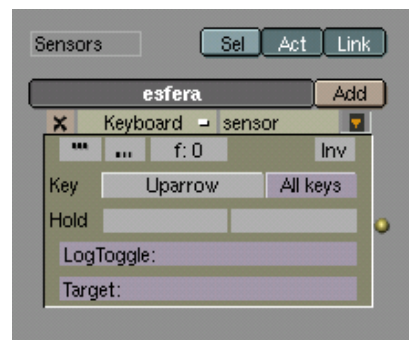
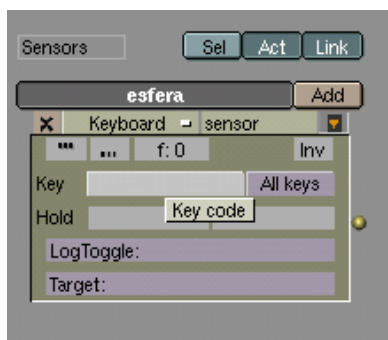
- a. Vamos a controlar el movimiento de la esfera mediante el teclado. Para ello vamos a fijarnos en las tres columnas de botones; representan el cuerpo de construcción de la interactividad: **sensores (sensors)**, **controladores (controllers)** y **actores (actuador)**. Una metáfora de cada uno de ellos puede ser la siguiente: los sensores representan una forma de vida determinada; los controladores, el cerebro y, finalmente, los actores representan el sistema muscular.
- b. Con el **BIR**, pulsamos en el botón **Add** para crear un **LogicBrick (bloque lógico)** por cada columna.



- c. Para cambiar el tipo de **LogicBrick en la primera columna (sensores)**, pulsa **BIR** sobre el la etiqueta **"Always"**. Mantén apretado el **BIR** para poder ver todas las posibilidades de **Logic Bricks** que permite la columna de sensores. De todas ellas, elegimos **keyboard**.



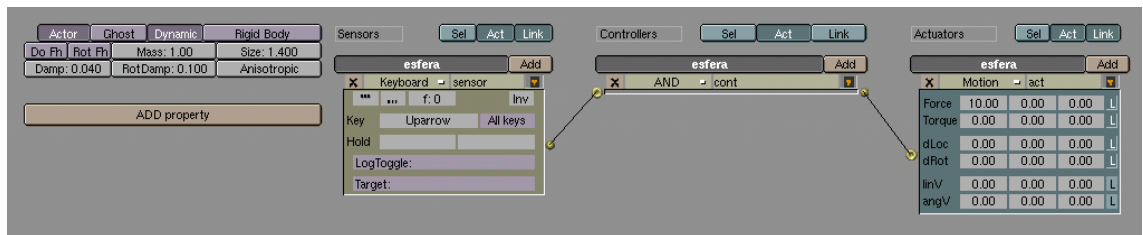
- d. Dentro del menú desplegable de **Keyboard**, haz un click con el **BIR** sobre el campo **"Key"** y aparecerá el siguiente texto: **"Press any Key"**. Pulsa cualquier tecla que quieras utilizar para mover hacia delante la esfera (sugerencia: Cursor Arriba=**Uparrow**).



- e. Ahora debemos asignar a esta función (en nuestro caso, cada vez que pulsemos **cursor –arriba**) un tipo determinado de acción. Vamos a la columna de **actuators** donde aparece el menú **Motion** desplegado. Lo que pretendemos es que la esfera se mueva en una dirección (hacia delante) por la fuerza (**force**) asignada a uno de sus ejes X, Y y Z. (Para visualizar el eje local de la esfera, activa menú **buttons (F9)** y **pulsa sobre el botón “axis”** que se encuentra en el lado izquierdo de la ventana). Si miramos desde el punto de vista del jugador (es decir, desde la cámara), el eje X apunta hacia “enfrente” del objeto. Para hacerlo, pulsa en la primera casilla (eje X) de la fila de **Force (Fuerza)**.



- f. Finalmente, sólo nos queda “vincular” los tres **aparatos lógicos** o **LogicsBricks** para que quede de la siguiente manera:



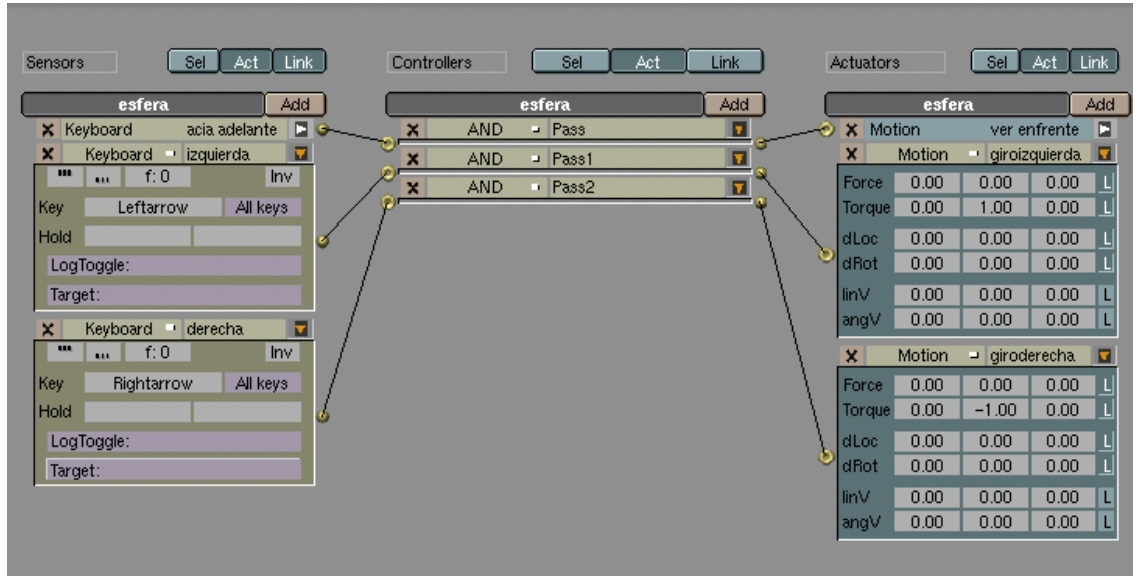
- g. Para conseguir enlazar los aparatos lógicos, pulsa y manten el **BIR** sobre el botón amarillo, a la izquierda del bloque “**keyboard**”, y arrastra el cursor hasta el aro del bloque “**AND**”. Verás como se va dibujando la línea-vínculo. Si quieres borrar la línea-vínculo, pasa el cursor por encima de ella y, cuando se ilumine, pulsa **X**.



- h. Pulsa **P** para activar el **gameEngine** y observa cómo se mueve la esfera cuando aprietas el cursor-arriba.

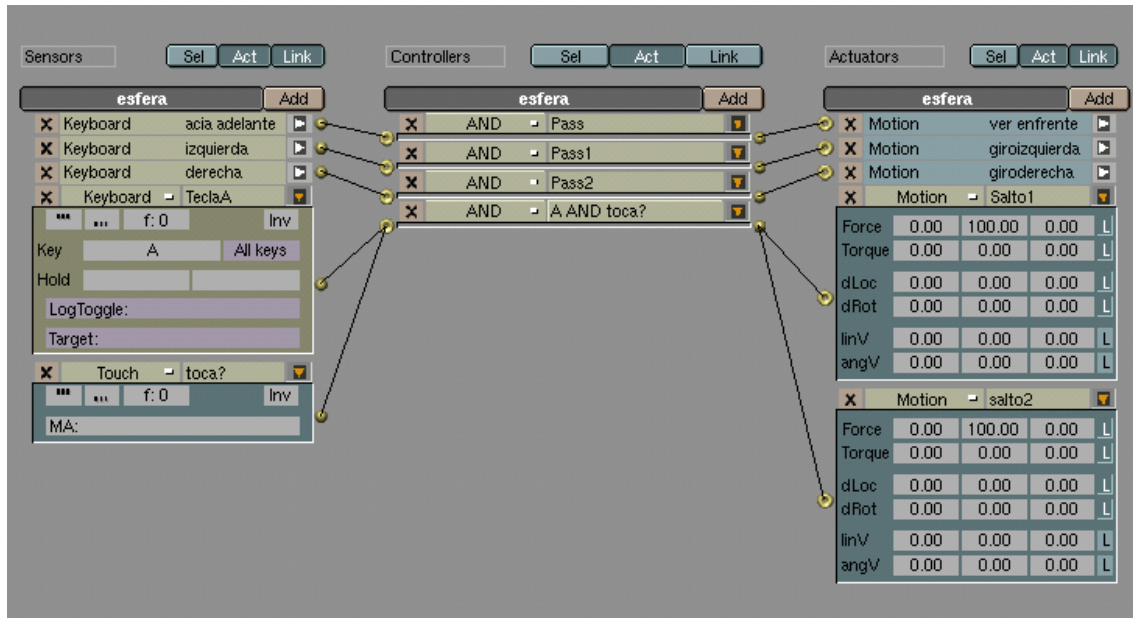
MAYOR CONTROL DE LA ESFERA (extraído de la documentación)

Vamos añadiendo más **LogicBricks**, como muestra la figura inferior, para conseguir un mayor control sobre el movimiento de la esfera. En este caso, configuramos el “giro” a la izquierda-derecha a través de la fila **Torque** de **Motion**.



SALTA! (extraído de la documentación)

También podemos añadir más de un sensor como entrada para un controlador: a través de la acción de hacer “saltar” a la esfera, podemos mostrar el ejemplo:



En esta situación, la esfera sólo podrá saltar justo cuando esté tocando el suelo o cualquier cubo de la escena. Si hacemos la lectura a través del editor, diríamos que “Si la tecla **A** es pulsada **Y (AND)** la esfera (**player**) toca (**touch**) el suelo, **THEN (Entonces)** da un impulso en el movimiento del actor (**motion-actuador**).”

Con esta manera te aseguras que nunca podrás saltar mientras estés en el aire: borra el vínculo que enlaza el sensor **Touch** con el controlador **AND** y observa el resultado...

Si hemos añadido a dos controladores de movimiento (**Motion Controllers**), ambos tendrán una fuerza de 100x2 para el eje Y. Esto es debido a que 100 es el valor máximo para cada **motion actuador** y usamos dos para conseguir un salto más potente.

FICHA VIII. Blender y Python.

Esta ficha representa una verdadera introducción a esta gran y acertada elección por parte del equipo de NAN desde su versión 1.68: servirse de **Python como script para Blender**. **Python** es un lenguaje interpretado, orientado a objetos y realmente fácil de aprender para **newbies** en la programación (“novatos” es una acepción que suele traer malos recuerdos!) y muy sencillo de implementar.

Con Python podemos acceder a gran parte de los objetos que definen una escena en Blender. Para una mayor información sobre Python (www.python.org) y sobre Python en Blender tenéis esta magnífica página en constante desarrollo: <http://www.blender.nl/python/>.

Para trabajar correctamente con Python dentro de Blender es más que recomendable instalarse el intérprete de Python. Blender incorpora aquellos módulos que pueden resultar necesarios para el desarrollo de scripts básicos, tales como “**math**” o “**whrandom**” (permite generar números aleatorios).

Vamos a iniciarnos en Blender-Python con el script de manual:

```
#Mantiene el objeto sobre el suelo!!!  
#Script propiedad de Carsten Wartmann?  
  
import Blender  
  
if (Blender.bylink):  
    objeto = Blender.link  
    if (objeto.LocZ < 0.0): objeto.LocZ = 0.0
```

Si conocéis la programación, comprobaréis que este script es muy sencillito. Para aquellos que todavía no habéis tenido la oportunidad, es un buen momento para adentrarse en este mundo de grandes posibilidades. Vamos a “diseccionarlo”:

1. Las almohadillas en Python se emplean para añadir comentarios que faciliten la comprensión del código. En este caso se ha utilizado como enunciado.
2. “Import Blender” incluye una llamada al módulo **Blender** —en realidad, una **librería**—, el cual define la interficie entre Blender y Python.
3. La iteración “**if**” nos indica si la condición “**Blender.bylink**” es cierta, es decir, si tenemos el script “linkado” a un objeto. Si es así, pasemos al interior de la iteración.
4. La primera proposición es asignar el valor del atributo link (el objeto que tenemos seleccionado) del módulo Blender a la variable “objeto”.
5. De nuevo otra iteración del tipo “**if**”: en este caso comprobamos si el valor de la posición Z del objeto es menor que cero. Si es así, le asignamos un valor igual a cero.

Vamos a ponerlo a prueba en Blender a través de la escena básica: un plano y la cámara. Selecciona el plano. Seguidamente dividimos la ventana 3D y accedemos en una de ellas a la ventana de texto:



Para poder añadir el script a la ventana de texto tenemos dos posibilidades: **OPEN** y **NEW**. Para acceder a las dos opciones, pulsa con el **BIR** sobre el botón con el símbolo “menos”.



Con **OPEN NEW** podemos abrir un archivo de texto o archivo .py (de python) mientras que, con **ADD NEW** podemos escribir directamente sobre el editor de texto de Blender.

- a. Una vez que nuestro primer script está escrito correctamente en el editor de Blender, dirígete a la ventana de botones de **Script** o **Script buttons**.



- b. Selecciona el icono para vincular scripts a objetos (dentro de **Script buttons**).



- c. Haz click con el **BIR** sobre el botón **New** de la fila izquierda.



- d. En el botón vacío, escribe el nombre del script que has editado.
- e. Ahora selecciona el objeto (el plano) y muévelo con **G**. Si el script funciona, verás que el plano no podrá moverse por debajo del límite cero del valor Z.
- f. Si el script no funciona, mira los mensajes de error en el terminal abierto de Blender.

Creación de un toro (en construcción)

En esta ocasión desarrollaremos un script en el que podamos dibujar en pantalla una nueva primitiva geométrica: el toro.

Partimos de su parametrización como base donde podemos especificar la posición de $P(x, y, z)$ a partir de las constantes r, R y los ángulos θ, ϕ , que utilizaremos como parámetros.

$$\begin{aligned} x &= R \cos \theta \cos \phi \\ y &= R \sin \theta \cos \phi \\ z &= r \sin \phi \end{aligned}$$

Una primera implementación sería:

```
#Primitiva Toro v0.1 Javier Belanche, Septiembre 2001
#a partir de un script de C.Wartmann.

#jbelanche@campus.uoc.es

from Blender import*
from math import*

xw=0.331
yw=0.331
```

```

xmax=20 #valor resolucion x del toro
ymax=20 #valor resolution y del toro

r=2
R=2
d=3

me2= NMesh.GetRaw()

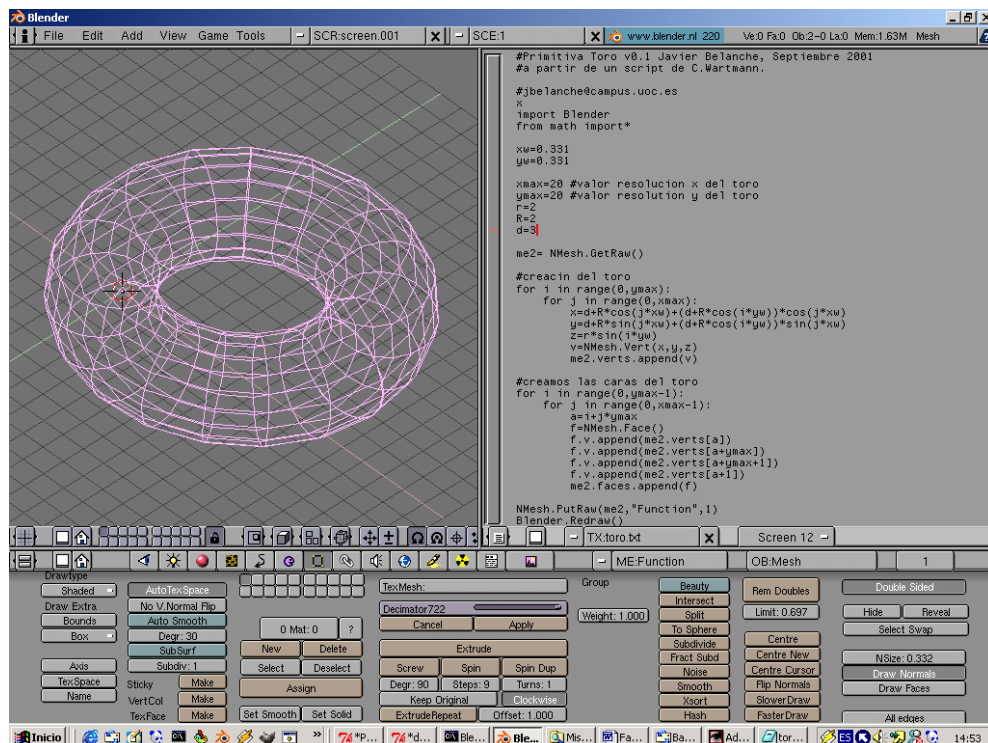
#creación del toro
for i in range(0,ymax):
    for j in range(0,xmax):
        x=d+R*cos(j*xw)+(d+R*cos(i*yw))*cos(j*xw)
        y=d+R*sin(j*xw)+(d+R*cos(i*yw))*sin(j*xw)
        z=r*sin(i*yw)
        v=NMesh.Vert(x,y,z)
        me2.verts.append(v)

#creamos las caras del toro
for i in range(0,ymax-1):
    for j in range(0,xmax-1):
        a=i+j*ymax
        f=NMesh.Face()
        f.v.append(me2.verts[a])
        f.v.append(me2.verts[a+ymax])
        f.v.append(me2.verts[a+ymax+1])
        f.v.append(me2.verts[a+1])
        me2.faces.append(f)

NMesh.PutRaw(me2,"Function",1)
Blender.Redraw()

```

Si ejecutas el script en la ventana de texto (**lleva el cursor a la ventana de texto y pulsa Alt+P**) podrás comprobar cómo se ha creado el toro, tal como aparece en pantalla:



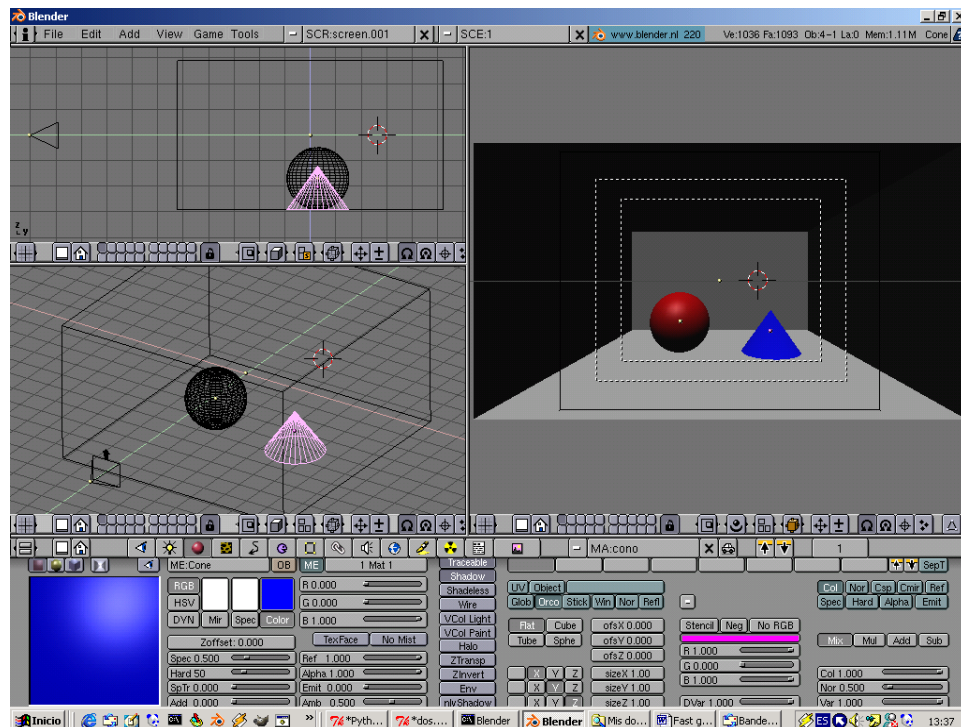
FICHA IX. Radiosity.

La Radiosidad es un algoritmo de render donde intenta simular la reflexión difusa de la luz y el color entre las superficies de una escena. La técnica de la radiosidad se conoció en un principio como “**interfection reflection**” y fue planteada en 1946 para simular matemáticamente la luz en espacios arquitectónicos.

Blender incorpora un motor de radiosidad que recoge la solución más optimizada: el método “**progressive refinement**” con el esquema “**adaptive subdivision**”. Para una mayor explicación de este método, recomiendo las lecturas de Foley & vanDam “*Computer Graphics*” y la obra de Andrew Glassner “*Principles of digital image synthesis*”.

Aquí no entraremos en detalle de cómo se ha estructurado la radiosidad en Blender (desarrollado en el manual). Por ahora tenéis un ejemplo de cómo podemos obtener un resultado de radiosidad en Blender a través de una escena muy sencilla.

- En primer lugar, para aplicar la radiosidad, **es necesario que todos los objetos sean polígonos, es decir MESH**. Hemos creado esta escena donde una de las caras del cubo (la que apunta la cámara) ha sido eliminada (selecciona los vértices->**X->Only Faces**). Redimensiona el cubo de tal forma que adquiera el aspecto de una habitación.
- Sitúa ahora dos objetos -una uvsphere y un cono- dentro del cubo. Aplícales un material en el que la esfera sea roja y el cono azul. Finalmente, suaviza las superficies (**F9->Set smooth**). El resultado podría ser el siguiente:



- En términos de radiosidad, cualquier objeto de la escena es susceptible de emitir luz (**area lights**). En nuestra escena nos serviremos de un plano como emisor de luz. Para que el plano emita luz tenemos que asignarle un material donde daremos un valor al **Emit**. (Ver figura 1)

- c. **EMIT** es el valor de emisión de luz del objeto. La dirección queda determinada por la normal o normales del objeto. Para ver la dirección, en **EDITMODE (edición de vértices->tabulador)**, pulsa **F9** y, en la parte derecha del panel de botones, encontrarás **DRAW NORMALS**. Si no ves casi nada, aumenta el tamaño de las mismas subiendo el valor **NSIZE**.

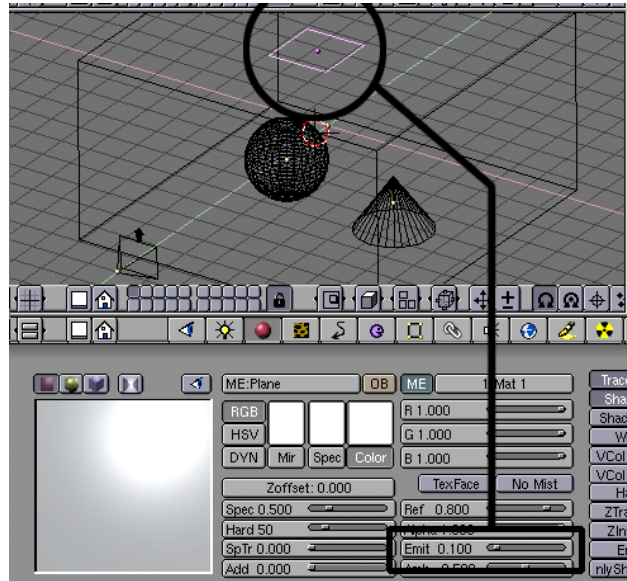
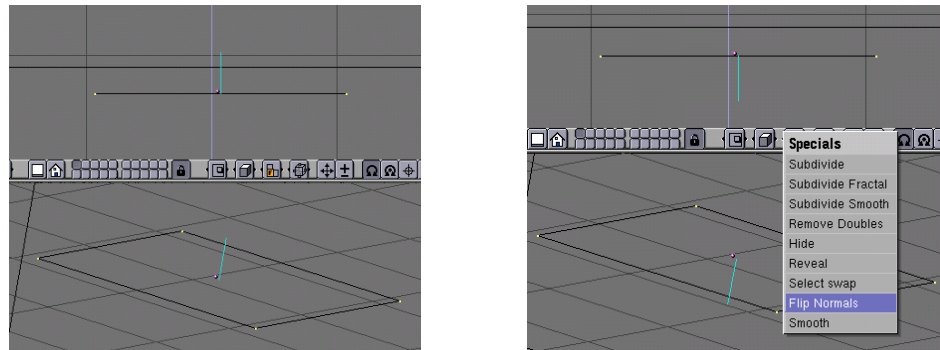
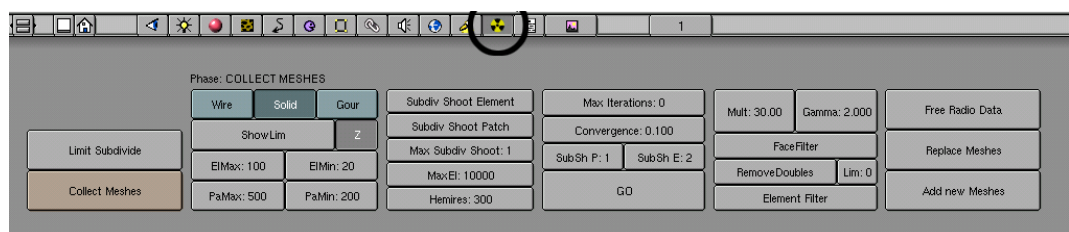


Figura 1

- d. No hace falta girar el objeto para invertir la dirección de la normal: pulsad **W->Flip Normals**...qué detalle ¿no? En nuestro caso, si no hubiésemos cambiado la dirección de la normal del plano, la emisión de la luz se hubiera concentrado en el techo en lugar de iluminar la escena de nuestra "habitación".

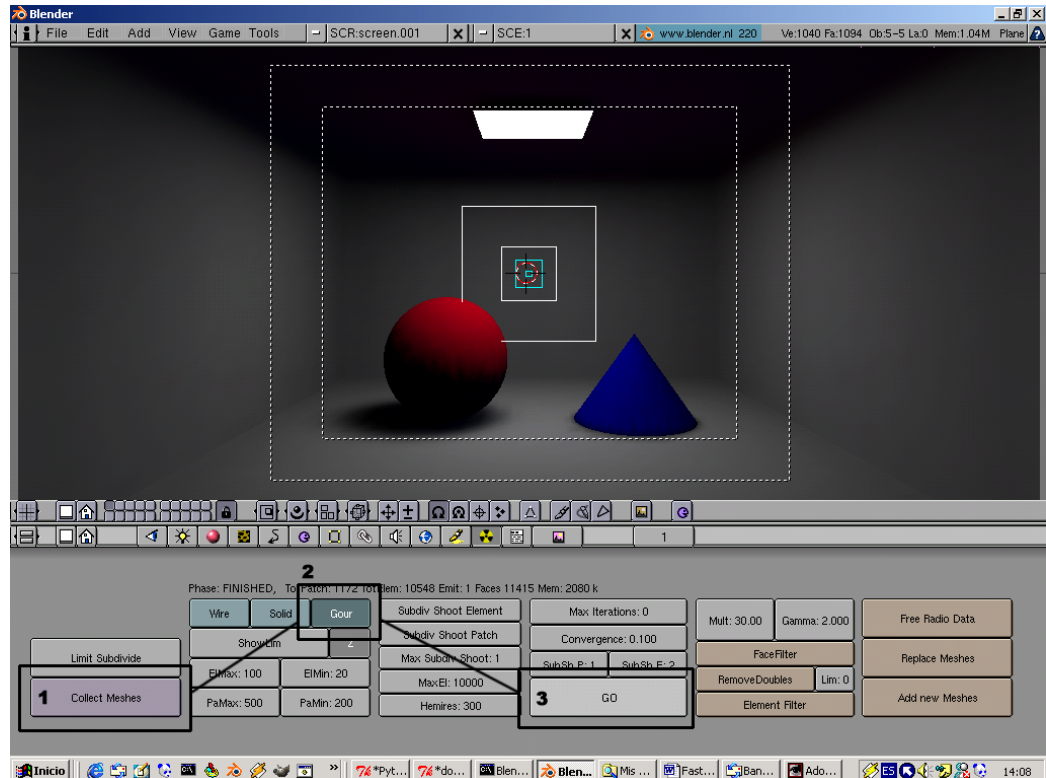


- e. Cuando ya tengáis la escena preparada, **seleccionad todos los objetos de la escena** y apretad el botón que tiene como símbolo el icono de la radioactividad. Os aparecerá el siguiente menú:



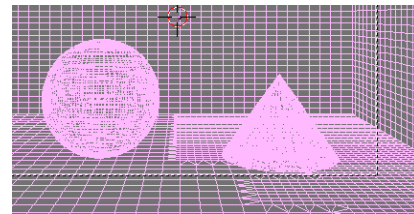
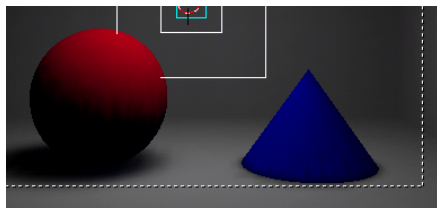
- f. Acto de fe: **Pulsa COLLECT MESHES->GOUR->GO**. La escena comenzará a trabajar...si creéis que la escena ya está acabada...pulsa **ESC** para salir del cálculo de radiosity.

- g. Este sería un primer resultado. De paso, he resaltado los tres pasos que tenéis que realizar para ejecutar el cálculo de radiosity.



- h. Todavía no se ha acabado: ahora falta salir convirtiendo la escena en otra malla: en la misma ventana de radiosity. **REPLACE MESHES->FREE RADIO DATA.**

Desde este momento, tendremos una malla nueva en lugar de la original. Si activamos **ALT+F** volveremos a la visualización realista de la escena. Lo único que ha realizado la radiosity es “pintar” la superficie: prueba a moverte con la cámara por el nuevo espacio. Ya no hace falta que calcules de nuevo la radiosity debido a su comportamiento como “material”.



Ahora tienes una malla “global”. Para recuperar la identidad de los objetos, deberás seleccionarlos y separarlos de esta malla. ¿Cómo? Con la malla seleccionada, activa **edit mode (TAB)** para visualizar los vértices de la malla. Lleva el cursor a la esfera o al cono y pulsa **L**. Comprueba que sólo te ha seleccionado el objeto. Una vez que el objeto está seleccionado, pulsa **P (Separate)** para separar el objeto de la malla. A partir de aquí ya puedes editar el objeto de forma independiente (por ejemplo, para añadir texturas).

Nota!: Recuerda también que los objetos comparten el material de radiosity, es decir, que tiene activado en el menú de botones de material la opción **Vcol Light**. Si desactivas la opción, desaparecerá el resultado de la radiosity.



FICHA X. Técnicas de modelado poligonal: extrusion, spin, subdivision surfaces y decimator.

EXTRUSION (EXTRUDE)

Es la herramienta estrella de modelado poligonal en Blender. Según comentarios de sus creadores, con la extrusión “puedes crear cubos a partir de rectángulos y cilindros a partir de círculos”. Su modo de trabajo es crear un modelo tridimensional a partir de un contorno bidimensional extendiéndola a través de un recorrido (path) o de un eje. En algunas herramientas, la extrusión recibe el nombre de **lofting o solevado**. En Blender, la extrusión es un proceso altamente intuitivo, rápido y de fácil manejo que sigue unos sencillas reglas:

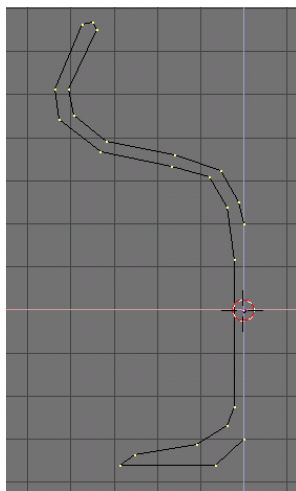
- Si tenemos un plano y seleccionamos todos sus vértices (por lo tanto, seleccionamos todas sus aristas) y los extruimos, las aristas formaran nuevas caras. De esta operación se deduce la creación de un cubo a partir de un plano.
- En el momento de extruir se activa por defecto la función Grabber o Desplazamiento. Una vez ejecutada la extrusión, podemos escalar o rotar.

SPIN

Viene a traducirse como torno o revolución. Blender rota alrededor de un eje (el cursor 3D) los vértices seleccionados y los conecta mediante aristas para crear caras. Prácticamente, una gran mayoría de objetos cotidianos tienen como base esta forma de modelado: a partir de medio perfil del objeto que deseamos “revolucionar”, la función SPIN nos completará el recorrido circular entero o parte de él.

De SPIN podemos decir que es otra técnica muy popular entre las distintas herramientas de 3D. Podemos encontrarla bajo nombres como **lathe**, **revolve** o **spin** en el caso de Blender. Las superficies creadas con esta técnica reciben el nombre de **superficies de revolución (surfaces of revolution)**.

- Abrimos Blender. Diseñaremos una copa
- Si tenemos por defecto la vista superior, hemos de activar la vista frontal (**Key Pad 1**)
- Si no tienes el plano que aparece por defecto en pantalla, créalo con el menú de herramientas (**Space**) **ADD->MESH->PLANE**
- El plano aparece en EditMode. Por ahora sólo nos interesa trabajar a partir de un solo vértice. Para ello, borra los vértices del plano mediante **X->ERASE->Vertices**.
- Ahora podemos insertar libremente vértices en la ventana 3D con **CTRL+BIR**.
- Intenta dibujar (un poco de entrenamiento no va nada mal) el perfil de la copa tal como aparece en la imagen.



- g. Un consejo: es más rápido dibujar primero una aproximación del perfil y luego, seleccionar y mover vértices hasta conseguir el contorno deseado. Este modo de modelado se le conoce por **direct point** manipulation, es decir, la manipulación directa de vértices.
- h. Entramos en el menú **Edit Buttons (F9)**. Aquí encontramos las diferentes opciones de **SPIN**. Vamos a comentarlas:



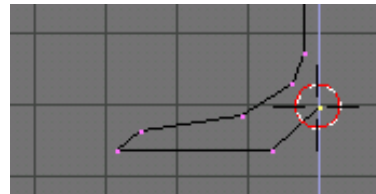
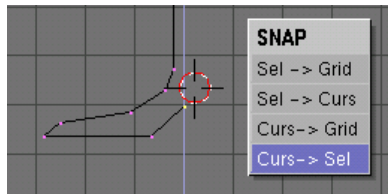
El botón **Spin** está situado justo en el centro del menú Edit Buttons. Debajo de él encontramos tres opciones:

Degr: Es el valor en grados que cubrirá el “spin” del perfil u objeto. 360° indica que dará la vuelta entera.

Steps: Indica el nº de veces que el perfil u objeto se duplicará en el rango de grados que hemos indicado con Degr. A mayor número, mayor será la resolución del objeto, pero mayor será el peso de vértices y caras.

Clockwise: Es la dirección de la “revolución”. Activado representa en la misma dirección de las manecillas del reloj y desactivado en la dirección contraria.

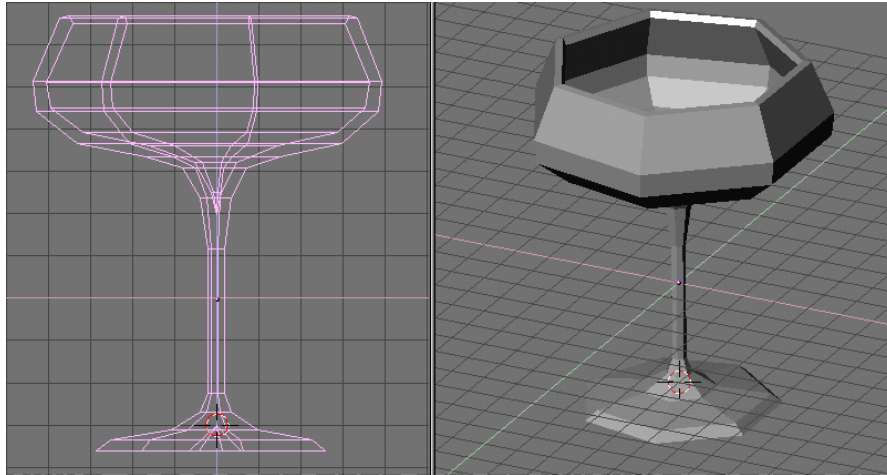
- i. Para nuestro primer “Spin” vamos a dar los siguientes valores: **Degr->360 y Steps->7**
- j. **IMPORTANTE:** el eje de la revolución o del Spin vendrá indicado **siempre** por el cursor 3D. Donde esté situado levantará un eje perpendicular a la vista activa. Fíjate en la imagen:
- k. Para una mayor precisión, el cursor 3D puedes situarlo justo en uno de los vértices que estarán contenidos en el eje de revolución: seleccionamos el vértice en la vista frontal (**KeyPad 1**) y con **SNAP (Shift+S)->Curs Sele.**



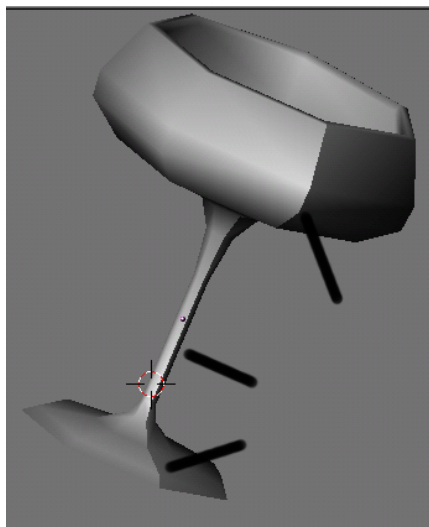
- l. Para que Blender dibuje nuestra copa, tendremos que activar la vista de planta o superior (**KeyPad 7**). **¿Por qué?** Hemos dicho que Blender levanta un eje perpendicular a la vista activa en la posición del cursor. Si hemos dibujado el perfil de la copa en la vista frontal, nuestro eje de “spin” o de revolución será perpendicular al plano del suelo, es decir, al eje X. Si activamos la vista superior o **KeyPAD 7**, el eje de rotación que trace el spin será perpendicular al eje X.



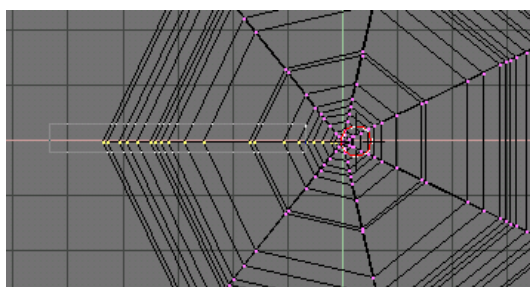
- m. Ya lo tenemos todo preparado para ejecutar el **SPIN**. Activamos la vista superior (**KeyPAD 7**) y seleccionamos todos los vértices del perfil con A.
- n. Pulsamos SPIN y obtenemos la copa!



NOTA: La operación SPIN deja un duplicado de los vértices de la silueta. Si quieres comprobarlo, activa el modo shade (**Z**), suaviza el modelo (**F9->Set Smooth**) y gira el ángulo de visión de la vista actual (Alt +Botón izquierdo del ratón). En el modelo verás dibujado la línea del perfil original: esto es debido a que el duplicado de los vértices no se encuentran justo en la posición de los vértices del perfil original. Para eliminar estos vértices utilizaremos la función **Rem Doubles** que puedes encontrar en el menú Edit Buttons (F9). Sigue los pasos que ahora te indicamos:



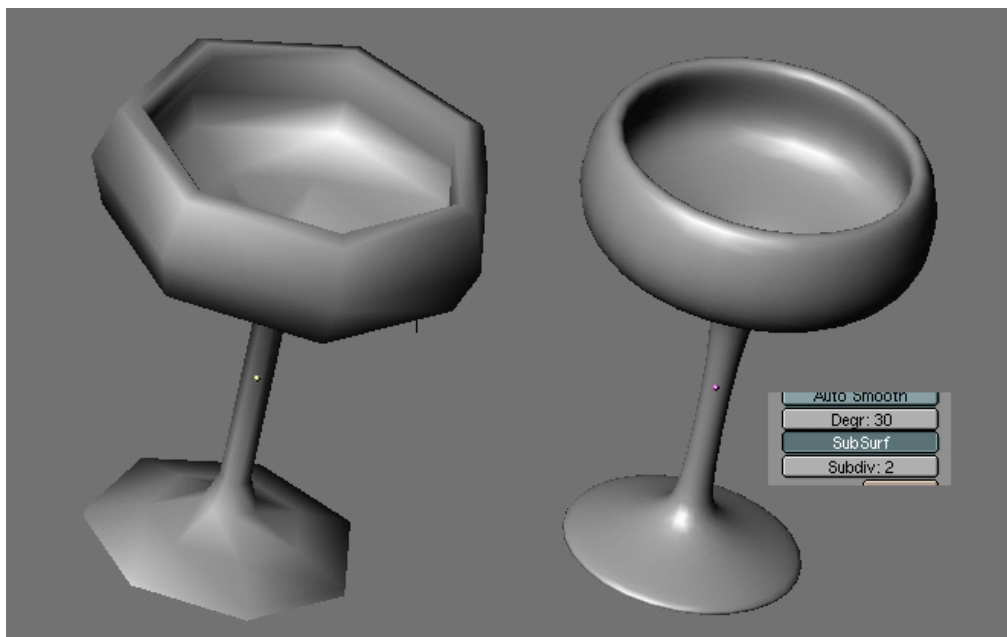
- a. El objeto en modo edición de vértices (**TAB**), activamos la vista superior (**Key PAD 7**).
- b. Hacemos una selección rectangular de los vértices del perfil original y del duplicado, tal como se muestra en la imagen.



- c. Una vez seleccionados los vértices, pulsamos **Rem Doubles** en el menú Edit Buttons (**F9**) o a través del menú flotante **Specials (W)**->**Remove Doubles**.



- d. El efecto es el siguiente: **Rem Doubles** eliminará vértices “innecesarios” o “duplicados” para dejar sólo uno. El límite o distancia de acción de Rem Doubles vendrá dado por **Limit**.
- e. Una vez pulsado **Rem Doubles**, aparecerá el número de vértices eliminados. Si el límite es demasiado pequeño, la operación tendrá como resultado 0 vértices eliminados, por lo que debemos aumentar poco a poco su valor e ir probando hasta conseguir eliminar los vértices duplicados.
- f. El resultado será el siguiente:

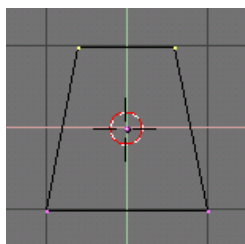


- g. Para conseguir el suavizado como la copa de la derecha, sólo tenemos que pulsar sobre el botón de **Sub Surf** y darle un valor de 2.

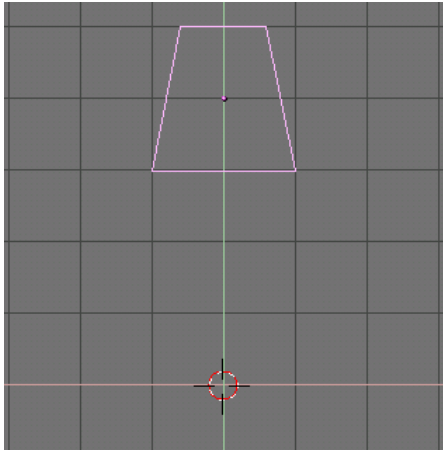
SPIN DUP

En este caso, Blender sólo duplica los vértices sin conectarlos. Es un buen método de obtener copias de un objeto a partir de un recorrido circular. **Vamos a intentar modelar una sencilla tuerca a partir del plano que nos viene dado por defecto:**

- a. Abrimos Blender. Si no tenemos el plano, activa la vista superior (**Key Pad 7**) y crea un plano **ADD->MESH->Plane** en el centro de la ventana.
- b. Activa el modo Edit Mode del plano y selecciona los dos vértices de la vista superior.
- c. Con **S** los escalamos hasta conseguir la forma de la imagen.



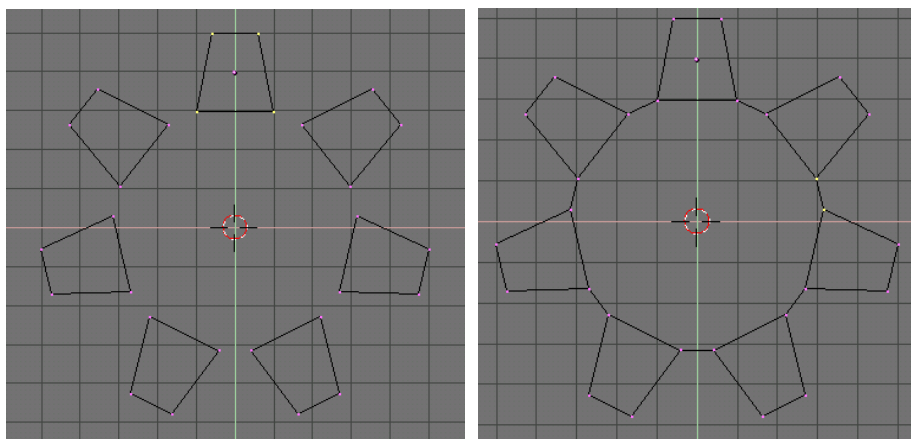
- d. Desactivamos Edit Mode y desplazamos (**G**) el plano cuatro unidades hacia arriba. Recuerda que si quieres “restringir” el movimiento del objeto a las cuadrícula de la rejilla, mantén apretado **CTRL** mientras mueves el ratón.



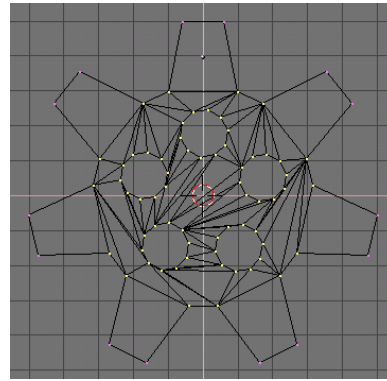
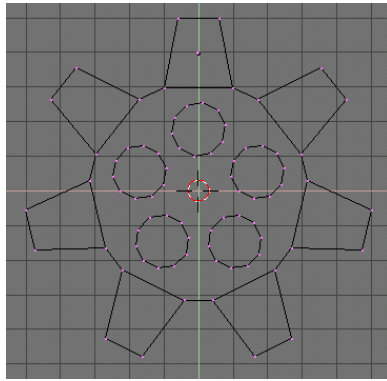
- e. Volvemos a Edit Mode y seleccionamos todos los vértices del plano (**A**). Nos vamos a Edit Buttons (**F9**) para hacer el Spin Dup.



- f. Recuerda que, como en la operación de Spin, el cursor 3D será quien indique el eje de rotación del Spin Dup. Fíjate en las dos ilustraciones.
- g. Con **Steps** indicamos el número de objetos que se duplicarán en la operación Spin Dup y con **Degr** indicamos el grado de rotación del Spin Dup. En nuestro caso daremos a Steps un valor de 7 y en Degr un valor de 360° (cubrirá todo el recorrido).
- h. Pulsa en el botón Spin Dup y aparecerá en pantalla los siete duplicados. Al igual que en el caso de SPIN, se cuenta siete duplicados más el original. El original y el último duplicado se solapan. Para eliminar esos “vértices” innecesarios nos servimos de **Rem Doubles (ver SPIN)**.



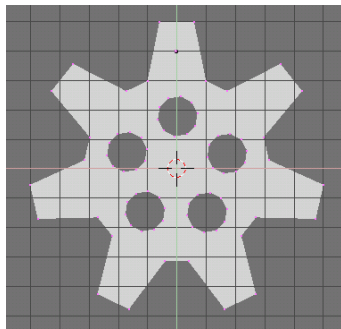
- i. Para completar la base, conectamos manualmente los vértices de la base dos a dos (**selecciona dos y pulsa F**), tal como aparece en la imagen de la derecha.
- j. Vamos a hacer las perforaciones del interior de la tuerca. Seguimos el mismo proceso, pero en este caso creamos un círculo (**ADD->MESH->CIRCLE**) de pocos vértices (**10**).
- k. Repite la misma operación hasta conseguir un resultado parecido al de la imagen.



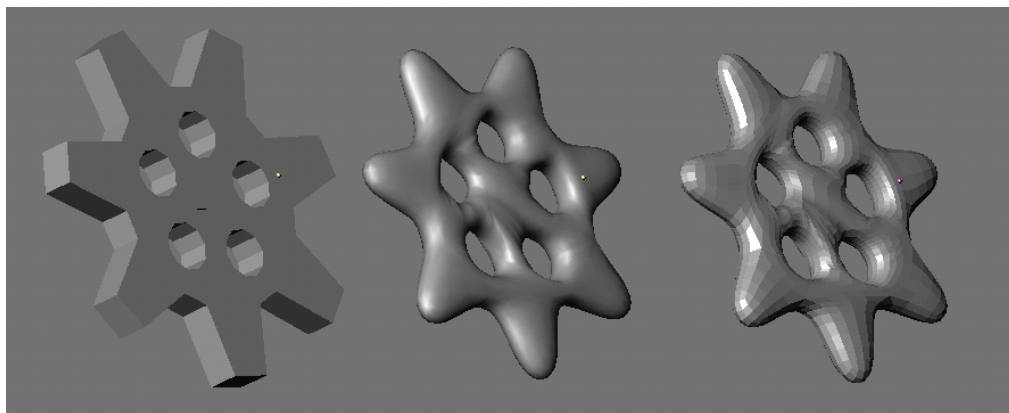
- l. Para rellenar el espacio que va de los “dientes” de la tuerca a las “perforaciones” (pulsas **Z**, modo shade para comprobarlo), selecciona los vértices base de los dientes y todos los círculos. Ayúdate del modo **BB** para una selección rápida.

NOTA: Con los vértices seleccionados, prueba a pulsar **CTRL+F** (varía la disposición de las aristas) y **J** para eliminar aquellas aristas innecesarias.

- m. Una vez seleccionados los vértices, tal como se ve en la ilustración, pulsa **SHIFT+F** para rellenar la superficie. Blender dibujará un nº mínimo de caras para esta operación.



- n. Dibujada la superficie, sólo nos falta extruir la cara de la tuerca para darle volumen. Para ello activaremos una vista en la que veamos el contorno de la figura; seleccionamos todos sus vértices y extruimos (**E**) hacia una dirección vertical.
- o. Si rotamos la vista veremos la pieza acabada tal como se muestra en la ilustración. La imagen central es mediante la función **Sub Div** activada en modo **Set Smooth**. La de la derecha corresponde a la anterior pero en modo **Set Solid**.



SUBDIVISION SURFACES

La **Subdivisión de superficies (subdivision surfaces)** es una técnica bastante reciente que ha ganado terreno como solución rápida y flexible para el modelado de superficies orgánicas. Fue introducida en producciones como *A Bug's Life* y *Geri's Game*. Consiste, básicamente, en presentar una malla poligonal como superficie curva (curved patches). De esta forma, el aspecto de la malla no corresponde a la baja resolución propia del modelado poligonal (donde se distingue las caras del modelo) sino a las superficies generadas por curvas Bèzier o NURBS. Y por otro lado se evita tener que trabajar con un nº elevado de polígonos.

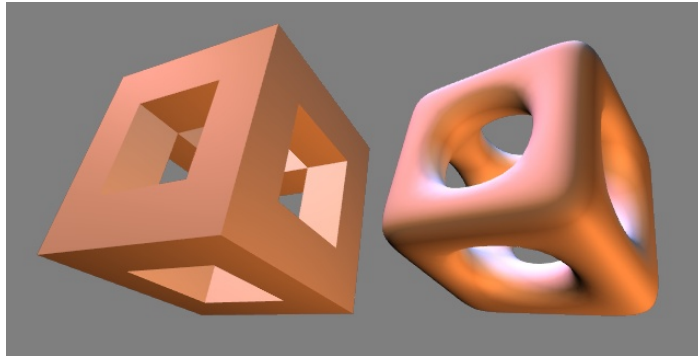
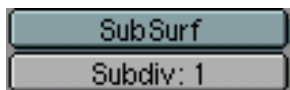


Ilustración que demuestra la técnica **Subdivision Surfaces**. NAN? 2001

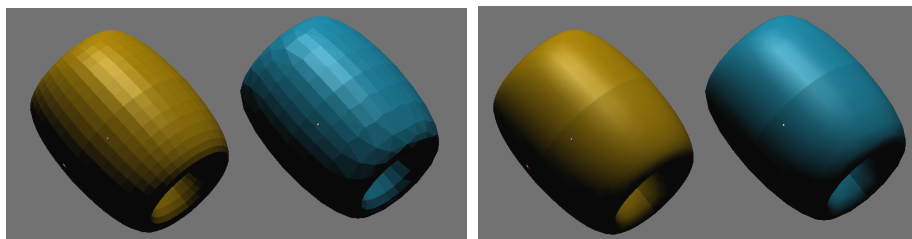
El algoritmo básico de la **Subdivision Surfaces**, conocidas como **Catmull-Clark surfaces** es dividir cada cara del modelo en cuatro sub-caras, donde el vértice central de aquellas se ubica justo en el centro de la cara del modelo. Un ejemplo de ello será la creación de un cubo y activar la operación **SubSurf** a través del **menú Edit Buttons (F9)** con el valor de la **Subdiv** en uno. Si aumentamos el valor de **Subdiv** a tres, comprobaremos cómo adopta la forma esférica manteniendo el mismo valor de vértices y aumentando muy poco la memoria.



Es muy probable que, una vez aplicada la operación de **SubSurf**, aparezcan extrañas manchas negras en la superficie del modelo (**activar modo Shade con Z**). Esto es debido a la dirección de las normales de las caras del modelo: para “recalcular” las normales, pulsaremos **CTRL+N**, seleccionando todos los vértices (**A**) en modo **Edit Mode (TAB)**. Si deseamos convertir a malla poligonal el modelo **SubSurf**, desactivamos **Edit Mode** y con **ALT+C** lo convertirá a polígonos manteniendo el original. Esto nos puede ser de utilidad si deseamos posteriormente trabajar con las funciones de **VertexPaint** y **Radiosidad** de Blender.

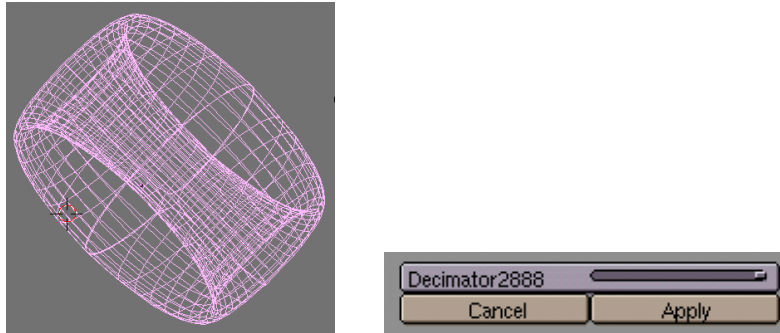
DECIMATOR

Esta función tiene como objetivo reducir el nº original de polígonos de un modelo de la escena, pero intentando conservar su identidad lo máximo posible. Un ejemplo:

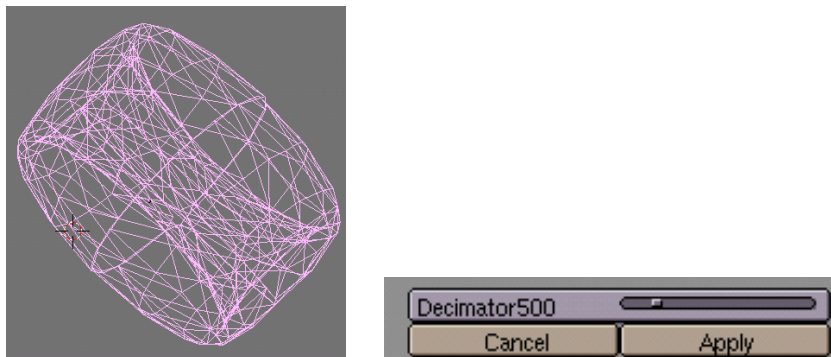


El modelo azul es un duplicado del modelo de color amarillo. En un principio, los dos tienen 2888 vértices. A través de la función **decimator**, hemos reducido a 1200 el número de vértices del modelo azul manteniendo su identidad o forma. Si aplicamos el suavizado (**set smooth**) a los dos, el resultado no difiere en absoluto:

¿Cómo aplicar la función decimator? Seleccionamos el modelo en modo selección objeto y nos dirigimos al menú **Edit Buttons (F9)** donde aparece el botón de **decimator** con el nº de vértices del objeto.



En este caso, tenemos seleccionado el modelo amarillo. Para una mejor comprensión, vamos a reducir el objeto de 2888 vértices a 500 vértices. Pulsamos **Apply** para ejecutar la operación.



Es conveniente probar diferentes resultados hasta conseguir el más óptimo: que equilibre el nº de vértices y el aspecto final del modelo.