

DESCRIPCIÓN DEL VIDEOJUEGO

El videojuego consiste en un minigolf. Cada jugador podrá lanzar la pelota en la dirección y con la fuerza que quiera para alcanzar el hoyo evitando los obstáculos. Habrá 3 niveles y cada uno será distinto, pero todos tendrán los mismos elementos. Por un lado, tendrán obstáculos que dificultarán el juego (un muro, una tabla de madera, bunkers de arena, agua...) y por otro lado el hoyo donde habrá que meter la pelota para ganar. El objetivo principal será realizar cada nivel en el menor número de golpes posibles. En el modo multijugador o jugando contra la IA, se contabilizará tanto el número de golpes como el tiempo que se tarda en meter la pelota en el hoyo desde el primer golpe. Una vez el primer jugador haya superado el nivel, habrá una cuenta atrás para que el resto meta la pelota en el hoyo. En caso de no hacerlo, a los golpes que lleven se le sumará 2 como penalización.

JUSTIFICACIÓN UML

Tenemos tres componentes principales:

La capa de jugador contiene tanto al jugados como la pelota. Para ambos se ha hecho una factory para hacer más fácil su creación. La persona también contiene la pelota con la que juega (una por jugador). Por un lado, la pelota tiene un color y tiene una opción de incluirle un sombrero según el jugador quiera. Aquí se ha usado el patrón bridge para separar la funcionalidad del propio objeto del color. Para añadirle el sombrero y el color, se ha usado un decorador. Por otro lado, el jugador hace golpes (con los que tira la pelota).

La capa de nivel tiene el nivel que se está jugando, los obstáculos de dicho nivel (para lo que se ha creado una factory también), el hoyo del nivel y los suelos que hay en el nivel (arena, agua o calle).

La capa de gestión tiene tanto la gestión de la puntuación, como un registro de las jugadas para entrenar a la IA, gestión de estados, gestión del escena, gestión de sonido, los eventos y la IA.

HISTORIAS DE USUARIO

Historia 1: Como jugador quiero meter la bola en el hoyo para pasar al siguiente nivel:

- Análisis del Objetivo
 - Permitir que el jugador complete un objetivo en el juego (meter la bola en el hoyo) para poder progresar al siguiente nivel.
- Extracción de Requisitos
 - Control del jugador sobre la bola: El jugador debe tener la capacidad de mover la bola lanzándola.
 - Detección de colisión con el hoyo: El juego debe detectar cuándo la bola entra en el hoyo.
 - Progresión al siguiente nivel: Al completar la acción (meter la bola en el hoyo), el juego debe desencadenar un cambio de nivel.
 - Sistema de objetivos y puntuación: Actualizar puntuación.
 - Tiempo de respuesta: La transición al siguiente nivel debe ocurrir rápidamente tras completar el objetivo, para mantener el flujo del juego.
 - Interfaz y feedback visual/auditivo: El juego debe proporcionar un feedback claro (visual o sonoro) cuando la bola entra en el hoyo, para indicar al jugador que ha completado el objetivo.
- Extracción de tareas a realizar:
 - Implementar el sistema de movimiento de la bola.
 - Permitir que el jugador pueda controlar la dirección y fuerza de la bola.
 - Asegurarse de que la física de la bola sea realista.
 - Programar la detección de colisión con el hoyo.
 - Configurar una zona de detección en el hoyo que registre cuando la bola entra.
 - Validar que el sistema funcione para diferentes ángulos o velocidades de entrada.
 - Desarrollar la transición al siguiente nivel.
 - Configurar un sistema que, al detectar que la bola está en el hoyo, cargue el siguiente nivel.
 - Si el juego tiene una estructura de niveles con dificultad creciente, asegurarse de que el progreso se registre adecuadamente.

- Programar una animación, sonido o efecto visual que confirme que el jugador ha metido la bola en el hoyo.
- Mostrar un mensaje o transición que indique el avance al siguiente nivel.
- Probar el movimiento de la bola y el sistema de colisiones para asegurarse de que la experiencia es fluida.
- Asegurarse de que la transición de nivel es estable y rápida.

Historia 2: Como jugador quiero realismo del juego para una experiencia mejor:

- Análisis del objetivo
 - Aumentar el realismo del juego para mejorar la experiencia de inmersión y satisfacción del jugador.
 - Este objetivo implica que los aspectos visuales, físicos y de interacción del juego deben sentirse lo más realistas posible, lo que podría incluir gráficos, física, sonidos, animaciones, y la forma en que el jugador interactúa con el entorno.
- Extracción de requisitos
 - La física de los objetos (como el movimiento, las colisiones, y la gravedad) debe reflejar cómo se comportan en la vida real, para que el juego se sienta auténtico.
 - Los modelos y texturas deben ser detallados y realistas.
 - Los movimientos de la bola deben ser fluidos y creíbles.
 - Los efectos de sonido deben coincidir con las acciones y el ambiente del juego.
 - El jugador debe poder mover la bola de manera realista.
 - La adición de realismo debe mejorar la experiencia sin hacer el juego más complicado o menos divertido.
- Extracción de tareas a realizar
 - Implementar física avanzada
 - Aplicar motores de física realistas para simular movimientos de objetos, colisiones, rebotes, y gravedad.
 - Ajustar la fricción, la velocidad y otras variables para reflejar el comportamiento de los materiales en la vida real.
 - Usar modelos 3D de alta calidad y texturas detalladas para los elementos clave del juego.
 - Incorporar animaciones de transición, para que los cambios de movimiento o posición no se sientan bruscos.

- Integrar sonidos para cada acción relevante (por ejemplo, el sonido de una bola rodando o rebotando).
- Realizar pruebas de rendimiento para asegurar que el realismo no afecta la jugabilidad.
- Probar el juego para asegurar que las mejoras en el realismo realmente aumentan la inmersión sin afectar la diversión.
- Recoger feedback de los jugadores para ajustar el realismo según las expectativas y preferencias.

Historia 3: Como jugador quiero tener la opción de jugar multijugador o contra la máquina para competir.

- **Análisis del Objetivo**

- Permitir a los jugadores competir en un modo multijugador donde puedan enfrentarse tanto a otros jugadores como a la CPU (inteligencia artificial).
- Este objetivo implica que el juego debe integrar un sistema multijugador que incluya tanto jugadores humanos como oponentes controlados por la CPU. Los oponentes de la CPU deben comportarse de manera desafiante y ofrecer una experiencia competitiva comparable a la de jugar contra otro jugador humano.

- **Extracción de Requisitos**

- La CPU debe ser capaz de actuar de forma autónoma, participando en la competencia en función de las reglas del juego.
- Los oponentes de la CPU deben comportarse de forma desafiante para ofrecer una experiencia similar a jugar contra un humano.
- Los jugadores deben poder interactuar y competir con la CPU en el mismo entorno y bajo las mismas reglas.
- El juego debe incluir un sistema de puntuación que permita medir el desempeño de los jugadores y la CPU.
- Al final de la partida, debe determinarse un ganador, ya sea un jugador humano o la CPU.
- La inteligencia artificial debe ofrecer una competencia justa sin ser demasiado fácil o difícil, para que la experiencia sea desafiante y divertida.
- El sistema debe permitir que varias instancias de la CPU y jugadores humanos se unan a la partida sin problemas.

- Extracción de Tareas a Realizar
 - Sincronizar las acciones de los jugadores y la CPU en tiempo real para mantener la coherencia de la partida.
 - Crear un sistema de IA para que los oponentes controlados por la CPU puedan competir en el juego, siguiendo las mismas reglas que los jugadores humanos.
 - Ajustar la dificultad de la IA para que se adapte al nivel de habilidad deseado.
 - Configurar la partida para incluir oponentes controlados por la CPU en el modo multijugador.
 - Asegurarse de que la IA interactúe y compita correctamente con los jugadores humanos.
 - Implementar un sistema de puntuación que mida el desempeño tanto de los jugadores humanos como de la CPU.
 - Establecer reglas para determinar el ganador de la partida, ya sea un jugador humano o la CPU.
 - Probar diferentes configuraciones de dificultad para garantizar que la IA sea desafiante pero justa.
 - Asegurarse de que el juego funcione fluidamente al incluir tanto jugadores humanos como oponentes de la CPU en la misma partida.
 - Realizar pruebas de carga para confirmar que el juego puede manejar varias conexiones y procesos de IA.