



Fantasy Racing

Simulador de carreras multijugador online

13/06/2022

Memoria Proyecto Final Desarrollo de Aplicaciones Multiplataforma

Antonio José de la Torre Pérez

2º DAM - 21/22

.....	0
1. Introducción.....	4
1.1 Motivación.....	4
1.2 Objetivos.....	5
1.3 Juegos de referencia.....	5
2. Tecnologías usadas. Estado del Arte	6
2.1 Motor de videojuegos	6
2.2 Lenguaje de programación.....	6
2.3 Software adicional	7
2.4 Herramienta de planificación	7
3. Planificación.....	8
3.1 Hitos.....	8
4. Presupuesto.....	9
4.1 Recursos humanos	9
4.2 Recursos materiales	9
4.3 Presupuesto total	10
5. Diseño	11
5.1 Plan de juego.....	11
5.2 Experiencia de juego	11
5.3 Coches y Habilidades.....	12
5.4 Mecánicas.....	13
5.5 Circuitos.....	14
5.6 Mecanismos	14
5.7 Enemigos	15
5.8 Multijugador	15
5.9 Interfaz de usuario	16
5.10 Guardado y carga de datos.....	18

6.	Implementación	19
6.1	Modelado 3D.....	19
6.2	Clase Coche y clases hijas	22
6.3	Habilidades, clase Prop y clases hijas	24
6.4	Control físicas.....	25
6.5	Partículas	25
6.6	Interfaz HUD.....	27
6.7	Control de checkpoints	29
6.8	Online	29
6.9	Spawns.....	30
6.10	Interfaz Menús y Gameflow.....	31
6.11	Gestión Usuarios	32
6.12	Leaderboard.....	34
6.13	Sonido.....	35
7.	Manual de usuario	36
7.1	Menús	36
7.2	Pantalla de juego	36
7.3	Controles	36
7.4	Consejos	37
8.	Recursos utilizados	38
8.1	Modelos 3D	38
8.2	Audio.....	39
9.	Conclusiones.....	40
9.1	Conocimientos aplicados	40
9.2	Dificultades encontradas	40
9.3	Resultados obtenidos.....	40
10.	Trabajo futuro.....	41



11.	Bibliografía	44
-----	--------------------	----

1. Introducción

El uso de las tecnologías de la información en la época actual está creciendo a pasos agigantados. Actualmente cualquier persona tiene acceso a un ordenador e incluso tienen un mini ordenador en su bolsillo, su smartphone.

A su vez la industria de los videojuegos no deja de evolucionar, con el paso del tiempo, el número de jugadores aumenta de manera notable. En los últimos cinco años este incremento de jugadores se ha visto disparado. El valor de la industria mundial de los videojuegos supera los 250 mil millones de euros.

Actualmente los videojuegos son claramente una de las principales formas de entretenimiento entre el público joven. Cabe destacar las competiciones profesionales de videojuegos, conocidas como e-Sports, seguidas mundialmente por millones de usuarios.

La evolución de los videojuegos ha sido rápida, muy ligada a los avances tecnológicos del momento. El lanzamiento de nuevos componentes de ordenador o consolas de nueva generación viene de la mano con lanzamientos de nuevos videojuegos con cambios en gráficos importantes. Podemos ver notables diferencias entre los videojuegos de hace tan solo 5 o 10 años y los actuales, desde la implementación de Ray Tracing¹ hasta herramientas de creación de personajes ultra realistas²

1.1 Motivación

Desde pequeño los videojuegos han sido mi principal fuente de entretenimiento. El mundo de los videojuegos me interesa hasta el punto de estar al tanto de las noticias de los últimos lanzamientos y seguir los avances del desarrollo de nuevas entregas.

Mi proyecto se relaciona con este campo debido a la necesidad de comprobar si sería capaz de conseguir entretener e impresionar a la gente con algo creado por mí.

En este proyecto he querido plasmar los conocimientos adquiridos en programación durante estos dos últimos años, así como mi creatividad personal. Los retos son algo que me motiva y me mantiene activo, considero que este proyecto se presentaba como uno de los más grandes.

¹ **Ray Tracing:** Técnica de renderizado basada en el trazado de una gran cantidad de rayos desde distintas fuentes de luz, calculando y analizando los rebotes de cada rayo sobre los distintos objetos.

² **[MetaHuman Creator](#):** Aplicación basada en la nube que permite a cualquier persona crear seres humanos digitales fotorrealistas

1.2 Objetivos

El objetivo principal de este proyecto es desarrollar un videojuego desde cero. La idea es crear una primera versión totalmente jugable de un juego de carreras online. Por una parte, implementaremos las características de un juego de carreras y por otro lado le sumaremos lo más icónico de los juegos MOBA³, como las habilidades disponibles para los jugadores.

Este videojuego será desarrollado para PC. El motor de videojuegos nos permitirá exportar este juego para plataformas como Windows, Mac OS, WEB GL o consolas. Para esta primera versión del juego se exportará para Windows.

1.3 Juegos de referencia

En la industria de los videojuegos existe una amplia variedad de juegos de conducción, desde juegos arcade hasta juegos de conducción realista. En mi opinión los mejores son aquellos que juegas en familia de manera casual y crean momentos de diversión.

Entre los juegos que me han inspirado encontramos:

- Mario Kart 8 Deluxe
- Crash Tag Team Racing
- La saga Need For Speed
- Hot Wheels Unleashed
- La saga Forza



³ MOBA: Género de videojuegos multijugador de arena de batalla en línea

2. Tecnologías usadas. Estado del Arte

2.1 Motor de videojuegos

Para este proyecto me decanté por utilizar el motor de videojuegos Unity. Entre las alternativas disponibles encontramos Unreal Engine, CryEngine o Amazon Lumberyard.

Entre otras razones elegí Unity por:

- La gran cantidad de información, documentación, tutoriales y componentes que la comunidad mantiene en internet.
- Su curva de aprendizaje. Para un proyecto de estas dimensiones su complejidad es la adecuada, pudiendo hacer cosas interesantes con conocimientos de principiante.
- Su lenguaje de programación.
- El hecho de haberlo usado anteriormente, tanto en clase como en antiguos proyectos personales.



2.2 Lenguaje de programación

En Unity se puede crear un videojuego casi totalmente desde la interfaz visual, pero sin un script de programación no podríamos hacer que el juego tuviera vida. Para crear la lógica del juego es necesario utilizar Scripts.

Para crear estos scripts utilicé C# ya que, pese a tener otras opciones, ya lo conocía con anterioridad y es el principal lenguaje soportado. La mayoría de información de scripting de Unity en internet es sobre C#.

C# es un lenguaje de programación orientado a objetos desarrollado por Microsoft como parte de su plataforma .NET. Su sintaxis básica deriva de C/C++ y utiliza un modelo de objetos similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

2.3 Software adicional

Photon 2

PUN (Photon Unity Networking) es un paquete de Unity para videojuegos multijugador. Los jugadores se conectan a un servidor central y se reúnen en salas dónde los objetos pueden ser sincronizados a través de la red.

Playfab

Playfab es un potente y reciente servicio online de Microsoft que nos permite gestionar los datos de los usuarios registrados. Utilicé esta forma de almacenamiento de datos por su sinergia con los videojuegos. Este servicio se asemeja a Firebase de Google, pero se especializa en los videojuegos.

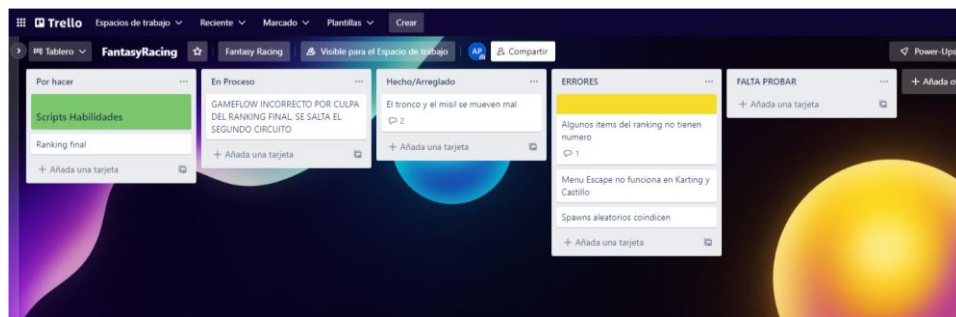
2.4 Herramienta de planificación

Para organizar el proyecto he utilizado la herramienta Trello.

Trello es un organizador de tareas tanto personales como grupales. Gracias a su gran variedad de opciones es posible manejar la información fácilmente. La aplicación se basa en el método Kanban, organiza la información por tableros, listas y tarjetas personalizables.

En mi caso dividí el proyecto en tareas pequeñas que asigné a tarjetas individuales. Creé 5 listas denominadas “Por hacer”, “En proceso”, “Hecho”, “Errores encontrados” y “Errores arreglados”.

Gracias a esta herramienta he podido organizar el proyecto fácilmente, sabiendo en todo momento las tareas que aún tenía que hacer, qué fue lo último que estaba haciendo o apuntando errores cuando surgían.



3. Planificación

Inicialmente dediqué la primera semana a identificar los objetivos principales del proyecto, seguidamente dividí el proyecto en hitos u objetivos a cumplir semanalmente para tener las distintas partes del proyecto totalmente identificadas y avanzar de manera que las partes iniciales fueran la base del juego y las partes posteriores fuesen mejoras sobre esa base.

3.1 Hitos

- Semana 1: Game Design Document
El Documento de Diseño del Videojuego debe de ser lo primero para establecer los objetivos y límites del proyecto, en ese documento se recoge la primera idea de lo que será nuestro juego.
- Semana 2: Mapas
Utilizando el editor visual de Unity modelé cada uno de los mapas utilizando formas simples y modelos 3D descargados de internet.
- Semana 3: Movimiento Básico del coche
Este primer script de movimiento nos permitía acelerar, frenar, andar marcha atrás y girar.
- Semana 4: HUD⁴ y Puntos de control
Se diseñó el aspecto visual de lo que el jugador vería durante la partida, así como el sistema de puntos de control a lo largo del circuito.
- Semana 5: Online
Implementé la posibilidad de ver a otros jugadores en el mismo mapa.
- Semana 6: Menús y Gameflow⁵
Se crearon las ventanas de inicio del juego y selección de sala y se comprobó que los jugadores acababan la carrera para pasar al siguiente mapa.
- Semana 7: Habilidades, gestión de usuarios
Se implementaron las habilidades de cada uno de los coches y el acceso de los usuarios utilizando e-mail y contraseña.
- Semana 8: Guardado de datos y sonido
Añadir sonido a las habilidades, impactos de objetos y puntos de control. También se almacenarán los mejores tiempos de los circuitos en nuestra base de datos.
- Semana 9: Desarrollo de esta memoria

⁴ **Head-Up Display:** Información que se muestra en pantalla durante la partida.

⁵ **Gameflow:** Organización y cambio de una pantalla a otra durante el juego.

4. Presupuesto

4.1 Recursos humanos

Para poder estimar los costes del trabajo personal he supuesto una jornada de 8 horas diarias 5 días por semana, obteniendo así los siguientes resultados:

Tarea	Horas dedicadas	Precio por hora	Coste
Planificación	32	6,00€	192,00€
Diseño	15	8,00€	120,00€
Modelado 3D	45	8,00€	360,00€
Implementación	153	8,00€	1.224,00€
Sonido	12	6,00€	72,00€
Pruebas	62	6,00€	372,00€
Documentación	41	6,00€	246,00€
	360		2.586,00€

4.2 Recursos materiales

- Unity: El motor de videojuegos dispone de una versión para uso personal. Esta versión es gratuita si nosotros o nuestra compañía ha facturado menos de cien mil dólares durante los últimos 12 meses. Al iniciar el juego aparecerá el logo de Unity por usar esta versión gratuita.
- Modelos 3D: He utilizado los modelos 3D gratuitos que nos ofrece la página web [Kenney](#). Estos assets⁶ son completamente gratuitos tanto para uso personal como comercial.
- Audio: Los audios utilizados para este juego han sido descargados de la página web [Kenney](#). Son completamente gratuitos para uso personal y comercial.
- Música: La música del juego proviene de la web del autor [Patrick de Arteaga](#). Esta música es gratuita y puede utilizarse en proyectos comerciales o personales, el único requisito que pide el autor es mencionar su nombre o la web en los créditos del proyecto.
- Photon 2: El framework de conexión en tiempo real para juegos multijugador tiene un plan básico para desarrollo y evaluación que resulta suficiente para esta primera versión del juego. Permite la conexión de hasta 20 jugadores concurrentes.

⁶ **Assets:** Recursos multimedia que utiliza el juego

- Playfab: El servicio online de almacenamiento de datos de usuarios tiene un plan inicial gratuito disponible para juegos en desarrollo. Nos permite almacenar hasta 10 juegos diferentes con un volumen de cien mil jugadores por título.
- Ordenador: Para desarrollar este proyecto se ha utilizado un ordenador sobremesa compuesto por un procesador i5 8600K y una tarjeta gráfica GTX 1060 6GB valorado en 1200€. Se estima un coste de amortización de 150€.

4.3 Presupuesto total

Gracias al uso de recursos con licencias de uso gratuitas el coste total del proyecto se limita a los recursos humanos y la amortización del ordenador utilizado.

Recursos	Coste
Humanos	2586€
Materiales	150€
	2736€

5. Diseño

Fantasy Racing se trata de un juego de carreras en el que cada vehículo tiene diferentes habilidades ligadas a su propia naturaleza. El único objetivo es ser el primero en llegar a la meta, pero tus contrincantes te complicarán la tarea usando sus habilidades contra ti.

5.1 Plan de juego

Una vez iniciado el juego nos encontraremos con un menú de inicio de sesión. Tras iniciar sesión veremos un menú donde buscar la sala a la que conectarnos o consultar el ranking de puntuaciones. Cuando entremos a una sala podremos ver hasta un total de 6 jugadores y podremos elegir nuestro coche. Elegiremos entre los 6 vehículos inicialmente disponibles utilizando los botones situados sobre nuestro avatar. Una vez que todos los jugadores hayan elegido su vehículo comenzará la partida.

Una partida estará compuesta por tres carreras. Cada carrera comenzará con una cuenta atrás y finalizará cuando el último jugador cruce la meta. Una vez finalizada la carrera se asignará una puntuación a cada jugador en función de su posición final en la carrera. Una vez completadas las tres carreras se procederá a la entrega de premios dónde se podrá ver la clasificación final, los jugadores quedarán ordenados por la suma de las puntuaciones en cada una de las carreras.

5.2 Experiencia de juego

La música podrá variar según el escenario en el que nos encontremos, desde música alegre y divertida en un circuito de carreras estándar hasta música frenética cuando nos encontremos en la carrera final.

El juego estará repleto de efectos de sonido. Habrá efectos de sonido para las habilidades, choques, aumentos de velocidad, ralentizaciones, paralizaciones, etc.

La cámara perseguirá al coche en tercera persona, es decir una vista superior desde la parte trasera del coche como si la cámara flotara en el aire. El jugador podrá cambiar la cámara puntualmente para ver lo que hay detrás de él, como si se tratase de un espejo retrovisor.

5.3 Coches y Habilidades

Los vehículos dispondrán de cuatro habilidades para obtener ventajas las cuales bien utilizadas supondrán una mejor posición en la carrera.

Estas habilidades podrán ser:

- Beneficios para el propio jugador (Velocidad, Escudo...)
- Perjuicios para un solo enemigo (Ralentización, Aturdir...)
- Perjuicios para todos los enemigos

Habilidades de cada vehículo:

- Ambulancia
 1. Usa las sirenas y recibe un impulso de velocidad
 2. Deja en el suelo un charco de sangre que resbala
 3. Regenera su vida un 30%
 4. Electrocuta al resto de jugadores y los deja inmovilizados
- Coche de Policía
 1. Usa las sirenas y recibe un impulso de velocidad
 2. Lanza un misil en línea recta que al impactar produce daño
 3. Deja en el suelo una cadena de pinchos, si otro vehículo la pisa quedará ralentizado
 4. Lanza una ráfaga de veinte misiles que al impactar producen daño
- Fórmula 1
 1. Limpia los efectos negativos sobre él
 2. Deja un rastro de fuego que al ser pisado produce daño
 3. Aumenta su velocidad máxima para el resto de la carrera
 4. Recibe un impulso de velocidad alto
- Camión de Basura
 1. Deja caer basura al suelo que al ser pisada ralentiza
 2. Activa un escudo que le protegerá de la siguiente habilidad enemiga
 3. Deja en aire una nube de gases tóxicos que impiden ver al enemigo
 4. Lanza una bomba que al explotar lanza por los aires a los enemigos
- Camión de Bomberos
 1. Usa las sirenas y recibe un impulso de velocidad
 2. Deja un rastro de agua que al ser pisado hace que el coche resbale

3. Lanza una bomba de agua hacia delante que al impactar con el suelo deja un charco de agua grande que resbala
 4. Se vuelve maligno y crea un vórtice de fuego a su alrededor que quema a los enemigos
- Pickup
 1. Lanza un tronco en línea recta que si impacta con el enemigo lo deja inmóvil
 2. Deja caer una caja de herramientas que al impactar con ella te deja inmóvil
 3. Catapulta una bomba que al impactar con el suelo incendia una zona y al pasar por ella los enemigos reciben daño
 4. Activa el modo 4x4, lo que le permite circular por cualquier terreno y que no le afecten los efectos negativos

5.4 Mecánicas

Durante la carrera los jugadores podrán aprovechar las siguientes mecánicas para ser el mejor.

Estadísticas

Para hacer más equilibrado el juego, y no sea frustrante que un mismo vehículo gane siempre todas las carreras, se dispondrá de un sistema de estadísticas por vehículo. Estas estadísticas vendrán dadas a razón de la naturaleza del modelo del vehículo.

Ejemplo:

- Fórmula 1: 50 Vida, 200 Velocidad Máxima, 100 Aceleración
- Camión de Basura: 200 Vida, 100 Velocidad Máxima, 50 Aceleración

Derrape

Los jugadores podrán obtener una bonificación de velocidad de 3 niveles diferentes en función de la duración de su derrape en curvas. Será fácil identificar el nivel de aumento de velocidad gracias al color de las partículas que desprende el vehículo.

5.5 Circuitos

El juego tendrá un total de tres circuitos diferentes:

Circuito Óvalo

El primer circuito se tratará del clásico circuito oval de la Nascar. En él encontraremos rampas, plataformas de aumento de velocidad y decoración temática del mundo de las carreras.

Circuito Karting

El segundo circuito estará basado en los circuitos de Karting Indoor comunes que resultan ser una atracción familiar segura en la que personas de cualquier edad pueden participar. Estará lleno de curvas y la decoración está compuesta de vallas, neones y una iluminación tenue.

Circuito Castillo

El tercer circuito será un camino por un prado que se introduce en el interior de un castillo. La parte exterior estará decorada con hierba, árboles, etc. La parte interior del castillo será un trazado con largos pasillos zigzagueantes.

5.6 Mecanismos

Dentro del juego nos encontraremos con multitud de objetos de distintos tipos:

Potenciadores

Plataformas de Velocidad: Estos objetos serán plataformas situadas en el suelo que al ser pisadas nuestro coche obtendrá una mejora de velocidad de movimiento durante un corto periodo de tiempo. Estas plataformas serán representadas con flechas paralelas, suelen ser similares en la gran mayoría de juegos por lo que el usuario las podrá reconocer fácilmente.

Obstáculos

El asfalto será la zona de circulación habitual, cuando un vehículo se salga de la pista su velocidad de movimiento se verá reducida drásticamente.

Según la naturaleza del circuito es posible que nos encontremos con paredes verticales que al chocar con ellas reduzcan la velocidad del vehículo.

En el caso de que el circuito no tenga borde por alguno de sus lados y el vehículo se caiga este se verá penalizado cuando tenga que reaparecer una distancia por detrás de donde se cayó.

Las habilidades de los vehículos pueden producir obstáculos: basura, cadenas de pinchos, suelo resbaladizo etc.

Puntos de control

Estos objetos serán arcos situados por encima del asfalto que al ser atravesados por el jugador se guardará el último punto de control por el que se pasó. Esto nos permitirá establecer puntos de reaparición y nos ayudará a controlar las posiciones de los jugadores dentro de la carrera.

Los puntos de control otorgarán al usuario una sensación de progresión y les indicarán que están siguiendo el camino correcto dentro del circuito.

Será necesario atravesar todos los puntos de control en cada vuelta para poder completarla.

5.7 Enemigos

En este juego no existe un enemigo ideado como tal con el objetivo de acabar contigo, sin embargo, se presentan dos agentes que dificultan tu partida.

El primero de ellos es el propio circuito en sí. El trazado de la calzada, las rampas, los obstáculos y el hecho de tener que atravesar todos los checkpoints no será tarea fácil.

En segundo lugar, tendremos al resto de jugadores. Estos serán los enemigos principales de este juego, ellos serán los que te hagan la vida imposible con sus habilidades. Los objetos por el suelo ralentizan tu vehículo, te dejarán sin poder moverte o incluso sin poder ver, todo esto sumado a la complejidad del propio circuito será la mezcla suficiente para complicarte la partida.

5.8 Multijugador

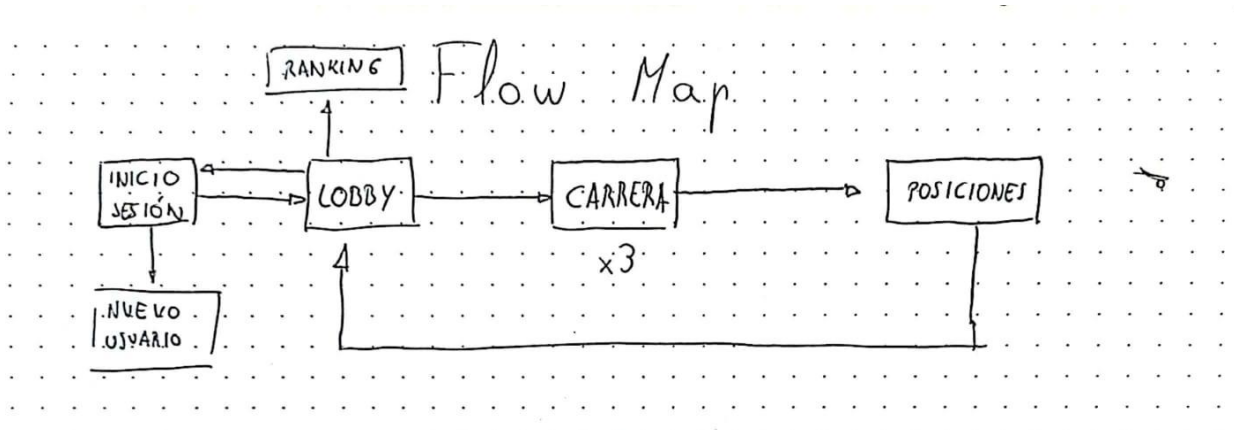
Este juego será jugable tanto de manera individual como en modo online

- Individual: El jugador podrá jugar sólo intentando batir el récord de mejor tiempo del circuito.
- Online: En este modo de juego se enfrentará un total de hasta 6 jugadores conectados a un mismo servidor dirigidos por el dueño de la sala.

5.9 Interfaz de usuario

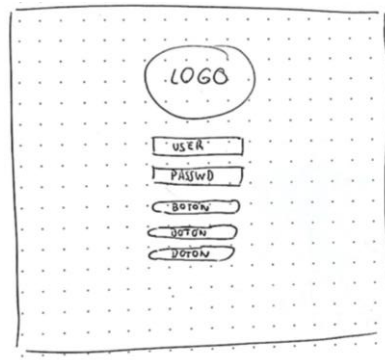
Respecto al tema de la Interfaz de usuario tendremos que hablar de dos grandes grupos de interfaces:

Menús



Como hemos comentado anteriormente, el usuario tendrá que navegar por una serie de menús antes de poder comenzar la partida. Concretamente tendremos 5 ventanas de menús:

- Inicio de sesión: En esta pantalla podremos entrar al juego introduciendo un email y contraseña registrados.



- Nuevo usuario: En caso de no tener un usuario, desde la pantalla de inicio de sesión podremos acceder a la ventana de nuevo usuario. En esta ventana se controlarán todos los datos introducidos por el usuario y si todos son correctos se creará un nuevo usuario.

Hand-drawn sketch of a user registration form titled "Usuario". It contains four input fields labeled "NOMBRE", "EMAIL", "PASSWD", and "PASSWD", followed by an oval button labeled "CREAR".

- Selección de sala: Si el usuario ha iniciado sesión correctamente podrá ver esta ventana donde crear una nueva sala o incorporarse a una ya existente.

Hand-drawn sketch of a room selection screen. It has two main sections. The left section has a "NOMBRE" label and a "BOTON" button. The right section is a box containing three labels: "SALA 1", "SALA 2", and "SALA 3".

- Selección de vehículo: Cuando nos incorporamos a una sala veremos al resto de jugadores que la componen, una vez que el líder de la sala compruebe que todos los jugadores están listos podrá dar comienzo a la partida.

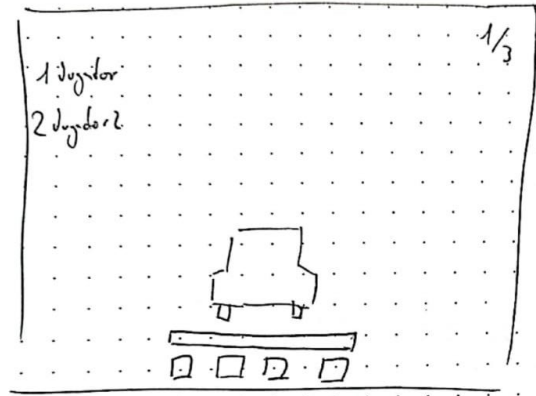
Hand-drawn sketch of a room screen titled "SALA". It displays six car icons arranged in two rows of three, representing players in the room.

- Ranking: En esta ventana podremos ver los mejores tiempos de cada circuito.

Hand-drawn sketch of a ranking screen titled "TOP". It shows two entries: "Jugador 01:20" and "Jugador 02:20".

HUD

Durante la partida el usuario tendrá información visible en la pantalla. Se podrá visualizar la vuelta por la que estamos, nuestra cantidad de vida, las habilidades y su tiempo de reutilización y finalmente el ranking de posiciones de los jugadores. Todo esto estará organizado de manera que no interfiera con la visión del jugador y se visualice toda la información de manera clara.



5.10 Guardado y carga de datos

En este proyecto se ha decidido controlar los usuarios registrados. El motivo de ello es tener la capacidad de tener información y estadísticas de los jugadores, así como un control de los accesos de los usuarios.

A su vez se mantendrá un ranking con los mejores tiempos de cada circuito. Esta estadística permitirá saber si el desarrollo de los circuitos es el correcto y si los jugadores están haciendo trampas o no.

6. Implementación

6.1 Modelado 3D

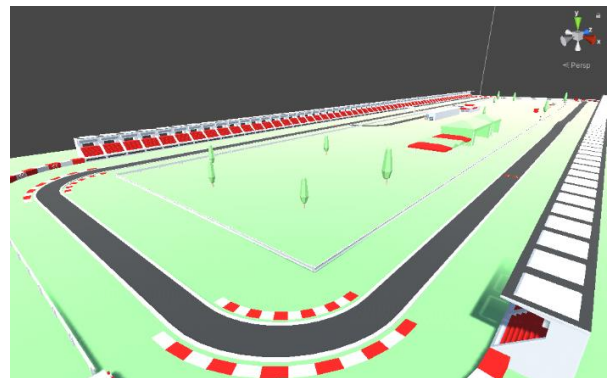
Como se comentó en el apartado de planificación, durante la primera semana de proyecto el reto era crear los mapas o circuitos que iban a componer el juego.

Desde el editor visual de Unity, utilizando polígonos básicos y modelos 3D se ha logrado crear los tres mapas, decoración del escenario y objetos propios de las habilidades.

Para decorar los tres circuitos he utilizado los modelos de las carreteras, curvas, pianos, línea de salida y posiciones de salida.

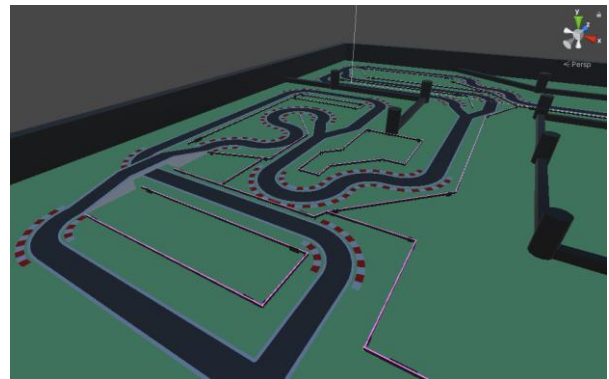
Circuito Óvalo

Para decorar todo se han incorporado vallas, gradas, árboles, carpas de equipos, una zona de boxes, así como cámaras y modelos de coches estáticos que no interfieren en la partida.



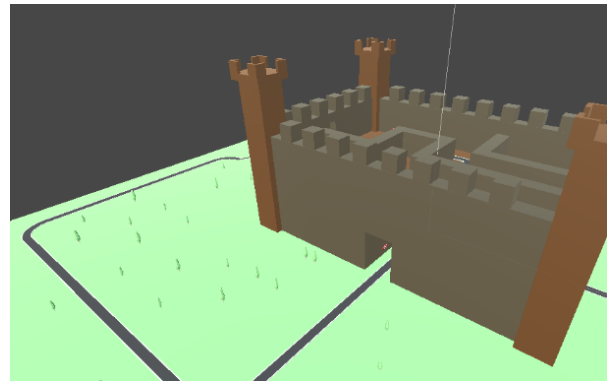
Circuito Karting

En este segundo circuito podemos apreciar un claro cambio en la iluminación. He utilizado modelos de puentes y bifurcaciones. Para decorar he modificado el modelo de la valla y le he incorporado dos cilindros que simulan ser una luz de neón gracias al material que utilizan.



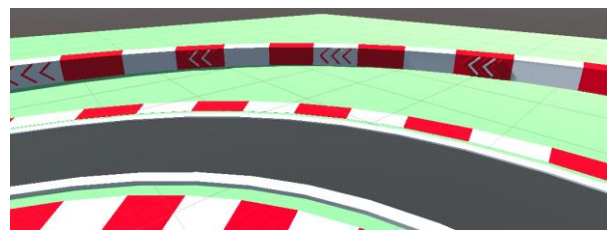
Circuito Castillo

Para el último circuito he creado un castillo utilizando formas geométricas básicas, por el interior del castillo he incorporado muros altos simulando los pasillos de un castillo. He utilizado los modelos de flechas de dirección y árboles para decorar.



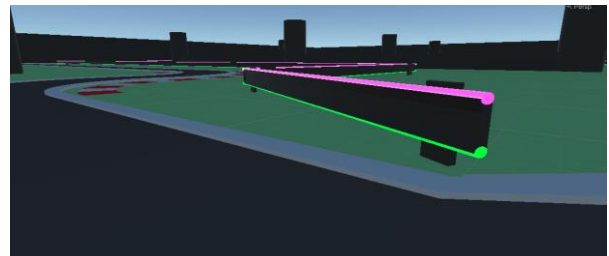
Flechas dirección

El paquete de modelos que utilizo no incorpora algo para indicar el sentido del circuito así que he creado yo mismo estas flechas con polígonos.



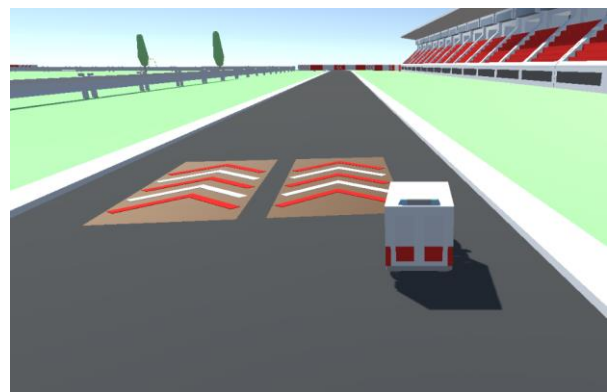
Barreras neón

Para darle un toque retro he incorporado estos cilindros con materiales que imitan al neón al modelo de la valla.



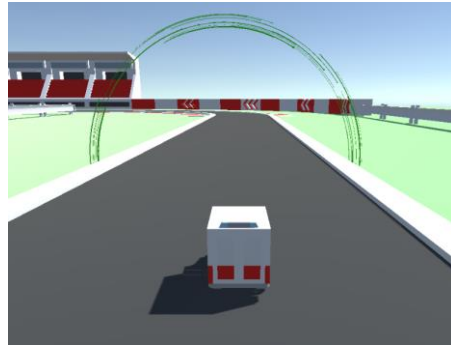
Plataformas potenciadoras

Estas plataformas cogen el modelo utilizado para las flechas de dirección e incorporan una base dando la sensación de ser una baldosa.



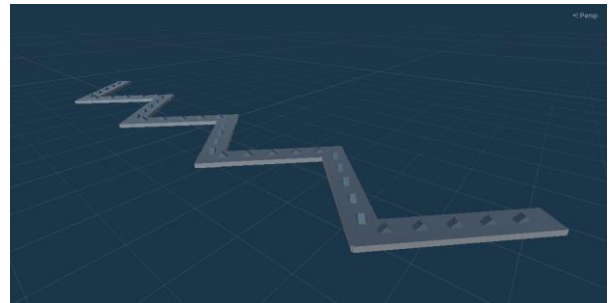
Puntos de control

Los puntos de control son objetos que contienen un sistema de partículas. Posteriormente se profundizará en este tema.



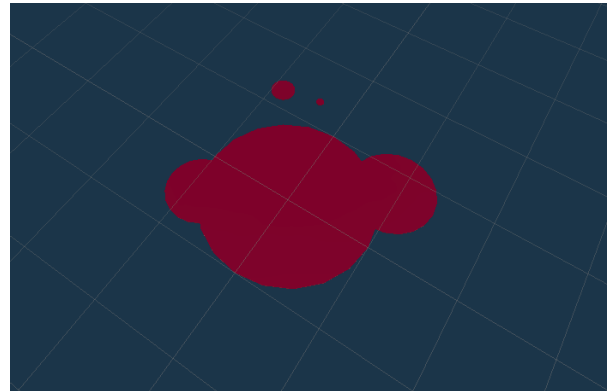
Cadena de pinchos

Para crear este modelo he utilizado rectángulos que transmiten la sensación de estar unidos. Se asemeja a la idea de una cadena de pinchos policial con una forma muy básica.



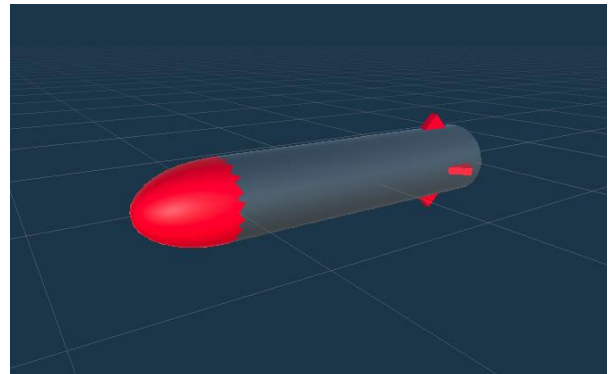
Charco

Los charcos tienen diferentes variantes, pueden ser de agua o sangre y están formados por cilindros de poca altura.



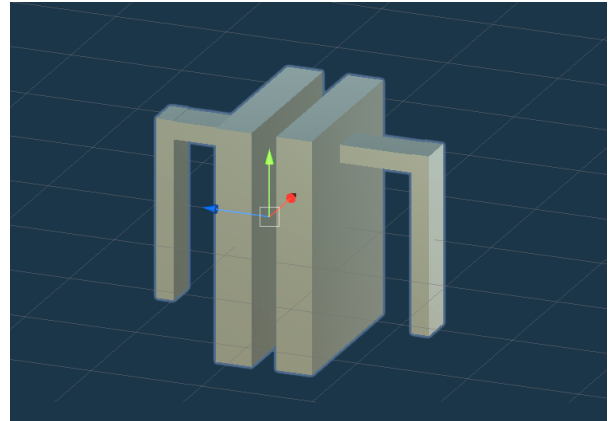
Misil

Este modelo está creado a partir de un cilindro y una esfera achatada. Su forma simple identifica a la perfección la idea de disparo.



Palas desfibriladoras

En este modelo he utilizado cubos de distintas dimensiones. Esta forma simple unida a una animación y sonidos dará la sensación de ser un desfibrilador.



6.2 Clase Coche y clases hijas

El control del jugador es uno de los aspectos más importantes del juego. En este caso el jugador es un coche y a pesar de ser diferentes, toda la parte básica de los coches funcionan igual por lo que todos partirán de un mismo objeto base.

El objeto Jugador se compone de:

- Modelo Vehículo: Cada coche tiene un modelo 3D diferente dependiendo de si es una Ambulancia, un coche de Policía, etc.
- Posiciones importantes: Para tener referencias durante la partida existen puntos que se almacenan en el objeto.
- Esfera: Verdaderamente, si no viéramos el modelo del coche, el jugador manejaría una esfera. Para un juego de este calibre manejar una esfera es suficientemente satisfactorio.
- Canvas: Contiene la barra de vida, las vueltas, las habilidades y una pantalla cegadora.
- Partículas: En la posición de las ruedas traseras se implementan los sistemas de partículas para dar el feedback del derrape.

Cada tipo de coche tendrá asignada una clase con su nombre, pero todas ellas heredan de una clase padre llamada Coche. Esta clase Coche es la encargada de:

- Avisar al controlador de checkpoints de que le vigile.
Al cargar la escena le pide al controlador de checkpoints que controle su paso por los diferentes puntos de control.
- Manejar su propio HUD.
Esto se debe a que al instanciar objetos de manera online el prefab del jugador debe tener incorporado todo lo que vaya a utilizar excepto las cámaras.
- Cargar todas las variables necesarias

- Capturar el input del jugador
 - Captura del movimiento básico
Para ello controlamos si el jugador pulsa las teclas WASD para incrementar la velocidad o rotar el vehículo.
 - Captura del derrape
Controlamos si el jugador pulsa la tecla espacio mientras gira.
 - Captura del uso de habilidades
Controlamos si el jugador pulsa las teclas "UIOP"
- Aplicar velocidad
Dependiendo del input recogido y de si el jugador está sobre la calzada se aplicará la velocidad necesaria.
- Gestionar tiempos de reutilización de habilidades
El tiempo de reutilización se debe reducir paulatinamente y al ser algo común en todos los vehículos se puede hacer desde esta clase
- Gestionar efectos de habilidades
Como todos los coches tienen que recibir los mismos efectos se codifica aquí efectos como ralentizar, inmovilización, ceguera...
- Gestionar sus datos en red
El propio coche será el encargado de actualizar sus datos en red, datos como su vuelta, su punto de control, la posición en la que ha finalizado, etc.

Para finalizar las clases que heredan de coche implementan los métodos de habilidades, esto es lo único que diferencia una clase que hereda de vehículo de otra.

Las habilidades, a grandes rasgos, se encargan de instanciar efectos u objetos. La manera más sencilla de provocar efectos en otros jugadores es que esos jugadores interactúen con objetos que se comparten a través de la red. De esta forma un vehículo puede soltar en la carretera un objeto en una posición y este objeto aparecerá en la pantalla del resto de jugadores provocando efectos al aparecer, al colisionar, o al desaparecer.

6.3 Habilidades, clase Prop y clases hijas

Todos los objetos que son lanzados por una habilidad deben de heredar de la clase Prop y estar etiquetados en la capa Prop también. Esta clase es muy básica ya que solo almacena un sonido que reproduce el objeto al aplicar su efecto, y gestiona la desaparición del propio objeto tras un periodo de tiempo.

Todos los objetos tienen un componente Collider que controla las colisiones, y sólo unos pocos tienen un componente Rigidbody para aplicarles fuerzas cómo un lanzamiento parabólico.

Casi todos los objetos funcionan de una de las siguientes maneras:

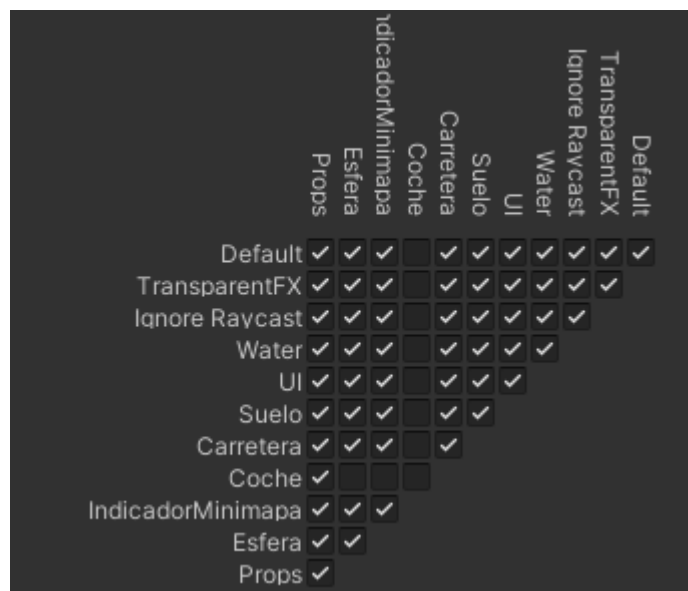
- Aplica efectos al aparecer
Existen algunos objetos que nada más ser lanzados empiezan a aplicar efectos en los jugadores. Un ejemplo de este grupo puede ser el desfibrilador que aplica una inmovilización al aparecer.
- Aplica efectos al desaparecer
Estos objetos, por lo general, invocan otros objetos cuando son destruidos, este es el caso de las bombas que sueltan agua, fuego o una explosión al chocar contra el suelo.
- Aplica efectos por colisión
Este último grupo es el más común. Los coches saben dónde están los objetos porque tienen una copia de ellos en su pantalla, en el caso de que alguien se choque contra ellos recibirán una penalización. En este grupo encontramos objetos como el charco de sangre que resbala o el misil que reduce la vida.

6.4 Control físicas

Unity se encarga del cálculo de físicas como la gravedad, los impulsos de fuerza o la aceleración. Para que un objeto pueda tener físicas es necesario que tenga un componente Rigidbody, y para que el objeto sepa si ha chocado necesita tener un collider. En caso de que un objeto pueda ser atravesado tendrá que tener un collider del modo trigger.

Los objetos que acabamos de comentar necesitan saber cuándo les ha impactado algún objeto, este objeto sólo puede ser un coche para que se produzca el efecto.

Para facilitarle la vida a Unity haremos que no se preocupe por comprobar algunas combinaciones físicas entre objetos de capas diferentes. Concretamente haremos que los coches solo puedan recibir colisiones de objetos que estén en la capa "Props". Esta opción la podemos modificar desde Project Settings en la pestaña Physics.



6.5 Partículas

Los sistemas de partículas son componentes que nos permiten crear efectos como gotas de agua cayendo, fuego o vórtices. Las partículas tienen una configuración compleja pudiendo llegar a tener velocidad variable, colores en función de la velocidad o incluso gravedad.

He utilizado las partículas en dos ocasiones para este proyecto:

- Puntos de control: Este sistema hace que las partículas roten sobre un eje dejando un rastro difuso tras ellas, esto provoca el efecto de portal mágico y etéreo que se deseaba.
- Derrape: El derrape puede tener tres niveles de fuerza, según el tiempo que dure el derrape la recompensa de impulso será variable.

Para implementar el feedback visual he creado dos sistemas de partículas para cada rueda trasera, el primero provoca el efecto de brillo en el suelo aumentando y disminuyendo el tamaño de unas partículas estáticas y el segundo lanza chispas que tienen gravedad.

En último lugar, para que el jugador sea consciente del beneficio adquirido se cambia el color de las partículas desde amarillo para el nivel 1, azul para el nivel 2 y magenta para el nivel 3.

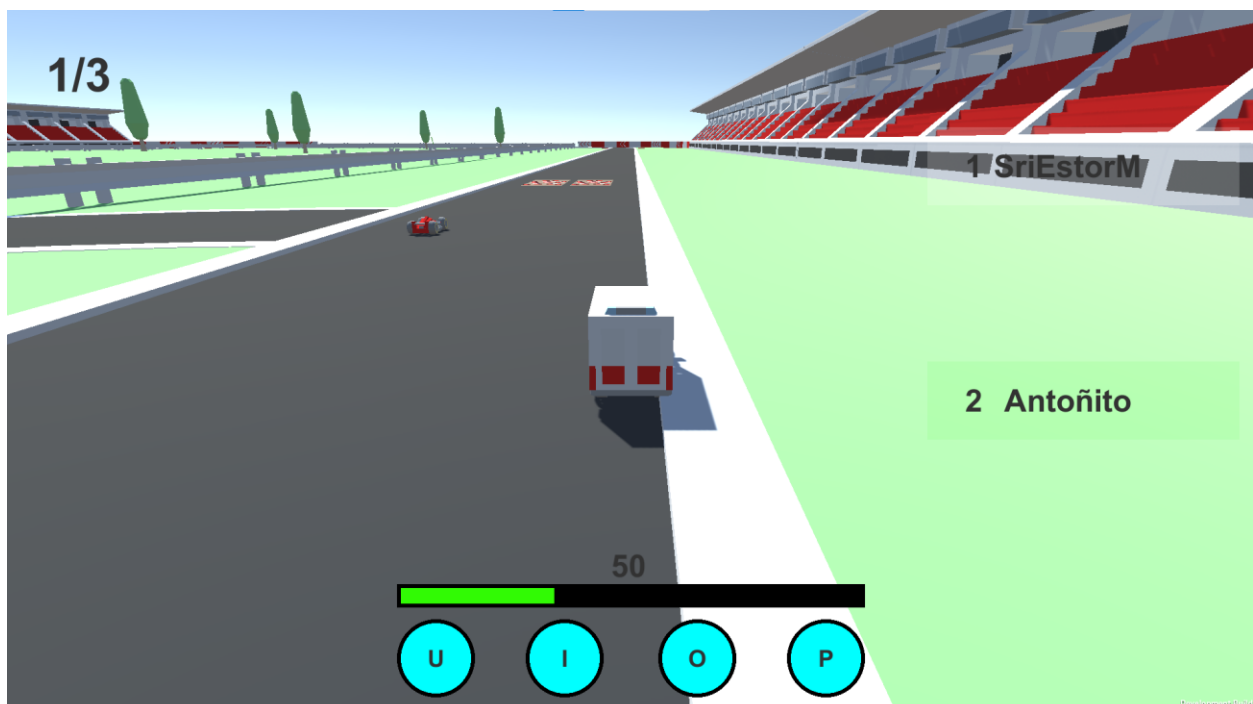


Las partículas pueden iniciarse al instanciar el objeto o bien pueden iniciarse desde un script, este es el caso de las partículas de derrape que se activan cuando se detecta que el jugador está derrapando. El color también se cambia desde el script.

6.6 Interfaz HUD



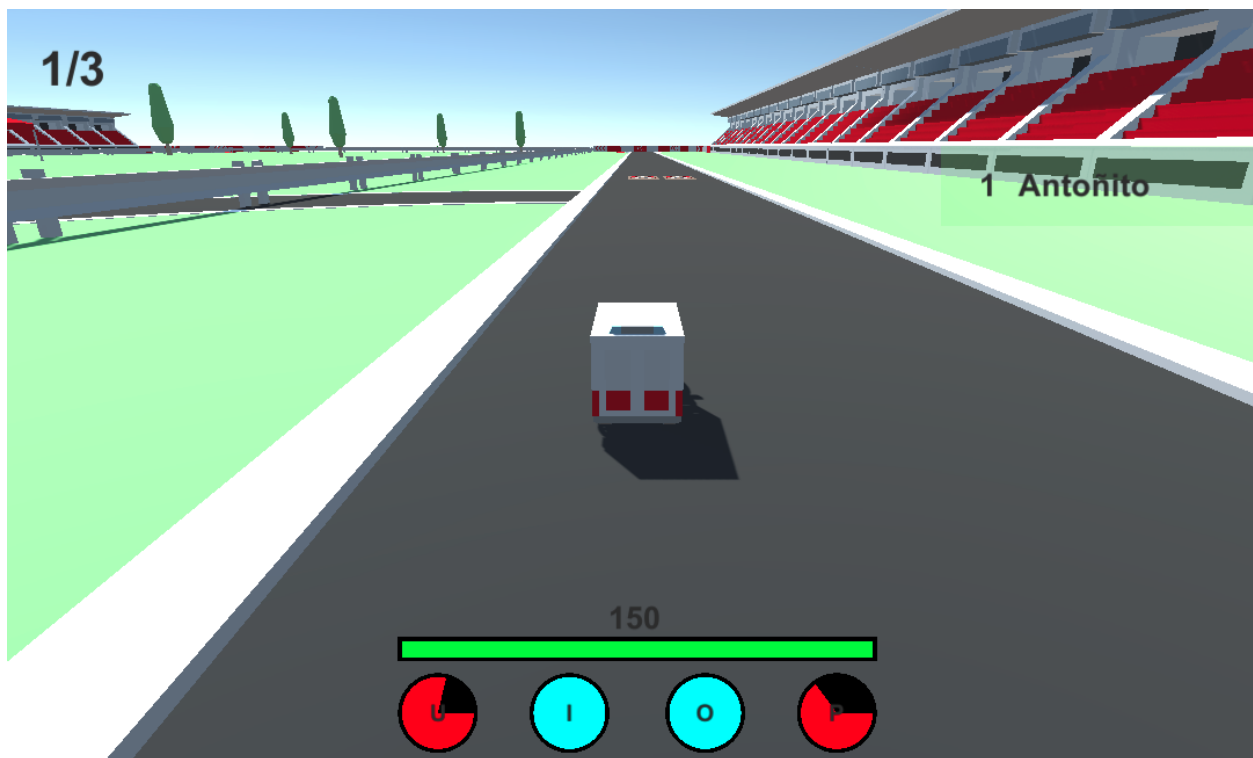
Al iniciar la partida en el centro de la pantalla podremos ver un contador que va decrementando desde 5, una vez que este contador acaba los jugadores podrán manejar el vehículo.



En la parte derecha de la pantalla podemos ver un ranking de posiciones de los jugadores ordenados. El elemento que representa al jugador que está mirando la pantalla está representado con un color de fondo de tono verdoso, el resto de jugadores tienen un fondo transparente. Este contraste de color está pensado para facilitar al jugador identificar su puesto.

Para mostrar los datos del ranking se ha creado un prefab de un ítem de la lista y se ha usado un panel con un layout vertical al que se van añadiendo instancias de este prefab. El listado se modifica cuando un usuario atraviesa un checkpoint o cada 2 segundos y los datos de los ítems se modifican según los datos de los usuarios de la sala.

Si miramos en la parte inferior vemos una barra de vida que se completa proporcionalmente a la vida del jugador. El texto encima de esta barra nos ayuda a representar con exactitud la cantidad de vida del jugador.



En esta última imagen podemos ver los contadores de las habilidades en la parte inferior. Cuando una habilidad está disponible para ser utilizada el círculo aparecerá completo y de color azul cian. En el caso de que la habilidad esté preparándose para poder ser lanzada este segmento de círculo aparecerá en color rojo. Los colores y el uso de fragmentos que se completan en relación al tiempo se han utilizado para mostrar al usuario un temporizador de una forma sencilla.

Por último, en la esquina superior izquierda nos encontramos un contador de vueltas. Este contador tiene un truco sencillo pero que ayuda bastante:

Si nosotros pusiéramos en todo momento la vuelta en la que está el jugador, esté empezaría en la vuelta 0 ya que las carreras suelen empezar por detrás de la línea de meta. Para que el jugador sepa que empieza dando la primera vuelta pondremos el texto inicialmente en 1/3 de manera que cuando pase por la meta el contador de vueltas se incremente al valor que ya está representado, pero en ningún momento veamos 0 vueltas.

6.7 Control de checkpoints

Este objeto se encarga de comprobar que los coches atraviesen sus puntos de control. Al instanciarse busca los puntos de control que hay en el escenario, los puntos de control individualmente son capaces de avisar a este objeto de qué coche ha pasado a través de él.

Cuando un coche pasa por un punto de control y el tracker recibe la información comprueba si el checkpoint era o no el correcto y emite un sonido como feedback al usuario, a continuación, carga cual es el siguiente punto de control que debe de atravesar el coche.

Para que el coche termine una vuelta completa debe de haber pasado por todos los puntos de control. Cuando el coche pasa por un punto de control correcto el tracker le avisa de que actualice sus datos para informar al resto de jugadores.

Este objeto está preparado para trabajar con distinta cantidad de puntos de control y se prepara para empezar desde el primer punto de control automáticamente cuando el usuario completa la vuelta.

6.8 Online

Para implementar la posibilidad de jugar en línea con otros jugadores he utilizado el componente Photon 2, concretamente su servicio PUN (Photon Unity Networking).

Para poder utilizar este componente tenemos que descargarlo desde la [web](#). Una vez descargado tendremos que importarlo y una vez completada la importación nos solicitará un “APPID” o identificador de nuestra aplicación. Sin embargo, esto no es suficiente para que funcione.

El siguiente paso fue incorporar los métodos de conexión del servidor a nuestro código cuando un jugador hacía login correctamente. Los métodos y callbacks de cambios de datos en el servidor nos permiten pintar en pantalla el listado de salas disponibles.

Cuando un jugador entra en una sala avisa al resto de jugadores para que lo tengan en cuenta en el siguiente pintado. Mediante las “Custom-properties”, que son propiedades de cada jugador que se almacenan en el servidor, el resto de jugadores son capaces de conocer datos sobre ti. Un uso que hago de las custom properties es decir cuál es mi Display Name al resto de jugadores, así como mi coche seleccionado, así ellos pueden pintar mi nombre y mi coche elegido.

Todos los jugadores han elegido coche, ahora esperaremos a que el líder de la sala decida pulsar el botón, ya que para sincronizar la partida adecuadamente el líder guiará al resto de jugadores.

Ya ha empezado la partida y para sincronizar los coches el objeto debe de instanciarse en cada una de las instancias del juego conectado a esa sala. Para que un objeto se pueda compartir en red necesitamos que el prefab se encuentre alojado en un directorio llamado Resources y que el objeto incorpore el componente Photon View y Photon Transform View Classic para compartir su posición, rotación y escala.

Una vez que el jugador completa las 3 vueltas le dice al resto de jugadores a través de las Custom Properties que ya ha acabado. Cuando el líder detecta que todos los jugadores han acabado se pasa al siguiente circuito.

Durante las 3 carreras sucede lo mismo, también se aprovechan estas custom properties para almacenar la posición de cada jugador, su número de vueltas, número de checkpoints, la distancia al siguiente punto de control, puntuación obtenida tras la carrera y la posición final en la carrera.

Cuando se completan las tres carreras los jugadores vuelven al lobby donde estarán listos para entrar en otra sala.

6.9 Spawns

Al cargar un escenario el objeto Spawner se encarga de instanciar una copia de todos los coches de los jugadores, cada uno con su modelo elegido, para ello revisa las propiedades de cada jugador en la sala.


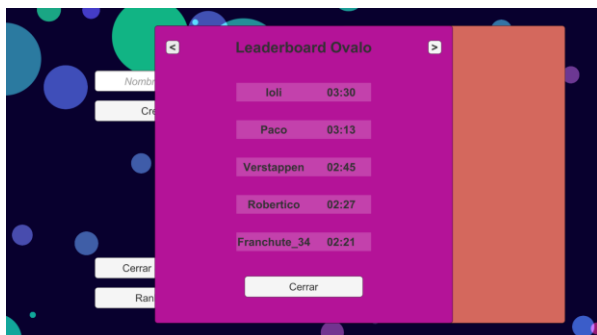
Para que cada coche tenga una posición diferente, el líder de la sala asigna previamente una posición a cada jugador, al entrar en los circuitos cada jugador será consciente de su punto de aparición y de la del resto de jugadores.

Un hecho curioso es que las cámaras no pueden ir adjuntas al prefab del jugador, sino que tienen que existir previamente en el escenario y desde este objeto se le dice al jugador que

tome esta cámara y la utilice. En caso de no trabajar de esta manera es posible que un jugador vea en su pantalla el movimiento de otro jugador.

6.10 Interfaz Menús y Gameflow

<p>Inicio de Sesión</p> <p>En esta primera ventana le pediremos al usuario los datos de inicio de sesión, es decir el correo electrónico y la contraseña. También tenemos un botón de “Iniciar”, otro para crear un nuevo usuario llamado “Registrar” y el último botón llamado “Salir” para cerrar el juego.</p>	
<p>Nuevo usuario</p> <p>En la ventana de creación de usuarios le pediremos al usuario datos como su nombre, el correo electrónico y la contraseña. Para efectuar el registro pulsaremos el botón “Crear” y si necesitamos volver al menú inicial lo haremos mediante el botón “Atrás”. Todos los campos están comprobados y en caso de que alguno no fuera válido se muestra un texto en rojo como feedback al usuario.</p>	
<p>Gestor de salas</p> <p>Esta ventana nos permite crear salas rellenando el nombre y pulsando el botón “Crear” o podemos acceder a una sala ya creada haciendo click en ella en la lista izquierda. En la parte inferior podremos pulsar el botón “Cerrar Sesión” para volver a iniciar sesión de nuevo o consultar el ranking de mejores tiempos pulsando el botón “Ranking”</p>	

<p>Selección de Vehículo</p> <p>En esta pantalla veremos los jugadores que hay dentro de la sala. Nuestro vehículo aparecerá resaltado en amarillo y podremos elegir otro usando los botones laterales. Si somos el líder de la sala podremos iniciar la partida pulsando "Jugar"</p>	
<p>Leaderboard</p> <p>Esta pequeña ventana nos muestra el ranking de mejores tiempos. Si hacemos uso de los botones superiores podremos ver los mejores tiempos de otros circuitos.</p>	

6.11 Gestión Usuarios

La gestión de usuarios es una parte importante en un juego online. Los usuarios deben de estar identificados claramente para poder representar sus nombres en partida, por ello se precisa tener un registro de nuevos usuarios.

Para llevar a cabo esta tarea utilizo la herramienta Playfab. Playfab es un servicio online de Microsoft que nos permite almacenar datos de manera permanente en un servidor. Esta herramienta está especialmente diseñada para trabajar con videojuegos, por ello incorpora métodos para guardar tipos de datos string dónde almacenar cualquier cosa en ellos.

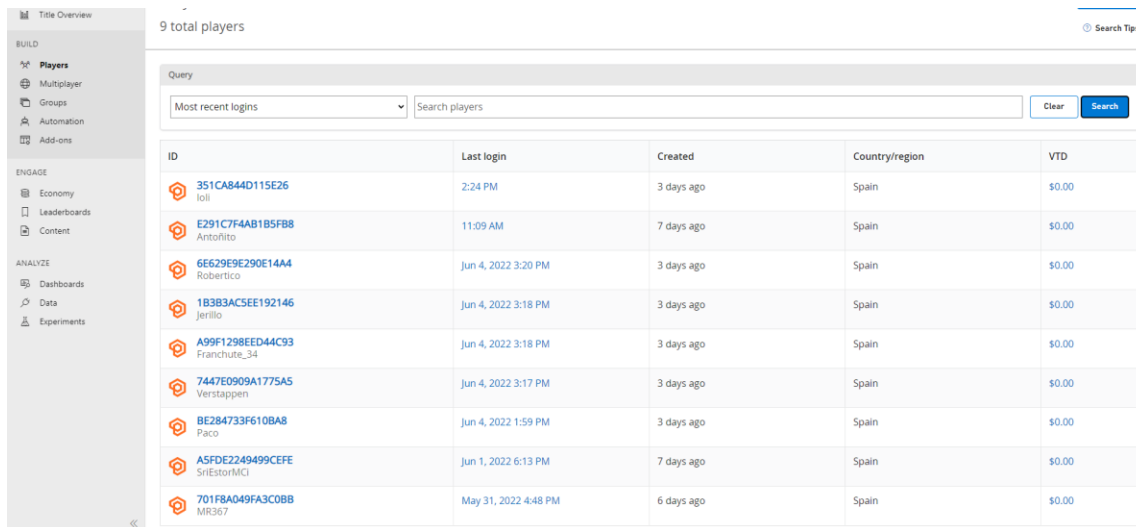
Para poder usar esta herramienta empezamos registrando nuestro juego en su página [web](#), a continuación, descargamos el sdk de playfab y lo importamos en nuestro proyecto de Unity. Playfab abrirá automáticamente una pestaña de configuración dónde tendremos que iniciar sesión y ahora sí que nos permitirá instalar el sdk, a continuación, introduciremos los datos de nuestro estudio y nuestro título.

Una vez completada la instalación podremos pasar a gestionar el guardado de datos desde un script.

Desde la interfaz de creación de usuarios llamaremos a un método del script que se encargará de lanzar requests al servicio de playfab. Nosotros tendremos que encargarnos

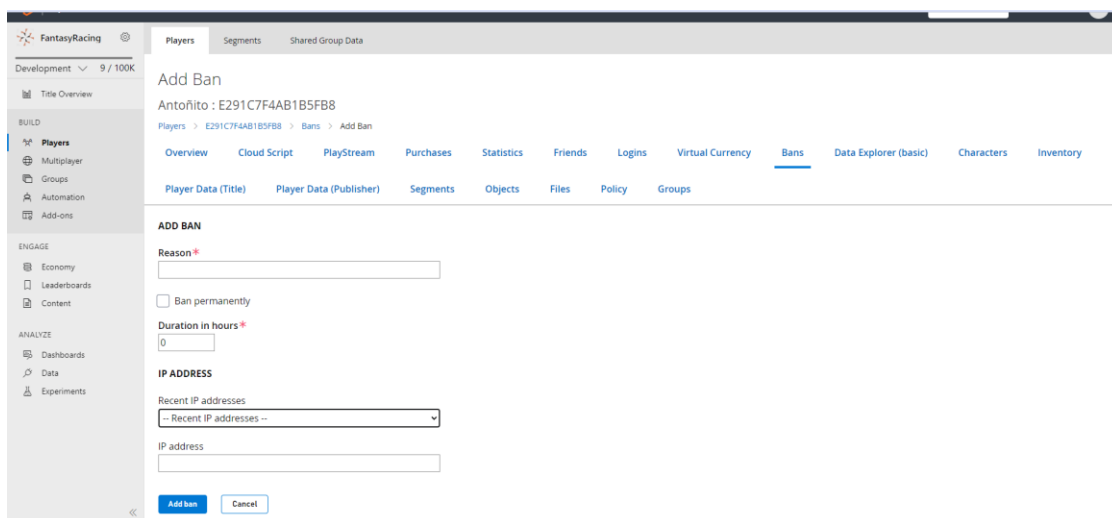
de recibir los datos del usuario, comprobar inicialmente que son correctos y mandarlos por la petición, aun así, Playfab volverá a revisar los datos enviados y nos devolverá un mensaje de error si fuera necesario.

Si ya tenemos un usuario creado podemos iniciar sesión. La primera vez que iniciemos nos pedirá que ingresemos el nombre que queremos que otros jugadores puedan ver, a partir de esta vez el registro de los datos del usuario habrá finalizado y solo registrará estadísticas básicas.



ID	Last login	Created	Country/region	VTD
351CA844D115E26 loli	2:24 PM	3 days ago	Spain	\$0.00
E291C7F4AB1B5FB8 Antoñito	11:09 AM	7 days ago	Spain	\$0.00
6E629E9E290E14A4 Robertico	Jun 4, 2022 3:20 PM	3 days ago	Spain	\$0.00
1B3B3AC5EE192146 Jerillo	Jun 4, 2022 3:18 PM	3 days ago	Spain	\$0.00
A99F1298ED44C93 Franchute_34	Jun 4, 2022 3:18 PM	3 days ago	Spain	\$0.00
7447E0909A1775A5 Verstappen	Jun 4, 2022 3:17 PM	3 days ago	Spain	\$0.00
BE284733F610BA8 Paco	Jun 4, 2022 1:59 PM	3 days ago	Spain	\$0.00
ASFDE2249499CEFE SnEstorMCI	Jun 1, 2022 6:13 PM	7 days ago	Spain	\$0.00
701F8A049FA3C0B8 MR367	May 31, 2022 4:48 PM	6 days ago	Spain	\$0.00

Por si fuera poco, Playfab nos brinda la posibilidad de gestionar todos los datos de nuestros usuarios desde su Dashboard, aquí podremos analizar los datos y detectar actividad maliciosa por parte del usuario. Como parte fuerte de este servicio podremos también prohibir la entrada a usuarios problemáticos, tanto en una cuenta concreta como todas las que provengan de la misma dirección IP.



Add Ban

Antoñito: E291C7F4AB1B5FB8

Players > E291C7F4AB1B5FB8 > Bans > Add Ban

Overview Cloud Script PlayStream Purchases Statistics Friends Logins Virtual Currency **Bans** Data Explorer (basic) Characters Inventory

Player Data (Title) Player Data (Publisher) Segments Objects Files Policy Groups

ADD BAN

Reason *

☐ Ban permanently

Duration in hours *

0

IP ADDRESS

Recent IP addresses

-- Recent IP addresses --

IP address

Add ban Cancel

6.12 Leaderboard

Para gestionar la Leaderboard utilizamos nuevamente Playfab.

Podríamos haber registrado los mejores tiempos de los usuarios nosotros mismos, almacenando en cada usuario su mejor tiempo en cada circuito y añadiendo o eliminando tiempos de una lista local ordenada por menor tiempo.

Playfab nos facilita la tarea de gestión de Leaderboards, precisamente tiene un apartado dedicado a este tema. Desde su dashboard nosotros crearemos una Leaderboard, introduciremos el nombre y que método de guardado queremos que utilice: menor cantidad, mayor cantidad, última cantidad... ella misma detectará el orden que queremos usar según este método para mostrarla posteriormente. Nuestra tarea es la de mandar el tiempo obtenido por el jugador al acabar la carrera y Playfab se encargará de saber si el dato que tiene que conservar es el que había anteriormente o el nuevo.

Cuando queramos mostrar la leaderboard haremos una petición a la API de playfab solicitando información de la lista con el nombre dado y la cantidad de registros que queremos.

Leaderboard Ovalo
Minimum (always use the lowest value)

Buttons: [Edit leaderboard](#) [Reset now](#)

Rank	Name	Value
0	Paco	193
1	Ioli	167
2	Verstappen	165
3	Robertico	147
4	Franchute_34	141
5	Antofito	128
6	Jerillo	111
7	SriEstorMCI	1

Version	Start	End	Archive log
0	May 31, 2022 2:51 PM		

Por último, vamos a mencionar cómo se muestran los datos. Al igual que cómo el ranking in-game o la selección de vehículo, para mostrar estos datos se ha creado un prefab de un ítem de la lista y se ha usado un panel con un layout vertical al que se van añadiendo instancias de este prefab. El listado se modifica según el circuito seleccionado con los botones superiores de la Leaderboard y los datos de los ítems se modifican según los datos de los usuarios cargados.

6.13 Sonido

Música

Para gestionar la música del juego he utilizado un objeto vacío que contiene un Audio Source y un Script que gestiona la música.

Este script detecta si ya existe un objeto reproductor de música en la escena y si es así destruye el creado en la nueva escena. Al cargar una escena según el nombre de la escena reproducirá una u otra canción.

Efectos de Sonido

Todos los objetos, así como los coches tienen un Audio Source y un script de control. En el caso de los vehículos algunas habilidades reproducen un sonido al ser lanzadas, los objetos pueden reproducir un sonido al ser instanciados, al chocar contra un coche o al destruirse.

Los sonidos son asignados a los prefabs desde el editor visual de Unity.

7. Manual de usuario

7.1 Menús

Para poder rellenar los campos de texto en las pantallas de Inicio de sesión y creación de nuevo usuario la persona debe de hacer click sobre ellos. Una vez que estén rellenos todos los campos, si no recibimos una alerta de color rojo de que algo está fallando podremos continuar.

En la pantalla de selección de sala podemos crear una nosotros mismos rellenando el campo “Nombre Sala” y haciendo click en crear o haciendo click sobre uno de los botones con el nombre de una sala ya creada.

En la vista de selección de vehículo podemos hacer click en los botones laterales del jugador con nuestro nombre y resaltado en color amarillo para cambiar entre un coche y otro.

En la ventana de Leaderboards podemos consultar los diferentes rankings utilizando los botones situados arriba, concretamente a izquierda y derecha del nombre del ranking visualizado actualmente.

7.2 Pantalla de juego

En la pantalla de juego veremos el contador de vueltas situado arriba a la izquierda, el ranking de posiciones situado a la derecha, la barra de vida situada abajo y las habilidades con su tiempo de reutilización representado por un fragmento de circulo en la parte inferior.

Si pulsamos la tecla escape podremos acceder al un menú que nos permitirá abandonar la partida o cerrar el juego.

7.3 Controles

W	Aceleración del vehículo
S	Frenado y marcha atrás
A	Giro a la izquierda
D	Giro a la derecha
Espacio	Derrape
U	Uso de habilidad 1
I	Uso de habilidad 2
O	Uso de habilidad 3
P	Uso de habilidad 4
Escape	Apertura menú en partida

7.4 Consejos

Para disfrutar del juego hazlo acompañado de amigos y/o familiares, este juego pretende entretener y divertir, no crear una competición tóxica.

A la hora de manejar el vehículo la velocidad no es lo más importante, si conduces a una velocidad excesiva lo más probable es que te choques con paredes u obstáculos.

Dependiendo el circuito será más o menos aconsejable el uso del turbo en ciertas zonas, ya que es probable que si lo utilizas en una rampa acabes saliendo del circuito y reapareciendo en el último punto de control.

Utiliza tus habilidades con calma, no es mejor utilizarlas cada vez que las tengas, sino en el momento adecuado. Gestiona los tiempos de reutilización acorde a tu posición en la carrera y espera cualquier acción del enemigo posible.

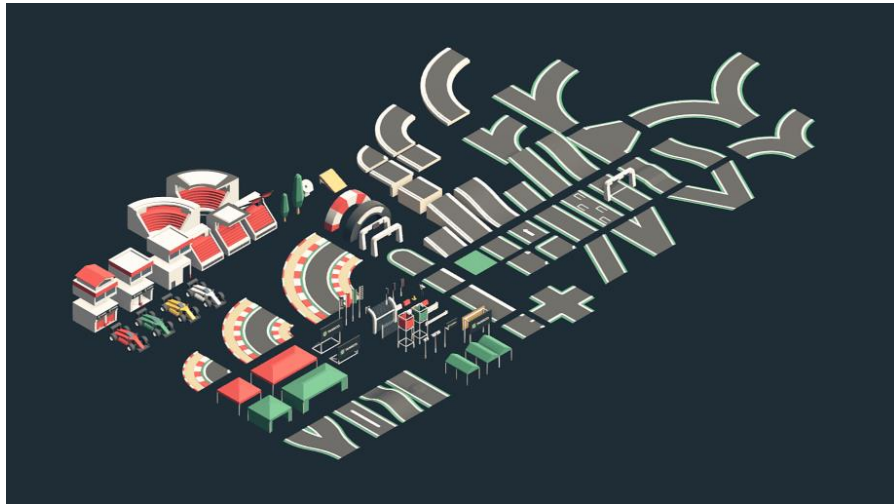
No existe un coche mejor que otro, se ha previsto el balanceo de estadísticas y lo que un coche tiene de bueno o malo lo compensa por otra parte.

Diviértete con este juego y no seas tóxico con el resto de jugadores, al final de la partida la clasificación deja de importar tras unos minutos.

8. Recursos utilizados

8.1 Modelos 3D

Para construir el diseño de los circuitos, el trazado de la calzada y la decoración del escenario he utilizado modelos del paquete [“Racing Kit”](#) de la compañía Kenney. Este paquete es bastante completo y su diseño Low Poly⁷ hace que no sean pesados en almacenamiento ni en rendimiento del juego.



Para los modelos de los coches he utilizado el paquete "Car Kit" también de la compañía Kenney. Al ser de la misma compañía tienen un diseño semejante que encaja a la perfección con los modelos 3D del circuito.

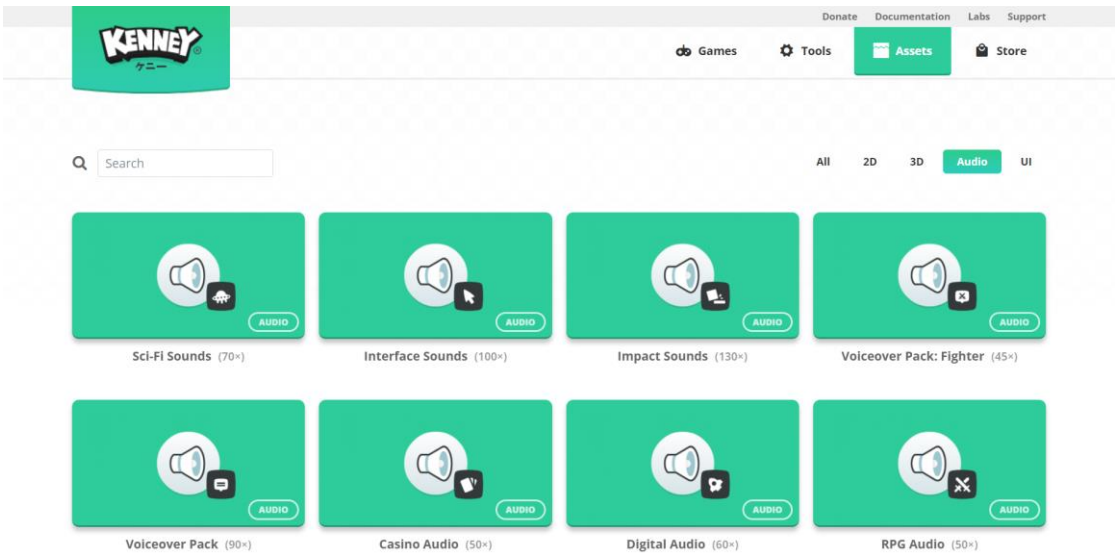


⁷ Low Poly: Malla poligonal en gráficos 3D por computadora que tiene un número relativamente pequeño de polígonos.

En referencia a los objetos utilizables por los jugadores como la bomba, el fuego o la basura he utilizado modelos procedentes de la página web [Sketchfab](#).

8.2 Audio

Para los efectos de sonido del videojuego he utilizado los archivos de audio descargables de la compañía [Kenney](#). He utilizado los paquetes “Sci-Fi Sounds”, “Digital Audio” e “Impact Sounds”.



La música del videojuego proviene del autor [Patrick de Arteaga](#) que en su página web nos ofrece una amplia variedad de temas. Concretamente he utilizado las canciones “Chiptonical”, “Electronic Fantasy” e “Intergalactic Odyssey”.

PATRICK DE ARTEAGA Music for Games and Multimedia			Inicio	Música GRATIS	
NOMBRE	DESCRIPCIÓN	ESCUCHAR			
Child's Nightmare DESCARGAR	Juguetera y energética, esta canción vale para juegos de plataformas.				
The Biggest Discovery DESCARGAR	Música de misterio para realzar tus escenas más enigmáticas.				
No Place For Straw Cowboys DESCARGAR	Basada en el conocido estilo de películas Western. ¡Desenfunda!				
Battleship DESCARGAR	Creada para tus batallas o escenas tensas. Huye o luchemos contra el peligro.				
Not Giving Up DESCARGAR	Canción épica in crescendo. No es medieval pero sí orquestal con añadidos electrónicos.				
Intergalactic Odyssey DESCARGAR	Otro track futurista con un estilo de rock, adecuado para batallas o juegos arcade.				
Spring Village DESCARGAR	Una canción de estilo celta, perfecta para ambientar la Edad de Hierro o tiempos medievales.				
Chase At Rush Hour DESCARGAR	Este track de estilo cinemático es ideal para escenas de acción o momentos jugables de mucha adrenalina.				

9. Conclusiones

9.1 Conocimientos aplicados

Durante este periodo de proyecto he aprendido nuevas capacidades de la herramienta Unity, sobre todo he adquirido bastantes conocimientos en el apartado de scripting con C# y el propio uso del lenguaje.

También he aplicado nuevos conocimientos como el uso de Playfab para almacenar los datos, hasta el momento pensaba que se recomendaba utilizar siempre Firebase.

Embarcarme en un proyecto online ha sido todo un reto, pero he podido sacarlo adelante aprendiendo a utilizar el componente Photon.

9.2 Dificultades encontradas

En general la mayor dificultad que ha tenido este proyecto ha sido la sincronización de los jugadores en línea. En un principio no era capaz de conseguir que unos jugadores visualizaran el coche de los demás, poco después lo conseguí, pero un jugador manejaba el coche del otro y viceversa.

El control de los datos de usuario era un tema ambiguo ya que la información que había en internet podía ocasionar confusión. Unos usuarios te aconsejaban utilizar el guardado de datos mediante el jugador local, pero eso nunca ha funcionado de manera correcta, la solución fue utilizar las propiedades personalizadas de cada jugador en lugar de las locales.

Finalmente, habilidades de control global como el desfibrilador fueron complicadas de implementar ya que el componente no está preparado para seleccionar jugadores en concreto fácilmente, sino que tienes que instanciar objetos y aplicar efectos.

9.3 Resultados obtenidos

Finalmente puedo decir que a grandes rasgos estoy contento con el resultado del proyecto. Ha sido un gran reto y puedo decir que he logrado superarlo.

Sea como sea este proyecto ya ha sido suficiente para mí ya que he podido disfrutarlo en persona con mis amigos, algo que para mí siempre ha sido un sueño.

10. Trabajo futuro

Para cumplimentar este proyecto sería interesante incorporar las siguientes características en futuras versiones:

Modos adicionales

Algunos de los modos disponibles pueden ser partidas por equipos, arenas de batalla con otro tipo de objetivos o incluso partidas clasificatorias dónde los jugadores competirán por tener la mejor puntuación posible compitiendo con otros jugadores de su mismo nivel.

Mapas

Inicialmente el juego tendrá tres circuitos, pero esto será escalable pudiendo crear infinidad de circuitos con distinto trazado, obstáculos, vueltas, etc.

Vehículos

Inicialmente el juego tendrá seis vehículos, para que el jugador se vea incentivado a seguir jugando será posible conseguir otros coches pagando por ellos con una moneda del juego.

Eventos

El juego tendrá eventos de temporada relacionados con la época del mundo real, estos eventos modificarán artísticamente los circuitos incluyendo modelos 3D relacionados con el evento.

- Evento de VERANO con pelotas de playa, sombrillas, etc.
- Evento de INVIERNO con nieve por la pista.
- Evento de CHERRY BLOSSOM con árboles de cerezo, pétalos de flor de cerezo, etc.
- Evento de HALLOWEEN decorando el circuito con calabazas, fantasmas y esqueletos.

Los jugadores podrán obtener recompensas como skins temáticas, banners de jugador, etc.

Posteriormente, con el transcurso del juego, podrán idearse infinidad de eventos a petición de los usuarios

Multiplataforma

El juego está casi totalmente preparado para ser exportado a las siguientes plataformas:

- Nintendo Switch
- PlayStation
- Xbox
- Web GL
- Dispositivos Móviles

Sin embargo, sería necesario hacer algunos ajustes para que pueda ser jugado:

- Asignar el uso de habilidades a los botones del un mando de consola
- Asignar la función de derrape a un botón de mando de consola
- Asignar un botón del mando de consola al menú in-game
- Implementar un joystick y botones en pantalla para la versión de móvil

Si se utilizase cualquier otra plataforma diferente a un ordenador o un navegador web sería necesario implementar un puntero para interactuar con los menús u ofrecer la posibilidad de navegar utilizando las flechas del mando. A la hora de introducir texto también sería necesario que aparezca el teclado en móviles y consolas.


Monetización

Ya que el juego se trata de un Free-to-play la fuente de ingresos serán los micro pagos. El jugador podrá conseguir monedas dentro del juego en proporción a su posición final en la partida. Con estas monedas podrá conseguir vehículos además de los iniciales, su precio será elevado por lo que tendrá que jugar bastantes partidas para conseguirlos o usar dinero real para conseguir esas monedas rápidamente.

Para obtener beneficio también se venderán monedas virtuales diferentes a las conseguidas dentro del juego. Con estas monedas el jugador podrá adquirir antes los siguientes ítems:

○ Vehículos

Los jugadores más impacientes podrán conseguir los vehículos invirtiendo las monedas en ellos. Aunque se trate de un recurso gratuito algunos jugadores prefieren invertir dinero en esos recursos



○ Skins

Los jugadores podrán personalizar su coche para diferenciarse del resto con skins. Estas skins modificarán desde el color del coche hasta completamente su forma, según el grado de personalización el precio variará. El precio de las skins traducido a euros oscila entre los 2 y 10 euros.

El pago por ítems virtuales dentro de los videojuegos se ha convertido en una de las principales fuentes de ingresos de esta industria. Grandes títulos como League of Legends es capaz de facturar millones de dólares con una tirada quincenal de skins para sus personajes.

11. Bibliografía

[1] Unity: <https://docs.unity.com/>

[2] C#: <https://docs.microsoft.com/es-es/dotnet/csharp/>

[3] Photon: <https://doc.photonengine.com/en-us/fusion/current/getting-started/fusion-intro>

[4] Playfab: <https://docs.microsoft.com/en-us/gaming/playfab/>

[5] Kenney: <https://www.kenney.nl/>

[6] Sketchfab: <https://sketchfab.com/>

[7] Música: <https://patrickdearteaga.com/>