

Universidad Nacional de San Agustín de Arequipa

Facultad de Ingeniería de Producción y Servicios

Escuela Profesional de Ingeniería de Sistemas

DOCUMENTACIÓN DEL PROYECTO

Videojuego Educativo de Carreras en Unity

Curso: Tecnología de Objetos

Profesor: Ing. Richart Smith Escobedo Quispe

Integrantes del Equipo

Choquehuanca Zapana Hernan Andy (Rama: hz07s/workspace)

Larico Rodriguez Bryan Fernando (Rama: BryanLarico/workspace)

Portugal Portugal Eduardo Sebastian (Rama: Eduardo-P/workspace)

Mamani Anahua Victor Narciso (Rama: victorma18/workspace)

Semana Documentada: Semana 3

Fecha: 26 de noviembre de 2025

Introducción

En la tercera semana del desarrollo del videojuego educativo en Unity, se avanzó en la integración de la lógica de preguntas y respuestas en los portales, el sistema de selección de autos y el manejo de modos de juego (un jugador y multijugador). Se implementaron scripts para gestionar el banco de preguntas, la interacción con los portales, la visualización de preguntas y respuestas, así como la selección y atributos de los autos. A continuación, se presenta el detalle y explicación de los principales archivos desarrollados durante esta semana.

1. MultiplayerManager.cs

1.1. Rol General

Gestiona el modo de juego multijugador y la inicialización de los vehículos y cámaras según el modo seleccionado.

1.2. Responsabilidades

- Instancia los vehículos en posiciones específicas para cada jugador.
- Aplica los atributos seleccionados de cada vehículo.
- Configura la cámara principal y el modo de pantalla (completa o dividida).
- Lee las preferencias de modo de juego y selección de autos.
- Desactiva la cámara principal de Unity para evitar conflictos.

1.3. Métodos Principales

Start(): Determina el modo de juego (un jugador o multijugador), instancia los vehículos y aplica sus datos.

ApplyCarData(): Busca y asigna los atributos del vehículo seleccionado a cada controlador de auto.

1.4. Notas

Utiliza PlayerPrefs para recuperar las preferencias de modo y selección de autos. El sistema está preparado para expandirse y soportar más jugadores o modos en el futuro.

2. CarButton.cs

2.1. Rol General

Representa el botón de selección de autos en la interfaz del menú.

2.2. Responsabilidades

- Muestra la imagen y nombre del auto.
- Visualiza las estadísticas del auto seleccionado.
- Permite seleccionar y resaltar el auto.
- Notifica al sistema cuando se selecciona un auto.

2.3. Métodos Principales

Initialize(): Configura el botón con los datos del auto.

OnClick(): Marca el auto como seleccionado y actualiza la visualización.

Deselect(): Quita la selección del botón.

UpdateVisual(): Actualiza el color y escala según el estado de selección.

3. CarStatsDisplay.cs

3.1. Rol General

Muestra las estadísticas detalladas de un auto en el menú.

3.2. Responsabilidades

- Visualiza atributos como velocidad, aceleración, giro, frenado y velocidad máxima.
- Actualiza las barras y valores numéricos según el auto seleccionado.

3.3. Métodos Principales

UpdateStats(): Recibe los datos del auto y actualiza la visualización.

GetNormalizedValue(): Obtiene el valor normalizado de cada atributo para mostrar en la barra correspondiente.

4. CarData.cs

4.1. Rol General

Define los datos y atributos de cada auto disponible en el juego.

4.2. Responsabilidades

- Almacena nombre, imagen y atributos principales del auto.
- Permite mapear valores normalizados a valores reales usados por el controlador del auto.

4.3. Atributos Principales

- Nombre del auto.
- Imagen representativa.
- Velocidad base, aceleración, giro, frenado y velocidad máxima.

4.4. Métodos Relevantes

GetRealAcceleration(): Convierte la aceleración normalizada a valor real.

GetRealMaxSpeed(): Convierte la velocidad máxima normalizada a valor real.

5. MenuManager.cs

5.1. Rol General

Gestiona la lógica y navegación del menú principal del juego, permitiendo la selección de modo de juego y autos.

5.2. Responsabilidades

- Controla la visibilidad de los paneles de menú para uno o dos jugadores.
- Administra la selección de autos para cada jugador.
- Habilita los botones de jugar según las selecciones realizadas.
- Instancia los botones de autos en los ScrollView.
- Guarda las selecciones en PlayerPrefs y carga la escena principal.

5.3. Métodos Principales

SelectMode(): Cambia entre modo de juego individual o multijugador.

PopulateScrollView(): Instancia los botones de autos en el menú.

OnCarSelected(): Actualiza la selección de autos y habilita los botones de jugar.

LoadGameScene(): Guarda las selecciones y carga la escena de juego.

6. SwipeMenu.cs

6.1. Rol General

Permite la navegación horizontal entre elementos del menú usando gestos de arrastre (swipe).

6.2. Responsabilidades

- Gestiona el desplazamiento del ScrollView mediante el scrollbar.
- Ajusta la escala de los elementos para resaltar el seleccionado.
- Suaviza el movimiento y la selección visual de los elementos.

6.3. Métodos Principales

Update(): Calcula posiciones, gestiona el scroll y ajusta la escala de los elementos.

7. StatisticsBar.cs

7.1. Rol General

Representa una barra visual para mostrar valores de estadísticas (como velocidad, aceleración, etc.) en la interfaz.

7.2. Responsabilidades

- Actualiza el valor mostrado y el llenado de la barra.
- Sincroniza el valor numérico con la visualización gráfica.

7.3. Métodos Principales

BarFiller(): Calcula el llenado de la barra según el valor actual.

Update(): Actualiza el texto y la barra cada frame.

8. PortalAnswerTrigger.cs

8.1. Rol General

Detecta cuando un vehículo selecciona una respuesta en el portal de preguntas.

8.2. Responsabilidades

- Configura el collider como trigger para respuestas.
- Identifica si la respuesta es izquierda o derecha.
- Notifica al QuizManager cuando un vehículo selecciona una respuesta.

8.3. Métodos Principales

OnTriggerEnter(): Detecta la entrada del vehículo y comunica la respuesta seleccionada.

9. PortalAnswerDisplay.cs

9.1. Rol General

Muestra el texto de la respuesta (izquierda o derecha) en el portal de preguntas.

9.2. Responsabilidades

- Visualiza el texto de la respuesta correspondiente.
- Permite limpiar o actualizar el texto mostrado.
- Se registra en el QuizManager para ser gestionado dinámicamente.

9.3. Métodos Principales

SetAnswer(): Actualiza el texto de la respuesta.

Clear(): Limpia el texto mostrado.

10. MetaTrigger.cs

10.1. Rol General

Detecta cuando un vehículo cruza la meta y notifica al sistema de preguntas.

10.2. Responsabilidades

- Configura el collider como trigger.

- Llama a QuizManager al detectar que un vehículo cruza la meta.

10.3. Métodos Principales

OnTriggerEnter(): Detecta la entrada del vehículo y comunica el evento de meta.

11. QuizManager.cs

11.1. Rol General

Gestiona toda la lógica de preguntas y respuestas del juego, coordinando los portales, displays y triggers.

11.2. Responsabilidades

- Administra los bancos de preguntas para cada portal.
- Registra y gestiona los displays de preguntas y respuestas.
- Selecciona preguntas aleatorias y muestra opciones en los portales.
- Evalúa respuestas y aplica efectos (boost o penalización) al vehículo.
- Resetea preguntas y displays al cruzar la meta.

11.3. Métodos Principales

OnPreportalEnter(): Activa y muestra una pregunta en el portal correspondiente.

OnPortalAnswer(): Evalúa la respuesta seleccionada y aplica el efecto al vehículo.

OnMeta(): Resetea preguntas y limpia los displays tras cruzar la meta.

12. QuestionBank.cs

12.1. Rol General

Define la estructura de las preguntas y almacena los bancos de preguntas para cada portal.

12.2. Responsabilidades

- Define el formato de pregunta con dos opciones y la respuesta correcta.
- Proporciona bancos de preguntas para Portal 1 y Portal 2.

12.3. Estructura Principal

- **TwoOptionQuestion:** Clase que representa una pregunta con dos opciones.
- **QuestionsBank:** Clase estática con arrays de preguntas para cada portal.

13. PreportalTrigger.cs

13.1. Rol General

Detecta cuando un vehículo se aproxima a un portal y solicita una pregunta al sistema de quiz.

13.2. Responsabilidades

- Configura el collider como trigger.
- Llama a QuizManager para mostrar una nueva pregunta en el portal correspondiente.
- Realiza logs de depuración para seguimiento de eventos.

13.3. Métodos Principales

OnTriggerEnter(): Detecta la entrada del vehículo y solicita una pregunta.

14. PortalQuestionDisplay.cs

14.1. Rol General

Muestra el texto de la pregunta en el portal correspondiente usando TextMeshPro.

14.2. Responsabilidades

- Visualiza el texto de la pregunta en el portal.
- Permite limpiar o actualizar el texto mostrado.
- Se registra en el QuizManager para ser gestionado dinámicamente.

14.3. Métodos Principales

SetQuestion(): Actualiza el texto de la pregunta.

Clear(): Limpia el texto mostrado.