COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

Proyecto Final
Videojuego Taxi

Jaime Pedrosa Comino
Guzmán Ignacio Pérez Ibarz
Grupo B

Paradigmas y Técnicas de Programación
3º Grado en Ingeniería Matemática e Inteligencia Artificial

Índice

Introducción

Contexto

Este proyecto final se centra en el desarrollo de un videojuego simulador en Unity, como parte del aprendizaje de arquitectura de software y principios de programación orientada a objetos. El proyecto está alineado con los conceptos fundamentales de la asignatura, incluyendo el uso de patrones de diseño y principios SOLID.

Descripción del Problema

El objetivo principal es construir un videojuego funcional donde un taxi y un coche de policía interactúan en un entorno virtual. Se busca implementar funcionalidades clave como el seguimiento del taxi por parte del coche de policía, la detección de velocidad mediante un radar y la gestión de eventos en el juego a través de una cámara en primera persona.

Metodología

Descripción del Entorno de Desarrollo

El proyecto ha sido desarrollado en Unity 6000.0.25f1 utilizando el lenguaje de programación C#. Se han creado distintos scripts para implementar las funcionalidades del juego.

- **Unity** como motor de desarrollo.
- **C#** como lenguaje de programación principal.
- **GitHub** para control de versiones y gestión del repositorio del proyecto.

Diseño de la Solución

El diseño del proyecto sigue un enfoque modular, donde cada funcionalidad está encapsulada en un script independiente. A continuación, se describen los scripts más relevantes en detalle:

MainMenu.cs

Este script gestiona las opciones del menú principal del juego, incluyendo las funcionalidades de empezar el juego y salir de la aplicación.

- **Empezar()**: Cambia a la siguiente escena en la secuencia del juego.
- **Quit()**: Cierra la aplicación.

Taxi.cs

Este script controla la lógica del taxi, incluyendo su movimiento, detección de colisiones y estado de vida.

- Atributos clave:
    - - rb: Rigidbody (Gestiona las físicas del taxi).
    - - speed: float (Velocidad del taxi).
    - - lifeTaxi: int (Estado de vida del taxi).
- Métodos importantes:
    - **Start()**: Inicializa el Rigidbody y ajusta el centro de masa.
    - **FixedUpdate()**: Gestiona el movimiento y la rotación del taxi basándose en la entrada del jugador.
    - **OnCollisionEnter()**: Detecta colisiones y disminuye la vida del taxi si impacta contra un obstáculo.

PoliceCar.cs

Este script implementa la lógica del coche de policía, encargándose de patrullar y perseguir al taxi cuando el radar detecta una velocidad superior a la permitida.

- Atributos clave:
    - - navMeshAgent: NavMeshAgent (Gestiona la navegación del coche de policía).
    - - patrolRadius: float (Radio de patrullaje).
- Métodos importantes:
    - **Start()**: Configura el agente de navegación y localiza el objeto Taxi en la escena.
    - **Update()**: Gestiona las diferentes lógicas de comportamiento del coche de policía (patrullaje o persecución).
    - **PredictAndAdjustPath()**: Ajusta la ruta para predecir la dirección del taxi.

SpeedRadar.cs

Este script detecta la velocidad del taxi y activa una alerta si el vehículo excede un límite establecido.

- Atributos clave:
  - - alert: bool (Estado de la alerta).
  - - detectionRadius: float (Radio de detección del radar).
  - - maximumvelocity: float (Velocidad máxima permitida).
- Métodos importantes:
  - **Start()**: Inicializa la referencia al objeto Taxi.
  - **Update()**: Calcula la distancia al taxi y activa la alerta si la velocidad es demasiado alta.

FirstPersonCamera.cs

Este script controla la cámara en primera persona que sigue al taxi durante el juego.

- Atributos clave:
  - - taxi: Transform (Referencia al objeto Taxi).
  - - offset: Vector3 (Distancia relativa entre la cámara y el taxi).
- Métodos importantes:
  - **Start()**: Calcula el desplazamiento inicial entre la cámara y el taxi.
  - **LateUpdate()**: Ajusta la posición y la rotación de la cámara para seguir al taxi.

PantallaFin.cs

Este script gestiona la pantalla final del juego.

- **Start()**: Invoca el cambio de escena al menú principal después de un tiempo.
- **LoadMainMenu()**: Carga la escena principal del menú.

Pruebas Realizadas

Las pruebas del videojuego se realizaron en un entorno de desarrollo local en PC. Se verificaron las siguientes funcionalidades:

- Movimiento del taxi controlado por el jugador.
- Seguimiento del taxi por parte del coche de policía.
- Activación del radar de velocidad y persecución.
- Cambios de escena desde el menú principal hasta la pantalla final.
- Ajuste de los parámetros del NavMeshAgent para optimizar el comportamiento del coche de policía.

Se realizaron pruebas adicionales para asegurar la estabilidad del sistema en diferentes escenarios, incluyendo rutas de patrullaje y detección de obstáculos.

Resultados

Descripción de los Resultados

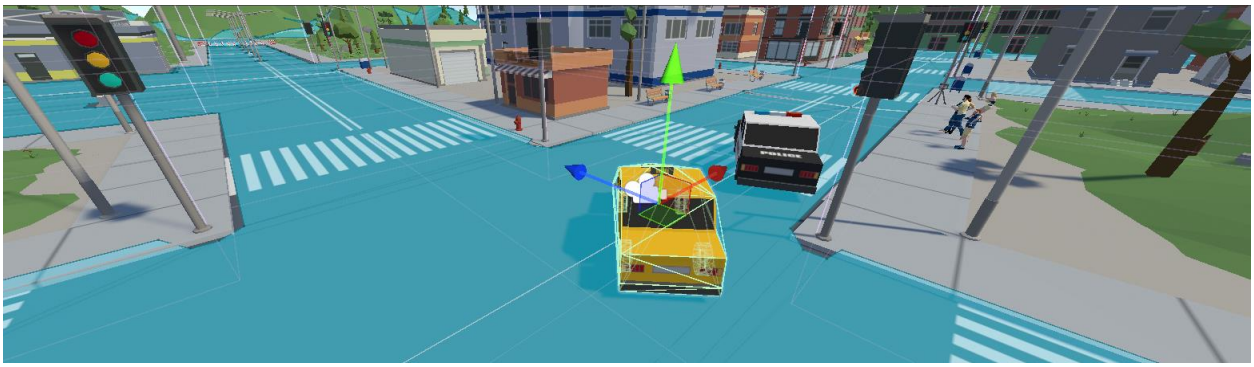
El videojuego desarrollado cumple con los requisitos planteados en el enunciado. Se han implementado todas las funcionalidades descritas y el sistema se comporta de manera estable durante la ejecución.

Pantallazos de la Ejecución

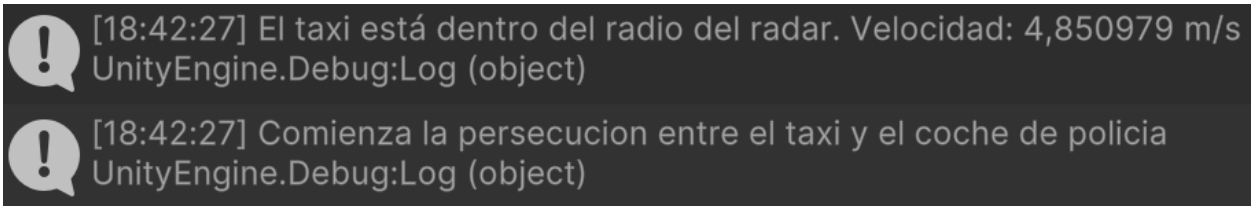
Se incluyen capturas de pantalla que muestran las distintas etapas del juego:

1. Menú principal del juego.

2. Escena principal con el taxi y el coche de policía.



3. Activación del radar de velocidad y comienzo de la persecución.



4. Pantalla de finalización del juego.

Discusión de los Resultados

El sistema se comporta como se esperaba en la mayor parte de los casos. Sin embargo, se identificaron algunos problemas menores relacionados con la detección de colisiones y la configuración de las rutas de patrullaje del coche de policía. Estos problemas se solucionaron ajustando los parámetros del NavMeshAgent y las distancias de detección del radar.

Se observó un comportamiento estable durante la persecución y la correcta transición entre las distintas escenas del juego. Además, la cámara en primera persona mejoró la experiencia inmersiva del jugador.

Conclusión

Resumen del Proceso

El desarrollo del videojuego simulador se llevó a cabo siguiendo una metodología de desarrollo iterativo. Se identificaron los requisitos, se implementaron las distintas funcionalidades y se realizaron pruebas para garantizar el correcto funcionamiento del sistema.

Este proyecto ha permitido reforzar los conceptos de arquitectura de software y ha mejorado las habilidades en el desarrollo de videojuegos en Unity, sentando una base sólida para proyectos futuros.