

nombre:

Oscar Zúñiga, andre piffaut, jesus vidal, francisco zamorano

# 1.introduccion

Antiverso es un videojuego 2D estilo metroidvania, ambientado en un universo oscuro, terrorífico y decadente. En este mundo sin esperanza, el jugador encarna a Osttinger, un ente sombrío atrapado en una dimensión alternativa conocida como el Antiverso.  
  
La historia gira en torno a la confrontación entre Osttinger y su gemelo, Krivering, un ente blanco que representa su contraparte y quien amenaza con borrar su existencia. Ambos poseen habilidades únicas y poderosas, y están destinados a enfrentarse en una lucha definitiva que trasciende la materia y el tiempo.  
  
A lo largo del juego, el jugador deberá avanzar por diferentes etapas ubicadas en un planeta en ruinas, enfrentando enemigos como Quimeras y Goliats, desbloqueando habilidades especiales y enfrentando jefes en combates desafiantes. Cada etapa forma parte del mismo mapa general, permitiendo backtracking al adquirir nuevas habilidades.  
  
El objetivo principal del videojuego es ayudar a Osttinger a sobrevivir y superar las pruebas del Antiverso, culminando en un enfrentamiento con Krivering, su hermano y reflejo antagónico, en una batalla que definirá el destino de ambos.  
  
En términos técnicos, el proyecto busca implementar conocimientos clave en el desarrollo de videojuegos con Unity, tales como:  
- Programación en C#  
- Manejo de físicas 2D y colisiones  
- Integración de assets visuales y sonoros  
- Control de animaciones con Animator  
- Diseño de UI funcional  
- Estructuración modular del proyecto

2.Requerimientos Técnicos

Motor de desarrollo: Unity 2022.X

Lenguaje de programación: C#

Assets utilizados:

- Sprites originales estilo pixel art (personaje principal Osttinger, enemigos, entorno)

- Escenario principal con ambientación oscura y temática espacial

- UI personalizada (menú principal con botones de iniciar y salir)

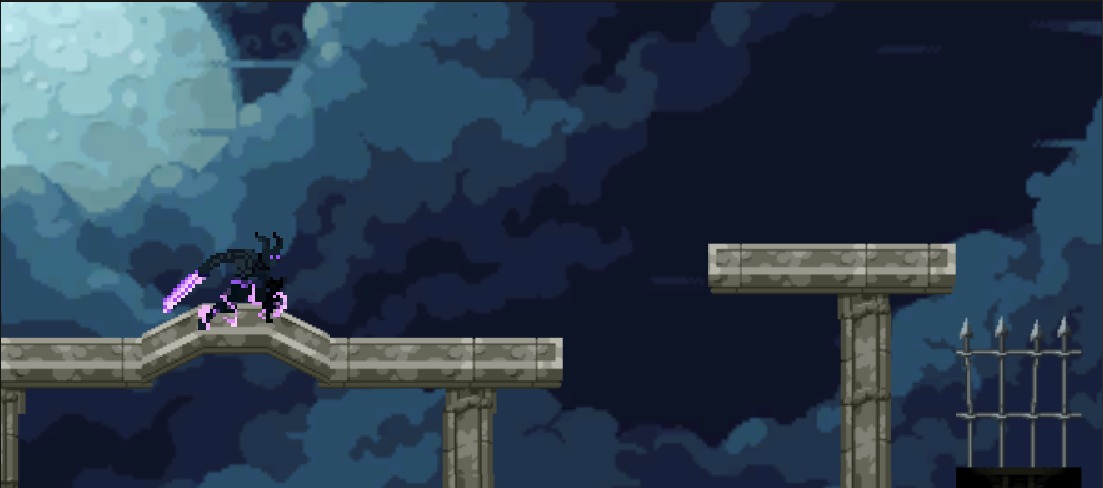
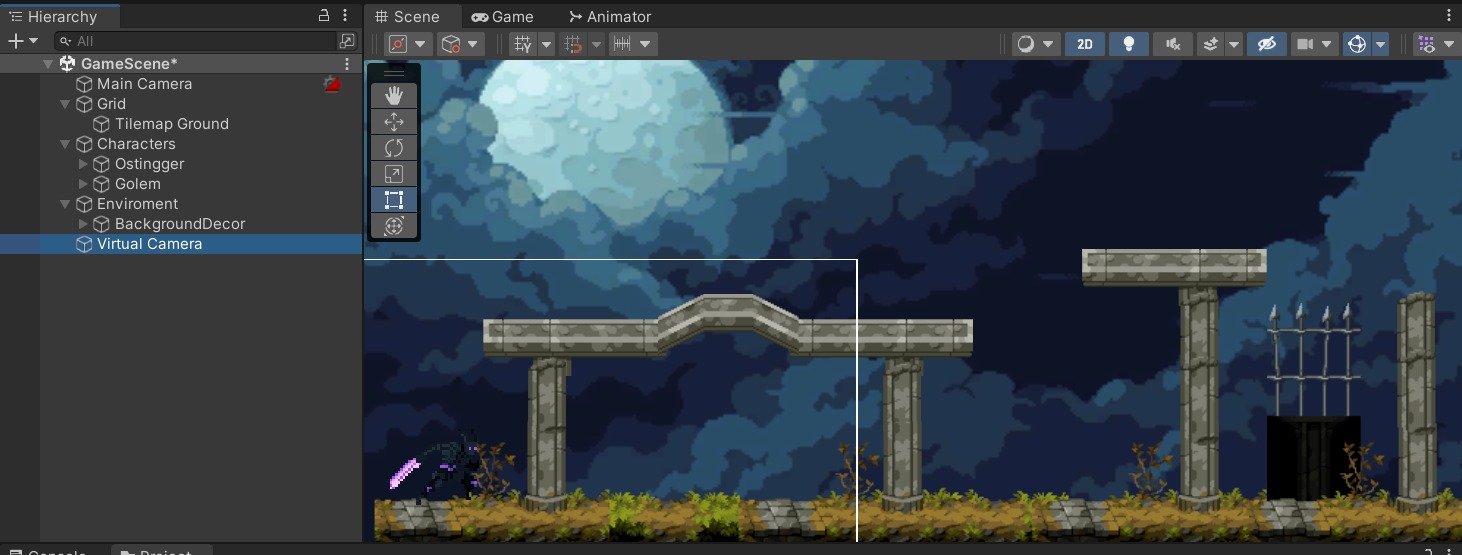
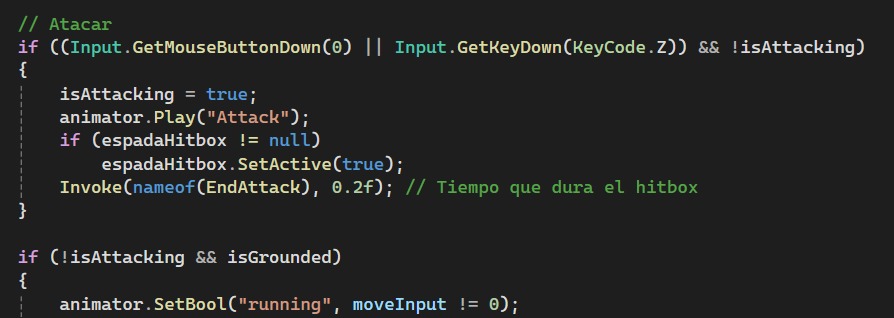
Plataforma de prueba: Windows PC

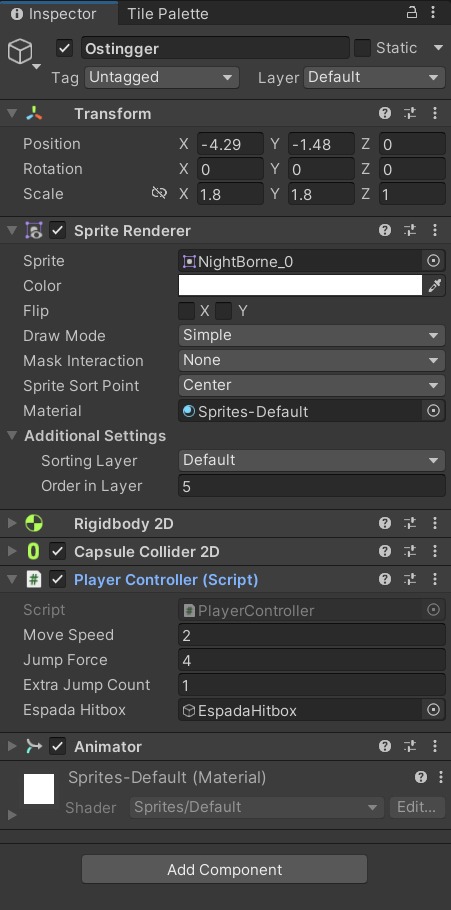
# 3. Desarrollo del Proyecto

Estructura del proyecto en Unity:  
Assets/  
├── Scenes/  
│ └── MainScene.unity  
├── Scripts/  
│ ├── PlayerController.cs  
│ ├── EnemyController.cs  
│ └── CollisionManager.cs  
├── Sprites/  
│ └── Osttinger, Krivering, enemigos, escenario  
├── Prefabs/  
│ └── Personaje principal, enemigos, plataformas  
├── UI/  
│ └── Menú principal con botones funcionales  
  
Principales GameObjects creados:  
- Player (Osttinger): incluye Rigidbody2D, Animator, BoxCollider2D y control de animaciones.  
- Krivering (enemigo en etapa 0 y jefe final): movimiento básico y ataques.  
- Quimera: enemigo volador que dispara.  
- Goliat: enemigo terrestre que golpea.  
- Elementos del entorno: plataformas, colisionadores, checkpoints.  
- Menú principal: con botones de "Iniciar Juego" y "Salir".  
  
Programación básica implementada:  
- PlayerController.cs: movimiento, salto y ataque del jugador.  
- EnemyController.cs: comportamiento de enemigos simples.  
- CollisionManager.cs: detección de colisiones con enemigos y escenario.  
  
Mecánica básica del juego:  
- El jugador puede moverse, saltar y atacar.  
- Hay un sistema de daño por colisión con enemigos.  
- El juego inicia con un tutorial (Etapa 0) donde se enfrentan por primera vez Osttinger y Krivering.  
- El objetivo es avanzar a través de un mundo fragmentado, derrotar enemigos y desbloquear nuevas zonas mediante habilidades.

**4. Capturas de Pantalla**

Aquí debes incluir **4 capturas obligatorias**:

1. **Escena principal del juego** (lo que se ve al jugar).
2. **Jerarquía (Hierarchy)** con los objetos en la escena.
3. **Código relevante** (un fragmento de un script importante como PlayerController.cs o manejo de colisiones).
4. **Inspector** de uno de los GameObjects principales (por ejemplo, el jugador).



# 5. Reflexión y Autoevaluación

Durante el desarrollo de este proyecto aprendimos a estructurar un videojuego en Unity desde cero, incluyendo el manejo de GameObjects, animaciones, scripts en C# y diseño de interfaces. También profundizamos en cómo organizar un proyecto de forma modular, reutilizando prefabs y scripts para distintos elementos del juego.  
  
Uno de los principales desafíos fue lograr que las animaciones del personaje Osttinger se sincronizaran correctamente con sus acciones como caminar, saltar y atacar. Además, ajustar el movimiento de los enemigos y detectar colisiones precisas entre el jugador y los obstáculos fue un proceso de ensayo y error que requirió bastante depuración.

También fue clave aprender a implementar un menú funcional con botones interactivos y a manejar correctamente la jerarquía de objetos en Unity, lo que permitió mantener el proyecto organizado.