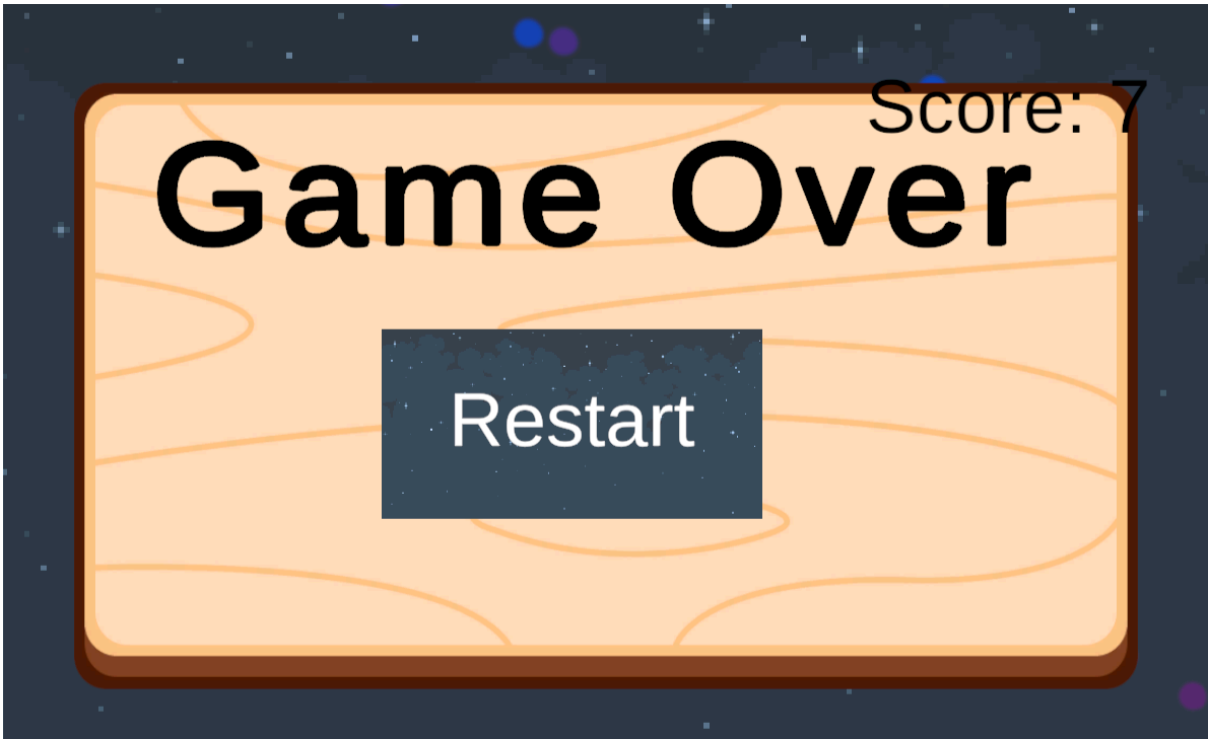
Diego García González A01198976
07/06/25LIBERTY BIRD by DiegoGarcia_G https://diegogarcia-g.itch.io/integradora

Descripción General del Videojuego

El presente documento expone el diseño, funcionamiento y características técnicas de un videojuego desarrollado en Unity, cuyo objetivo principal es brindar una experiencia interactiva sencilla pero dinámica. El juego tiene como protagonista a un pájaro animado

que debe esquivar obstáculos en forma de jaulas a lo largo de un entorno de desplazamiento continuo.



Mecánicas del Juego

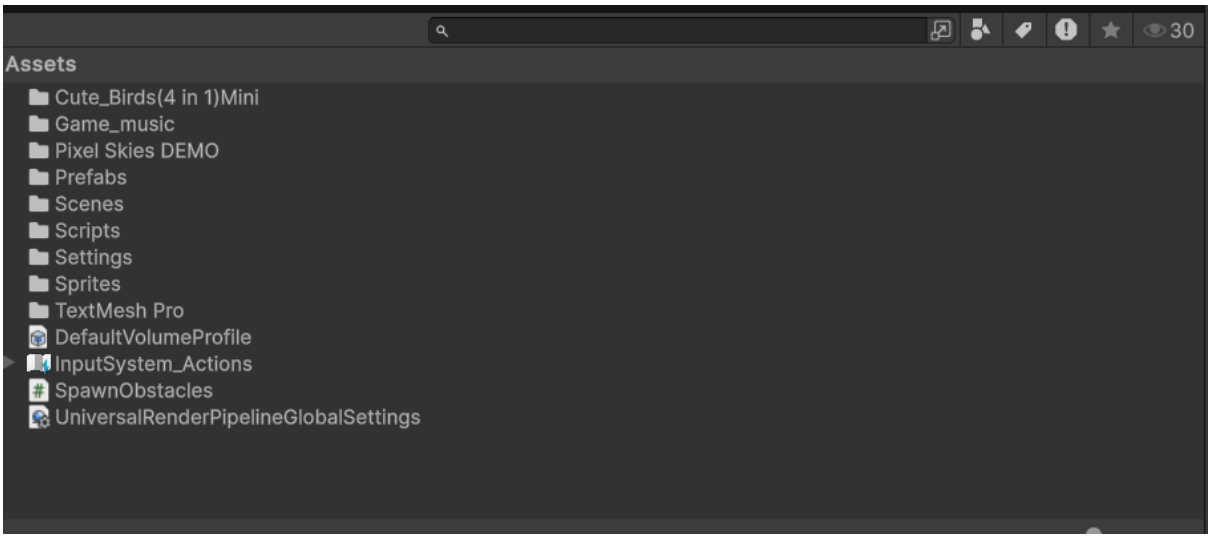
El jugador controla un pájaro que puede moverse exclusivamente en dirección vertical (hacia arriba y abajo), utilizando las teclas del teclado asignadas para tal fin. La movilidad del personaje se logra mediante el uso del componente `Rigidbody2D`, lo que permite un movimiento suave y controlado dentro de los límites del área visible de la pantalla.

Los obstáculos que deben evitarse son jaulas que se generan de forma aleatoria y se desplazan horizontalmente desde el borde derecho de la pantalla hacia el izquierdo. Este comportamiento se logra mediante instanciación periódica y controlada de objetos con movimiento predefinido, proporcionando al jugador una experiencia de desafío progresivo.

Entorno Visual y Animaciones

El fondo del juego está compuesto por una imagen o animación que se desplaza continuamente en bucle infinito, simulando un escenario sin fin. Este sistema permite mantener al jugador inmerso en una experiencia de desplazamiento continuo sin interrupciones.

Además, se han incorporado efectos de partículas que aparecen de forma constante en el fondo, contribuyendo a la ambientación visual del juego y reforzando su estética dinámica y animada.



Organicé mis carpetas de estas maneras.

Interfaz de Usuario

Durante el juego, en la parte superior de la pantalla se muestra un contador de puntaje o **score**, el cual se actualiza automáticamente cada segundo, en función del tiempo que el jugador logra mantenerse con vida esquivando los obstáculos. Este valor se visualiza mediante un componente de texto, que incrementa en tiempo real para ofrecer retroalimentación inmediata al usuario.

Cuando el jugador colisiona con uno de los obstáculos, el personaje es destruido o desactivado, y se activa un panel de madera con la leyenda “**Game Over**”. Este panel, diseñado con un estilo visual coherente con el resto del entorno gráfico, incluye también un botón funcional de **"Restart"** que permite reiniciar la escena actual, facilitando así la reanudación inmediata del juego sin necesidad de regresar al menú principal.

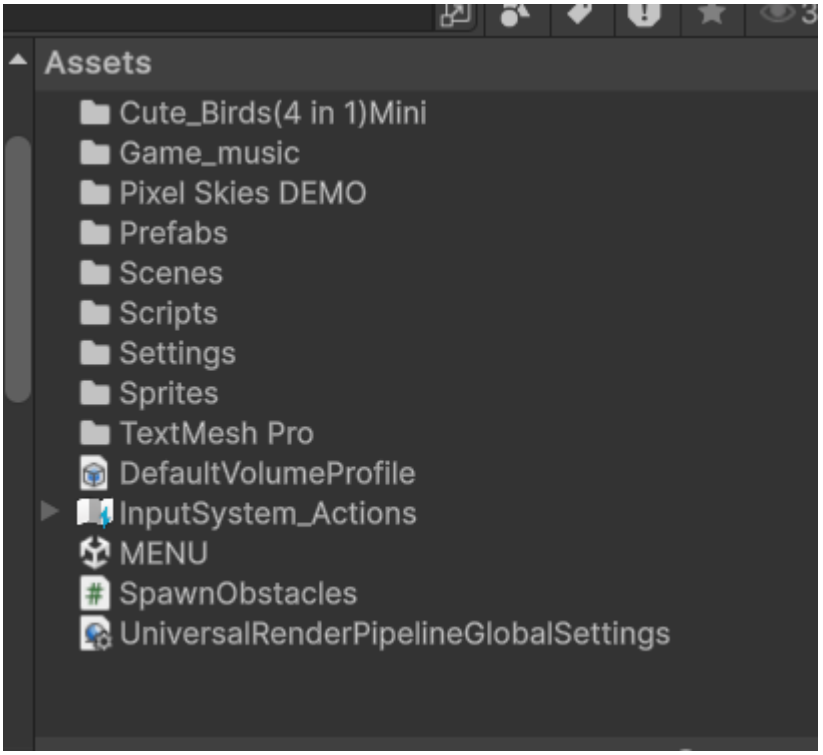
Sonido y Ambientación

El videojuego incluye una pista de música de fondo que se reproduce durante toda la sesión de juego. Esta música ha sido cuidadosamente seleccionada para complementar el ritmo y la atmósfera general del entorno, contribuyendo a una experiencia envolvente. Asimismo, se pueden incluir efectos sonoros opcionales para colisiones u otras interacciones, aunque esto dependerá de las decisiones finales de diseño.

Conclusión

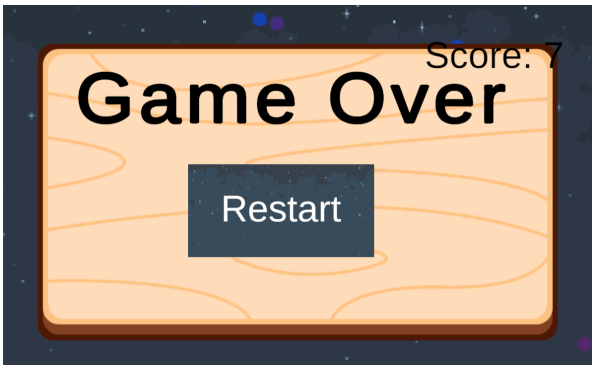
Este videojuego representa una implementación funcional de un minijuego tipo “arcade” en Unity, con un sistema de control intuitivo, mecánicas de generación aleatoria de obstáculos, entorno visual atractivo y una interfaz clara. Su estructura modular permite futuras mejoras, como la incorporación de niveles, distintos tipos de obstáculos, recompensas, rankings o integraciones con bases de datos externas para el almacenamiento de puntuaciones.

1.Estructura del proyecto y organización: Carpetas Scenes, Scripts, Sprites, Prefabs, Animations, etc…



Se separó en diferentes carpetas los scripts, sprites, prefabs, etc

2.Uso de escenas y navegación: Mínimo 3 escenas: Menú principal, Nivel del Juego, Pantalla Victoria o Derrota. Y transición entre escenas.



3.Implementación de Parallax: Fondo con efecto parallax en el nivel principal.

El fondo se hizo con parallax pero tmb un script para que fuera fondo infinito, y el juego no tenga fin.

4.Uso correcto de Colliders: Uso de 2D colliders ya sea en personajes, enemigos, plataformas, etc.

Se usó colliders para detectar las colisiones con los obstáculos, también unos bordes arriba y abajo para que el personaje no pueda salir de escena y un collider afuera de la escena que va destruyendo las jaulas que ya no están en escena.

5.Interfaz de usuario: Uso de mínimo 2 tipos de components de UI: Botones, Text, Input Text, Sliders, etc.

Se usaron botones y text

6.Lectura de Teclado o Mouse: Eventos o Movimiento de sprites con teclado

Se leen las flechas del teclado.

7.Código: Scripts bien estructurados y comentados.

8.Animation y Animator: Implementa como mínimo dos animaciones.

Sí del pájaro son 2 y encima animaciones de las partículas y fondo.

9.Generación y uso de Prefabs: Genera Prefabs que sean utilizados en tu juego.

Generé un prefab del obstáculo

10.Spawner: Crear un game object encargado de crear objectos en tu juego.

Hice un spawner de obstáculos, crea obstáculos en lugares random.

11.APIs: Invocación de APIs en el Juego.

Se puede invocar a un api para guardar el puntaje del usuario pero en ITCHIO no, solo en despliegue web.

12.Despliegue del video juego en Web.

En la web se puede guardar puntaje en ITCHIO no esta esa funcionalidad.