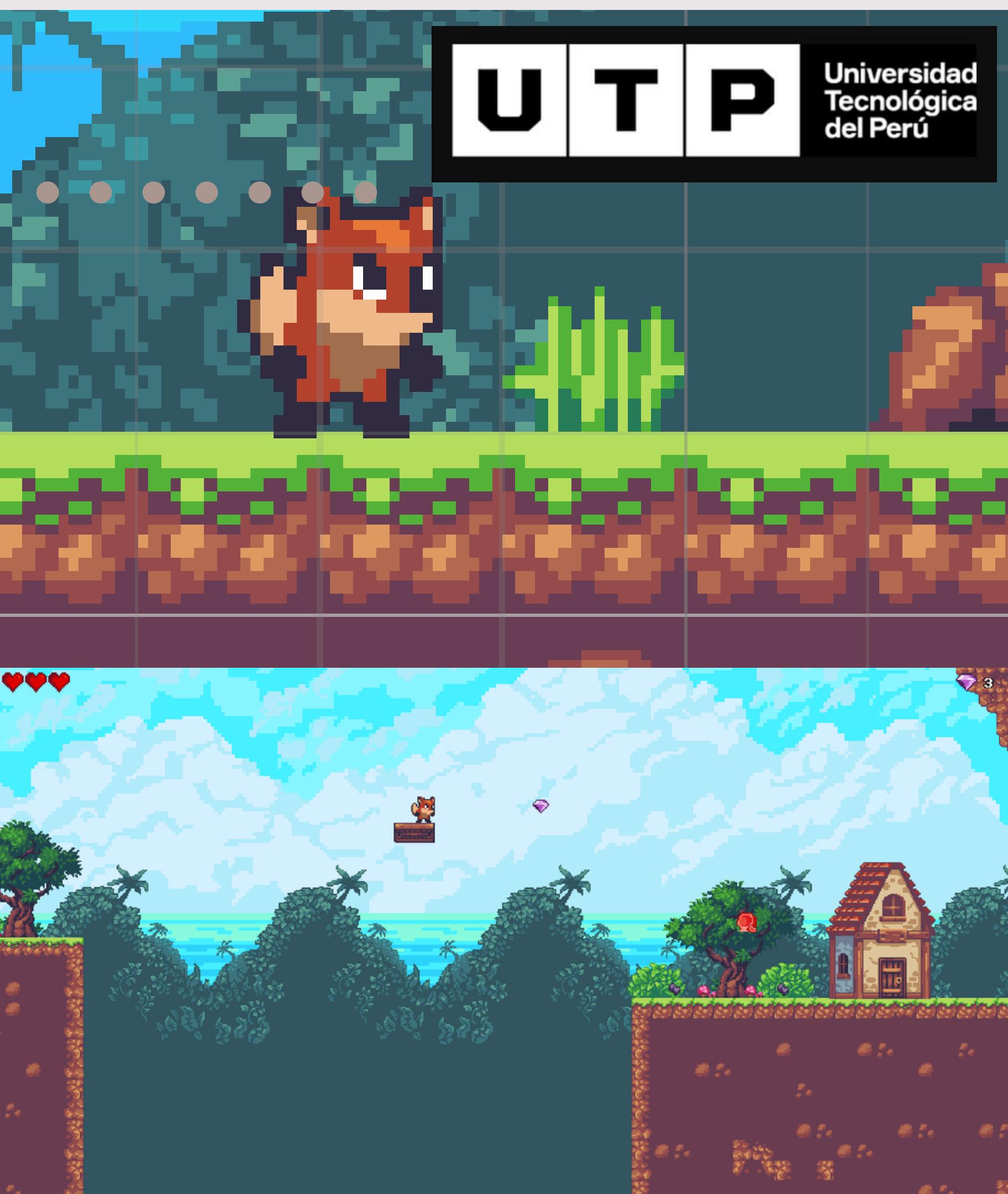




GAME FOX

INTEGRANTES :

- Jefferson Martin Cruz Flores
- Renzo Gustavo Reyes Carbajal
- Rogelio Espinoza Cabezas



TECNOLOGÍAS



Unity



*Visual
Studio 2022*



GitHub Desktop

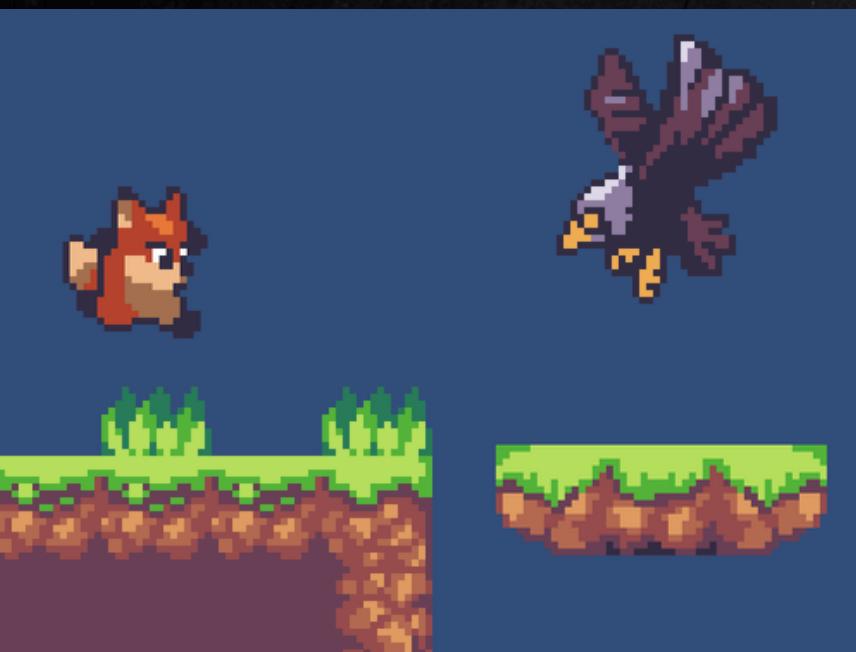
Descripción General:

- FoxGame es un juego de plataformas en 2D donde el jugador controla Foxi el zorro, que está perdido y debe avanzar a través de niveles llenos de obstáculos, enemigos y plataformas móviles. El objetivo es llegar al final de cada nivel evitando caer en trampas o ser derrotado, para que el personaje logre llegar a casa con su familia.



Características Principales:

- **Movimiento básico:** Correr a izquierda y derecha, saltar, y doble salto opcional.
- **Física sencilla:** Gravedad constante, colisiones con plataformas y enemigos.
- **Enemigos:** Enemigos que patrullan zonas específicas y dañan al jugador al tocarlo.
- **Recolectables:** Monedas o gemas dispersas en el nivel para sumar puntos.
- **Plataformas móviles:** Plataformas que se mueven en dirección horizontal o vertical.
- **Sistema de vidas:** El jugador tiene 3 vidas; pierde una si cae o es dañado.
- **Meta del nivel:** Llega a la puerta de salida o bandera para completar el nivel.



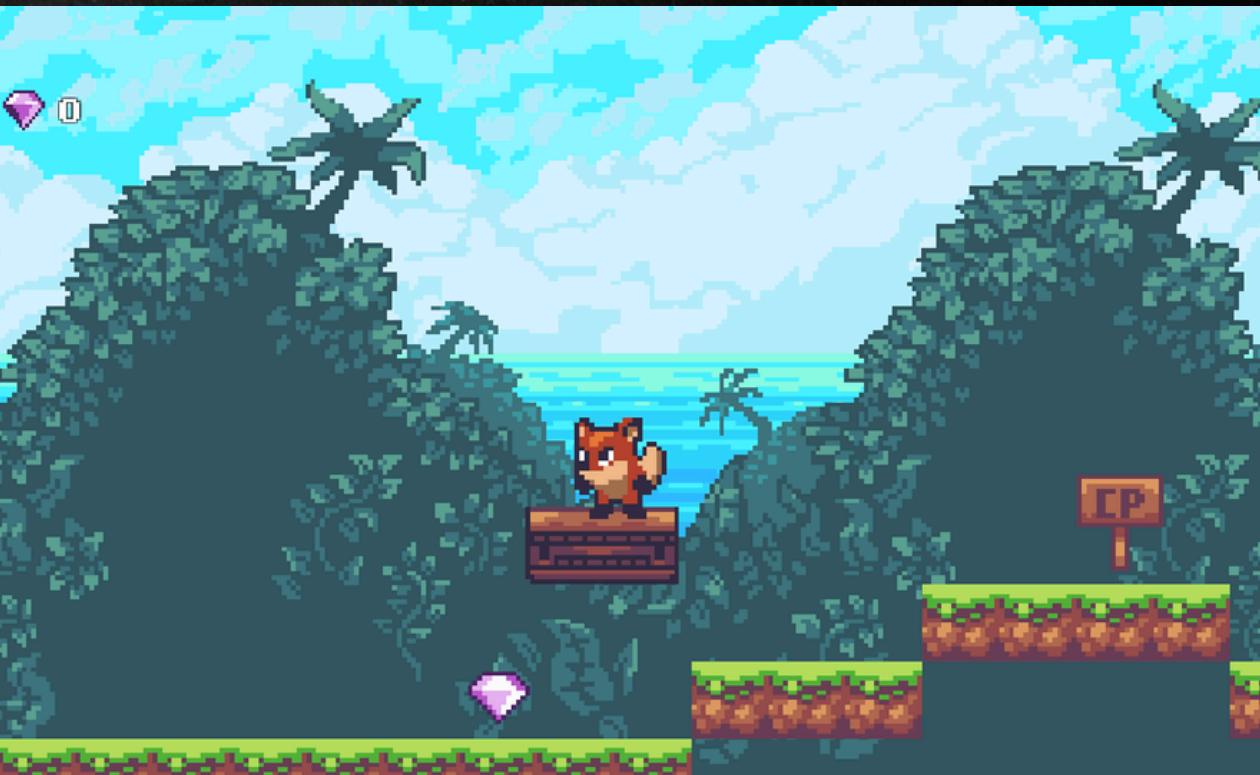
Entrega 1: Configuración Inicial y Movimiento del Jugador

Objetivo:

- Establecer las bases del proyecto y la jugabilidad básica.

Contenido:

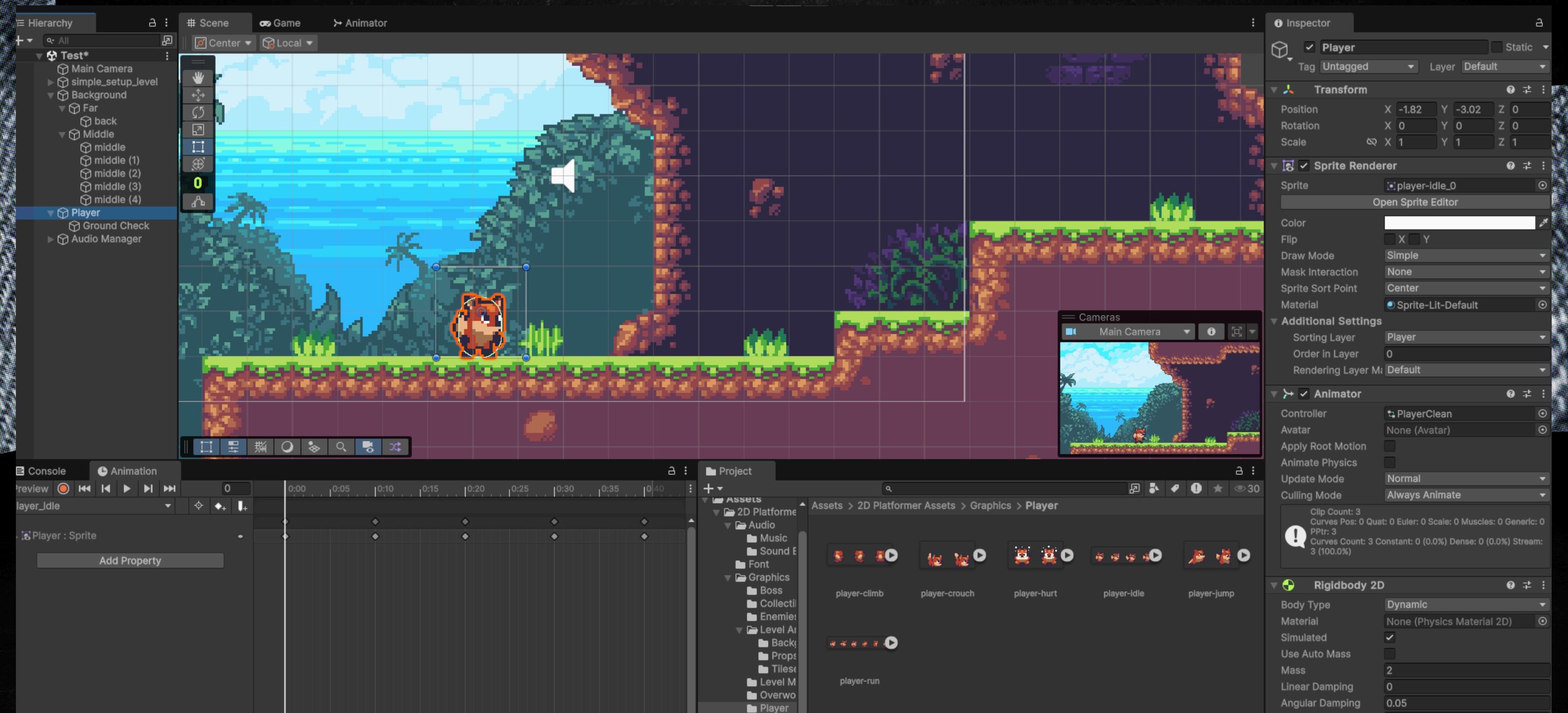
- Instalación de Unity y creación del proyecto.
- Configuración del entorno 2D.
- Importación de assets (sprites, tilesets, etc.).
- Creación del jugador:
 - Animaciones (idle, walk, jump).
 - Movimiento horizontal.
 - Salto básico.
- Diseño del primer nivel sencillo con plataformas estáticas.



Funcionalidades:

- Control del personaje con teclado.
- Animación sincronizada con el movimiento.
- Gravedad y detección de suelo.

CAPTURAS DEL JUEGO



```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  // Script de Unity (1 referencia de recurso) | 0 referencias
6  public class PlayerController : MonoBehaviour
7  {
8      [Header("Movimiento")]
9      public float moveSpeed;
10
11     [Header("Salto")]
12     private bool canDoubleJump;
13     public float jumpForce;
14
15     [Header("Componentes")]
16     public Rigidbody2D theRB;
17
18     [Header("Animator")]
19     private Animator anim;
20     private SpriteRenderer theSR;
21
22     [Header("Grounded")]
23     private bool isGrounded;
24     public Transform groundCheckpoint;
25     public LayerMask whatIsGround;
26
27     // Mensaje de Unity | 0 referencias
28     void Start()
29     {
30         anim = GetComponent<Animator>();
31         theSR = GetComponent<SpriteRenderer>();
32     }
33     // Mensaje de Unity | 0 referencias
34     void Update()
35     {
36         theRB.linearVelocity = new Vector2(moveSpeed * Input.GetAxis("Horizontal"), theRB.li
37
38         isGrounded = Physics2D.OverlapCircle(groundCheckpoint.position, .2f, whatIsGround);
39
40         if (isGrounded)
41         {
42             canDoubleJump = true;
43         }
44
45         if(theRB.linearVelocity.x < 0)
46         {
47             theSR.flipX = true;
48         }else if(theRB.linearVelocity.x > 0)
49         {
50             theSR.flipX = false;
51         }
52
53         anim.SetFloat("moveSpeed", Mathf.Abs(theRB.linearVelocity.x));
54         anim.SetBool("isGrounded", isGrounded);
55     }
56 }
```

THANK
you