

MEMORIA



Programación concurrente

GDDV 4.3

Convocatoria ordinaria
Primera Práctica del Primer Semestre - 2023/24.

Simulador de conducción con carreras en línea

Profesor: Maximiliano Miranda Esteban

Contenido

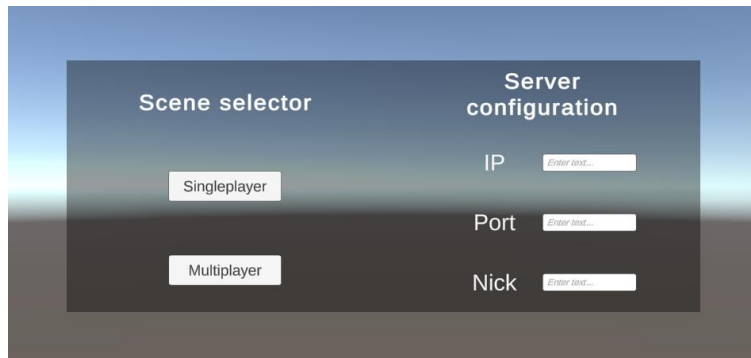
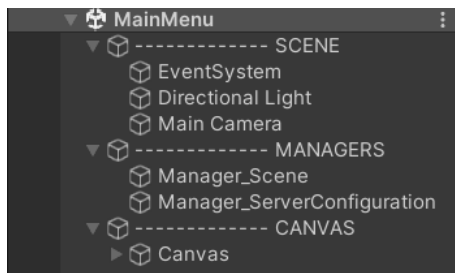
Contenido.....	2
Desarrollo del proyecto	3
Gestión de escena - Elementos del juego	3
Menú principal	3
Singleplayer	3
Multiplayer	4
Sistemas	4
Cámara	4
Controlador de vehículos	5
Uso de físicas.....	5
Controlador de IA.....	6
Gestión del circuito y carreras	6
Lógica de juego en red en el proyecto de Unity	8
Implementación	9
Complicaciones	9
Sistema de vueltas	9
Sistema de evasión de obstáculos	9
Aspectos de mayor interés	9
Conclusiones	10

Desarrollo del proyecto

Gestión de escena - Elementos del juego

3 escenas...

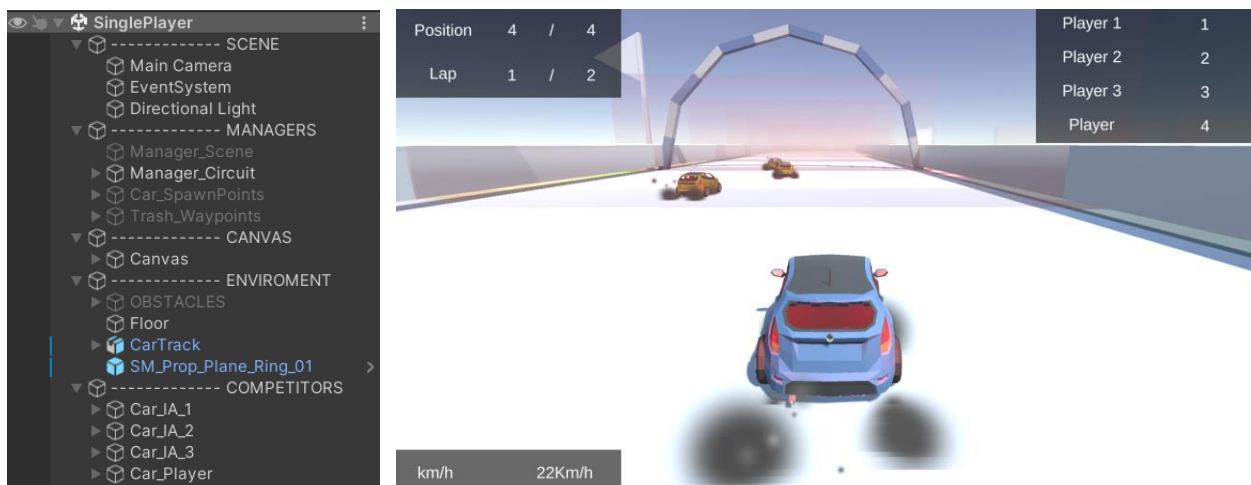
Menú principal



Gestiona la partida a la que quieres entrar, y si entras a una online puedes gestionar la información para conectarte de servidor y cliente introduciendo la IP y el Puerto, los cuales se gestionan mediante un scriptable object.

El nickname no logré implementarlo finalmente por pequeños fallos a pesar de que está introducido.

Singleplayer

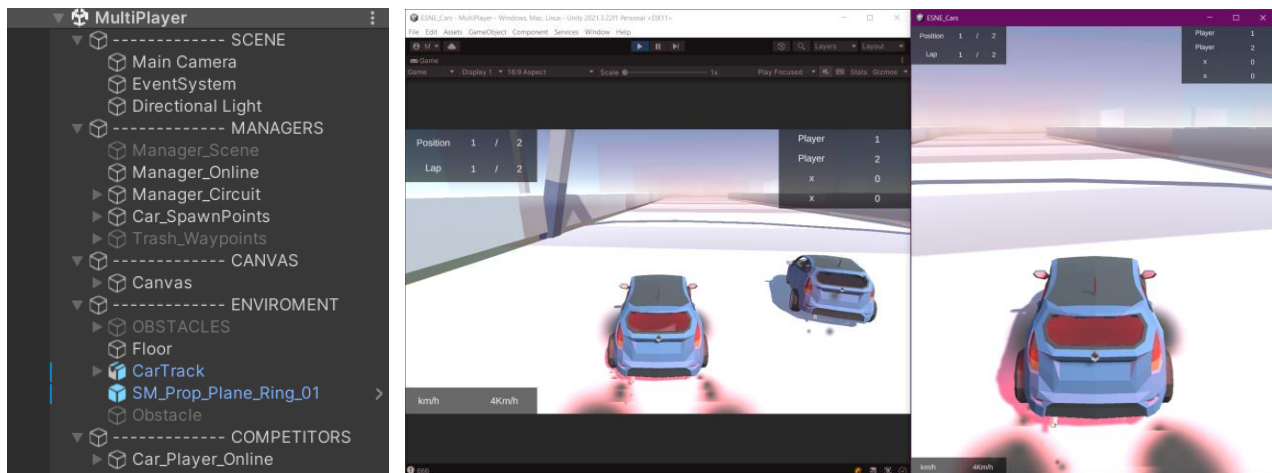


Compite con otros 3 jugadores que son unas IAs con su sistema de seguimiento de waypoints y de esquivar obstáculos hasta pasar todas las vueltas y ganar.

Tiene un canvas que gestiona perfectamente la tabla de resultados en tiempo real.

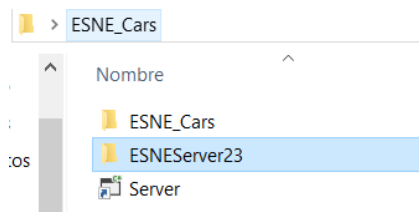
El escenario fue creado rápidamente en blender, aunque no logré tiempo para texturizarlo.

Multiplayer



El sistema de nombres no funciona correctamente ya que no logré implementarlo correctamente y todos se llaman igual, pero si que coge la posición bien de cada 1 y el ganador.

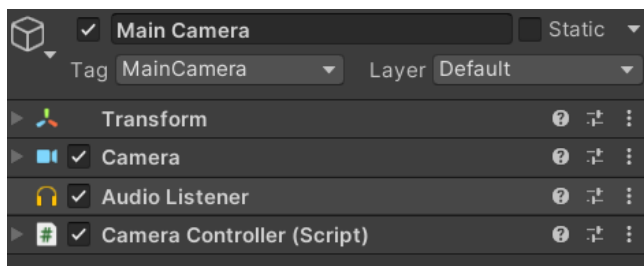
Funciona al ejecutar el host que se encuentra en el proyecto de la carpeta “ESNEServer23” al lado de la carpeta del proyecto de Unity.



Sistemas

Cámara

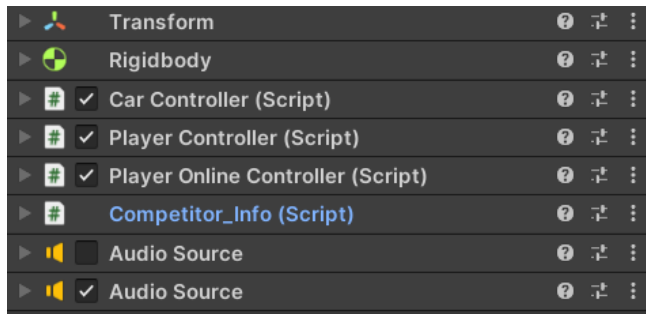
Componentes:



- **CameraController:** Logica para coger la posición del target y seguirlo con la cámara.

Controlador de vehículos

Componentes:



Componentes:

- **CarController:** Control del coche y gestión de la información de vueltas y waypoints.
- **PlayerController:** Coge el input del jugador para transmitirlo a la logica de “CarController”.
- **PlayerOnlineController** (en el caso solo de juego online, en el coche del jugador en singleplayer no está):
- **CompetitorInfo:** Tiene la información del jugador de posición, waypoints, nombre...

Usa las físicas y fuerzas con el peso del coche con un rigidbody para mover el coche.

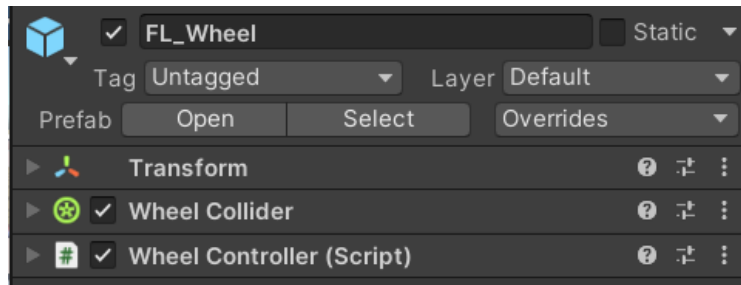
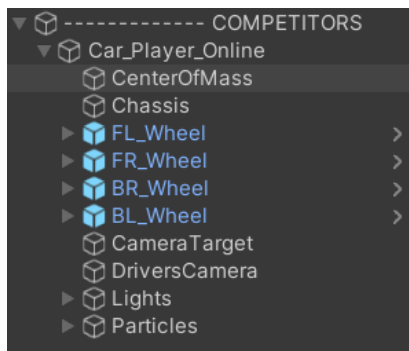
Sistemas internos:

- Sistema que controla el movimiento de la cámara según la velocidad del jugador.
- Sistema que controla las luces y sonido que genera el coche.
- Sistema que calcula la velocidad del coche y lo muestra en un canvas por pantalla.

Uso de físicas

Uso de la mesh del coche en “CarController” en si con un mesh collider para moverse y avanzar.

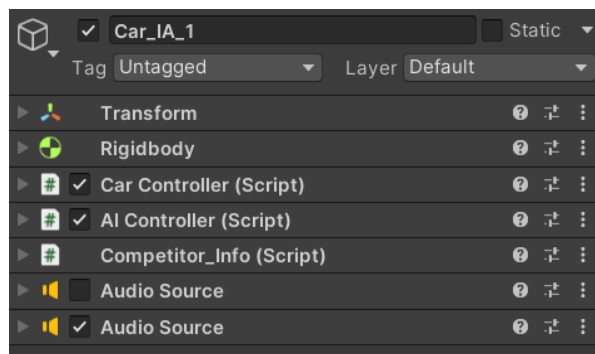
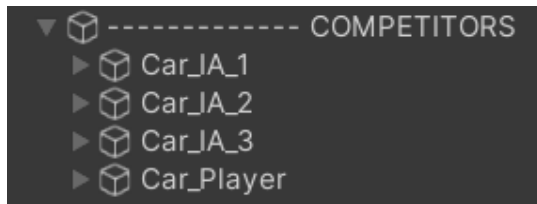
Las ruedas cuentan con colliders propios y un sistema de unity propio mediante un componente atribuible mediante el inspector llamado “Wheel Collider”, el cual permite controlar físicas adaptadas a ruedas de coches.



- **WheelController:** Maneja la posición y rotación de la rueda.

Controlador de IA

Componentes:



Son como el coche normal del jugador pero sin el player controller y con un AIController aparte.

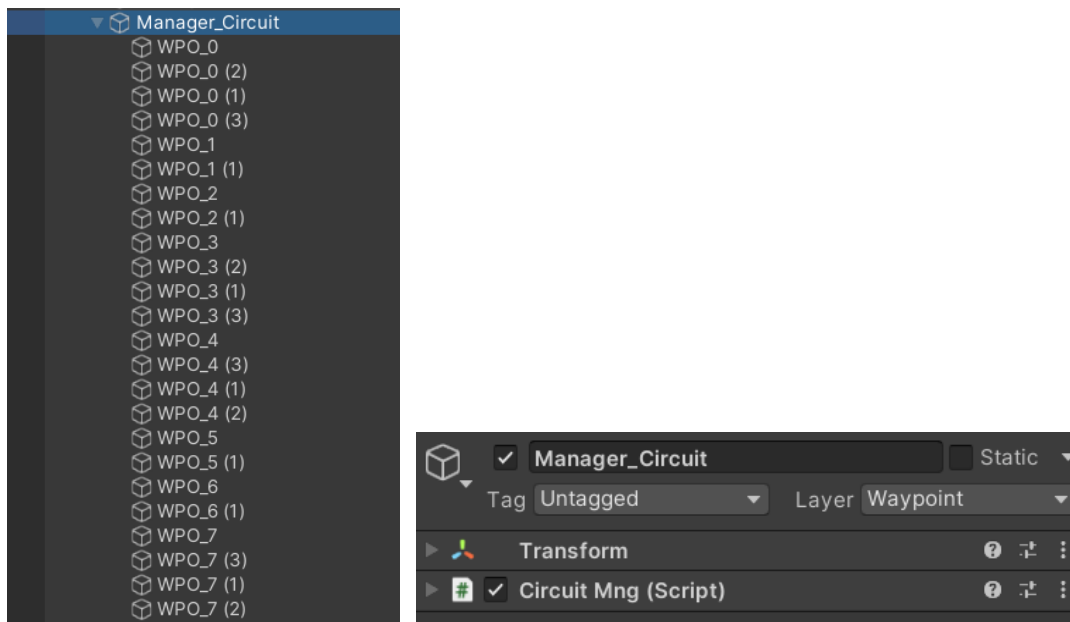
- **CarController:** Gestiona la lógica del coche, waypoints, vueltas y carrera en general del coche.
- **AIController:** Gestión de movimiento por el mapa entre waypoints y sistema de esquite de obstáculos mediante varios raycast en el frente del coche y en varias direcciones diferentes.
- **CompetitorInfo:** Tiene la información del jugador de posición, waypoints, nombre...

La IA persigue puntos establecidos por el mapa, y cuando llega a 1 pasa al siguiente así hasta llegar al último y reiniciarse.

Se desplaza con fuerzas mediante el script "CarController" como el jugador.

Gestión del circuito y carreras

Gestionador del circuito = Script "CircuitMng".



Para la gestión de la carrera, contiene todos los waypoints como hijos y los recoge en una lista “waypoints” nada más empieza la carrera.

- **CircuitMng:**
 - Contiene todas las listas principales de la partida:
 - **Jugadores**
 - **Waypoints**
 - **Spawnpoints**
 - Además gestiona el ranking, la lista de competidores, añadir jugadores online y cuando termina la partida.

Tiene una lista de los gameobjects de los competidores “competing” a la cual se añaden las entidades que se encuentran en la carrera

Se hace con la intención de gestionar las posiciones y sistema de cálculo de quien va por delante del resto gestionando las variables de gestión de carrera.

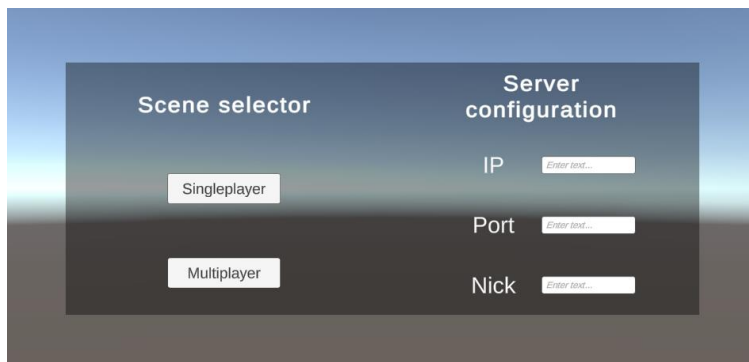
Cubos invisibles que cuentan como líneas de llegada a cada punto del circuito para que los coches de la IA tengan que pasar por ellos fijándolos como objetivo y comprobar si está por delante de otro coche

También sirve para que el jugador deba pasar por ellos y también contabilizar por qué zona del circuito va.

Sabiendo la zona en la que va cada uno, un manager de escena llamado “CircuitMng” comprueba quien va primero, cuantas vueltas llevan y quien va ganando.

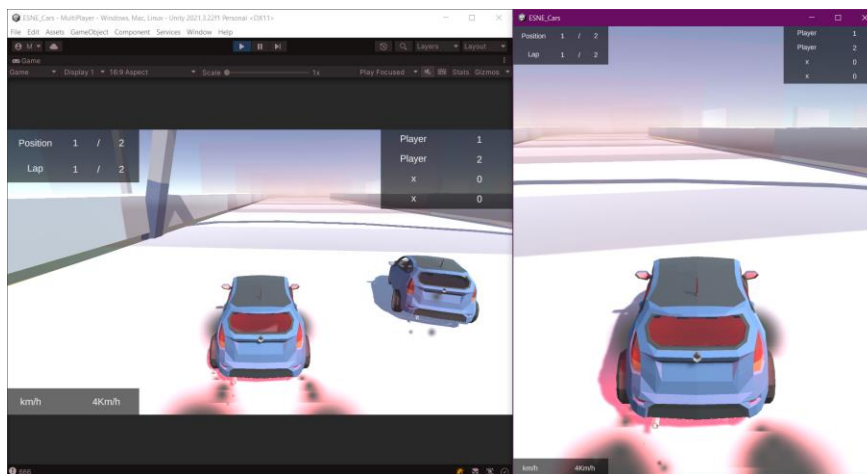
Lógica de juego en red en el proyecto de Unity

Se ejecuta el servidor externamente, los jugadores pueden desde el menú decidir la localización a la que se conectan, si se conectan a la misma aparecen los coches conectados y todos los movimientos y datos de carrera se guardan en el servidor, quien es el proyecto de Unity.



Ya ahí, el primero se une y aparece en la escena.

Ahora solo hace falta ejecutar desde la build otro juego y hacer lo mismo con la misma configuración de servidor y puerto y se unirá y añadirá a la lógica del controlador de la carrera “CircuitMng” y la carrera podrá empezar a tener lugar.



Implementación

Complicaciones

Sistema de vueltas

Para instanciar, gestionar y replicar las estadísticas teniendo en cuenta a todos los jugadores en todo momento fue un problema en una primera instancia del proyecto para el cual tras darle muchas vueltas al inicio, planteé que la mejor solución era utilizar una lista en el manager de circuito y utilizar aparte una gestión con parents teniendo al gameObject "----- COMPETITORS" como padre y a los coches que se fuesen a unir o instanciar en escena como todas las entidades competidoras.

Ahorré muchos problemas en algunos aspectos, pero creó otros al tener esos requerimientos base de funcionamiento como al instanciar los coches online que dio problemas para meterlos como hijos en un inicio.

Sistema de evasión de obstáculos

Logré uno que salvase ante las situaciones menos complejas de un obstáculo delante, pero si detecta una pared no hice que fuese capaz de dar marcha atrás ni nada por el estilo.

Con lo cual, la IA es bastante limitada y solo cumple los mínimos.

Aspectos de mayor interés

La parte del online era amplia e interesante, y me percaté cuando llegué al final de las mejoras que podría haber hecho de base con scriptable objects para la información de cada entidad por separado para ahorrar tiempo y complicaciones a la larga.

Aparte, el sistema de detección de colisiones es más complejo de lo que en un inicio parecía y podría mejorarse con sistema de marcha atrás y una mejor detección de si se queda quieto por un obstáculo o no lo sortea bien. Fue muy interesante y mejorable con un mayor tiempo y estudio.

Conclusiones

Práctica que llevó más tiempo del esperado.

Poco agradable de hacer al tener que comprobar entre servidor y clientes rehaciendo builds.

Muy interesante de cara a empezar proyectos online en Unity como primera experiencia y seguramente utilice esta base para ampliarla a algo más profesional de portfolio con más tiempo y una gestión online más allá del sistema cerrado de red LAN que planteamos.