

# Proyecto Final Paradigmas y Técnicas de Programación

Nicolás del Val y Santiago Martínez

Github: <https://github.com/ndelval/practica-final-paradigmas>

## Historias de Usuario

### AIController

- Como controlador de IA de coches, quiero controlar la posición y velocidad del coche de IA para competir en la carrera.
- Como controlador de IA de coches, quiero manejar la lógica de movimiento, incluyendo adelantamientos y maniobras de reversa, para adaptarme al entorno de la pista.

### SensorManager

- Como gestor de sensores, quiero detectar obstáculos frente al coche para evitar colisiones.
- Como gestor de sensores, quiero verificar los laterales del coche para determinar si es seguro adelantar o cambiar de carril.

### OvertakeManager

- Como gestor de adelantamientos, quiero identificar cuándo y cómo realizar un adelantamiento seguro en la pista.
- Como gestor de adelantamientos, quiero ajustar el ángulo de giro y la velocidad del coche durante el adelantamiento.

### TurnManager

- Como gestor de curvas, quiero ajustar la dirección y velocidad del coche para girar sin salirme de la pista.
- Como gestor de curvas, quiero detectar curvas cerradas y aplicar una estrategia de frenado o aceleración adecuada.

### MovementManager

- Como gestor de movimiento, quiero gestionar la aceleración y frenado del coche en función de la distancia a los puntos de control.
- Como gestor de movimiento, quiero mantener una velocidad constante en tramos rectos para optimizar el rendimiento en la carrera.

### ReverseManager

- Como gestor de reversa, quiero detectar si el coche está atascado y activar el modo reversa para desatascarlo.
- Como gestor de reversa, quiero ejecutar un movimiento en reversa de forma segura y luego retomar la dirección normal en la pista.

### **CarController**

- Como controlador de coche, quiero gestionar los controles de dirección, aceleración y frenado del coche para que el jugador pueda interactuar con el vehículo.
- Como controlador de coche, quiero activar y desactivar el efecto de nitro cuando el jugador lo solicite
- Como controlador de coche, quiero controlar las partículas de humo en el tubo de escape para aumentar la inmersión visual.

### **HumoTuboEscape**

- Como sistema de humo, quiero activar partículas en el tubo de escape en función de la aceleración y frenado del coche para simular el efecto de un coche real.

### **PlayerCarController**

- Como controlador del coche del jugador, quiero almacenar la posición de inicio y las configuraciones de control personalizadas para ofrecer una experiencia adaptada.
- Como controlador del coche del jugador, quiero permitir al jugador modificar las teclas de control en función de sus preferencias.

### **CarSetup**

- Como configurador de coches, quiero almacenar la configuración de motor, fuerza de frenado y atributos de velocidad para permitir ajustes de rendimiento.
- Como configurador de coches, quiero gestionar los valores de nitro y efectos de humo en función del estado del coche.

### **ControllInputManager**

- Como gestor de controles, quiero permitir que el jugador asigne y modifique las teclas de control para una experiencia de juego personalizada.

- Como gestor de controles, quiero mostrar al jugador las teclas asignadas en pantalla para mejorar la accesibilidad.

#### **PlayerPositionReset**

- Como sistema de reinicio de posición, quiero restaurar la posición del coche a un punto inicial cuando el jugador lo solicite para evitar atascos.

#### **PauseMenu**

- Como menú de pausa, quiero permitir al jugador pausar y reanudar la carrera para darle control sobre el flujo de juego.
- Como menú de pausa, quiero proporcionar acceso al menú de configuraciones y la opción de reiniciar la carrera.

#### **PlayerReset**

- Como sistema de reinicio del jugador, quiero restablecer los datos de posición y tiempos de vuelta para que el jugador pueda comenzar una nueva partida.

#### **LapTimerReset**

- Como sistema de reinicio de vueltas, quiero borrar los tiempos de vuelta anteriores para iniciar un nuevo registro.

#### **SettingsMenu**

- Como menú de configuraciones, quiero permitir que el jugador acceda a las configuraciones de sonido, controles y gráficos para ajustar la experiencia de juego.
- Como menú de configuraciones, quiero ofrecer opciones de ajuste de volumen y otras configuraciones para personalizar el juego.

#### **FinishLine**

- Como línea de meta, quiero registrar el tiempo y las vueltas completadas por el jugador para mostrar su progreso en la carrera.

#### **LapTimer**

- Como temporizador de vueltas, quiero registrar y mostrar el tiempo de cada vuelta para que el jugador pueda mejorar su rendimiento.
- Como temporizador de vueltas, quiero mostrar el tiempo total de la carrera y cada tiempo parcial.

#### **GameManager**

- Como gestor de juego, quiero controlar el flujo general de la partida, incluyendo inicio, pausa y finalización, para una experiencia fluida.
- Como gestor de juego, quiero actualizar el velocímetro y otros elementos de la interfaz para que el jugador tenga feedback en tiempo real.

### **ITimer**

- Como interfaz de temporizador, quiero proporcionar un sistema unificado para gestionar los diferentes tiempos del juego, como el tiempo de vuelta y el tiempo total.

### **UntouchableCarController, JumpingCarController, CrashingCarController**

- Como coche especial, quiero tener habilidades únicas como ser intocable, saltar o colisionar sin ser afectado, para diversificar la experiencia de juego y las estrategias del jugador.

## **Análisis de Historias de Usuario**

### **AICarController**

**Objetivo:** Gestionar el control y movimiento del coche de IA para competir de forma autónoma en la carrera.

- **Requisitos:**
  - El controlador debe ser capaz de ajustar la velocidad y dirección del coche en función de la pista y posición de los otros coches.
  - El controlador debe manejar situaciones de adelantamiento y maniobras de reversa cuando el coche esté atascado.
- **Tareas:**
  - Implementar lógica de movimiento de IA que ajuste velocidad y dirección.
  - Integrar con OvertakeManager y ReverseManager para gestionar adelantamientos y reversa.
  - Coordinar con TurnManager para optimizar giros y evitar salir de la pista.

### **SensorManager**

**Objetivo:** Detectar obstáculos y proporcionar información al controlador de IA para evitar colisiones.

- **Requisitos:**
  - Detectar obstáculos al frente y en los laterales del coche.
  - Informar al controlador de IA sobre la proximidad y ubicación de los obstáculos para decisiones de maniobra.
- **Tareas:**
  - Configurar sensores de proximidad para detectar obstáculos.
  - Implementar métodos para verificar el estado de los laterales y decidir sobre adelantamientos.
  - Integrarse con OvertakeManager para facilitar maniobras seguras.

## **OvertakeManager**

**Objetivo:** Gestionar los adelantamientos de forma segura en la pista.

- **Requisitos:**
  - Identificar cuándo es seguro adelantar basándose en los datos de SensorManager.
  - Ajustar el ángulo de giro y la velocidad durante el adelantamiento.
- **Tareas:**
  - Implementar la lógica de detección de espacios seguros para adelantar.
  - Coordinar con TurnManager para optimizar las maniobras en curva durante el adelantamiento.

## **TurnManager**

**Objetivo:** Ajustar la dirección y velocidad del coche en las curvas para evitar salirse de la pista.

- **Requisitos:**
  - Detectar curvas y aplicar una estrategia de frenado o aceleración adecuada.
  - Colaborar con el controlador de IA para mantener la dirección correcta en la pista.
- **Tareas:**

- Implementar lógica de detección de curvas y aplicar ajustes de velocidad.
- Integrarse con AICarController para coordinar el control de dirección en curvas cerradas.

## **MovementManager**

**Objetivo:** Gestionar el movimiento del coche, especialmente en aceleración y frenado.

- **Requisitos:**

- Asegurar una aceleración constante y frenado suave en tramos rectos y puntos de control.
- Ajustar la velocidad en función de la distancia a los puntos de control.

- **Tareas:**

- Implementar lógica de aceleración y frenado automático.
- Mantener la velocidad óptima en función del entorno de la pista.

## **ReverseManager**

**Objetivo:** Controlar el modo de reversa del coche cuando esté atascado o fuera de la pista.

- **Requisitos:**

- Detectar automáticamente si el coche está atascado.
- Activar y gestionar el movimiento en reversa para desatascar el coche.

- **Tareas:**

- Implementar detección de atascos y reversa automática.
- Coordinar con el controlador de IA para retomar la dirección normal tras la maniobra de reversa.

## **CarController**

**Objetivo:** Proporcionar control sobre los elementos básicos del coche, incluyendo aceleración, frenado y dirección.

- **Requisitos:**

- Permitir al jugador controlar la aceleración, frenado y dirección del coche.
- Gestionar el uso de nitro y efectos visuales como humo de escape.

- **Tareas:**

- Implementar controles básicos de movimiento.
- Configurar el sistema de nitro y sus efectos visuales.
- Coordinar con HumoTuboEscape para activar partículas de humo.

## **HumoTuboEscape**

**Objetivo:** Simular el efecto de humo en el tubo de escape del coche.

- **Requisitos:**

- Activar partículas de humo en función de la aceleración y frenado del coche.

- **Tareas:**

- Configurar partículas de humo para simular el escape del coche.
- Sincronizar la emisión de partículas con la aceleración del coche.

## **PlayerCarController**

**Objetivo:** Controlar las configuraciones y posición del coche del jugador.

- **Requisitos:**

- Permitir que el jugador modifique sus controles y mantenga configuraciones personalizadas.

- **Tareas:**

- Implementar personalización de controles para el jugador.
- Almacenar la posición inicial y configuraciones de control.

## **CarSetup**

**Objetivo:** Gestionar las configuraciones y atributos del coche.

- **Requisitos:**
  - Configurar parámetros como motor, frenos, nitro y efectos de humo.
- **Tareas:**
  - Implementar un sistema de configuración que permita ajustar los atributos del coche.
  - Integrarse con CarController para aplicar estas configuraciones.

## **ControllInputManager**

**Objetivo:** Gestionar y mostrar los controles de entrada del jugador.

- **Requisitos:**
  - Permitir al jugador modificar y visualizar las teclas asignadas.
- **Tareas:**
  - Implementar sistema de asignación y visualización de teclas.

## **PlayerPositionReset**

**Objetivo:** Permitir al jugador reiniciar la posición del coche.

- **Requisitos:**
  - Restaurar la posición del coche a un punto inicial.
- **Tareas:**
  - Implementar lógica de reinicio de posición en caso de atasco.

## **PauseMenu**

**Objetivo:** Ofrecer opciones de pausa y configuración durante el juego.

- **Requisitos:**
  - Pausar y reanudar el juego.
  - Acceder a configuraciones y reiniciar la carrera.



- **Tareas:**
  - Implementar menú de pausa y sus opciones.

### **PlayerReset**

**Objetivo:** Reiniciar los datos del jugador al comenzar una nueva carrera.

- **Requisitos:**
  - Restablecer posición y tiempos de vuelta.
- **Tareas:**
  - Implementar lógica de reinicio de posición y tiempos.

### **LapTimerReset**

**Objetivo:** Reiniciar el temporizador de vueltas.

- **Requisitos:**
  - Borrar los tiempos de vuelta anteriores.
- **Tareas:**
  - Implementar lógica de reinicio del temporizador de vueltas.

### **SettingsMenu**

**Objetivo:** Permitir ajustes de configuraciones del juego.

- **Requisitos:**
  - Configurar sonido, controles y gráficos.
- **Tareas:**
  - Implementar el menú de configuraciones.

### **FinishLine**

**Objetivo:** Registrar el tiempo y las vueltas al cruzar la meta.

- **Requisitos:**
  - Registrar el tiempo al cruzar la línea de meta.

- **Tareas:**
  - Implementar lógica para calcular el tiempo total de carrera y vueltas.

## **LapTimer**

**Objetivo:** Gestionar y mostrar el tiempo de cada vuelta.

- **Requisitos:**
  - Mostrar tiempos parciales y total de la carrera.
- **Tareas:**
  - Implementar temporizador de vuelta y lógica de cálculo de tiempos.

## **GameManager**

**Objetivo:** Controlar el flujo general del juego.

- **Requisitos:**
  - Manejar inicio, pausa y finalización del juego.
- **Tareas:**
  - Implementar la lógica del flujo de juego.
  - Coordinar el funcionamiento de sistemas como LapTimer, PauseMenu, y PlayerReset.

## **ITimer**

**Objetivo:** Proporcionar una interfaz unificada para gestionar los diferentes temporizadores del juego.

- **Requisitos:**
  - Ofrecer funciones comunes para los temporizadores.
- **Tareas:**
  - Crear una interfaz para manejar tiempos de vuelta y de carrera.

## **UntouchableCarController, JumpingCarController, CrashingCarController**

**Objetivo:** Implementar coches con habilidades únicas para diversificar la experiencia de juego.

- **Requisitos:**

- Cada tipo de coche debe tener una habilidad especial (intocable, saltar, colisión).

- **Tareas:**

- Implementar habilidades especiales en los respectivos controladores.
- Integrar estas funcionalidades con el sistema de control de IA y de jugador.