

Informe de Desarrollo del Videojuego "Taxi Drive"



Hecho por: Pedro Meseguer y Jorge Enebral

Introducción

"Taxi Drive" es un videojuego de simulación arcade inspirado en el clásico Crazy Taxi. El jugador controla un taxi en un entorno urbano, transportando pasajeros desde puntos de recogida a destinos finales mientras compite contra la paciencia de los pasajeros, con el fin de conseguir los máximos puntos en la partida. En el videojuego también a los policías, que el jugador debe esquivar para aguantar el máximo tiempo, ya que si el policía alcanza al jugador, el juego terminará.

Objetivo del videojuego

El principal objetivo del jugador es completar tantos viajes como sea posible mientras mantiene la paciencia del pasajero alta para maximizar la puntuación. La mecánica de puntuación sigue esta fórmula:

- +200 puntos por cada viaje completado.
- +100 puntos multiplicados por el porcentaje de paciencia restante del pasajero al finalizar el viaje.

Aparte de esto, el jugador tiene la posibilidad de recoger unas esferas que aparecen en la carretera con cada viaje (4 con cada viaje, no se acumulan) con las que puede subir la paciencia del pasajero.

Todo esto debe ser sin que el policía te alcance, debido a que si esto ocurre la partida terminará.

Comparación entre el diseño inicial y el final

1. Capa Manager

Las principales diferencias son que en la práctica no hemos implementado un dataManager, SoundManager ni un StateMachine. Como la dimensión de lo desarrollado es pequeña hemos considerado no necesario implementarlos.

Pero no solo hemos reducido funcionalidad, sino que hemos añadido nuevas: hemos añadido una escena de pausa y una barra de progreso (paciencia del cliente).

2. Capa Agentes

Las principales diferencias son que en la práctica no hemos implementado ciudadanos con lógica, por lo que esa parte no ha sido necesaria desarrollarla.

Solo hay un taxi y como mucho un policía a la vez, teniendo cada uno su controller. Además, hemos creado un PoliceCarFactory y un generador de rutas, que indica al jugador dónde está su próximo cliente y dónde debe dejarlo.

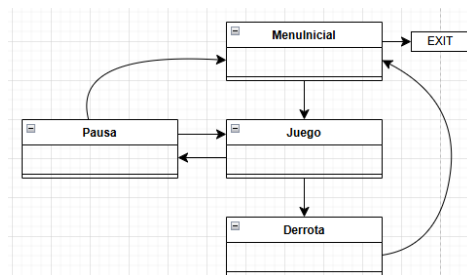
3. Capa Escenas

Las principales diferencias son que solo tenemos un obstáculo de tipo powerup, por lo que al ser uno nos hemos ahorrado el obstacleFactory y la lógica de abstracta de los objetos con su interfaz. El cameraController depende del taxi.

Scripts Principales y sus Funciones

1. GameManager

Este script es el que gestiona los cambios de escenas. Es un singleton. El diagrama UML de posibles cambios de escenas es el siguiente:



2. SceneManager

Este script que cambia de escenas. Hay 4 escenas: el menú inicial, el juego, pausa y derrota. Cada escena tiene asignada su música por lo que no se necesita de un SoundManager.

3. Route

Este script controla la lógica central del sistema de rutas del juego y es clave para la experiencia de navegación y transporte de pasajeros. Sus responsabilidades incluyen:

- Generar puntos de recogida y destino utilizando la función `GetRandomPointInArea`, que asegura que los puntos estén dentro de un área válida del NavMesh (carretera).
- Dibujar visualmente la ruta desde el taxi al punto de recogida (línea azul) y del taxi al destino (línea verde), usando `LineRenderer`.
- Controlar las etapas del viaje:
 - Etapa de recogida: el taxi se mueve hacia el pasajero.
 - Etapa de transporte: el taxi se dirige al destino.
- Reiniciar el ciclo después de cada viaje, incluyendo el reajuste de la barra de paciencia y llamando a la función que suma la puntuación total.

4. TaxiController

Controla el comportamiento del taxi en el entorno. Este script permite al jugador moverse usando controles estándar.

5. PatienceObjectSpawner

Se encarga de generar objetos de paciencia en la carretera durante el viaje con el pasajero. Estas son sus funciones clave:

- **Generación de objetos:**
 - Utiliza el método `GetRandomPointInArea()` del `Route` para generar posiciones aleatorias dentro de la carretera.
 - Los objetos de paciencia se generan cuando el viaje comienza y se limita a un máximo de 4 objetos.
- **Interacción con objetos:**
 - Comprobamos si el taxi está cerca de un objeto, y si lo está el objeto se recoge y se aumenta la paciencia mediante `ProgressBarController`.
- **Limpieza de objetos:**
 - Al finalizar un viaje, el script elimina todos los objetos generados y limpia la lista de objetos activos.

6. ProgressBarController

El script `ProgressBarController` controla la barra de progreso que refleja la paciencia del pasajero durante el viaje. Las funciones clave de este script son las siguientes:

- **Control de progreso:**
 - La barra de progreso (`progressBar`) se actualiza en la interfaz de usuario para reflejar la paciencia del pasajero, que varía entre 0 y 1.
- **Disminuir la paciencia:**
 - Cuando el viaje está en curso, la paciencia disminuye gradualmente con el tiempo a través del método `DecreasePatienceOverTime()`.
- **Incrementar paciencia:**
 - Al recoger objetos de paciencia, se usa `IncreasePatience()` para restaurar parte de la paciencia.
- **Resetear la barra:**
 - El método `ResetPatience()` restablece la barra al 100% cuando el viaje se completa o se reinicia.

7. CameraController

El CameraController gestiona la inclinación de la cámara en función del movimiento del vehículo. Sus funciones principales son:

- **Inclinación de la cámara:**
 - Se ajusta dinámicamente según la velocidad del coche en la dirección horizontal (steeringInput), proporcionando un efecto de cámara que se inclina mientras el coche gira.
- **Suavizado:**
 - La inclinación se suaviza utilizando las variables smoothSpeed y returnSpeed para una transición suave.

8. GameScoreController

Gestiona la puntuación total del juego y muestra la información relevante sobre el rendimiento del jugador. Sus funciones principales son:

- **Mostrar la puntuación:**
 - La puntuación se muestra en la pantalla utilizando TextMeshPro. El texto está anclado en la esquina superior izquierda y se ajusta dinámicamente.
- **Sumar puntos al completar el viaje:**
 - Al final de un viaje, el controlador calcula los puntos obtenidos tanto por completar el viaje (journeyPoints), como por la paciencia mantenida del pasajero, y los suma a la puntuación total.

9. PoliceAlertController

El PoliceAlertController gestiona la interfaz de alertas en el juego, mostrando mensajes cuando el jugador es perseguido por la policía:

- **Mostrar mensaje de alerta:**
 - Cuando la policía comienza a perseguir al jugador, el controlador muestra un mensaje en la pantalla usando TextMeshPro. El mensaje se configura y se activa para mostrarse en la pantalla durante un tiempo determinado, después se oculta.

10. PoliceCarController

Controla el comportamiento del coche de policía que persigue al jugador, haciendo que el juego finalice cuando el coche atrapa al taxi:

- **Seguir al jugador:**
 - El coche de policía utiliza un NavMeshAgent para seguir al taxi. Se asigna el taxi como el objetivo del coche de policía para que este lo persiga constantemente.
- **Detectar si el coche ha atrapado al jugador:**
 - Si el coche de policía está demasiado cerca del taxi, se activa una función de fin de juego.

11. PoliceCarFactory

El PoliceCarFactory es responsable de generar el coche de policía cuando el taxi supera una velocidad límite, además de gestionar su desaparición al finalizar el viaje:

- **Generar el coche de policía:**
 - Cuando la velocidad del taxi (accedida a través del CarController) supera un umbral, el PoliceCarFactory instancia un coche de policía en una ubicación fija en el mundo del juego.
- **Alertas al jugador:**
 - Al generar el coche de policía, el PoliceCarFactory también activa una alerta visual mediante el PoliceAlertController.
- **Desaparecer el coche de policía:**
 - Una vez que el taxi termina su viaje, se destruye el coche de policía, que vuelve a aparecer si el jugador vuelve a superar el umbral.

Aspectos Técnicos del Desarrollo

El uso de NavMesh fue fundamental para la navegación del taxi y la ubicación de puntos válidos en la carretera. Al principio esto ocasionó muchos problemas por la ciudad descargada, en la que para hacer que funcionara correctamente tuvimos que reemplazar algunos elementos como algunas carreteras que no eran detectadas correctamente, lo que hacía que no pudieran aparecer puntos o el policía no pudiera acceder. La generación de puntos de recogida, destinos y objetos de paciencia utiliza posiciones aleatorias validadas contra el NavMesh mediante NavMesh.SamplePosition, asegurando que todos los elementos aparezcan en la carretera y no fuera del área jugable. La interfaz proporciona retroalimentación visual constante al jugador, incluyendo una barra de paciencia para el pasajero, puntuación total en pantalla y una minimap con rutas dinámicas y objetivos visuales.

Conclusión

El juego combina navegación dinámica y desafíos. Usa NavMesh para la ruta del taxi y la generación de objetos de paciencia en la carretera. La interfaz muestra información clave como la puntuación y barra de paciencia. Además, la persecución del policía añade un reto extra. Es un juego entretenido y desafiante para el jugador.