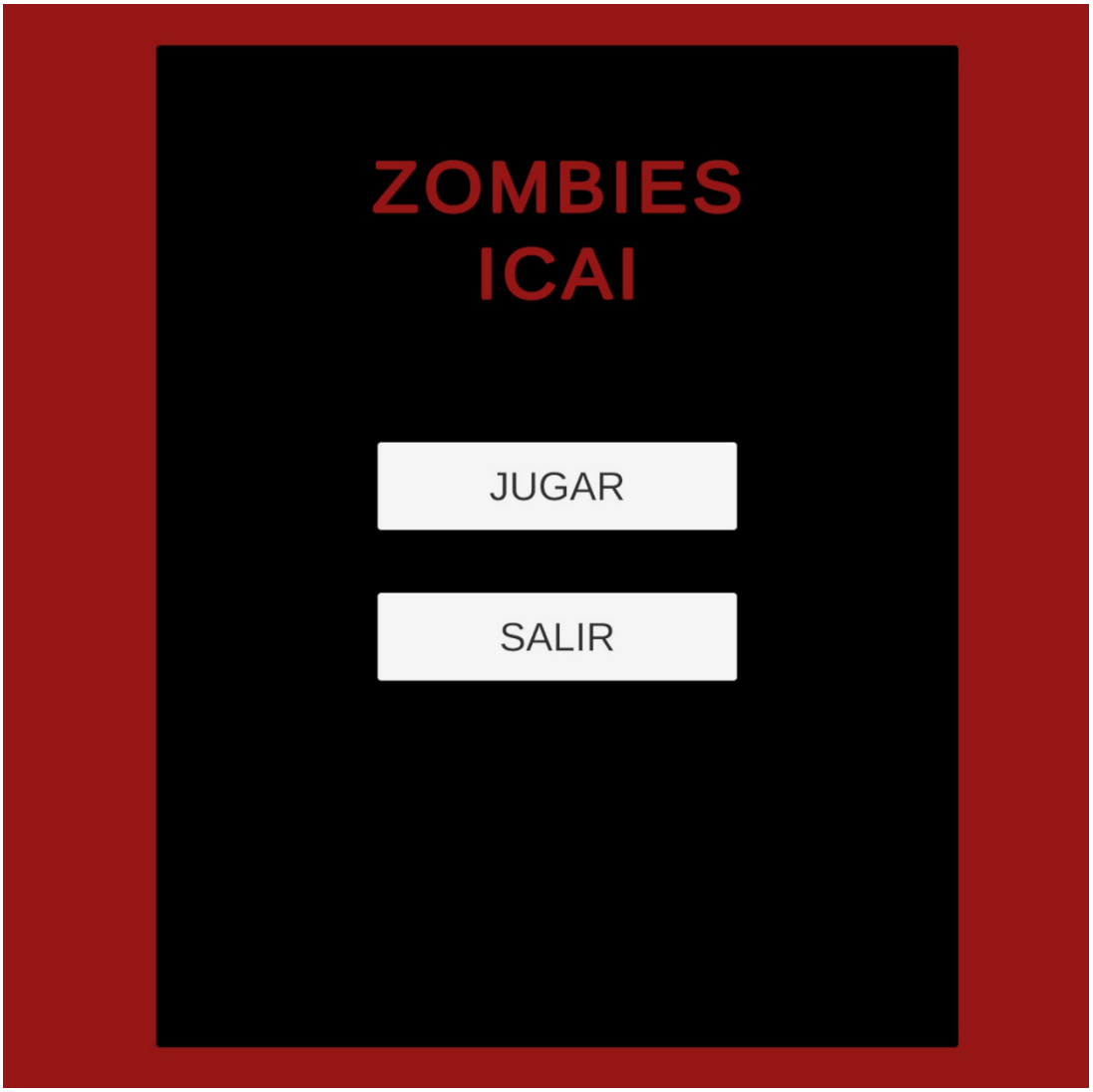
VIDEOJUEGO
PROYECTO FINAL



Pablo Rodríguez González y Álvaro Palomar Candel

INTRODUCCIÓN

Este proyecto consiste en el desarrollo de un videojuego de disparos en primera persona utilizando el motor Unity. El objetivo principal del juego es la supervivencia del jugador frente a oleadas sucesivas de enemigos, incrementando progresivamente la dificultad conforme avanza la partida (más zombies, con más daño, más vida y más rápidos), basándonos en la modalidad de zombies del conocido juego Call of Duty.

Durante el transcurso del juego, el jugador puede eliminar enemigos para obtener puntos, los cuales pueden ser utilizados en una tienda para mejorar diferentes atributos, como el daño del arma o la velocidad de movimiento. El sistema de rondas permite estructurar la progresión del juego, ofreciendo una experiencia dinámica y creciente en dificultad.

El proyecto ha sido desarrollado aplicando conceptos de programación orientada a objetos y siguiendo una arquitectura basada en componentes, propia del entorno Unity. Se han implementado distintos sistemas independientes, como la gestión de enemigos, el control de rondas, la interfaz de usuario y la gestión del estado de la partida, con el objetivo de mantener un código organizado, escalable y fácil de mantener.

CONTROLES:

WASD para movernos hacia delante, detrás, derecha e izquierda.

SHIFT para correr y aumentar la velocidad.

Movimiento de la cámara a través del ratón.

E para abrir la tienda al acercarse (hay que estar dentro del rigidbody que simula el área que rodea la tienda y marca desde donde se puede abrir).

Click izquierdo para disparar.

ESC para abrir el menú de pausa.

Movimiento del ratón para seleccionar la opción que queremos en los distintos menús (inicial, pausa, tienda, etc)

METODOLOGÍA DE DESARROLLO

El desarrollo del videojuego se ha llevado a cabo siguiendo una metodología incremental e iterativa, la cual nos ha permitido construir el proyecto de forma progresiva, comenzando por las funcionalidades básicas y ampliándolas gradualmente hasta alcanzar un producto funcional y completo.

En una primera fase establecimos la estructura general del proyecto, definiendo la escena principal de juego, el jugador y el entorno básico. Configuramos el sistema de cámara en primera persona y el movimiento del jugador, asegurando una base sólida sobre la que desarrollar el resto de mecánicas. Durante esta etapa realizamos pruebas constantes en el editor de Unity para verificar el correcto funcionamiento de los controles y la interacción con el escenario.

Posteriormente, implementamos el sistema de combate, centrado en la mecánica de disparo mediante raycast. Esta decisión nos permitió simplificar el sistema de armas, evitando la complejidad de la simulación de proyectiles físicos y facilitando la detección precisa de impactos sobre los enemigos. A su vez, desarrollamos el sistema de vida del jugador y de los enemigos, incorporando la gestión del daño, la muerte y las animaciones asociadas.

En una fase posterior diseñamos el sistema de enemigos, utilizando prefabs para garantizar la reutilización de los componentes y mantener la coherencia entre las instancias generadas en tiempo de ejecución. Integramos una inteligencia artificial mediante el uso de NavMeshAgent, permitiendo que los enemigos se desplazaran de forma autónoma hacia el jugador. Este sistema fue ajustado y depurado mediante múltiples pruebas, resolviendo problemas relacionados con la navegación, la detección del jugador y la correcta configuración de los prefabs.

Una vez establecida la mecánica principal, desarrollamos el sistema de rondas, encargado de controlar la aparición progresiva de enemigos y el aumento de la dificultad del juego. Para ello, implementamos un gestor de rondas que coordina el inicio y el final de cada oleada, apoyándose en eventos para detectar la eliminación de todos los enemigos activos. Este enfoque nos permitió desacoplar la lógica de los enemigos del control global del juego, mejorando la organización y mantenibilidad del código.

De forma paralela, implementamos el sistema de puntuación y la tienda del juego. El jugador obtiene puntos al eliminar enemigos, los cuales se gestionan a través de un sistema centralizado. La tienda permite intercambiar estos puntos por mejoras como el aumento del daño o la velocidad de movimiento, integrándose mediante un sistema de triggers que detecta la proximidad del jugador. Para esta funcionalidad utilizamos el sistema de físicas de Unity,

incorporando componentes Rigidbody y Colliders que garantizan una detección fiable de las interacciones.

Asimismo, desarrollamos la interfaz de usuario de manera progresiva, mostrando en pantalla información relevante como la vida del jugador, la ronda actual y los puntos disponibles. También añadimos pantallas específicas para la pausa del juego y el fin de la partida, mejorando la experiencia del usuario y proporcionando un flujo de juego completo. Ajustamos la interfaz tras múltiples pruebas para asegurar su correcta visualización y actualización en tiempo real.

Durante todo el proceso de desarrollo realizamos pruebas continuas y correcciones de errores, especialmente relacionados con la configuración de componentes en prefabs, la asignación de referencias entre scripts y la gestión de eventos. Abordamos estos problemas de forma sistemática, comprobando la correcta integración de cada sistema antes de avanzar en nuevas funcionalidades.

Finalmente, llevamos a cabo tareas de limpieza y organización del proyecto, eliminando recursos innecesarios y estructurando adecuadamente las carpetas de scripts, prefabs y assets. Este paso nos permitió mejorar la mantenibilidad del proyecto y facilitar su comprensión tanto a nivel de código como de estructura general.

Problemas afrontados durante el desarrollo y cambios realizados

A lo largo del desarrollo del proyecto nos enfrentamos a diversos problemas técnicos y de diseño que requirieron ajustes progresivos en la implementación inicial. Estos problemas formaron parte del proceso natural de desarrollo y permitieron mejorar tanto la estabilidad como la organización del código.

Uno de los principales problemas estuvo relacionado con la configuración de los prefabs, especialmente en el caso de los enemigos generados dinámicamente. En varias ocasiones, los prefabs no incluían todos los componentes necesarios, como agentes de navegación, colliders o referencias a scripts, lo que provocaba comportamientos incorrectos durante la ejecución del juego. Para solucionar este problema, revisamos y unificamos la configuración de los prefabs, asegurando que todas las instancias compartieran la misma estructura y componentes.

También surgieron dificultades en la sincronización entre la lógica del juego y las animaciones de los modelos, principalmente en los estados de muerte y movimiento de los enemigos. Estos problemas se resolvieron ajustando la gestión de estados y los disparadores de animación, garantizando que las transiciones visuales reflejaran correctamente el estado lógico de cada entidad.

Otro problema relevante fue la detección de colisiones y disparos y sus representaciones visuales (como puede ser la sangre de los zombies). Inicialmente se produjeron impactos imprecisos debido a una mala alineación entre los colliders y los modelos gráficos. Para

corregirlo, simplificamos los colliders y ajustamos su tamaño y posición, mejorando la coherencia entre la representación visual y la lógica de combate basada en raycast.

Por último, se detectaron problemas relacionados con la organización del proyecto y la acumulación de recursos innecesarios. Para resolverlos, realizamos una limpieza del proyecto y una reorganización de carpetas, lo que facilitó el mantenimiento y la comprensión del código.

En conjunto, los problemas afrontados y los cambios realizados contribuyeron a obtener una versión final más estable, coherente y alineada con los objetivos del proyecto