Proyecto VRML

El misterio de la casa del lago

1. Introducción

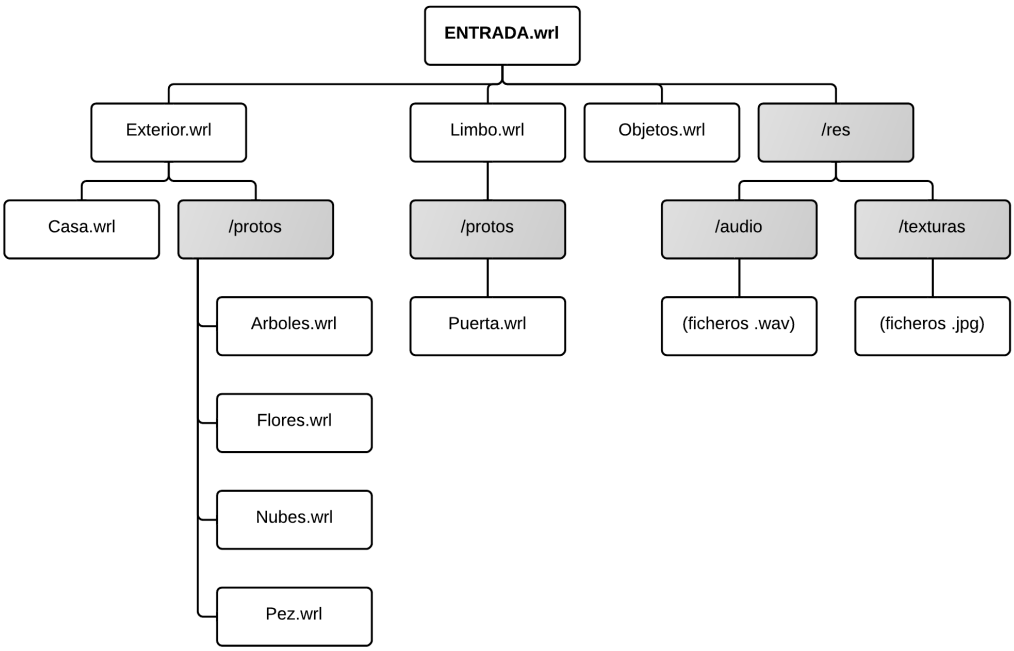
El mundo VRML construido consta de dos escenarios principales, *Exterior* y *Limbo*, conectados entre ellos. El fichero desde el cual se accede al mundo es *ENTRADA.wrl*, en el que se especifican los parámetros iniciales del usuario y otros parámetros generales.

A modo de minijuego en primera persona, al acceder al mundo el avatar se encuentra en la escena Exterior, que representa un terreno cubierto de hierba, con una casa sobre una colina, dos estanques, nubes en el cielo y algunos grupos de árboles y flores. El avatar puede interactuar con un cartel situado junto a uno de los lagos, y escuchar sus chapoteos al caminar por los estanques.

Al intentar abrir la puerta de la casa misteriosa sobre la colina, el avatar se ve transportado al Limbo, en el que hay una gran cantidad de puertas que rotan de forma que siempre están de cara hacia él. Sólo una de las puertas lleva de vuelta al Exterior, todas las demás llevan de nuevo al centro del Limbo… ¿Podrá el usuario volver al mundo real? ¿O quedará atrapado en el Limbo para siempre?

2. Estructura de ficheros

Al descomprimir el proyecto, aparecen todos los ficheros y recursos necesarios para visualizar el mundo. El siguiente esquema muestra, aproximadamente, las dependencias entre los ficheros, más que la forma exacta en la que están organizados. Los elementos sombreados representan directorios, y el resto representa tanto los ficheros VRML como los recursos de imagen y sonido.



3. Descripción de componentes

A continuación se describen los elementos más representativos utilizados para construir las escenas, clasificados según su tipo. El código se ha comentado y formateado de forma que resulte más fácil localizar cada una de las partes de cada componente.

3.1. Componentes visuales

En ambas escenas, la mayoría de elementos han sido definidos a partir de primitivas básicas, aunque también se han incluido muestras de elementos poligonales propios y de formas creadas mediante extrusión. A continuación se detallan las características de cada uno, organizados según el fichero en el que están definidos.

Exterior.wrl

Los únicos componentes visibles propios del fichero de esta escena (que es la principal del mundo) son el terreno, los dos lagos y el fondo.

El terreno verde se ha definido mediante una *ElevationGrid* de 30x20, asignando alturas distintas a cada vértice para conseguir las elevaciones y depresiones. Se ha especificado un *crease angle* elevado para suavizar los bordes, y se ha asignado una textura de hierba, punto por punto, para conseguir una apariencia de hierba.

En cuanto a los lagos, consisten en nodos *Box* con textura de agua, que al ser intersectados con las depresiones de la malla de elevación producen un buen efecto en las orillas.

El fondo, por último, consiste en un nodo *Background,* con un degradado de tonos azules para el "cielo" y otro de verdes para el "suelo". Se probó con imágenes de fondo, pero el efecto resultante no era mejor.

El resto del fichero consta de instanciaciones de prototipos definidos en otros ficheros (árboles, peces, etc.) y de elementos no gráficos (sonidos, sensores y scripts) que se describen más abajo.

Arboles.wrl

Se han definido dos prototipos diferentes de árboles. En ambos el tronco está formado simplemente por un cilindro, y el follaje está hecho en uno mediante 3 esferas deformadas, y en otro mediante 3 conos. Los dos modelos de árbol tienen texturas diferentes para el tronco y para el follaje.

También se han definido prototipos de dos tipos de tronco cortado, con las mismas texturas que los dos tipos de árboles, para ser utilizados a modo de tocones.

Por último, tanto para los árboles como para los tocones se han utilizado nodos *LOD*, con el fin de que a partir de cierta distancia los modelos no se dibujen, reduciendo el número de puntos y texturas a dibujar.

Flores.wrl

Las flores se han definido mediante un prototipo con cinco parámetros, que permiten elegir las características de una flor (altura, colores, rotación, etc.) en el momento de instanciarla. En cuanto a la geometría, todas los componentes de la flor están hechos mediante cilindros, en este caso sin ningún tipo de texturas, ya que taparían los colores asignados.

Se ha utilizado un nodo *LOD* para definir tres niveles de detalle: desde cerca se ven todos los pétalos, desde distancias medias se muestra sólo un disco, y desde lejos, al igual que en el caso de los árboles, no se muestra nada.

Nubes.wrl

Se han creado dos prototipos de nubes, cada una con un parámetro que permite elegir la posición inicial en la que aparecerá. Ambos modelos están formados por unos pocos discos (cilindros) de color blanco, sin texturas.

Pez.wrl

El prototipo de pez está creado mediante la extrusión de un octógono a lo largo del eje X, escalándolo en diversos puntos para lograr la silueta adecuada. Se le ha puesto una textura de escamas, y se ha rotado de manera que la junta entre los dos extremos de la textura, en la que se producía un salto de color demasiado visible, quedase en la parte inferior del modelo, ya que ésta no será visible.

Objetos.wrl

Este fichero describe los objetos que puede llevar el avatar en la mano: una caña de pescar, un hacha de leñador, o nada. Describe también un cartel de madera que, al ser clicado, permite cambiar entre los objetos.

La caña está compuesta íntegramente por cilindros de diferentes tamaños y grosores: cuatro para la caña y uno para el sedal. Todos ellos tienen colores planos: negro y amarillo para el mango y la caña, y gris para el sedal.

El hacha, por su parte, consta de un cilindro con textura de madera para el mango, y una figura poligonal de 9 caras (definida mediante un *IndexedFaceSet*) y un material plateado para la hoja.

A ambos elementos se les ha asociado un sensor *ProximitySensor* para que, mediante un *Script*, sigan el movimiento del avatar, produciendo el efecto de que los lleva en la mano. Además, los objetos están englobados dentro de un nodo *Switch*, de manera que mediante el script también se controla cuál de los objetos se dibuja delante del avatar: la caña, el hacha o ninguno de los dos.

Además, para incrementar un poco el realismo, se han colocado copias de ambos objetos junto al cartel, y al igual que en el caso anterior, se han usado nodos *Switch* controlados desde el script para hacer que desaparezca el objeto que el avatar lleve en la mano.

El cartel, por último, consiste en un cilindro y una caja, ambos con textura de madera, así como de un nodo *Text* que define un texto multilínea de color marrón oscuro, simulando un grabado en la madera. Toda la estructura del cartel está asociada a un *TouchSensor*, que es el que al ser clicado activa el script para cambiar entre objetos.

Casa.wrl

La casa no está definida como prototipo, ya que no se iba a utilizar más de una, por lo que ha sido definida en un fichero aparte simplemente para reducir el número de líneas del fichero de la escena principal, en la que está incluida por medio de un nodo *Inline*.

El modelo consta de tres partes: las paredes, hechas mediante un nodo *Box* con textura de listones de madera; el techo, definido como un polígono de cinco caras por medio de un nodo *IndexedFaceSet* y con una textura de chapa metálica; y la puerta, consistente en un nodo *Box* para la hoja (con una imagen de puerta de madera como textura) y una esfera dorada para el pomo.

Puerta.wrl

Consiste en una puerta similar a la de la casa descrita anteriormente, pero sin textura, sólo con un color gris plano en su lugar. Se le ha asociado un nodo *Anchor*, y el prototipo dispone de dos parámetros que definen los atributos de dicho nodo. Además, la figura está incluida dentro de un nodo *Billboard*, por lo que las puertas creadas a partir de este prototipo rotarán para estar siempre de cara al avatar.

Limbo.wrl

El Limbo es la otra escena del mundo, mucho más simple –y misteriosa– que la anterior. Consta de una gran superficie plana de color blanco (definida mediante un nodo *Box*) sobre un *Background* negro, y 401 puertas distribuidas en forma de malla –oh, ¡eso es imposible!– sobre la superficie.

3.2. Componentes no visuales

Los componentes no visuales de la escena Exterior se definen en el fichero *ENTRADA.wrl*. Consisten en un nodo *WorldInfo* con el título e información sobre el mundo, un nodo *NavigationInfo* con la especificación del avatar, dos nodos *ViewPoint* (uno para la primera vez que se entra a la escena, y otro para cuando se vuelve a ella desde el Limbo), y una luz *PointLight* situada en lo alto que representa el sol y provoca que haya zonas de los objetos en sombra. No se ha considerado apropiada la inclusión de más fuentes de luz, ya que se está representando una escena al aire libre, ni la inclusión de más *viewpoints* predefinidos, ya que el escenario es lo suficientemente plano y reducido como para que se pueda explorar entero en pocos segundos.

En la escena Limbo, del mismo modo, se han especificado únicamente un nodo *NavigationInfo* y una luz *PointLight* iguales a los anteriores, y un único *ViewPoint* al que se accede cada vez que se entra (o se vuelve) a la escena. De nuevo, no se ha considerado necesario incluir más luces ni más puntos de vista.

3.3. Animaciones

Las nubes descritas anteriormente se desplazan sobre la escena principal por medio de interpoladores lineales. Mediante el uso de nodos *TimeSensor* con distintos valores de intervalo y de nodos *PositionInterpolator*, se pueden asignar movimientos distintos a cada nube o grupo de nubes, pudiendo además indicar una traslación inicial al instanciar las nubes.

Por otro lado, los peces del estanque grande se mueven en una ruta en forma de cuadrado mediante un script, definido mediante un prototipo y con parámetros que permiten determinar propiedades como el tamaño y la rotación de dicho cuadrado o la velocidad de movimiento del pez. De esta forma, al usar prototipos tanto para los peces como para el script de movimiento, se ha podido definir un movimiento diferente para cada pez. *(Nota: en el script hay código comentado porque se intentó crear un movimiento aleatorio dentro de dicha área cuadrada, pero no llegó a funcionar correctamente.)*

Por último, y aunque quizás no se considere del todo como “animación”, las puertas del Limbo descritas más arriba rotan automáticamente para estar siempre de cara al avatar, gracias nodos tipo *Billboard*.

3.4. Interactividad

Aunque no muchos, se han incluido algunos ejemplos de elementos interactivos en ambas escenas.

Por un lado, y como se ha detallado antes, el cartel de madera situado en la escena *Exterior* permite, al ser clicado, cambiar el objeto que lleva el avatar en la mano. Esto se ha conseguido asociando un nodo *TouchSensor* al cartel, que al ser clicado con el ratón activa un script que modifica el valor de varios nodos *Switch*, mostrando u ocultando tanto los objetos que lleva el avatar como los que hay apoyados junto al cartel.

Por otro lado, todas las puertas en las dos escenas poseen enlaces Anchor, que al ser clicados llevan a un *viewpoint* determinado. En el caso de la puerta de la casa, sólo el pomo es clicable, y el enlace lleva al centro de la escena Limbo. En éste, todas las puertas son clicables en toda su superficie, pero todas llevan de nuevo al centro del Limbo… Todas excepto una, situada justo en el centro, en la que sólo el pomo es clicable, y lleva de nuevo a la escena Exterior. La vuelta al mundo real está mucho más cerca de lo que parece en un principio…

3.5. Otros componentes

Como se ha descrito en varios de los puntos anteriores, se ha hecho uso en repetidas ocasiones de sentencias *PROTO* para definir diversos elementos (con o sin parámetros), que después pueden ser redefinidos en otro fichero mediante el uso de *EXTERNPROTO*, y pueden ser instanciados como el resto de nodos del lenguaje.

También se han descrito dos ejemplos del uso de nodos *Script*: uno para el movimiento de los peces que nadan en uno de los estanques, controlando la posición y la rotación del pez en cada instante, y con posibilidad de especificar, entre otras cosas, el tamaño del cuadrado que recorre o la velocidad; y otro para la selección de los objetos que puede llevar el avatar, controlando tanto el objeto que lleva como los que quedan apoyados en el cartel.

No se han usado programas externos (SkecthUp, Blender, etc.) para crear ninguno de los modelos. Las únicas “herramientas” externas que se han utilizado han sido pequeños scripts propios en lenguaje Python, escritos para crear vectores o matrices grandes (para mallas de elevación, etc.) con menos esfuerzo. Sin embargo, la mayoría se usaron solamente para pruebas y fueron descartados, por lo que sólo se ha podido incluir uno de ellos en el proyecto (carpeta “/res”).

4. Problemas encontrados

Los principales problemas encontrados durante el desarrollo del proyecto han sido los relacionados con las limitaciones del lenguaje VRML y las herramientas para trabajar con él.

Al trabajar con Mac OS X, era imposible utilizar tanto el visualizador Cortona3D como el editor VRMLPad, con los inconvenientes que ello implica, ya que VRMLPad es sin duda el mejor editor para escribir VRML rápidamente y con coloreado de sintaxis, y Cortona es el visualizador más estable. Por tanto, la mayor parte del desarrollo ha sido realizado con el visualizador InstantPlayer, bastante inestable a veces y con notables diferencias en la renderización respecto a Cortona; y el editor de textos Sublime Text 2, con el que con un poco de configuración se pudo conseguir un buen coloreado de sintaxis, pero no otras funciones como la de autocompleción. Cada cierto tiempo, por tanto, se ha tenido

que usar Windows sobre una máquina virtual para poder comprobar el progreso realizado y corregir errores que el visualizador InstantPlayer no revelaba.

Por otra parte, la propia antigüedad del lenguaje VRML ha implicado que la mayoría de enlaces a recursos online estuviesen muy desactualizados o no funcionaran en absoluto, por lo que acceder a ejemplos y modelos VRML de muchos sitios ha sido prácticamente.

5. Conclusiones

La intención y la orientación de las prácticas y el proyecto con VRML me ha parecido muy interesante, ya que se trabaja muchísimo el manejo de figuras y transformaciones 3D, que son los fundamentos de los gráficos por computador, y que además ayudan mucho al desarrollo de la visión espacial, al tener que visualizar mentalmente qué transformaciones aplicar para hallar los parámetros adecuados.

Sin embargo, opino que el uso de VRML como lenguaje no es lo idea, no por sus posibilidades o rendimiento, sino porque su antigüedad ha hecho, como se ha comentado antes, que la mayoría de recursos online ya no estén disponibles, y que apenas haya información actual sobre el lenguaje ni comunidad de desarrolladores. Por tanto, creo que la asignatura podría ser mucho más interesante si se sustituyera VRML por alguna tecnología más moderna como openGL o webGL, o, al menos, por su sucesor X3D, que sí que parece estar más estandarizado y tener una comunidad de usuarios mayor.

6. Bibliografía

No se han utilizado elementos ni librerías externos, a excepción de las texturas, seleccionadas desde el buscador de imágenes de Google, y los archivos de audio. Por tanto, las fuentes consultadas son en su mayoría tutoriales online de VRML, así como alguna web de recursos. *(Nota: No se especifican las fuentes que han sido consultadas de forma casual o puntual, sólo las que han sido utilizadas de forma recurrente.)*

- AudioMicro, librería de audio de *stock* con algunos sonidos gratuitos.
- The Annotated VRML 97 Reference, referencia online de VRML.
- VRML Interactive Tutorial, tutorial de VRML, similar al anterior.
- RGB Color Wheel, selector de colores RGB.