A close-up of a logo

Description automatically generated

**Proyecto Final Paradigmas**

**Ignacio Viadero y Pablo Díaz**

**Paradigmas y Técnicas de Programación**

**3º Grado en Ingeniería Matemática e Inteligencia Artificial**

# Índice

[Introducción 3](#_Toc177138896)

[Problemas encontrados 4](#_Toc177138897)

[Resultados 5](#_Toc177138898)

[Conclusión 6](#_Toc177138899)

Imagen que contiene lego, interior, juguete, edificio

Descripción generada automáticamente

# Introducción

El proyecto "Taxi Driver" consiste en el desarrollo de un videojuego de simulador de conducción diseñado como parte de una asignatura académica. Este juego se desarrolla en el motor Unity y utiliza el lenguaje de programación C#, aprovechando herramientas específicas de Unity como prefabs, NavMesh, colisiones, y físicas avanzadas para crear una experiencia dinámica y desafiante para los jugadores. El objetivo del proyecto es construir un juego que sea funcional, atractivo y que permita al jugador vivir la emoción de una persecución policiaca mientras demuestra sus habilidades de conducción y estrategia.

La propuesta de "Taxi Driver" se aparta del modelo proporcionado como ejemplo en la asignatura, introduciendo una lógica de juego completamente diferente. En lugar de controlar un taxi que recoge y deja clientes mientras es perseguido por la policía si excede el límite de velocidad, en este proyecto el jugador controla un taxi que ya comienza siendo perseguido por la policía. El objetivo principal es sobrevivir el mayor tiempo posible sin ser atrapado, recorriendo un mapa lleno de obstáculos y evitando colisiones con edificios u otros elementos del entorno.

El juego tiene varios niveles de dificultad que influyen directamente en la velocidad de la policía que persigue al jugador. El jugador comienza con 100 puntos de vida, que se reducen en 20 unidades cada vez que el taxi colisiona con un obstáculo o edificio. El juego termina cuando la policía atrapa al taxi o cuando la vida del jugador llega a cero, lo que motiva a los jugadores a combinar velocidad, precisión y estrategia para superar el reto.

Desde el punto de vista técnico, el proyecto incluye la implementación de elementos clave como la sincronización de las ruedas del coche con su movimiento general, la configuración de una superficie de navegación (NavMesh) para el comportamiento de la policía, y el manejo de colisiones a través del método OnCollisionEnter. Además, se diseñaron varios prefabs para representar a los personajes y elementos del entorno, cada uno con componentes específicos como colliders, rigidbodies y scripts personalizados en C#.

En términos de diseño, se creó una arquitectura básica del juego en UML para representar los elementos clave y su interacción. Este proyecto se ha documentado en un repositorio de GitHub, siguiendo prácticas de desarrollo ágil mediante historias de usuario, análisis de requisitos y la asignación de tareas. El repositorio en el que se ha guardado nuestro proyecto es:

El desarrollo del videojuego no solo ha permitido explorar las herramientas que ofrece Unity, sino también aplicar conceptos avanzados de programación, diseño de videojuegos y trabajo en equipo, enfrentándonos a diversos problemas técnicos que detallamos a continuación.

# Problemas Encontrados

# **Configuración del Box Collider para el Taxi y la Policía**

# Una de las primeras dificultades surgió al intentar ajustar el box collider de los prefabs del taxi y la policía. Estos prefabs tienen geometrías complejas que no encajan perfectamente en un collider estándar. La precisión en esta configuración era crucial para evitar colisiones erróneas que pudieran impactar negativamente la experiencia del jugador. Fue necesario probar diferentes tipos de colliders, ajustando manualmente sus dimensiones para que coincidieran de manera óptima con las formas de los prefabs, manteniendo el equilibrio entre realismo y rendimiento.

# **Movimiento del Coche y Sincronización Visual**

El taxi es un prefab compuesto por múltiples partes, incluidas las ruedas y el cuerpo principal. Esto complicó la implementación de su movimiento, ya que no solo debía trasladarse por el mapa, sino también mostrar un comportamiento visual coherente. Fue necesario sincronizar las físicas del coche con el giro de las ruedas mediante un mesh renderer, asegurando que el movimiento fuera fluido y realista. Esto implicó trabajar con scripts que controlaban tanto las animaciones visuales como la lógica del movimiento, garantizando que los elementos estuvieran perfectamente coordinados.

# **Configuración de la NavMesh Surface**

El desarrollo de la NavMesh Surface para definir las áreas navegables del mapa fue otro reto. Este sistema permite que la policía persiga al taxi evitando obstáculos como edificios y otros elementos del entorno. La complejidad residía en delimitar correctamente las áreas transitables sin comprometer la jugabilidad, ajustando parámetros como la resolución de la malla y los márgenes alrededor de los objetos. Cualquier error en esta configuración podía causar comportamientos no deseados, como que la policía atravesara objetos o se quedara atascada en zonas intransitables.

# **Ajuste del Radio, Base Offset y Altura del NavMesh AI**

Una vez configurada la NavMesh Surface, fue necesario ajustar los parámetros del NavMesh Agent que controla el movimiento de la policía. La configuración del radio, base offset y altura era esencial para que la policía interactuara correctamente con el entorno y con el taxi. Un radio demasiado grande hacía que las colisiones fueran imprecisas, mientras que un base offset incorrecto provocaba que el agente pareciera flotar o hundirse en el suelo. Ajustar estos valores requirió numerosas pruebas y ajustes finos para garantizar un comportamiento natural y preciso.

# **Implementación del Método OnCollisionEnter**

La implementación del método OnCollisionEnter, encargado de detectar las colisiones entre el taxi y la policía, presentó múltiples desafíos. Este método debía ser capaz de identificar correctamente las colisiones relevantes y, en caso de que el taxi fuera atrapado, invocar la función GameOver. Fue necesario optimizar este sistema para evitar falsos positivos y garantizar que el juego respondiera de manera inmediata y precisa a las colisiones, manteniendo la coherencia de las reglas del juego.

# **Rediseño de Escenas**

En este proyecto, se planteó rediseñar las escenas iniciales del juego, ya que las originales presentaban un diseño muy simple, limitado principalmente a texto. Esta falta de atractivo visual y de interacción generaba una experiencia poco inmersiva, dificultando la conexión del usuario con el juego.

Las escenas iniciales tenían fondos estáticos o, en algunos casos, carecían de ellos. El texto era plano y sin formato, y no existían elementos interactivos que pudieran captar la atención del usuario, sobre todo en escenas como la de GameOver en la que ni siquiera estaba la puntuación, así como la del menú y la de la elección de dificultad. Si bien cumplían con su función básica, estas escenas eran visualmente poco llamativas, lo que afectaba la experiencia general del jugador y restaba profesionalismo al proyecto.

Para abordar estas limitaciones, se enfocó el proceso de rediseño en tres áreas principales. En primer lugar, se mejoraron los aspectos visuales mediante la incorporación de, imágenes, y modelos en 3D, además de trabajar en una iluminación más cuidada para enriquecer la atmósfera del juego. En segundo lugar, se añadió interactividad, incluyendo botones, transiciones fluidas entre escenas y elementos dinámicos como efectos de clic, lo que permitió que las escenas fueran más atractivas y funcionales.

El resultado del rediseño muestra escenas mucho más dinámicas y llamativas. Los gráficos animados y los colores vibrantes dan vida al juego, mientras que los elementos interactivos invitan al jugador a explorar y participar. Además, la integración de efectos visuales y sonoros en la presentación del texto enriquece la experiencia narrativa, haciéndola más envolvente y profesional.

Este rediseño no solo mejoró significativamente la calidad visual e interactiva del proyecto, sino que también dejó como aprendizaje la importancia de diseñar pensando en la experiencia del usuario.

# Resultados

Como resultado del desarrollo del proyecto "Taxi Driver", se ha creado un ejecutable funcional del videojuego, el cual permite a los jugadores disfrutar de la experiencia completa del juego en cualquier dispositivo compatible. Este ejecutable incluye todas las funcionalidades diseñadas, como la persecución policiaca, la gestión de colisiones, la disminución de vida al chocar con obstáculos y edificios, y los diferentes niveles de dificultad que ajustan la velocidad de la policía.

Además, para demostrar el correcto funcionamiento del videojuego, se ha incluido un video de pruebas que muestra cómo se desarrolla la jugabilidad en diferentes escenarios. En este video se pueden observar ejemplos prácticos del comportamiento del taxi y la policía, la interacción con los elementos del entorno, y el cumplimiento de las reglas de juego establecidas, como la detección precisa de colisiones y la finalización del juego al cumplirse las condiciones de "Game Over".

Este video no solo valida la funcionalidad del videojuego, sino que también sirve como evidencia del cumplimiento de los objetivos planteados en el proyecto. Junto con el ejecutable, proporciona una solución completa y accesible que destaca por su diseño técnico y por ofrecer una experiencia de juego dinámica y entretenida.

# Conclusión

sEl desarrollo del proyecto "Taxi Driver" ha sido una experiencia enriquecedora y desafiante que ha permitido aplicar una variedad de conocimientos técnicos y creativos. A través de este proceso, hemos superado numerosos problemas relacionados con la configuración precisa de colisiones, físicas y navegación, logrando construir un videojuego funcional y entretenido. Este proyecto no solo ha fortalecido nuestras habilidades en el uso de Unity y C#, sino que también ha demostrado la importancia del diseño detallado y la iteración constante en el desarrollo de videojuegos.