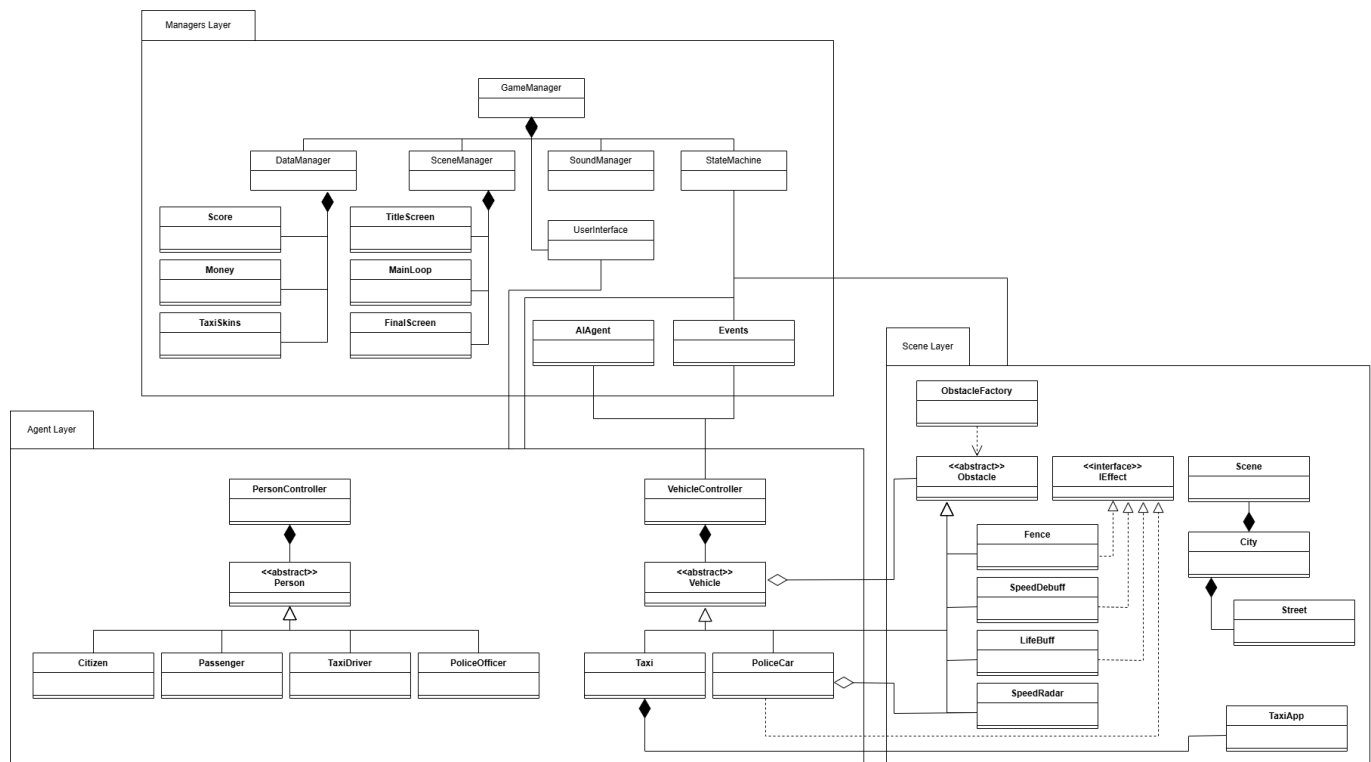


UML



https://drive.google.com/file/d/1tuYep9ebA8KG3bC1lohHM3_8Ulj7zx5/view?usp=sharing

Razonamiento de la Arquitectura

La estructura modular permite dividir las responsabilidades en clases específicas, favoreciendo el mantenimiento y escalabilidad del juego. Patrones de diseño usados:

1. Singleton para **GameManager** y **UserInterface**, permitiendo una instancia global que gestione el estado y la interfaz de usuario
2. Factory Method para la generación de obstáculos, enemigos e ítems, facilitando la creación dinámica de estos elementos en cada partida.

Historias de Usuario

1. Historia 1: Inicio de Partida

Como jugador, quiero ver un menú principal con mi puntaje y dinero. Quiero poder elegir entre comenzar una partida, elegir la skin de mi taxi o acceder a las configuraciones.

2. Historia 2: Taxista

Como taxista, quiero poder controlar un taxi en la ciudad para recoger pasajeros y llevarlos a su destino de forma segura. Además, quiero evitar se atrapado por la policía si excedo el límite de velocidad.

3. Historia 3: Pasajero

Como pasajero, quiero llegar a mi destino rápidamente y sin que se ponga en peligro mi seguridad, para dar una buena propina. Tengo una barra de confort que baja con choques, tiempo.

4. Historia 4: Ítems

Como taxista quiero poder recolectar ítems durante el trayecto que mejoren o perjudiquen el estado del taxi, para aumentar la dificultad del juego.

5. Historia 5: Policía

Como policía quiero que ningún taxista supere el límite de velocidad, tengo un radar que si detecto la infracción le persigo.

6. Historia 6: Final Screen

Cuando pierda como usuario, quiero que una bomba nuclear caiga sobre la ciudad, destruyendo todo y apareciendo un “menú de jugar de nuevo” o “volver al menú principal”.

Objetivo del Juego

1. Historia 1: Permitir al jugador iniciar y navegar el menú.
2. Historia 2: Controlar el taxi y llevar a los pasajeros a su destino.
3. Historia 3: Llegar al destino de manera rápida y segura, además de poder evaluar con el confort el viaje
4. Historia 4: Permitir al jugador recoger ítems para cambiar la dificultad del videojuego
5. Historia 5: Controlar que el taxista respete los límites de velocidad, y perseguirlo en caso de infracción
6. Historia 6: Ofrecer una pantalla de fin de juego impactante y que el jugador decida si jugar de nuevo o si volver al menú

Requisitos

1. Historia 1:
 - a. Crear una interfaz de menú principal.
 - b. Botones para iniciar partida, cambiar skin, ver puntaje y configuraciones.
2. Historia 2:
 - a. Sistema de control intuitivo para el taxi (teclado o pantalla táctil).
 - b. Lógica de detección de pasajeros y asignación de destinos.
 - c. Implementar una mecánica de límite de velocidad y reacción policial.
3. Historia 3:
 - a. Barra de confort visible para el jugador que se actualice en tiempo real.
 - b. Cálculo dinámico de la propina según el estado de la barra de confort.
 - c. Penalización de la barra por colisiones y tiempo empleado.
4. Historia 4:
 - a. Sistema de generación aleatoria de ítems en el mapa.
 - b. Efectos específicos para cada tipo de ítem.
5. Historia 5:
 - a. Radar funcional que detecte la velocidad del taxi en tiempo real.
 - b. Consecuencias para el jugador si la policía lo alcanza.
6. Historia 6:
 - a. Escena final con bomba nuclear y destrucción de la ciudad.
 - b. Menú funcional con las opciones para jugar de nuevo o de volver al menú principal.

Tareas

1. Historia 1:
 - a. Implementar el sistema de menús en Unity.
 - b. Conectar el botón de “Iniciar” con el ciclo de juego.
2. Historia 2:
 - a. Implementar controles para el manejo del taxi dentro de la ciudad.
 - b. Diseñar un sistema de puntos de recogida y destino para los pasajeros.
 - c. Detectar y registrar cuando el taxi recoge o deja a un pasajero.
3. Historia 3:
 - a. Implementar una barra de confort para el pasajero, que cambie según los eventos que vayan ocurriendo durante el viaje
4. Historia 4:
 - a. Generar ítems en puntos aleatorios de la ciudad.
 - b. Programar los efectos de los ítems en el estado del taxi
5. Historia 5:

- a. Implementar un radar que detecte si el taxi excede el límite de velocidad.
 - b. Iniciar una persecución si el radar detecta la infracción.
- 6. Historia 6:
 - a. Diseñar una animación de bomba nuclear que se active al perder la partida.