ESCALEXTRIC EN OPENGL

Realidad Virtual

5 DE JULIO DE 2018

UNIVERSIDAD DE HUELVA – ETSI – 3° GRADO DE INGENIERÍA INFORMÁTICA

Jesús Campos Márquez

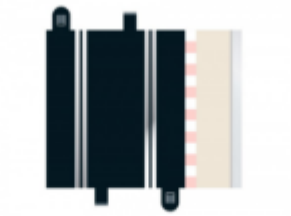
ÍNDICE

• • •

1. Descripción de los objetos que desarrollan cada pista.

Para desarrollar las siguientes pistas:

Recta Estándar
Longitud: 350 mm



Media recta
Longitud: 175 mm

Cuarto de recta
Longitud: 87.5 mm

Se ha creado un objeto Recta y un objeto Pieza_Recta.

El primero corresponde a una pieza del tipo de Figure3D, utilizado para crear lo que compondrá el material de la pista (Texturas y luces), y que básicamente creará una Pieza_Recta con el material correspondiente a esa pista.

El segundo objeto que corresponde a Pieza_Recta, creará una pista con el material asociado, y con el largo y ancho que deseemos. De esta forma, utilizamos los mismos objetos para las 3 rectas, pero usando el largo y ancho que les corresponda. La Recta se forma mediante la conexión entre vértices, formando triángulos.

Para desarrollar la siguiente pista:

Curva interior
Radio interior: 58 mm
Radio exterior: 214 mm
Ángulo: 45°

Curva estándar
Radio interior: 214 mm
Radio exterior: 370 mm
Ángulo: 45°

Curva exterior
Radio interior: 370 mm
Radio exterior: 526 mm
Ángulo: 22.5°

Se ha creado los objetos CurvaInterior, CurvaExterior y CurvaStd, y los objetos Pieza_CurvaInt, Pieza_CurvaExt y Pieza_CurvaSTD.

Para los primeros objetos, corresponde como en las rectas a una pieza del tipo de Figure3D, para darle las propiedades del material a la pista (su propia textura y luces).

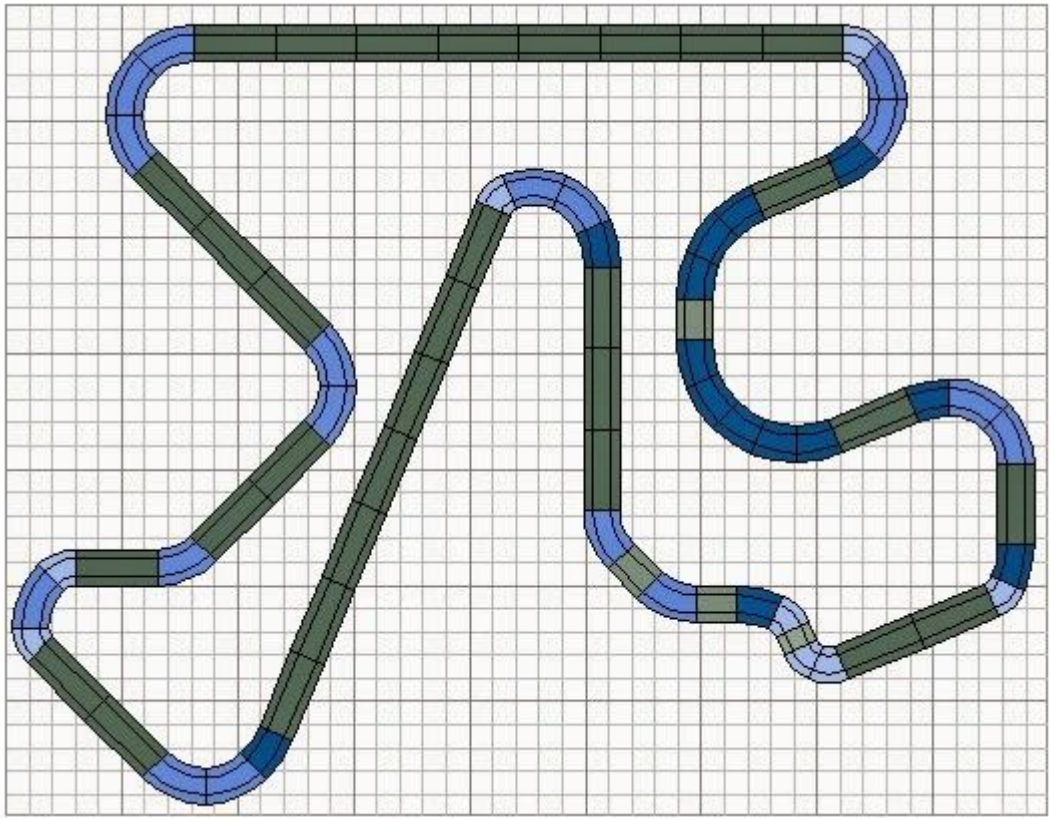
Los segundos corresponden a la formación de la curva con las características que deseemos, en este caso tenemos que tener en cuenta 2 radios y un ángulo para formarla. Para ello se ha dividido la pista en 12 vértices, 6 para el radio interior y 6 para el radio exterior, con el fin de conseguir calcular estos puntos con un ángulo determinado. Para formar estos triángulos sean indexado los vértices, por ejemplo, el 0 con el 2 y el 1, el 0 con el 3 y el 2, … Con ello se consigue que se vayan juntando los triángulos formados por esos vértices consiguiendo al final una curva total con radio interior y exterior, y ángulo como el especificado.

La particularidad que tiene este diseño es a la hora de texturizar la pista, se utiliza la textura de la Recta, diciendo que vértice corresponde a que parte de la textura y consiguiendo que cada triangulo tenga su textura.

No se ha utilizado las texturas de cada pista porque, al no tener un radio interior y exterior definido, el calcular cada punto de la textura con cada punto de los vértices que forman la pista se hace muy difícil, siendo esta una solución rápida y efectiva.

2. Descripción de la clase que desarrolla el circuito y la escena

Para la creación del circuito se utiliza totalmente la clase Scene3D. Esta, utiliza un skybox, un tapete (que es el propio suelo del skybox), la luz y las piezas descritas anteriormente, conectándolas entre sí, mediante un array de piezas indicado. En este caso utilizaremos el circuito de Jeréz.



Este está formado por un total de 31 rectas estándar, 3 medias rectas, 1 cuarto de recta, 8 curvas interiores, 17 curvas estándar y 14 curvas exteriores.

La construcción de la pista se lleva a cabo con una matriz que contiene las piezas en el orden correspondiente, en nuestro caso se ha empezado la pista por el punto rojo colocado en la foto anterior. Dentro de esta matriz se indica que pista es (1->RECTA ESTANDAR 2->MEDIA RECTA 3->CUARTO DE RECTA 4->CURVA INTERIOR 5->CURVA EXTERIOR 6->CURVA ESTANDAR) y el tipo de pista (1 si se gira hacia la derecha y un 2 si se gira hacia la izquierda).

Para montarlo todo se utiliza un for para recorrer las 74 piezas que forman el circuito, y un switch que dependiendo de que pista sea tendrá que crear la pieza con los parámetros correspondientes y hacer una serie de desplazamientos mediante las matrices de transformación para colocarse en el final de la pista anterior. La particularidad que tiene es que en la última pieza que consideramos que es una recta, debido a la perdida de

decimales al no encajar, la alargamos para que encaje el circuito perfectamente, y así evitar tener que crear un nuevo objeto que sea una pieza final.

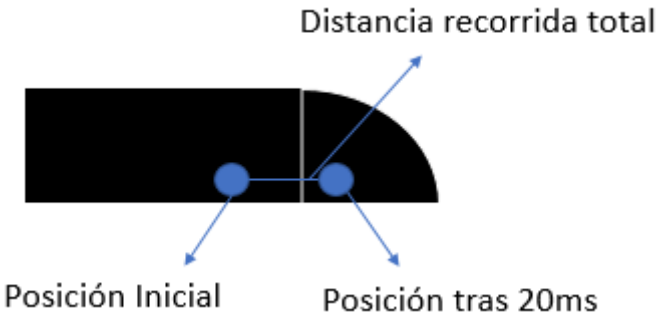
Esta clase, además tiene una jerarquía a la hora de pintar un modelo. Esto se indica en la función Draw, que colocará antes en nuestro mundo lo que esté antes, por ejemplo, el skybox se tendrá que pintar antes que las pistas para que quede por debajo y no se sobreponga a estas.

NOTA: Para reducir la dificultad desde mi punto de vista, he considerado que cada figura está formada por una pieza, en lugar de que una figura esta compuesta por 1 o muchas piezas, de esta forma cada pista será una figura y todo el ensamblado se lleva a cabo en el Scene3D

3. Descripción de la forma en la que se calcula la posición instantánea de cada vehículo.

La latencia de refresco de la ventana que visualiza todo el modelo es de 20ms, esto quiere decir que cada 0.02 segundos se actualiza la imagen en nuestra ventana. Además, para mover un coche necesitamos de velocidad por lo que, con lo anterior podemos saber, a que lugar le corresponde estar al coche cada 20ms dentro de nuestro circuito.

Todo esto se lleva a cabo dentro del método TimerAction de la clase Model3D. Para calcular la posición de cada vehículo tenemos que tener en cuenta en que pista se encuentra, de esta forma controlamos si tiene que girar un cierto ángulo si es una curva o ninguno en el caso de una recta. Todo ello, también tendrá que tener en cuenta que dada una pista si nos encontramos en una posición, y al siguiente frame se calcula que el coche debe estar en otra posición sobrepasando la pista anterior, el desfase producido de una pista a otra hay que guardarlo para que la siguiente pista sepa que el coche está en esa posición y no al principio.



Suponemos que l punto azul es el coche.

Se puede observar que la línea blanca que separa en dos la línea azul (distancia total recorrida) una corresponde a una parte de la recta y la otra a una parte de la curva, por lo que podemos ver que lo dicho anteriormente del desfase, hay que controlarlo.

Esto habrá que hacerlo para cada pista y coche.

NOTA: Tras muchos intentos de controlar el desfase, no he logrado corregir el error que se genera por lo que el coche puede salir de la pista.

El ángulo que debe de girar el coche en cada cual se calcula con una simple regla de tres. Supongamos que nos encontramos en una curva interior:

Angulo ---- Drecorrida
45º ---- x (desplazamiento total en x)

Para calcular el desplazamiento en ese frame con ese ángulo (pasado a radianes) se utiliza el método de desplazamiento de una curva que utiliza las siguientes ecuaciones:

z = radio * (1 - cos((angulo)));
x = radio * sin((angulo));

donde x y z son el desplazamiento del coche en su sistema de coordenadas.

Para que el simulado empiece a funcionar, se ha establecido la tecla ‘C’ para que se le dote a los coches inicialmente una velocidad igual y constante de 5.

Para que el coche 1 pueda aumentar o disminuir la velocidad, se han utilizado las teclas ‘Q’ y ‘A’ respectivamente.

Para que el coche 2 pueda aumentar o disminuir la velocidad, se han utilizado las teclas ‘O’ y ‘L’ respectivamente.

El límite de velocidad puesto es de 20. Cada pulsación de las teclas correspondientes, aumentarán o disminuirán la velocidad en una décima parte de la velocidad actual, siendo distinta a la pedida para que la velocidad sea mas progresiva.

4. Descripción de la forma en la que se desarrolla cada modo de visualización.

Para proveer al circuito de modos de visualización se han asignado teclas para ello.

Se ha creado el método TurnRightStep dentro de la clase Camera3D para poder hacer ciertos movimientos en la cámara a la hora de colocarla para ver el circuito y los coches.

La tecla 'F1' consiste en situar la cámara en una posición fija desde la que se pueda observar todo el circuito. Este modo de visualización pretende simular el punto de vista de un jugador de Scalextric. Para llevarlo a cabo al pulsar la tecla se usará la clase Camera3D que girará esta 90º y la situará a una altura de 600 por ejemplo con unos desplazamientos para poder apreciar el circuito completamente.

La tecla 'F2' consiste en situar la cámara detrás del primer coche, a una pequeña altura que permita apreciar el coche y los tramos de pista que se encuentren delante. Para llevar esto a cabo se ha colocado la cámara a una altura de 8 y 20 hacia atrás, consiguiendo así ver el coche completo y la pista.

La tecla 'F3' es similar a la anterior peor con la vista del segundo coche.

NOTA: Para hacerlo correctamente se debería de hacer con la inversa del modelo, pero al no conseguirlo lo he dejado para que simplemente siga al coche y, al menos, se vea el seguimiento.

5. Referencia a la procedencia de cualquier textura o modelo que se haya incorporado al trabajo.

Las texturas de las pistas han sido proporcionadas por el profesor, aunque solo se utiliza la textura de RectaStd.png dado que, al intentar colocar la textura de las curvas en su correspondiente curva, no lo he conseguido como he explicado anteriormente.

La textura del SkyBox ha sido obtenida de la página:

- http://www.humus.name

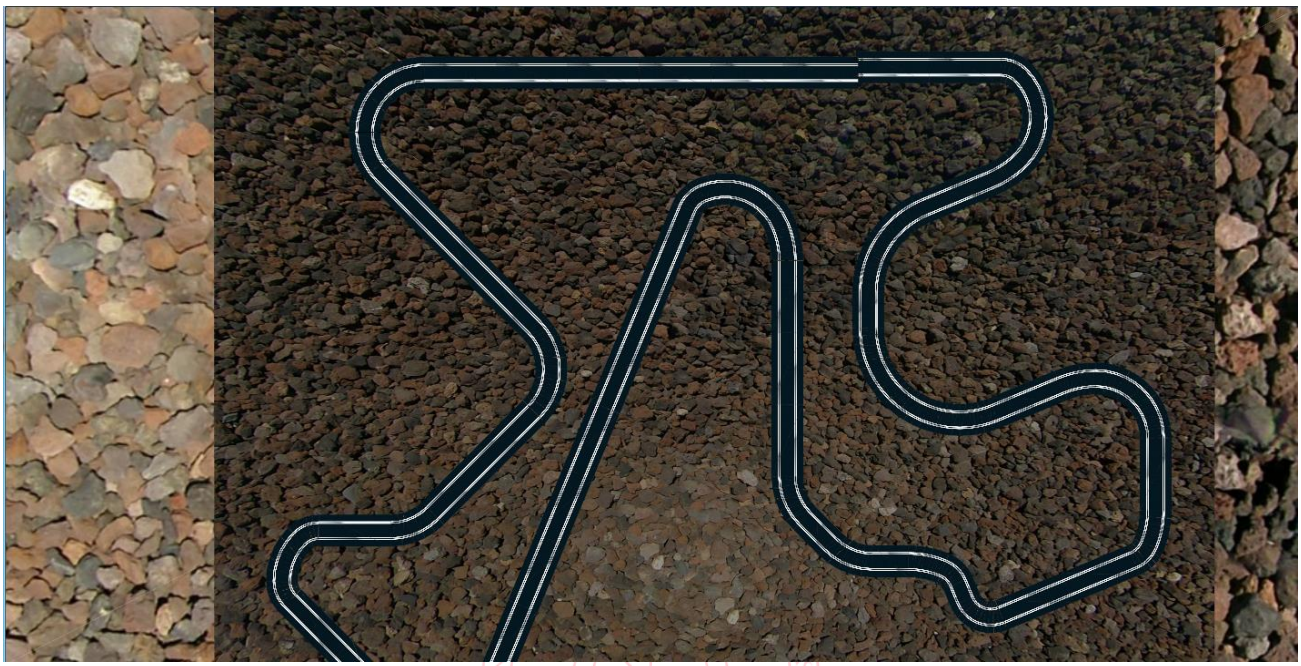
La textura de Césped (el tapete que se ve bajo la pista) se ha obtenido mediante el suelo del SkyBox, que básicamente es lo mismo, pero a un tamaño más reducido.

Los coches utilizados son los proporcionados por el profesor, obtenidos de la página:

- https://free3d.com/

6. Pruebas del funcionamiento de la aplicación (capturas de imágenes).

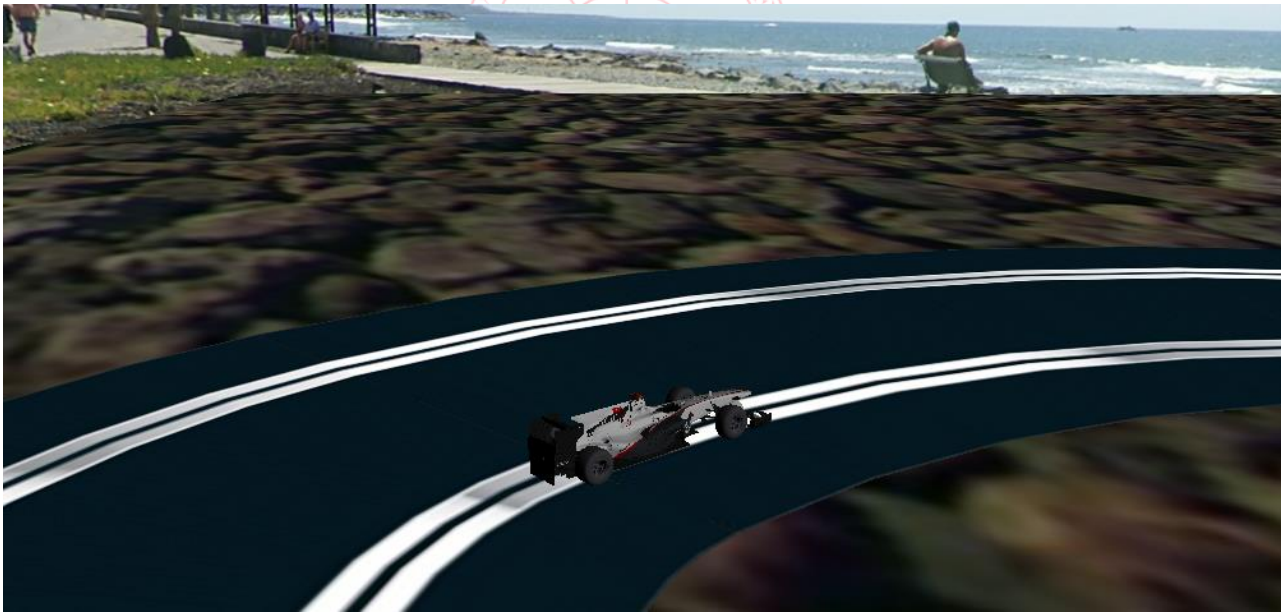
Cámara inicial y F1:



Cámara F2:

Cámara F3:

Cámara F2 en movimiento:



Cámara F3 en movimiento:

Salida de curva: