

Diseño de la aplicación

SuperKart

Víctor José Rubia López y Claudia Salado Méndez

Junio 2021

1. Diseño

A continuación, explicaremos las diferentes clases que componen el juego. La descripción es sencilla ya que puede encontrar más información consultando los comentarios que hay en el código.

- **MyPhysiScene**: esta clase es la principal y posee el funcionamiento básico de una escena que usa el motor de Physijs. Hemos usado una cámara con perspectiva, con la cual seguiremos al coche durante todo el juego desde detrás del mismo, una luz ambiental y eventos que manejan los controles mediante el teclado. Cuando termina el juego se muestra un ranking con las posiciones finales de los jugadores y las monedas obtenidas. Debemos recalcar que no hemos usado sombras porque ralentizaba la aplicación tanto que lo hacía casi injugable y no le aportaba la identidad visual que queríamos.
- **Coche**: es un vehículo conformado por un modelo jerárquico de varias cajas (al estilo voxel) y unas ruedas. Para otorgarle físicas y que las ruedas actúen como si fuesen del mundo real, usamos un tipo de objeto que brinda Physijs llamado **Vehicle**. Esto se encarga de darle movimiento mediante unas funciones para acelerar y frenar que se ejecutan cuando se recibe un evento desde la clase principal.
- **CochesEnemigos**: Esta clase es similar a la anterior, con la salvedad que creamos el coche como forma (**Object3D**) y esto se le añade a una caja física que nos proporciona Physijs. Además, cada coche tiene un spline asociado, el cual se genera aleatoriamente dentro de los límites establecidos de la carretera. Este spline hará que, mediante una animación **TWEEN** bastante medida, usando **random** para que tengan velocidades distintas, se simule la conducción por parte de estos coches.
- **Mapa**: Clase que crea el suelo, con el circuito, la meta, las barreras que han de formar los obstáculos del propio circuito, muros invisibles para que los objetos no puedan salirse del circuito, barreras para ir controlando la

posición relativa de los coches y un fondo cúbico para dar un efecto de profundidad. Todas las barreras se han creado usando Physijs.

- MuroPosicion: Clase en la que se apoya Mapa para crear las secciones para calcular las posiciones relativas.
- Moneda: Clase que crea monedas que se reparten a lo largo del circuito. En el Ranking final se indica el número de monedas que ha conseguido cada jugador. Si el jugador recoge las monedas le otorga un bonus de velocidad.
- Posiciones: Archivo que engloba funciones de JavaScript para calcular las posiciones relativas y así conformar un ranking final. Internamente las calcula en función de las secciones y, si se encuentran en la misma sección, se calcula la distancia relativa con la siguiente sección, siendo el que tenga menor distancia el jugador que va primero.

2. Diagrama de clases

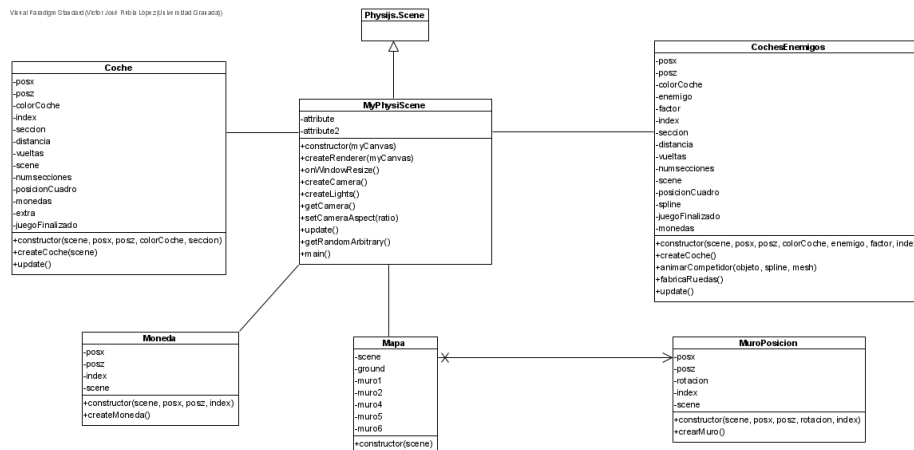


Figura 1: Este diagrama de clases no recoge con exactitud el contenido total del juego, ni tampoco cómo funciona. A pesar de poder haber usado JavaScript orientado a Objetos, no hemos podido, ya que el uso de la librería Physijs, que desde hace 6 años no se actualiza, no nos permite tratarla como un módulo.

Para terminar, destacamos que las clases Coche, CochesEnemigos, Moneda, Mapa y MuroPosicion, representan objetos físicos y por ende, han sido modelados mediante Physijs.

Referencias

- [1] Nintendo. *Imagen del circuito obtenido de Super Mario Circuit*. Super Mario Kart SNES, 1992. Recuperado de [Meristation](#)
- [2] Nintendo. *Fondo de la escena*. [*DL_VRHyrule_Alb.png*]. Mario Kart 8 Deluxe, 2017. Recuperado de [Textures-Resource](#)
- [3] Nintendo. *Audio del juego*. [*DL_VRHyrule_Alb.png*]. Mario Kart 8 Deluxe, 2017. Recuperado de [vgmsite](#)
- [4] Nintendo. *Modelo de la moneda*. [*Super Mario style golden coin Free low-poly 3D model*]. Super Mario, 1985. Recuperado de [cgtrader](#)
- [5] vladispavlov. *Biblioteca usada para el velocímetro*. Recuperado de [CSSScript](#)
- [6] Physijs. *Biblioteca usada para físicas*. Recuperado de [GitHub](#)