



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Desarrollo de un Videojuego de Carreras con  
Implementación de IA

Trabajo Fin de Grado

Grado en Tecnologías Interactivas

AUTOR/A: Campins Pérez, Marc

Tutor/a: Sendra Compte, Sandra

CURSO ACADÉMICO: 2022/2023

## Resumen y palabras clave

El objetivo principal de este proyecto es desarrollar un videojuego de carreras con implementación de inteligencia artificial para los oponentes. Para lograr este objetivo, se utilizará Unity 3D como motor de desarrollo de juegos, aprovechando sus capacidades para la creación de entornos y físicas.

El videojuego contará con vehículos controlados por IA, que competirán en carreras contra el jugador. Se explorarán diferentes técnicas de IA, como el uso de ML-Agents o la programación de comportamientos específicos, para lograr que los rivales **tomen** decisiones inteligentes y desafiantes.

Este TFG está pensado para una duración de un semestre, por lo tanto, se han establecido objetivos realistas que se ajustan al tiempo disponible. El enfoque del proyecto se centra en crear una demostración funcional completa del videojuego, que contará con un número limitado de pistas y tipos de vehículos.

Si bien el proyecto no tiene como objetivo final ser un videojuego completo listo para salir al mercado, se buscará que esté pulido y optimizado, identificando y corrigiendo errores durante el proceso de pruebas y depuración para brindar una experiencia satisfactoria al usuario.

**Palabras clave:** Unity 3D, videojuego, competición, carreras de coches, Inteligencia Artificial.

## Resum i paraules clau

L'objectiu principal d'aquest projecte és desenvolupar un videojoc de carreres amb implementació d'intel·ligència artificial per als oponents. Per assolir aquest objectiu, s'utilitzarà Unity 3D com a motor de desenvolupament de jocs, aprofitant les seues capacitats per crear entorns i físiques.

El videojoc comptarà amb vehicles controlats per IA, que competiran en carreres contra el jugador. S'exploraran diferents tècniques d'IA, com l'ús de ML-Agents o la programació de comportaments específics, per aconseguir que els rivals prenguen decisions intel·ligents i desafidores.

Aquest TFG està pensat per a una duració d'un semestre, per tant, s'han establert objectius realistes que s'ajusten al temps disponible. L'enfocament del projecte se centra a crear una demostració funcional completa del videojoc, que comptarà amb un nombre limitat de pistes i tipus de vehicles.

Tot i que el projecte no té com a objectiu final ser un videojoc complet llest per eixir al mercat, es buscarà que estiga polit i optimitzat, identificant i corregint errors durant el procés de proves i depuració per brindar una experiència satisfactòria a l'usuari.

**Paraules clau:** Unity 3D, videojoc, competició, carreres de cotxes, Intel·ligència Artificial.

## Abstract and keywords

The main objective of this project is to develop a racing videogame with artificial intelligence implementation for opponents. To achieve this goal, Unity 3D will be used as a game development engine, taking advantage of its capabilities for the creation of environments and physics.

The video game will feature AI-controlled vehicles, which will compete in races against the player. Different AI techniques will be explored, such as the use of ML-Agents or the programming of specific behaviors, to make rivals make intelligent and challenging decisions.

This TFG is intended for a duration of one semester, therefore, realistic goals have been set to fit the time available. The focus of the project is on creating a full functional demo of the video game, which will feature a limited number of tracks and vehicle types.

While the project's final goal is not to be a complete video game ready for market, it will be polished and optimized, identifying and correcting bugs during the testing and debugging process to provide a satisfactory user experience.

**Keywords:** Unity 3D, videogame, competition, cars racing, Artificial Intelligence.

# Índice

Resumen y palabras clave.....	1
Resum i paraules clau .....	2
Abstract and keywords.....	3
Índice.....	4
Índice de imágenes.....	5
Capítulo 1. Introducción.....	6
1.1    Presentación.....	6
1.2    Objetivos del proyecto .....	7
1.3    Estructura de la memoria.....	7
1.4    Presupuesto y distribución temporal .....	8
1.5    Trabajos relacionados .....	10
Capítulo 2. Desarrollo .....	12
2.1    Introducción al desarrollo .....	12
2.2    Metodología y herramientas de desarrollo.....	12
2.3    Diseño e implementación del videojuego.....	14
2.3.1    Implementación del controlador del coche .....	14
2.3.2    Implementación de una IA para los oponentes .....	16
2.3.3    Sistema de puntos de control.....	18
2.3.4    Implementación de los modos de juego .....	19
2.3.5    Interfaz de usuario .....	21
2.3.6    Audio y efectos visuales.....	22
2.4    Problemas de implementación .....	23
Capítulo 3. Testeo y resolución de errores.....	25
Capítulo 4. Manuales .....	26
4.1    Instalación .....	26
4.2    Guía de uso.....	28
Capítulo 5. Evaluación .....	34
5.1    Cumplimiento de objetivos .....	34
5.2    Conclusiones.....	34
5.3    Aportaciones personales .....	35
6. Referencias.....	37

## Índice de imágenes

Figura 1. Sección de assets 3D descargables de la Unity Asset Store .....	9
Figura 2. Diagrama de Gannt con la distribución temporal del proyecto .....	10
Figura 3. Interfaz de Unity.....	13
Figura 4. Interfaz del proyecto musical en FL Studio Mobile .....	14
Figura 5. Ejemplo de curva de torque en relación a las RPM del motor .....	15
Figura 6. Geometría de dirección de Ackermann .....	16
Figura 7. Disposición de sensores del coche .....	18
Figura 8. Distribución de distintos puntos de control en el circuito .....	19
Figura 9. Circuito generado con el paquete Road Track Generator .....	20
Figura 10. Curva de recompensa acumulada del agente .....	23
Figura 11. Curva de fricción usada por el WheelCollider.....	24
Figura 12. Paso 1 - Selección de Ruta de Instalación.....	26
Figura 13. Paso 2 - Creación de Acceso Directo.....	27
Figura 14. Paso 3 - Proceso de Instalación.....	27
Figura 15. Menú principal del juego .....	28
Figura 16. Menú de opciones del juego .....	29
Figura 17. Menú de configuración del juego .....	29
Figura 18. Menú desplegable para elegir la calidad gráfica del juego .....	30
Figura 19. Menú de controles del juego .....	30
Figura 20. Menú de pausa del juego.....	31
Figura 21. Cuenta atrás al inicio de una carrera.....	31
Figura 22. Interfaz en el modo "Race".....	32
Figura 23. Interfaz en modo "Practice" .....	32
Figura 24. Barra de restablecimiento del coche .....	33
Figura 25. Tabla de resultados mostrada al final de una carrera.....	33

# Capítulo 1. Introducción

## 1.1 Presentación

El presente trabajo consiste en el desarrollo de un videojuego de carreras que integra una inteligencia artificial para los oponentes. Este proyecto surge como respuesta a la creciente carencia de videojuegos de carreras en el género arcade, dirigidos a aquellos jugadores que buscan una experiencia de conducción emocionante sin la necesidad de adquirir dispositivos especializados costosos, como los volantes de conducción.

Durante la última década, los videojuegos de carreras han experimentado una notable transformación, alejándose de su esencia arcade para centrarse en simulaciones de conducción realistas. Si bien esta transformación ha satisfecho al segmento de jugadores aficionados a la simulación, ha creado una brecha en la escena 'mainstream'. Entre estas destacan las sagas Need For Speed y Gran Turismo. La primera experimentó un declive debido a la pérdida de características emblemáticas, como la personalización de los coches, una conducción que se volvió tediosa debido a sus físicas y cambios frecuentes de empresa desarrolladora. La segunda de estas sagas también sufrió pérdidas debido a la eliminación de modos de juego queridos por los fans, la reutilización de modelos de coches antiguos y el abuso de microtransacciones.

El videojuego, titulado "Sprint Rush", utiliza Unity 3D como motor de desarrollo y presenta la implementación de vehículos autónomos, que compiten en carreras contra el jugador, con el propósito de ofrecer oponentes desafiantes e inteligentes para mejorar la experiencia de juego.

El nombre "Sprint Rush", que se podría traducir como carrera de velocidad, es una representación de la experiencia que el juego pretende brindar a los jugadores. "Sprint" resalta la idea de una carrera rápida y enérgica, mientras que "Rush" sugiere la emoción y la adrenalina que caracterizan las carreras arcade.

El enfoque del presente proyecto se adapta a un marco temporal de un semestre académico, lo que implica la definición de objetivos realistas y alcanzables en dicho período. El objetivo principal consiste en la creación de una demostración funcional y pulida del videojuego, presentando un número limitado de pistas y vehículos.

Por tanto, es importante mencionar que el propósito no es la elaboración de un videojuego completo para su comercialización, sino la creación de una experiencia de juego satisfactoria que destaque por su jugabilidad y fluidez. A lo largo del proceso, se dedicará especial atención a la detección y corrección de errores, garantizando una experiencia de usuario óptima.

La perspectiva de este proyecto es, por lo tanto, abordar un problema actual en el ámbito de los videojuegos de carreras arcade, aplicando los conocimientos en desarrollo de videojuegos y programación adquiridos, y buscando proporcionar a los jugadores una experiencia de conducción emocionante y accesible.

## 1.2 Objetivos del proyecto

El objetivo principal de este proyecto abarca dos aspectos fundamentales. En primer lugar, se busca aumentar mi conocimiento y experiencia en el desarrollo de videojuegos, en concreto en el género de carreras. A través de la creación del videojuego "Sprint Rush", mi intención es adquirir una comprensión más profunda de los procesos involucrados en el diseño y la programación de juegos de este tipo.

En segundo lugar, este proyecto aspira a presentar una versión inicial funcional del videojuego. Aunque el enfoque actual se encuentra en el desarrollo de una demostración que demuestre la viabilidad del concepto, esta versión puede sentar las bases para futuras expansiones. La intención es que el juego pueda evolucionar con el tiempo, incorporando una gama más amplia de características y mejoras que enriquezcan la experiencia de juego.

Adicionalmente, esta demostración funcional puede desempeñar un papel vital como parte de un futuro portafolio que me permita exhibir esta creación, presentando de manera concreta mis aptitudes y logros en el ámbito del desarrollo de videojuegos.

En resumen, este proyecto no solo pretende incrementar mi conocimiento y experiencia, sino también establecer los cimientos de un videojuego funcional que, a lo largo del tiempo, podrá expandirse con nuevas características y mejoras.

Además de estos objetivos principales, también se busca atraer a jugadores que busquen una experiencia arcade sencilla y divertida en el contexto de un videojuego de carreras. En un entorno donde los títulos de carreras han evolucionado hacia simulaciones más realistas, este proyecto pretende llenar un vacío al ofrecer una alternativa accesible y emocionante para un público que busca la diversión inmediata propia de los juegos arcade.

## 1.3 Estructura de la memoria

La memoria está dividida en capítulos. Cada uno de ellos explica las diferentes partes del proyecto.

### **Capítulo 1: Introducción**

En el primer capítulo se realiza una presentación del proyecto, explicando en qué consiste y de donde surge la motivación de este. También se enumeran los distintos objetivos del proyecto.

### **Capítulo 2: Desarrollo**

En el segundo capítulo se proporciona una introducción al proceso de desarrollo del proyecto, seguida de una explicación sobre el motor de desarrollo y la metodología. A continuación, se desglosan los pasos seguidos para la creación del juego, incluyendo la implementación del controlador del coche, el sistema de inteligencia artificial, los distintos modos de juego y el diseño de los menús.

### **Capítulo 3: Testeo y resolución de problemas**

En el tercer capítulo, se describen las pruebas realizadas en el juego para encontrar y corregir errores. También se explican los problemas o dificultades encontrados durante el desarrollo y las alternativas que he usado para enfrentarlos.

### **Capítulo 4: Manuales**

En el cuarto capítulo se explica dónde encontrar el proyecto, cómo instalarlo y cómo usarlo de forma general.

### **Capítulo 5: Conclusiones**

En el quinto capítulo, se realiza una valoración final del proyecto, conclusiones y las aportaciones personales.

### **Capítulo 6: Referencias**

En este último capítulo se listarán todas las referencias consultadas, así como páginas o enlaces importantes para el entendimiento del trabajo.

## **1.4 Presupuesto y distribución temporal**

### **Presupuesto**

#### **1. Licencia de Unity**

Aunque el motor de desarrollo Unity ofrece un plan gratuito, siempre y cuando los ingresos obtenidos de su uso en los últimos 12 meses no superen los 100 mil USD, hay que tener en cuenta los posibles gastos asociados a algunos recursos necesarios para el desarrollo.

El plan de estudiante de Unity que he utilizado, ofrece ciertas características gratuitas. Si no contamos con dicha licencia, obtener dichas características requeriría el plan Unity Plus, el cual incluye algunas ventajas adicionales a un costo anual de 369 € por puesto.

#### **2. Unity Asset Store**

Además, he hecho uso de distintos 'assets' (items que pueden ser utilizados en un juego o proyecto) de la Unity Asset Store, que es la biblioteca de 'assets' comerciales y gratuitos creados por Unity Technologies y miembros de la comunidad. Éstos me han ahorrado tener que crear yo mismo los modelos 3D de los vehículos y pistas de carreras, así como distintos elementos de la interfaz, y centrarme en la estructura y programación del videojuego:

- Paquete de vehículos: 129.74 € [1]
- Generador de circuitos de carreras: 99.80 € [2]
- Set de botones: Gratuito [3]
- Fuentes de texto: Gratuito [4]

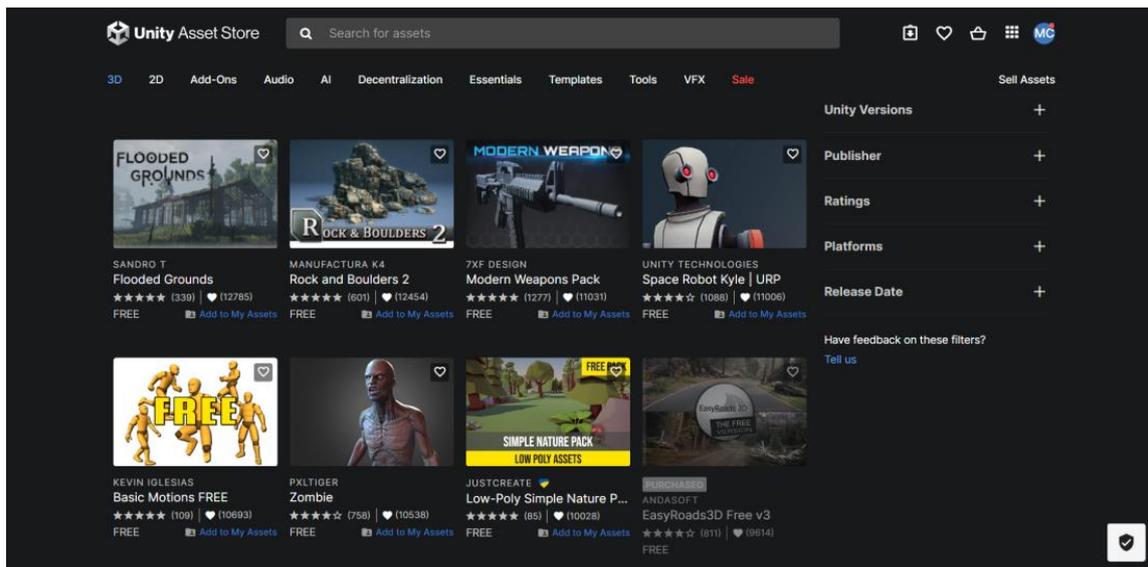


Figura 1. Sección de assets 3D descargables de la Unity Asset Store

Dado que no es demasiada mi experiencia en el desarrollo de juegos en Unity y se trata de un proyecto pequeño, posiblemente se esperaría que los ingresos no superasen los 100 mil USD en los primeros 12 meses de comercialización, por tanto contamos con la licencia gratuita de Unity, ya sea la versión de estudiante o el plan Personal. Por ello, en el presupuesto total solo consideraré el costo de los 'assets', que es un total de 229.54 €.

### Distribución temporal

Antes de iniciar el proceso de desarrollo del videojuego, establecí una distribución temporal detallada de las etapas clave del proyecto. La planificación inicial se presentó de la siguiente manera:

1. Investigar la documentación de Unity y tutoriales relevantes (20 h)
2. Diseño inicial del concepto del juego (10 h)
3. Implementación de las físicas y el manejo de los vehículos (60 h)
4. Diseño y construcción de las pistas y obstáculos (40 h)
5. Implementación de la lógica de la IA para los rivales (80 h)
6. Creación de la interfaz y menús del juego (30 h)
7. Añadir elementos de sonido y efectos de sonido (20 h)
8. Pulido final del juego y solución de errores (40 h)
9. Elaboración de la memoria del TFG (50 h)
10. Preparación de la presentación para su defensa (10 h)

Al final del proceso de desarrollo, se ha producido una adaptación en la distribución temporal originalmente planificada. El tiempo planeado para el diseño y construcción de pistas lo empleé para buscar, descargar y probar los items necesarios para esta parte, lo cual no llevó tanto tiempo como lo esperado. Sin embargo, necesité una inversión mayor de tiempo para la implementación de la IA, debido a ciertas complejidades inesperadas, y para la búsqueda y solución de errores en el juego.

Aquí dejo un diagrama de Gantt, donde se expone el tiempo de dedicación final aproximado del proyecto:



Figura 2. Diagrama de Gantt con la distribución temporal del proyecto

## 1.5 Trabajos relacionados

Para el estudio de los trabajos similares a este proyecto se ha accedido a la web de “RiuNet” [5] en busca de otros trabajos académicos que tengan similitudes.

**D.A.N.G.E.R. Disaster Prevention: videojuego educativo para la gestión de emergencias en Unity 3D: interfaz de usuario, gestión y tratamiento de la información** [6]

Es un trabajo de fin de grado hecho por Adrián Sánchez Lavarias presentado en la Escola Tècnica Superior d’Enginyeria Informàtica de la Universitat Politècnica de València en el año académico 2020-2021.

Este trabajo proponía realizar un videojuego de simulación social y estrategia en tiempo real en Unity 3D para ordenador, encargándose de la interfaz de usuario incluyendo los menús y ajustes necesarios, entre otras tareas del desarrollo de videojuegos como son el diseño y la programación de las distintas mecánicas.

Es un proyecto similar debido al uso del mismo motor de desarrollo de videojuegos y las mismas tareas principales, como son el diseño y programación de la interfaz, menús y mecánicas del juego.

**Desarrollo de un videojuego en Unity: implementación de las mecánicas de juego** [7]

Es un trabajo de fin de grado hecho por José Francisco Gómez Alemany también presentado en la Escola Tècnica Superior d’Enginyeria Informàtica de la Universitat Politècnica de València en el año académico 2020-2021.

Es el primer trabajo que aparece cuando haces una búsqueda de TFGs sobre videojuegos desarrollados en Unity en la UPV, aunque se trata de un proyecto mucho más ambicioso ya que se ha desarrollado por un equipo compuesto por cuatro

alumnos, en el marco de desarrollo de sus TFG, donde cada uno se ha centrado en un área del proyecto.

## Capítulo 2. Desarrollo

### 2.1 Introducción al desarrollo

En este capítulo profundizaré en el proceso de creación y programación que ha dado vida al videojuego de carreras ‘Sprint Rush’. Explicaré qué motor de desarrollo utilicé así como los detalles del diseño e implementación de los elementos esenciales del videojuego. De esta manera podrán obtener una visión detallada de cómo he plasmado mis ideas en código, destacando las decisiones y enfoques que he adoptado para cumplir con los objetivos.

También es importante reconocer el importante efecto que han tenido distintos canales de YouTube a la hora de ayudarme a aprender ciertos aspectos de la programación en Unity. Entre estas fuentes, menciono especialmente los canales “Nanousis Development” [8] y “pablos lab” [9], quienes me han servido para el desarrollo del controlador y el sonido del automóvil. “BlinkAChu” [10] me aportó un conocimiento útil sobre las ecuaciones para implementar el giro de las ruedas. Y por último, “Code Monkey” [11] e “Imphenzia” [12] contribuyeron en gran medida a la implementación del sistema de puntos de control y el cronómetro de vueltas. Estos recursos han contribuido significativamente a mi proceso de aprendizaje y desarrollo, dándome la dirección e instrucción necesarias para llevar a cabo este proyecto con éxito.

### 2.2 Metodología y herramientas de desarrollo

#### **Scrum**

Para la organización del trabajo en el proyecto, he utilizado la metodología “Scrum”. Aunque normalmente se emplea en equipos, he adaptado algunos principios a mi situación de trabajo individual.

“Scrum” se basa en dividir el trabajo en intervalos llamados ‘sprints’, que en mi caso han sido cada dos semanas. En cada sprint, he establecido unos objetivos realistas para ese período. Además, esta metodología me ha permitido mantener un registro claro de mi progreso y tareas pendientes para estar al tanto de mi trabajo. La autoorganización me ha permitido tomar decisiones sobre cómo manejar mi tiempo y decidir qué tareas priorizar en primer lugar.

#### **Unity**

Para la creación del videojuego he usado el motor de desarrollo multiplataforma Unity [13]. La elección de este software se debe a mi experiencia previa en los proyectos relacionados con videojuegos durante el Grado en Tecnologías Interactivas. Además, Unity es una opción asequible ya que ofrece una versión gratuita para uso personal y cuenta con un plan especial para estudiantes, lo que la hace una de las herramientas más populares para la creación de videojuegos.

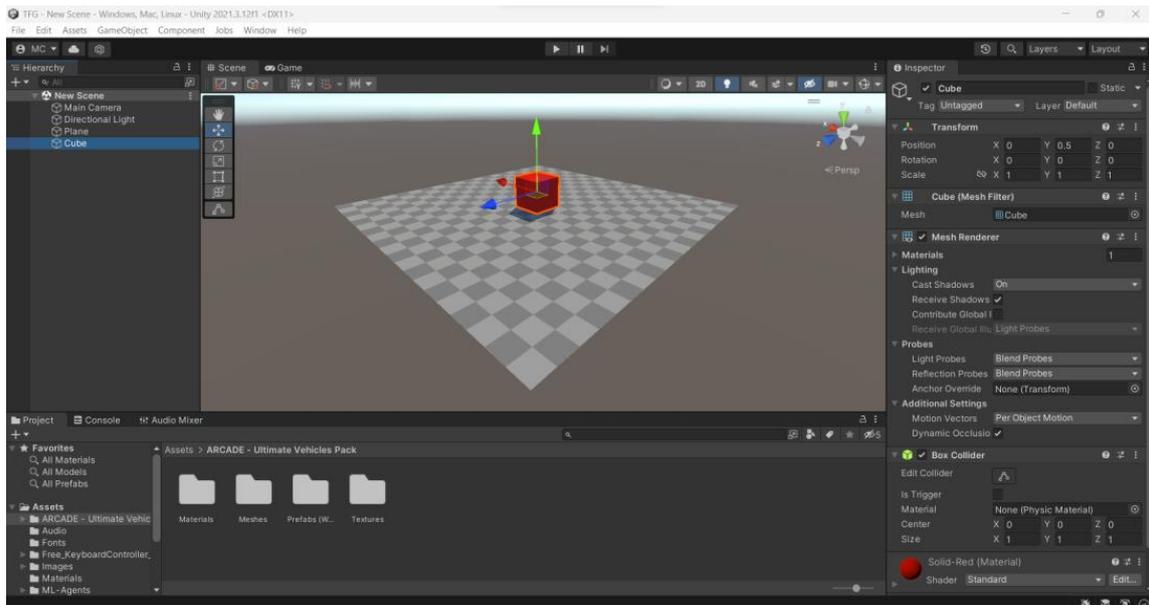


Figura 3. Interfaz de Unity

## Microsoft Visual Studio

Visual Studio [14] es un entorno de desarrollo integrado creado por Microsoft que soporta múltiples lenguajes de programación y es uno de los editores de código más utilizados para Unity, ya que ofrece una extensión que permite programar de forma más ágil.

En cuanto al lenguaje de programación, he utilizado C#. Esto se debe a que es el lenguaje estándar y el más ampliamente utilizado para los scripts o fragmentos de código en Unity. Su principal ventaja radica en que es un lenguaje de programación general muy completo y fácil de aprender.

## FL Studio Mobile

Para la música del menú principal del videojuego, decidí asumir la tarea personalmente y crearla por mí mismo. Para ello utilicé FL Studio Mobile [15], una aplicación para dispositivos móviles que uso diariamente para aprender y practicar uno de mis 'hobbies', la producción musical. Pude crear la música de manera rápida y sencilla debido a que ya estoy muy familiarizado con este programa. Esto me permitió tener más tiempo para concentrarme en la parte que realmente necesitaba más trabajo: pensar en la idea musical.



Figura 4. Interfaz del proyecto musical en FL Studio Mobile

## 2.3 Diseño e implementación del videojuego

### 2.3.1 Implementación del controlador del coche

El movimiento del coche se basa en los 'WheelCollider' de Unity, que son colisionadores en forma de circunferencia con detección de colisión integrada, física de ruedas y un modelo de deslizamiento basado en la fricción de la llanta. [16]

Estos 'WheelCollider' permiten aplicar el torque del motor expresado en Newton metro a la rueda, es decir, la fuerza de rotación; también permiten aplicar el torque de freno expresado en Newton metro, controlar el ángulo de giro de la rueda y ajustar la masa de la rueda en kilogramos. Además, ofrecen distintos parámetros de suspensión y fricción, así como la capacidad de obtener el radio de la llanta, las rotaciones por minuto del eje de la rueda o una propiedad que indica si la rueda está en contacto con una superficie sólida.

#### Aceleración

Para calcular las revoluciones por minuto del motor y la potencia total del motor me he basado en el blog "Unity3D WheelCollider and motorTorque" [17].

La siguiente fórmula sirve para calcular las revoluciones por minuto del motor:

$$RPM_{motor} = minRPM + (RPM_{ruedas} \times Relación\ de\ Transmisión\ Final \times GR_{actual}) \quad (1)$$

Donde:

- $RPM_{motor}$  son las revoluciones por minuto del motor actuales.
- $minRPM$  es el número mínimo de RPM del motor en ralentí.
- $RPM_{ruedas}$  es el promedio de RPM de las ruedas del vehículo.
- La Relación de Transmisión Final es un valor que representa la relación de todas las marchas.

- $GR_{actual}$  (de 'gear ratio') es la relación de transmisión actual, que ajusta la potencia en función de la marcha en la que se encuentra el vehículo.

La siguiente imagen muestra una curva de torque o potencia que alcanza un máximo de alrededor de 1200 Nm aproximadamente y luego cae bruscamente cerca de las 7000 rpm, lo que podría asemejarse a un motor real de un coche potente, por ejemplo, el de un Formula 1.

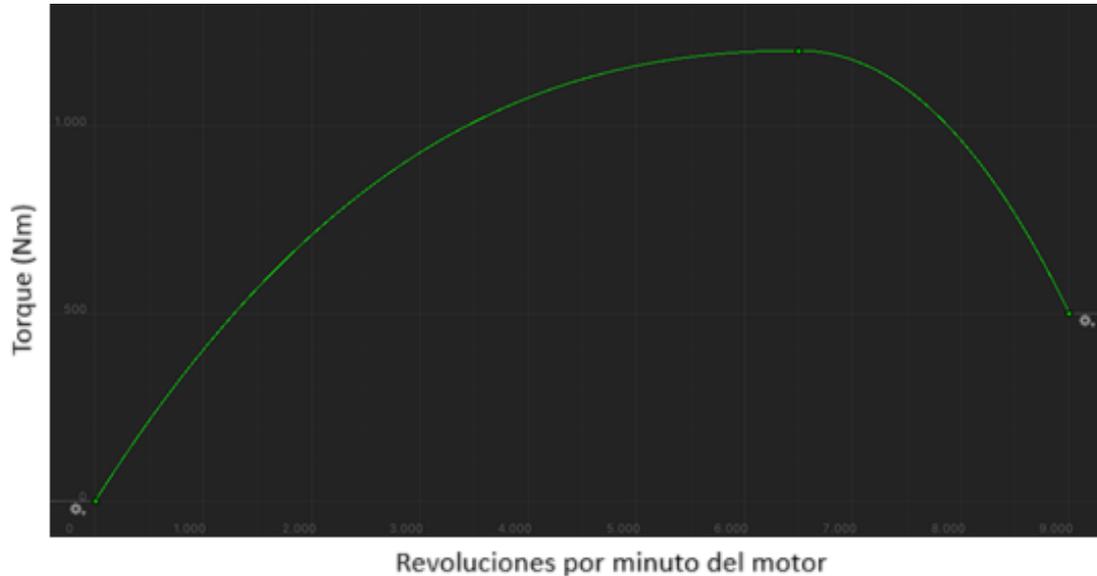


Figura 5. Ejemplo de curva de torque en relación a las RPM del motor

Para calcular la potencia total aplicada al coche he utilizado la siguiente fórmula:

$$P_{total} = P_{motor}(RPM_{motor}) \times GR_{actual} \quad ( 2 )$$

Donde:

- $P_{total}$  es la potencia total del motor.
- $P_{motor}(RPM_{motor})$  es el valor de la curva de potencia del motor en relación a las RPM del motor.

Habiendo calculado la potencia total del motor, ahora podemos distribuir esta potencia entre todas las ruedas motrices del vehículo para lograr su aceleración. Basándonos en la entrada del acelerador, que varía en un rango de -1 a 1, el jugador puede controlar la velocidad.

Cuando el jugador presiona el pedal del acelerador (con la entrada de aceleración siendo positiva), se aplicará el valor de la potencia total del motor multiplicado por el valor de la entrada del acelerador (es decir, cuando la entrada del acelerador esté completamente presionada, se aplicará la máxima potencia).

Por otro lado, al presionar el pedal del freno (con la entrada de aceleración en negativo), se aplicará una fuerza de frenado si el coche está en movimiento hacia adelante; en caso contrario, el coche se moverá hacia atrás.

## Dirección

Para calcular la dirección del vehículo me he basado en la Geometría de dirección de Ackermann [18].

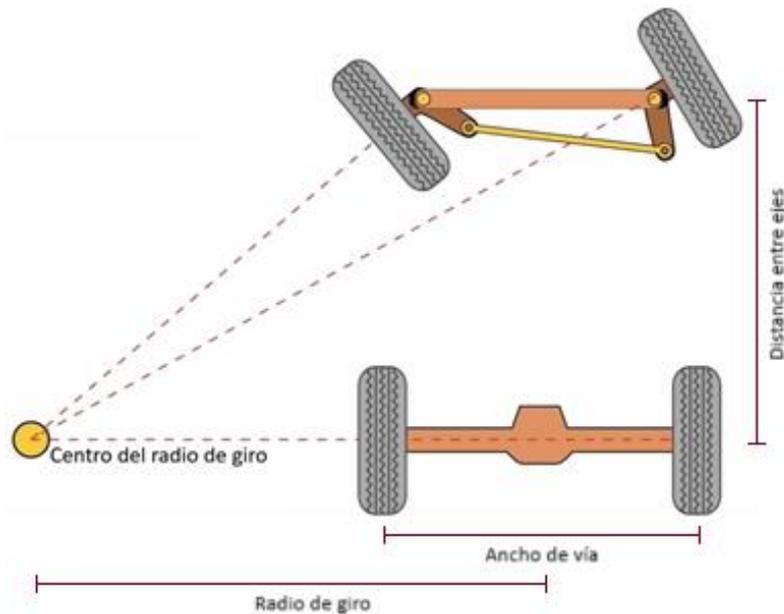


Figura 6. Geometría de dirección de Ackermann

Durante el recorrido de una curva, la rueda delantera exterior dibuja una trayectoria más amplia respecto a la rueda interior. La geometría de dirección de Ackermann es una configuración de los sistemas mecánicos de dirección de los coches cuyo fin es compensar el hecho de que las ruedas exteriores e interiores recorren circunferencias de diferentes radios cuando se traza una curva.

Para el cálculo teórico del ángulo de cada rueda se utilizan las siguientes fórmulas: [19]

$$\begin{aligned} \text{Ángulo de la rueda interior} \\ = \text{Distancia entre ejes} / (\text{Radio de giro} - (\text{Ancho de vía}/2)) \end{aligned} \quad (3)$$

$$\begin{aligned} \text{Ángulo de la rueda exterior} \\ = \text{Distancia entre ejes} / (\text{Radio de giro} + (\text{Ancho de vía}/2)) \end{aligned} \quad (4)$$

Donde:

- La distancia entre ejes es la distancia entre las ruedas delanteras y traseras.
- El radio de giro es la distancia entre el centro del ancho de vía y el centro del radio de giro.
- El ancho de vía es la distancia entre dos ruedas del mismo eje.

### 2.3.2 Implementación de una IA para los oponentes

Uno de los elementos cruciales para proporcionar una experiencia de juego desafiante en un videojuego de carreras es la implementación de la inteligencia artificial que

controla a los oponentes. En este caso, opté por implementar un enfoque basado en el seguimiento de una trayectoria definida por nodos a lo largo del circuito. Para ello tomé como base dos tutoriales. El primero de ellos, una IA para juegos de coches en Construct 2 [20], un motor de creación de videojuegos 2D para webs; y el segundo, una IA para oponentes de carreras en Unity [21].

## Aceleración

Para calcular la cantidad de aceleración que debe aplicar el coche para seguir la trayectoria de nodos se han seguido los siguientes pasos:

1. **Determinar el objetivo en la trayectoria:** En cada momento, el coche tiene un punto específico asignado, conocido como "nodo actual". Por ello, se calcula la dirección hacia dicho nodo, que es el punto que el coche debe alcanzar para mantenerse en la trayectoria.
2. **Ángulo de Dirección:** El coche compara la dirección en la que apunta con la dirección hacia el nodo actual. Si el ángulo entre estas direcciones es pequeño, como menos de 5 grados, se decide que el coche puede acelerar. Esto ayuda a asegurarse de que el coche solo acelere cuando esté bien alineado con la trayectoria.
3. **Decisión de Aceleración:** Si el ángulo es pequeño o si la velocidad actual del coche es baja, el coche acelera. Esto se hace para evitar que el coche se detenga cuando no está bien alineado con el nodo actual.
4. **Control en las Curvas:** Para evitar que el coche se estrelle contra el muro si entra a una curva con una velocidad demasiado alta, los nodos en la trayectoria se han colocado más cercanos entre sí en las curvas. Esto tiene un propósito clave: al calcular la distancia entre el último nodo alcanzado y el nodo actual, se puede detectar la cercanía a una curva. Si la distancia es reducida, la aceleración se reduce o incluso se frena, asegurando así que el coche entre en la curva con una velocidad segura y controlada.

## Dirección

La explicación sobre implementación del giro del coche se puede explicar en cuatro sencillos pasos:

5. **Determinar el objetivo en la trayectoria:** De igual manera que en el cálculo de la aceleración, se necesita conocer el nodo actual. Esto permite ajustar la dirección del coche para alinearse con ese punto y mantenerse en la trayectoria.
6. **Determinar la dirección:** Se calcula la dirección en la que se encuentra el nodo actual en relación con el coche. Esto ayuda al coche a entender si necesita girar a la izquierda o a la derecha para apuntar correctamente hacia el nodo.
7. **Controlar la magnitud de giro:** Una vez que se conoce la dirección hacia la que debe girar, se aplica una fuerza de giro que indica cuán "fuerte" debe girar el coche. Esta fuerza de giro influye en la rapidez con la que el coche cambia su dirección para acercarse al nodo actual.
8. **Suavizar el giro:** Por último, para evitar movimientos bruscos en la dirección del coche, se implementa un suavizado en el giro. Esto permite que el coche

ajuste su giro de manera gradual y más natural, evitando movimientos bruscos que puedan afectar negativamente la experiencia del jugador.

### Sensores de distancia

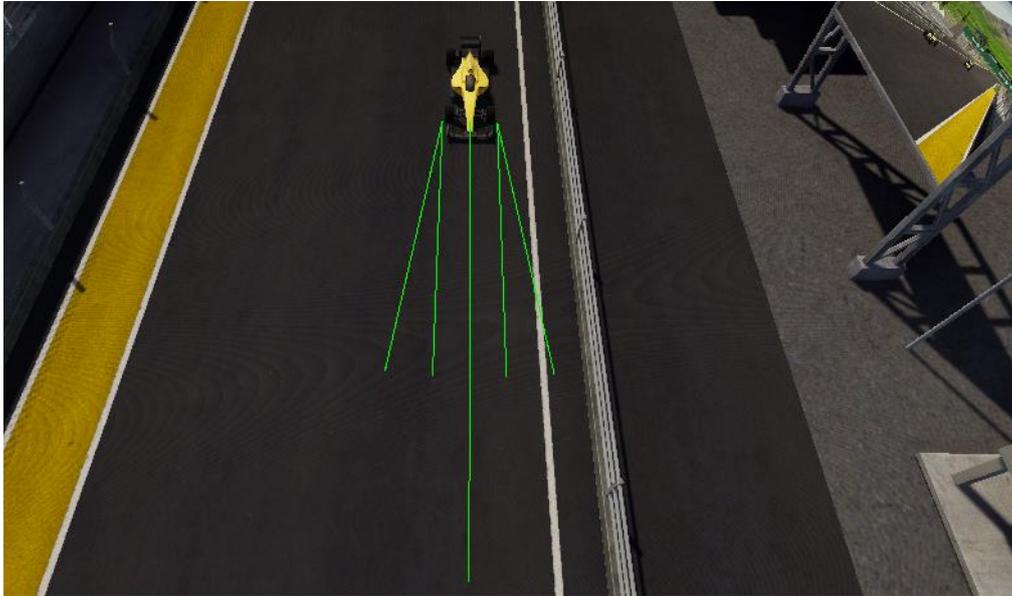


Figura 7. Disposición de sensores del coche

El coche de los oponentes está equipado con cinco sensores de distancia diseñados para la detección y evitación de obstáculos. La disposición de estos sensores se muestra en la 'Figura 6' (cabe destacar que estos sensores no son visibles durante el juego).

- **Sensor central:** Este sensor, al tener una longitud superior, es capaz de detectar obstáculos a mayor distancia, lo que proporciona al coche un mayor margen de tiempo para poder evitarlos. Además, este sensor permite generar un efecto de rebufo cuando detecta un coche delante a cierta distancia, lo que se traduce en un aumento de la velocidad máxima del coche para intentar adelantar al vehículo que lo precede.
- **Sensores frontales exteriores:** Se trata de dos sensores colocados en el borde frontal del coche. Si el sensor del borde izquierdo detecta un obstáculo, el coche girará a la derecha para intentar evitarlo. De igual manera, si el sensor del borde derecho detecta un obstáculo, el coche girará a la izquierda.
- **Sensores laterales en ángulo:** Se trata de dos sensores colocados también en el borde frontal del coche, pero formando un ángulo hacia el exterior del coche para tener más rango de detección. Funcionan de igual manera que los sensores frontales exteriores.

### 2.3.3 Sistema de puntos de control

En "Sprint Rush", he implementado un sistema de puntos de control a lo largo del circuito para determinar y calcular la posición relativa de cada coche en la carrera. Estos puntos de control se presentan como 'cajas' invisibles y sin colisión que ocupan

el ancho de la pista y son atravesados por el coche a medida que avanza en cada vuelta.

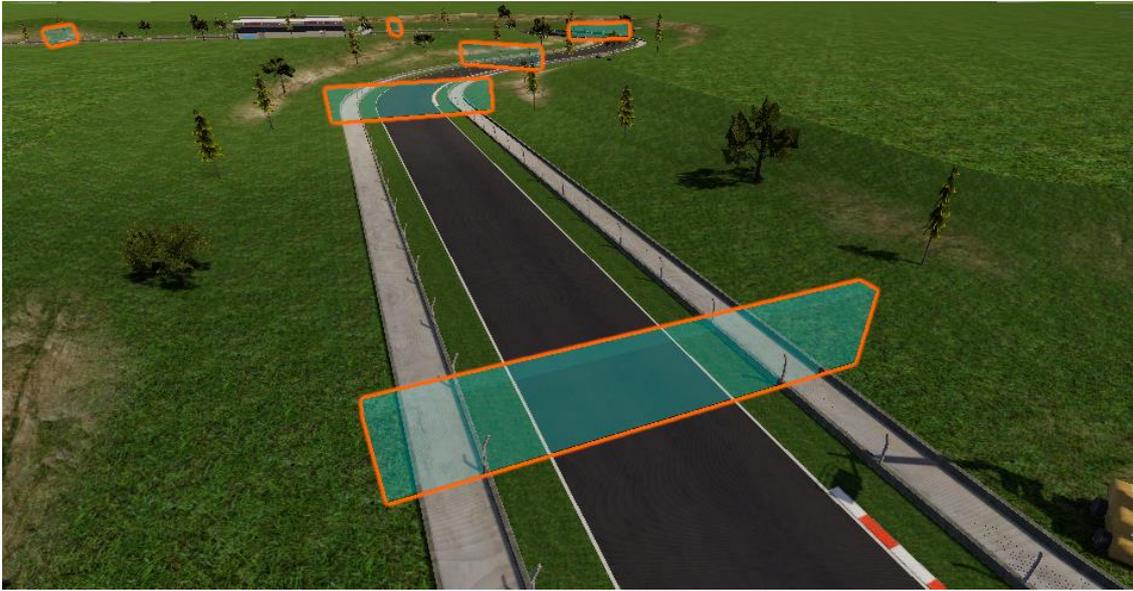


Figura 8. Distribución de distintos puntos de control en el circuito

El proceso de cálculo de la posición relativa de cada coche en carrera se basa en varios criterios:

- **Número de vueltas completadas:** En primer lugar, se compara el número de vueltas completadas por cada coche. Si un coche ha recorrido más vueltas que otro, se considera que está por delante en la carrera.
- **Número de puntos de control completados:** Si varios coches han completado la misma cantidad de vueltas, se procede a verificar cuántos puntos de control han sido superados en la vuelta actual. El coche que haya pasado por más puntos de control en la vuelta actual se coloca en una posición más avanzada.
- **Distancia al siguiente punto de control:** En caso de que dos coches hayan completado el mismo número de vueltas y el mismo número de puntos de control en la vuelta actual, se evalúa la distancia entre cada coche y el siguiente punto de control en la pista. El coche que se encuentre más cerca del siguiente punto de control se posiciona delante en la carrera.

### 2.3.4 Implementación de los modos de juego

El juego presenta dos modos de juego distintos para ofrecer a los jugadores diferentes experiencias al jugar: "Race" y "Practice". Para esta primera versión he generado un circuito utilizando el paquete del generador de circuitos de los 'assets' descargados.



Figura 9. Circuito generado con el paquete Road Track Generator

Para implementar los modos de juego, he utilizado un script que desempeña la función de 'Director de Carrera'. Si bien la versión actual del juego no admite la modificación del número de vueltas y rivales durante una carrera, este script permite la integración de la interfaz que inicialmente estaba prevista para permitir estas modificaciones. Estos ajustes podrán facilitar la personalización de la experiencia de juego al brindar a los jugadores mayor control sobre estos parámetros en futuras actualizaciones. También he implementado un temporizador o cuenta atrás de tres segundos que indica cuándo empieza la carrera.

### **Modo "Race" (Carrera rápida)**

En el modo "Race", los jugadores participarán en una carrera que consta de tres vueltas al circuito. En esta modalidad, competirán contra tres oponentes controlados por la IA. El objetivo es intentar superar a los rivales y alcanzar el primer puesto. En la tabla de resultados al final de la carrera,

### **Modo "Practice" (Práctica)**

El modo "Practice" está diseñado para permitir a los jugadores practicar y mejorar sus habilidades de conducción, ideal para familiarizarse con el comportamiento del coche y el circuito. Los jugadores pueden recorrer el circuito a su propio ritmo sin límite de vueltas. No hay competidores en este modo, lo que permite al jugador perfeccionar sus técnicas de conducción e intentar batir su propio récord de tiempo de vuelta.

Para lograr ganar a los oponentes y, sobre todo, mejorar el tiempo de vuelta en el circuito, el jugador deberá mantenerse dentro de la pista en todo momento, evitando pisar la hierba. Cuando un jugador se desvía de la pista y pisa la hierba, experimentará una ralentización en la velocidad del vehículo. Este efecto se implementa para reflejar las condiciones de la conducción en la vida real, donde el contacto con superficies no pavimentadas suele resultar en una pérdida de tracción y velocidad.

### 2.3.5 Interfaz de usuario

La interfaz de usuario del juego está compuesta por dos partes fundamentales: los menús y la interfaz utilizada durante las sesiones de carrera o práctica.

#### Menús

El menú principal proporciona a los jugadores las opciones necesarias para navegar por el juego. Aquí se encuentran los botones "Race" y "Practice" que permiten acceder a los modos de juego respectivos. También se incluyen opciones adicionales como el botón "Options", que conduce al menú de opciones, y "Exit" para salir del juego en cualquier momento.

Dentro del menú de opciones, el jugador puede ajustar configuraciones específicas mediante el botón "Settings". Este botón abre un menú donde puede modificar el volumen del juego, la calidad gráfica y el modo de pantalla completa. En este menú también se encuentra el botón "Controls", que proporciona información detallada sobre los controles disponibles tanto para teclado como para controladores. Asimismo, el botón "Back" permite regresar al menú principal en cualquier momento.

#### Interfaz en carrera o práctica

La interfaz en carrera o práctica proporciona información clave en tiempo real para mejorar la experiencia del jugador mientras está en el circuito. Los elementos esenciales de esta interfaz incluyen:

- **Número de vueltas completadas:** Indica cuántas vueltas ha completado el jugador con respecto al total de vueltas en la carrera.
- **Posición en carrera:** Muestra la posición actual del jugador en comparación con el total de coches en la carrera.
- **Tiempo total de carrera:** Muestra el tiempo total transcurrido desde el inicio de la carrera o de la práctica.
- **Tiempo de vuelta actual:** Indica el tiempo empleado en la vuelta actual.
- **Mejor tiempo de vuelta:** Muestra el tiempo más rápido logrado en una vuelta. En el modo "Practice", este tiempo se inicia con el mejor tiempo logrado en sesiones anteriores para que el jugador pueda comparar sus tiempos de vuelta actuales con su propio rendimiento anterior. En el modo "Race", el indicador de mejor tiempo de vuelta comienza vacío en cada carrera. Esto garantiza que el mejor tiempo registrado se refleje únicamente en la carrera actual.
- **Velocidad del coche:** Indica la velocidad actual del coche del jugador en kilómetros por hora.
- **Minimapa:** Proporciona una representación visual del circuito y la posición de los coches. El jugador está representado por una flecha roja, mientras que los rivales se muestran como círculos azules en el minimapa. Este minimapa estaba incluido en el paquete del generador de circuitos de los 'assets' descargados.

## 2.3.6 Audio y efectos visuales

### Música del menú principal

La música del menú principal la he creado usando el programa FL Studio Mobile. Me inspiré en la música de juegos arcade que suele ser rápida y enérgica, y le añadí un toque de estilo trance. La canción tiene un tempo constante de 150 bpm. Quería que evocara la sensación emocionante de los juegos arcade clásicos y la competencia de las carreras.

Incorporé varios samples del sonido del motor de un coche de Fórmula 1 para ayudar a los jugadores a identificar de inmediato la temática de carreras.

En cuanto a los elementos melódicos y el bajo, utilicé una variedad de sintetizadores para crear los diferentes sonidos electrónicos.

Para la percusión utilicé un kick 909 para un golpe potente, un clap para añadir ritmo, un sonido de güiro y una pandereta para crear un ambiente festivo, y un 'egg shaker' o gonzá para aportar un toque de vibración adicional.

Por último, decidí incluir un audio de fondo de una multitud emocionada gritando y silbando en las gradas de un circuito de carreras para aumentar la inmersión.

### Sonido del público

El efecto de sonido del público, que imita los murmullos y la emoción de la multitud en las gradas, agrega un ambiente competitivo al entorno de la pista. Este efecto de sonido proviene del paquete del generador de circuitos de los 'assets' descargados.

### Sonidos del coche

- **Sonido de arranque del motor, sonido del motor en ralentí, sonido del motor en marcha y sonido del motor en marcha atrás:** Estos sonidos los obtuve del repositorio de Github de Nanousis [22]. Además, me ayudé de los tutoriales en su canal de YouTube para la programación del sonido del motor.
- **Sonido de derrape:** El efecto de sonido de derrape lo extraje de la colección de 'Standard Assets' de Unity [23], y es usado cuando el coche pierde una tracción considerable en alguna de sus ruedas.
- **Sonido de coche sobre hierba:** El sonido al pisar la hierba se adquirió de la página web pixabay.com [24], usado cuando el coche se encuentra en marcha sobre una superficie de hierba.

### Marca de la rueda

Para crear el efecto de una "marca de la rueda" he usado el Trail Renderer de Unity. El Trail Renderer representa un rastro detrás de un objeto en movimiento. Aparece en las ruedas traseras cuando se pierde tracción, simulando el rastro que dejaría una rueda derrapando sobre el asfalto.

## Partícula de Polvo

Creé una partícula de polvo utilizando el sistema de partículas de Unity. Esto simula el efecto de polvo levantado cuando el coche pisa la hierba, mejorando la inmersión y la autenticidad visual del juego.

## 2.4 Problemas de implementación

A lo largo del desarrollo de este proyecto, me he enfrentado a diversos desafíos al implementar diferentes aspectos del videojuego. Para superar estos obstáculos, he tenido que recurrir a la búsqueda de información y considerar alternativas.

### ML-Agents [25]

El principal problema que encontré estuvo relacionado con el uso de ML-Agents. Mi objetivo principal era emplear esta extensión de Unity para implementar la inteligencia artificial de los oponentes, aprovechando su capacidad para entrenar agentes de aprendizaje por refuerzo. Siendo la primera vez que utilizaba esta herramienta, logré configurarla y poner en marcha el entrenamiento de los coches en el circuito sin mucha dificultad. Sin embargo, me encontré con dificultades debido a la lentitud del proceso de aprendizaje con esta herramienta y a la dificultad para poder evaluar la correcta configuración de entrenamiento hasta que hubiera progresado en gran medida. Después de horas de entrenar al agente y lograr que recorriera el circuito al completo, la curva de aprendizaje caía considerablemente, como se observa en la 'Figura 7', lo que resultaba en un mal funcionamiento del agente.



Figura 10. Curva de recompensa acumulada del agente

Estos problemas me llevaron a invertir una cantidad de tiempo mayor a la que esperaba en la implementación de la IA, lo que finalmente me llevó a buscar una alternativa para los oponentes: la técnica de seguimiento de un trazado de nodos a lo largo del circuito.

## WheelCollider de Unity

La tracción de las ruedas en los componentes WheelCollider de Unity viene definida por una curva de tracción llamada "WheelFrictionCurve" [26].

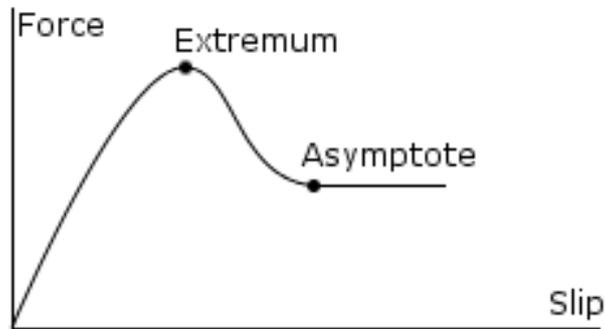


Figura 11. Curva de fricción usada por el WheelCollider

Un apunte valioso proviene del usuario 'Edy' [27], reconocido en el foro de Unity, quien destaca que, a pesar de que los valores de deslizamiento de las ruedas se reportan en la propiedad WheelHit, que proporciona información sobre el contacto de la rueda, la documentación oficial de Unity no detalla las especificaciones sobre cómo se calcula esta fuerza. En otras palabras, no se especifican las magnitudes que se utilizan para establecer los valores de estas propiedades de deslizamiento.

Es relevante mencionar que 'Edy' es un destacado desarrollador español certificado por Unity. Él es conocido por ser el creador del 'asset' de Unity 'Vehicle Physics Pro' [28], un modelo de simulación dinámica de vehículos de categoría AAA (clasificación utilizada para los videojuegos con un desarrollo de alto presupuesto) que puede equipararse a las principales franquicias de simulación de vehículos.

A pesar de todo lo expuesto, pude obtener una comprensión más clara sobre el funcionamiento de esta curva gracias a un video que encontré [29].

## Capítulo 3. Testeo y resolución de errores

Durante la fase de testeo y resolución de errores del videojuego, dediqué un tiempo considerable a jugar repetidamente con el propósito de identificar y solucionar errores. A medida que avanzaba, descubrí varios errores que tuve que solucionar:

El primer inconveniente que detecté ocurría cuando el coche estaba parado con el motor en marcha, y observé que se desplazaba lateralmente de un lado a otro debido a la fricción con el suelo. Para resolver este problema, ajusté la resistencia angular del componente RigidBody del coche (el cual pone el movimiento de un objeto bajo el control del motor de físicas de Unity), que controla la velocidad de rotación de un objeto, cuando el coche se encontraba en esta situación específica.

Aunque es complicado que un coche se salga del circuito debido a su diseño, consideré escenarios en los que esto podría ocurrir debido a colisiones fuertes o a errores no previstos. Para estos casos, implementé una función de reaparición del coche. Tanto el jugador como los oponentes pueden reaparecer en el último punto de control alcanzado. El jugador puede hacer reaparecer su coche utilizando el botón de reaparición que se muestra en la pantalla de controles, mientras que los oponentes reaparecen si sus ruedas no están en contacto con el suelo o si permanecen inmóviles debido a un accidente, durante más de 5 segundos.

Observé también un problema visual relacionado con la forma de los componentes WheelCollider de Unity, los cuales son circulares y no se ajustan al ancho de las ruedas. Esto resultaba en que las mallas de las ruedas de dos coches se superpusieran entre sí. Para solucionar este efecto, añadí colisionadores adicionales en la parte exterior de las ruedas, asegurándome de que no se atravesaran visualmente.

El último error que tuve que solucionar se encontraba en la actualización de la tabla de resultados al final de una carrera. Noté que los tiempos no se actualizaban correctamente cuando un coche cruzaba la meta al final de la carrera. Después de varias pruebas y correcciones, trabajé en la solución de este problema para garantizar que los tiempos se reflejaran correctamente en la tabla de resultados.

Estas correcciones ayudaron a que el juego se viese más pulido y sin errores inesperados que puedan influir negativamente en la experiencia de los jugadores.

## Capítulo 4. Manuales

### 4.1 Instalación

Para facilitar la instalación del juego, he creado un archivo instalador denominado "Sprint Rush Setup (x86).exe" utilizando el programa Inno Setup Compiler. Este instalador simplifica el proceso de compartir el juego, garantizando una instalación fluida para los usuarios.

Una vez ejecutamos el archivo instalador, nos aparecerá la siguiente ventana:

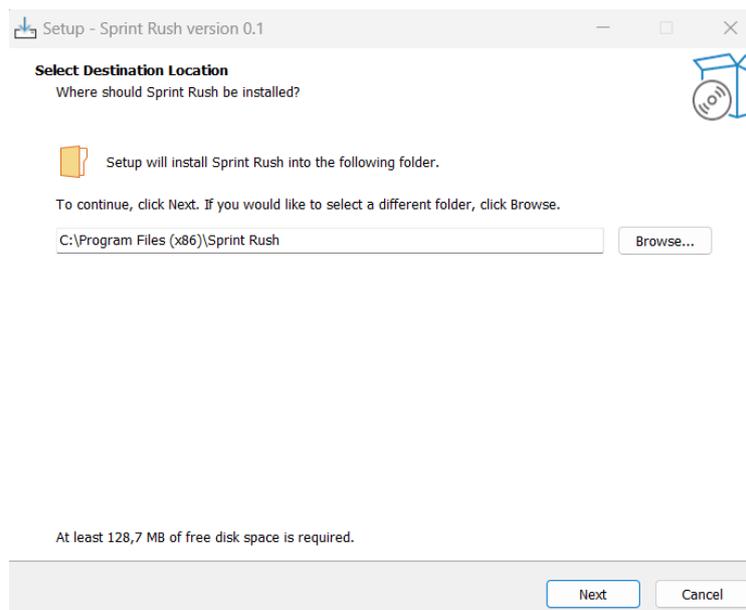


Figura 12. Paso 1 - Selección de Ruta de Instalación

En este apartado, el usuario puede elegir la ruta de instalación del juego o dejar la ruta por defecto. Para continuar, pulsamos el botón "Next" y nos aparecerá la siguiente ventana:

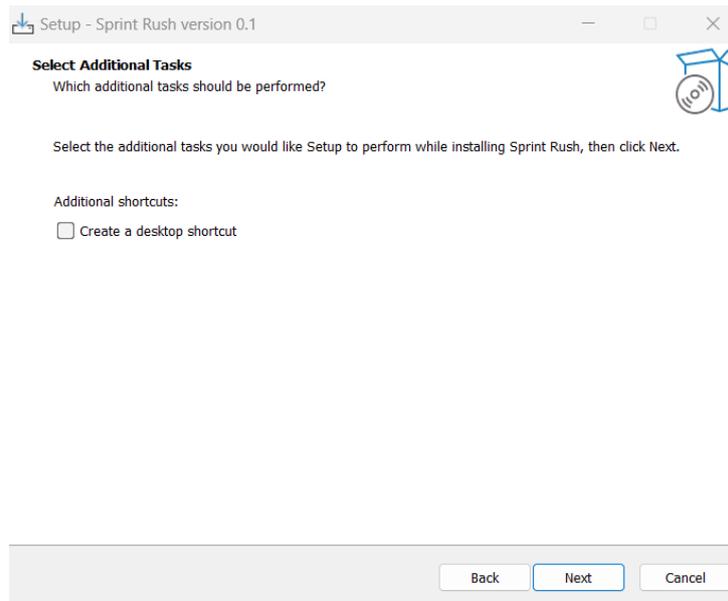


Figura 13. Paso 2 - Creación de Acceso Directo

En este apartado, el usuario decide si quiere crear un acceso directo en el escritorio marcando la casilla "Create a desktop shortcut". Luego, pulsamos el botón "Next" para pasar al último paso:

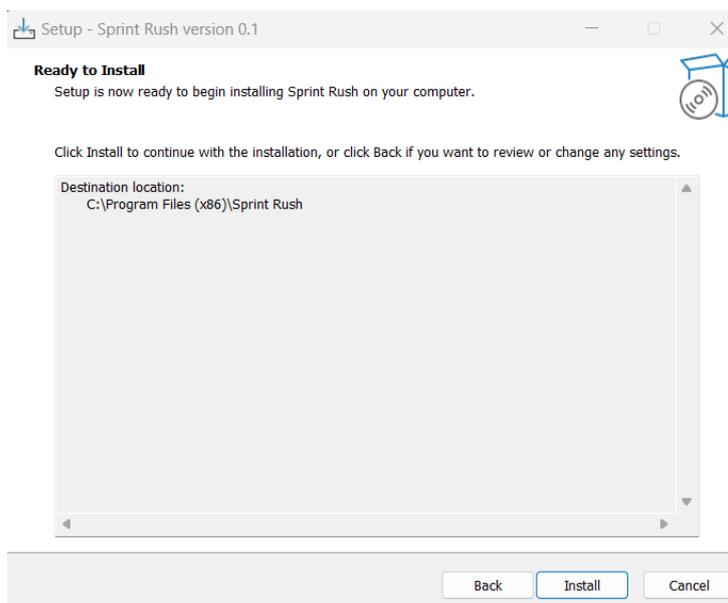


Figura 14. Paso 3 - Proceso de Instalación

En esta última ventana del instalador el usuario puede verificar cuál es la ruta de instalación. El último paso es pulsar el botón "Install" para llevar a cabo la instalación del juego en la ruta específica indicada.

Para abrir el proyecto en Unity, puedes seguir los siguientes pasos:

1. Ve al repositorio de GitHub. [30]
2. Haz clic en el botón verde "Code" en la parte superior derecha del repositorio.

3. Selecciona "Download ZIP" para descargar el repositorio como un archivo ZIP en tu ordenador.
4. Descomprime el archivo ZIP en la ubicación deseada.

Ahora que tienes el proyecto descargado:

1. Abre Unity Hub en tu ordenador.
2. Haz clic en "Open" para abrir un nuevo proyecto.
3. Navega hasta la carpeta donde descomprimiste el repositorio descargado.
4. Selecciona la carpeta del proyecto con nombre "SprintRush-main".
5. Unity Hub cargará el proyecto y lo mostrará en la lista de proyectos.
6. Haz clic en el proyecto para abrirlo en Unity.

## 4.2 Guía de uso

El juego se presenta en inglés, ya que este idioma se considera universal y muchos desarrolladores crean sus juegos con el objetivo de llegar a un público más amplio.

Al ejecutar el juego, el usuario se encontrará con la primera escena: el menú principal.



Figura 15. Menú principal del juego

Aquí, se muestran cuatro botones:

- "Race": Accede al modo de juego "Carrera Rápida".
- "Practice": Accede al modo de juego "Práctica".
- "Options": Abre el menú de opciones.
- "Exit": Sirve para salir del juego.

Dentro del menú de opciones, hay tres botones adicionales:

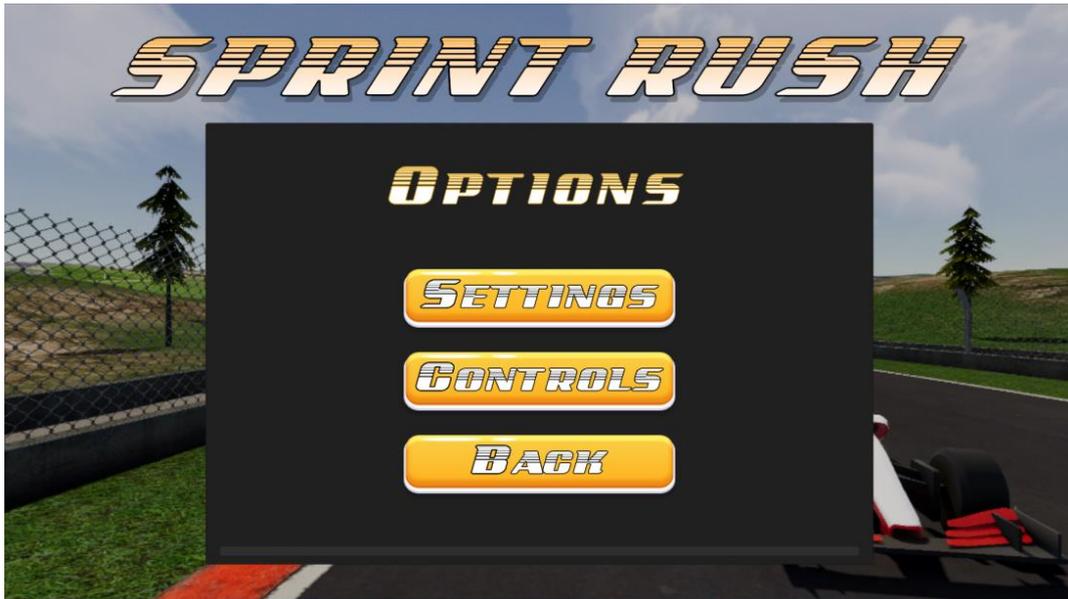


Figura 16. Menú de opciones del juego

- "Settings": Abre el menú de configuración del juego.
- "Controls": Muestra los controles del juego.
- "Back": Regresa al menú principal.

En el menú de configuración, se ofrecen tres opciones:



Figura 17. Menú de configuración del juego

- "Fullscreen": Una casilla de verificación para activar o desactivar el modo "Pantalla Completa".
- Control de Volumen: Un control deslizante para ajustar el volumen general del juego.

- "Graphics": Un menú desplegable para elegir la calidad gráfica del juego (Baja, Media, Alta).
- "Back": Regresa al menú de opciones.

Aquí se muestra el desplegable para elegir la calidad gráfica del juego:

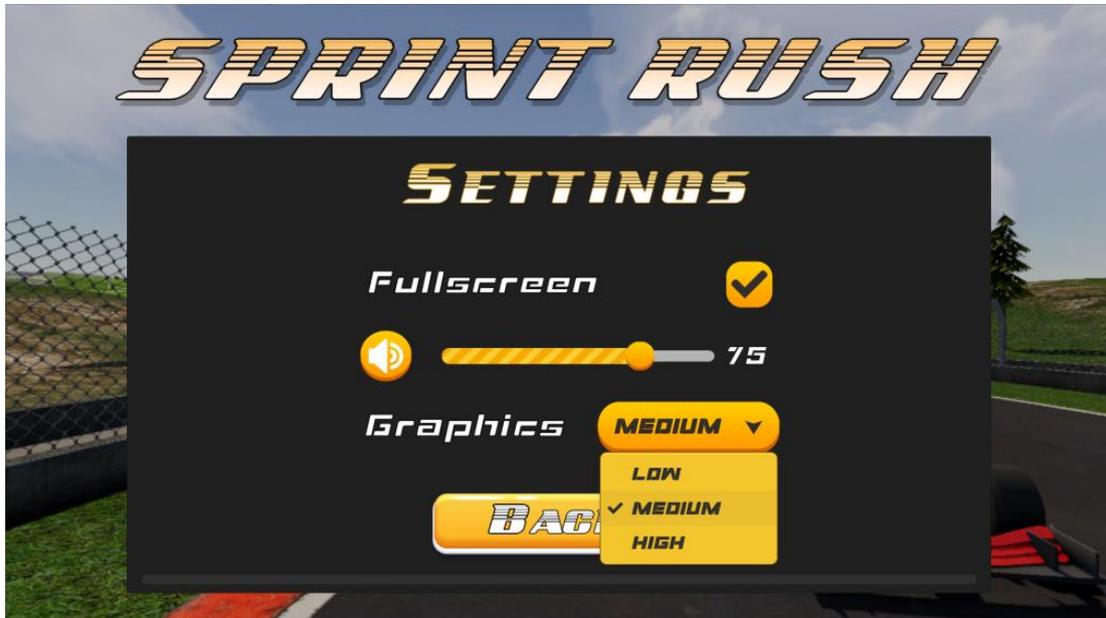


Figura 18. Menú desplegable para elegir la calidad gráfica del juego

En el menú de controles, se proporcionan los detalles de los controles del juego tanto para el teclado como para el controlador.



Figura 19. Menú de controles del juego

- "Back": Regresa al menú de opciones.

Dentro de los modos de juego "Race" y "Practice", se puede acceder al menú de pausa utilizando la tecla "Esc" o el botón "Options" del controlador.



Figura 20. Menú de pausa del juego

Desde este menú, el usuario puede:

- Reanudar la carrera o práctica.
- Acceder a los menús de configuración y controles.
- Salir y regresar al menú principal.

Antes de que comience una carrera o práctica, se muestra una cuenta regresiva de 3 segundos indicando el inicio cuando aparece la palabra "GO!".



Figura 21. Cuenta atrás al inicio de una carrera

A continuación, se presentan dos imágenes que muestran cómo se ve el juego y la interfaz durante una carrera y una práctica. La diferencia en la interfaz solo está en el número total de vueltas, ya que en la práctica este número es ilimitado o infinito.



Figura 22. Interfaz en el modo "Race"



Figura 23. Interfaz en modo "Practice"

Si el jugador sufre algún error inesperado o se sale del circuito, puede restablecer la posición de su coche en el último punto de control alcanzado, manteniendo el botón de reparación, que se muestra en la pantalla de controles, hasta llenar la barra circular que aparece en pantalla.



Figura 24. Barra de restablecimiento del coche

Al finalizar una carrera, se presenta una tabla de resultados con la opción "Exit" para regresar al menú principal.

POS	DRIVER	TOTAL TIME	BEST LAP
1	M. Schmitt	02:29.314	00:47.639
2	Player	02:28.566	00:47.199
3	F. Muller	02:30.973	00:47.940
4	S. Wagner	02:33.446	00:48.080

Figura 25. Tabla de resultados mostrada al final de una carrera

## Capítulo 5. Evaluación

### 5.1 Cumplimiento de objetivos

A lo largo de este proyecto, he logrado alcanzar la mayoría de los objetivos que me propuse inicialmente. Sin embargo, hubo algunos aspectos adicionales que me habría gustado incorporar, pero que, lamentablemente, quedaron fuera del alcance debido a ciertas circunstancias.

Uno de esos objetivos pendientes era la creación de una interfaz que permitiera a los jugadores seleccionar el número de vueltas y la cantidad de oponentes antes de comenzar una carrera o práctica. También había planeado la posibilidad de añadir más opciones, como la elección entre varios coches o circuitos, para brindar una experiencia de juego más diversa y personalizable. Estas características adicionales habrían enriquecido notablemente la experiencia del jugador.

El principal obstáculo que impidió la realización de estos objetivos adicionales fue la complejidad que encontré en el proceso de aprendizaje con ML-Agents. Como mencioné en el capítulo 2.4, el entrenamiento de los agentes no fue como esperaba y consumió una cantidad considerable de tiempo durante el desarrollo del juego, lo que afectó directamente mi capacidad para implementar características secundarias.

A pesar de estos desafíos, siento que he logrado alcanzar con éxito el objetivo central de este proyecto: la creación de un videojuego con inteligencia artificial para los oponentes. Además, estoy particularmente satisfecho con la creación de un controlador de coche que es fácil y divertido de manejar. Crear una experiencia de juego en la que los jugadores puedan disfrutar de un control intuitivo y entretenido del coche era algo que creo que es esencial para un videojuego, y considero que este objetivo también se ha cumplido de manera satisfactoria.

Si bien quedaron pendientes algunas características que habrían mejorado la experiencia de esta primera versión del juego, este proyecto representa un paso sólido hacia la creación de un videojuego de carreras estilo arcade. A pesar de que esta versión inicial del juego tiene un enfoque más limitado, sirve como una demostración jugable de la experiencia que se podría ofrecer y estoy abierto a la posibilidad de actualizar y mejorar este proyecto en el futuro, en caso de decidir continuar su desarrollo.

### 5.2 Conclusiones

Este proyecto de desarrollo de un videojuego de carreras estilo arcade me ha proporcionado valiosas perspectivas sobre la creación de experiencias de juego interactivas he podido extraer varias conclusiones.

En primer lugar, la creación de un videojuego de carreras es una tarea desafiante pero gratificante. La atención a los detalles en la física de los vehículos, la inteligencia artificial de los oponentes y la jugabilidad general es esencial para lograr una experiencia de juego entretenida y atractiva.

Además, creo que los videojuegos de carreras estilo arcade tienen el potencial de proporcionar entretenimiento a una amplia audiencia de jugadores. Su accesibilidad y mecánicas sencillas pueden atraer tanto a jugadores casuales como a entusiastas de este género.

En cuanto a la posibilidad de una versión completa de este juego, se abre una puerta interesante. Si bien este proyecto se ha centrado en una versión demostrativa, los comentarios de los jugadores podrían ser fundamentales para determinar la viabilidad de una versión comercial. Esto incluiría la adición de características fundamentales hoy en día como la selección de vehículos, pistas adicionales y opciones de personalización.

En una industria en constante evolución y cada vez más exigente, es importante recordar que productos como este, con un enfoque más indie o casual, desempeñan un papel crucial para mantener el crecimiento y la diversidad de la industria. Estas experiencias pueden ofrecer a los jugadores momentos de diversión y entretenimiento, al mismo tiempo que sirven como inspiración para futuras innovaciones en el mundo de los videojuegos de mayor presupuesto.

### 5.3 Aportaciones personales

Durante el proceso de desarrollo de esta primera versión jugable de mi videojuego de carreras, este proyecto me ha aportado una serie de conocimientos que considero muy valiosos para mi formación y crecimiento.

#### **Gestión del tiempo**

A nivel personal, ha sido todo un reto tener que desarrollar todo por mi mismo y sin ayudas. A diferencia de los proyectos anteriores, en los que trabajaba en equipo, esta vez tuve que enfrentar el desafío de planificar y administrar mi tiempo de manera independiente. Todo esto me fue posible gracias al uso de la metodología “Scrum” aprendida en el Grado de Tecnologías Interactivas.

#### **Experiencia en Unity**

Uno de los aspectos más notables de este proyecto ha sido la inmersión en la plataforma Unity. A lo largo del proceso, he ganado un profundo conocimiento de Unity, sus herramientas y su interfaz. Debido a que en el proyecto de videojuegos del grado me dediqué más a desarrollar la parte artística, como el modelado 3D, esta vez he decidido usar ‘assets’ descargados para este apartado y me he centrado más en la parte de programación, configuración y estructura del videojuego.

#### **Desarrollo de videojuegos**

Con este proyecto he querido sumergirme en el desarrollo de videojuegos, específicamente en el género de carreras. Mi interés en los videojuegos, especialmente en este género, siempre ha sido evidente, y este proyecto me ha brindado la posibilidad de adentrarme en la realización de un videojuego propio, marcando así un primer paso en esta dirección.

A través de la creación de este juego, he aprendido los principios fundamentales de diseño de juegos, la compleja lógica detrás de la física de los vehículos y cómo crear un sistema de juego interactivo y entretenido. También he tenido que abordar desafíos relacionados con la inteligencia artificial de los oponentes y la implementación de las demás mecánicas de juego.

### **Programación**

La programación ha sido una parte esencial de este proyecto. He tenido que escribir código para controlar el comportamiento de los vehículos, gestionar las mecánicas del juego, implementar la inteligencia artificial de los oponentes y crear interfaces de usuario.

En resumen, el desarrollo de este proyecto ha representado un paso significativo en mi formación como desarrollador. Aunque ha estado marcado por desafíos y obstáculos, estos también han servido para adquirir experiencia y conocimientos que considero valiosos.

## 6. Referencias

- [1] Mena. (2018, 13 noviembre). *ARCADE: Ultimate Vehicles Pack - Low Poly Cars | 3D Vehicles | Unity Asset Store*. Unity Asset Store. <https://assetstore.unity.com/packages/3d/vehicles/arcade-ultimate-vehicles-pack-low-poly-cars-132975>. [Último acceso: Junio 2023].
- [2] MasterPixel3D. (2018, 19 marzo). *Race Track Generator | 3D Roadways | Unity Asset Store*. Unity Asset Store. <https://assetstore.unity.com/packages/3d/environments/roadways/race-track-generator-113050>. [Último acceso: Junio 2023].
- [3] Kovalchuk, I., [Kartlnnka]. (2022, 7 febrero). *Buttons set | 2D GUI | Unity Asset Store*. Unity Asset Store. <https://assetstore.unity.com/packages/2d/gui/buttons-set-211824>. [Último acceso: Agosto 2023].
- [4] Teja, K., [Wecreatype]. (2015, 16 junio). *Racing Fonts | FontSpace*. fontspace. <https://www.fontspace.com/category/racing>. [Último acceso: Agosto 2023].
- [5] Universitat Politècnica de València. (2022, 17 octubre). *RiuNet Repositorio UPV. RiuNet*. <https://riunet.upv.es/>. [Último acceso: Agosto 2023].
- [6] Sánchez, A. (2021). *D.A.N.G.E.R. Disaster Prevention: videojuego educativo para la gestión de emergencias en Unity 3D: interfaz de usuario, gestión y tratamiento de la información*. Universitat Politècnica de València. <https://riunet.upv.es/handle/10251/174398>. [Último acceso: Agosto 2023].
- [7] Gómez, J. F. (2021). *Desarrollo de un videojuego en Unity: implementación de las mecánicas de juego*. Universitat Politècnica de València. <https://riunet.upv.es/handle/10251/173382>. [Último acceso: Agosto 2023].
- [8] Nanousis Development. (s. f.). *Nanousis Development [Canal de YouTube]*. YouTube. <https://www.youtube.com/@nanousis>. [Último acceso: Junio 2023].
- [9] Pablos Lab. (s. f.). *Pablos Lab [Canal de YouTube]*. YouTube. <https://www.youtube.com/@pabloslab>. [Último acceso: Junio 2023].
- [10] BlinkAChu. (2019, 5 mayo). *Unity Car Physics - Lesson 2 - Ackermann Steering [Vídeo]*. YouTube. <https://www.youtube.com/watch?v=ZwMa9g7lvT8>. [Último acceso: Junio 2023].
- [11] Code Monkey. (2020, 23 diciembre). *Simple checkpoint system in unity [Vídeo]*. YouTube. <https://www.youtube.com/watch?v=IOYNg6v9sfc>. [Último acceso: Julio 2023].
- [12] Imphenzia. (2020, 18 mayo). *Making of Low Poly Racing - EP 5 - Checkpoints and Lap Times - Unity [Vídeo]*. YouTube. <https://www.youtube.com/watch?v=h-sEtELIn9k>. [Último acceso: Julio 2023].

- [13] *Technologies, U.* (2018, 19 marzo). *Plataforma de desarrollo en tiempo real de Unity | Motor de VR, AR, 3D y 2D.* Unity. <https://unity.com/es>. [Último acceso: Agosto 2023].
- [14] *Microsoft.* (2022, 23 diciembre). *Visual Studio: IDE y editor de código para desarrolladores de software y teams.* Visual Studio. <https://visualstudio.microsoft.com/es/>. [Último acceso: Agosto 2023].
- [15] *Image-Line.* (2017, 1 marzo). *FL Studio Mobile.* <https://www.image-line.com/fl-studio-learning/fl-studio-online-manual/html/plugins/FL%20Studio%20Mobile.htm>. [Último acceso: Agosto 2023].
- [16] *Technologies, U.* (s. f.-b). *Wheel Collider - Unity Manual.* <https://docs.unity3d.com/es/2018.4/Manual/class-WheelCollider.html>. [Último acceso: Junio 2023].
- [17] *Wittber, S.* (2010, 7 diciembre). *Unity3D WheelCollider and motorTorque.* Entity Crisis. <http://entitycrisis.blogspot.com/2010/12/unity3d-wheelcollider-and-motortorque.html>. [Último acceso: Julio 2023].
- [18] *Colaboradores de Wikipedia.* (2022, 9 enero). *Geometría de Ackermann.* Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Geometr%C3%ADa\\_de\\_Ackermann](https://es.wikipedia.org/wiki/Geometr%C3%ADa_de_Ackermann). [Último acceso: Julio 2023].
- [19] *International Research Journal of Engineering and Technology (IRJET).* (2021, 5 mayo). *Anti-Ackermann Steering System of Formula Student car.* <https://www.irjet.net/archives/V8/i5/IRJET-V8I5648.pdf>. [Último acceso: Agosto 2023].
- [20] *Ixel Games.* (2017, 26 enero). *Tutorial Construct 2. Inteligencia artificial ( AI ) para juego de coches.* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=WWCCakeHmAQ>. [Último acceso: Agosto 2023].
- [21] *Pablos Lab.* (2020, 30 marzo). *Unity Tutorial - AI-System(AI-Controller).* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=Z7OzJA2B0P0>. [Último acceso: Julio 2023].
- [22] *Nanousis.* (2022, 7 diciembre). *GitHub - NanoUsis/EngineSound.* GitHub. <https://github.com/Nanousis/EngineSound>. [Último acceso: Julio 2023].
- [23] *Technologies, U.* (2018, 5 febrero). *Unity - Manual: Assets Estándar.* Unity Documentation. <https://docs.unity3d.com/es/530/Manual/HOWTO-InstallStandardAssets.html>. [Último acceso: Agosto 2023].
- [24] *Pixabay.* (2021, 2 septiembre). *90,000+ free sound effects for download - Pixabay - Pixabay.* <https://pixabay.com/sound-effects/>. [Último acceso: Agosto 2023].
- [25] *Technologies, U.* (2020, 11 mayo). *Agentes de aprendizaje automático.* Unity. <https://unity.com/es/products/machine-learning-agents>. [Último acceso: Julio 2023].

[26] Technologies, U. (2023, 25 agosto). *Unity - Scripting API: WheelFrictionCurve*. <https://docs.unity3d.com/ScriptReference/WheelFrictionCurve.html>. [Último acceso: Agosto 2023].

[27] García, A., [Edy]. (2015, 8 septiembre). *Wheel Friction Curve* [Publicación en un foro online]. Mensaje publicado en <https://forum.unity.com/threads/wheel-friction-curve.353421/>. [Último acceso: Junio 2023].

[28] García, A., [Edy]. (2016, 21 septiembre). *Vehicle Physics Pro*. <https://vehiclephysics.com/>. [Último acceso: Julio 2023].

[29] Nanousis Development. (2022, 2 noviembre). *Wheel colliders explained! Unity + GitHub link!* [Vídeo]. YouTube. [https://www.youtube.com/watch?v=c-yhZwXSx\\_c](https://www.youtube.com/watch?v=c-yhZwXSx_c). [Último acceso: Junio 2023].

[30] Campins, M. (2023, 2 septiembre). *GitHub - Campiins/SprintRush: Proyecto en unity de un juego de carreras para mi TFG*. GitHub. <https://github.com/campiins/SprintRush>. [Último acceso: Septiembre 2023].