

Memoria Descriptiva del Videojuego Educativo de Carreras en Unity

Equipo de Trabajo - Curso Tecnología de Objetos

2 de diciembre de 2025

1. Introducción

La presente memoria descriptiva documenta el funcionamiento general, la arquitectura técnica básica, el proceso de instalación y las principales características del videojuego educativo de carreras desarrollado en el motor Unity. El proyecto combina mecánicas de conducción con un sistema de preguntas y respuestas que introduce contenidos educativos durante la experiencia de juego.

El objetivo principal es ofrecer una herramienta lúdica que refuerce el aprendizaje mediante la toma de decisiones rápidas: el jugador avanza por un circuito de carreras y, en ciertos puntos se le presentan preguntas con varias opciones de respuesta integradas en portales interactivos. Según la opción seleccionada, el vehículo recibe un impulso de velocidad o sufre una penalización temporal, reforzando el feedback inmediato y la atención sostenida del jugador.

2. Arquitectura y Plataforma Tecnológica

El videojuego está desarrollado en Unity utilizando C# como lenguaje principal de programación. El proyecto se organiza en escenas, prefabs y scripts orientados a objetos, siguiendo una estructura modular que facilita la mantenibilidad y la incorporación de nuevas funcionalidades.

La arquitectura de software se basa en el uso de patrones de diseño como Singleton, State y Observer, empleados en componentes clave como gestores globales, sistema de eventos, control del vehículo y lógica de dificultad, dejando abierta la posibilidad de incorporar un patrón adicional en futuras iteraciones del proyecto. En particular, el patrón Observer se aplica al sistema de preguntas, respuestas y portales interactivos, donde los elementos visuales actúan como observadores que reaccionan automáticamente cuando se emite una nueva pregunta o se registra una respuesta por parte del jugador, manteniendo un acoplamiento bajo entre la lógica educativa y la presentación en pantalla. El juego está pensado para ejecutarse en equipos de escritorio con sistema operativo Windows, aunque la arquitectura de Unity permite su portabilidad a otras plataformas compatibles.

3. Funcionalidades del Sistema

El videojuego educativo de carreras proporciona un conjunto de funcionalidades orientadas tanto a la experiencia de juego como al componente formativo. Entre las principales se incluyen:

- Conducción de vehículos en una pista diseñada para integrar zonas de decisión con portales de respuesta.
- Sistema de preguntas y respuestas con bancos de preguntas configurables, que se muestran en tiempo real durante la carrera.
- Aplicación de impulsos de velocidad al responder correctamente y penalizaciones temporales al fallar, afectando directamente al rendimiento del vehículo.
- Menú principal con selección de modo de juego (un jugador o multijugador local) y elección de vehículos con diferentes atributos.
- Interfaz de usuario que presenta preguntas, opciones de respuesta y retroalimentación visual sobre el resultado de cada decisión.
- Soporte para estadísticas de los vehículos, mostrando atributos como velocidad, aceleración, giro y frenado durante la selección de autos.

4. Proceso de Instalación

Para trabajar con el proyecto desde el entorno de desarrollo, se siguen los pasos descritos a continuación.

4.1. Instalación para desarrolladores

1. Clonar el repositorio del proyecto desde GitHub, disponible en <https://github.com/rescobedoq/r>.
2. Abrir el proyecto en la versión de Unity recomendada por el equipo de desarrollo (LTS 2022), asegurando la correcta importación de escenas, scripts y assets.
3. Verificar las rutas de los recursos (assets, audio, texturas y prefabs) y permitir que Unity regenere librerías y metadatos necesarios.
4. Configurar la escena principal del juego como escena de inicio en la ventana de *Build Settings*.
5. Ejecutar el proyecto en modo *Play* para comprobar el correcto funcionamiento de la mecánica de conducción, el sistema de preguntas y la interfaz de usuario.

4.2. Instalación para usuarios finales

1. Descargar la versión compilada del videojuego (ejecutable y carpeta de datos) proporcionada por el equipo desarrollador.
2. Descomprimir el paquete en una carpeta local con permisos de lectura y escritura.
3. Ejecutar el archivo principal del juego (por ejemplo, `RyC.exe`) desde el sistema operativo.

5. Roles de Usuario y Modos de Juego

En el contexto del videojuego educativo, se consideran principalmente dos tipos de roles funcionales y varios modos de interacción.

5.1. Roles de usuario

- **Jugador:** usuario que controla uno de los vehículos, responde preguntas durante la carrera y experimenta los efectos de sus decisiones en tiempo real.
- **Facilitador o docente:** usuario que puede preparar o revisar bancos de preguntas, así como utilizar el juego como herramienta de apoyo en sesiones educativas presenciales o virtuales, incluso cuando no esté integrado un panel de administración completo dentro del software.

5.2. Modos de juego

- **Modo un jugador:** permite a un único jugador competir contra el entorno, poniendo énfasis en la mejora personal y el aprendizaje de contenidos.
- **Modo multijugador local:** habilitado mediante la gestión de cámaras y vehículos múltiples, permitiendo que dos jugadores compartan pantalla y compitan simultáneamente en el mismo circuito.

6. Gestión de Contenido Educativo y Recursos

El sistema de preguntas y respuestas se basa en bancos de preguntas estructurados, donde cada ítem contiene el enunciado, varias opciones de respuesta y la indicación de la opción correcta. Durante la carrera, se utilizan disparadores (*triggers*) ubicados en puntos específicos del circuito para solicitar una nueva pregunta y mostrar las opciones asociadas a través de portales. La respuesta seleccionada por el jugador se evalúa en tiempo real y genera un efecto directo sobre el estado del vehículo.

7. Conclusiones

El videojuego educativo de carreras en Unity constituye una propuesta que fusiona entretenimiento y aprendizaje mediante la integración de mecánicas de conducción con un sistema de preguntas y respuestas de impacto inmediato. La combinación de retroalimentación visual, efectos de velocidad y dinámica competitiva busca mantener la motivación del jugador mientras refuerza contenidos educativos.

La arquitectura de software se basa en el uso de patrones de diseño como Singleton, State y Observer, empleados en componentes clave como gestores globales, sistema de eventos, control del vehículo y lógica de dificultad, además de otros patrones complementarios que pueden incorporarse en futuras iteraciones del proyecto.