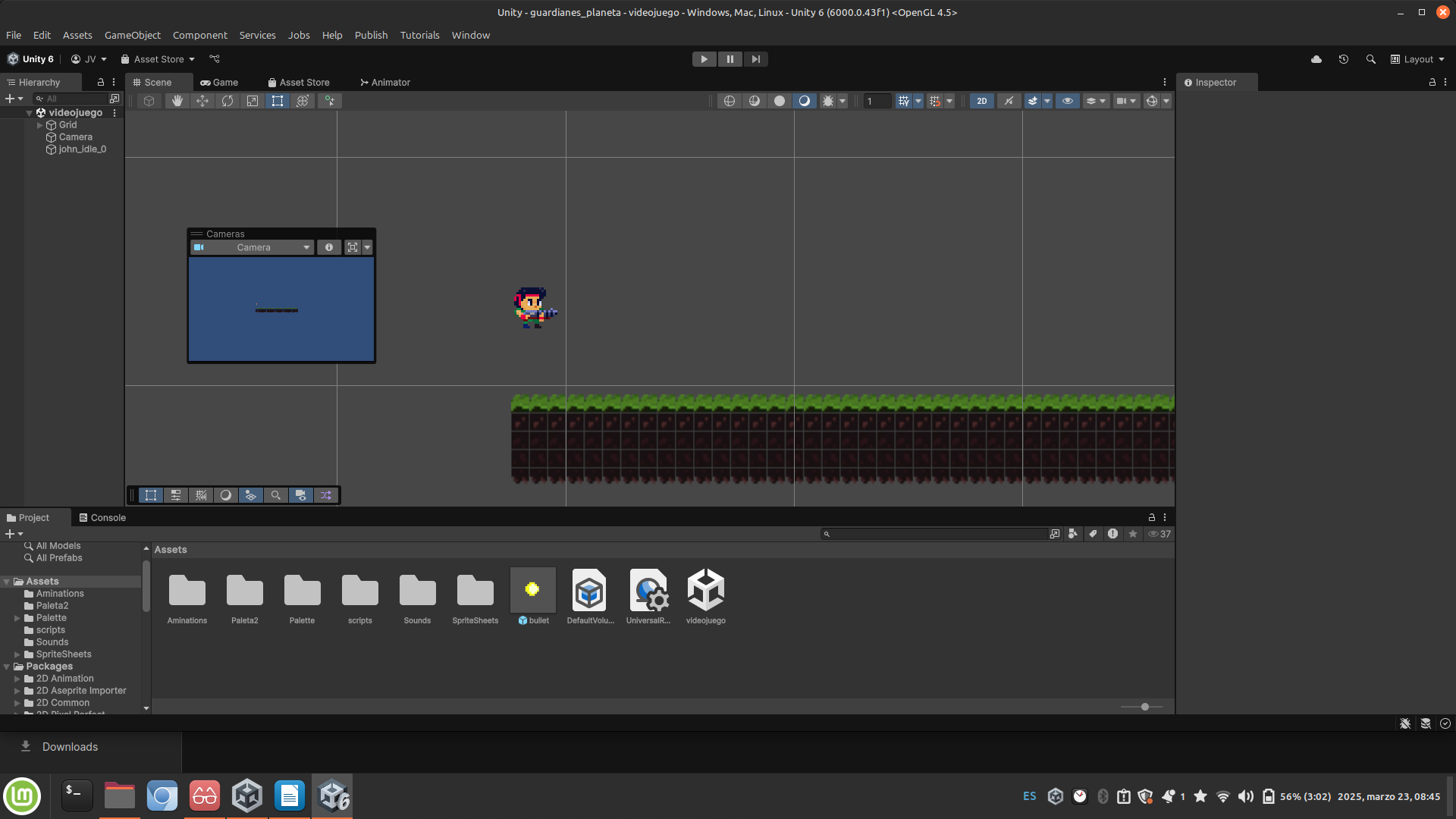
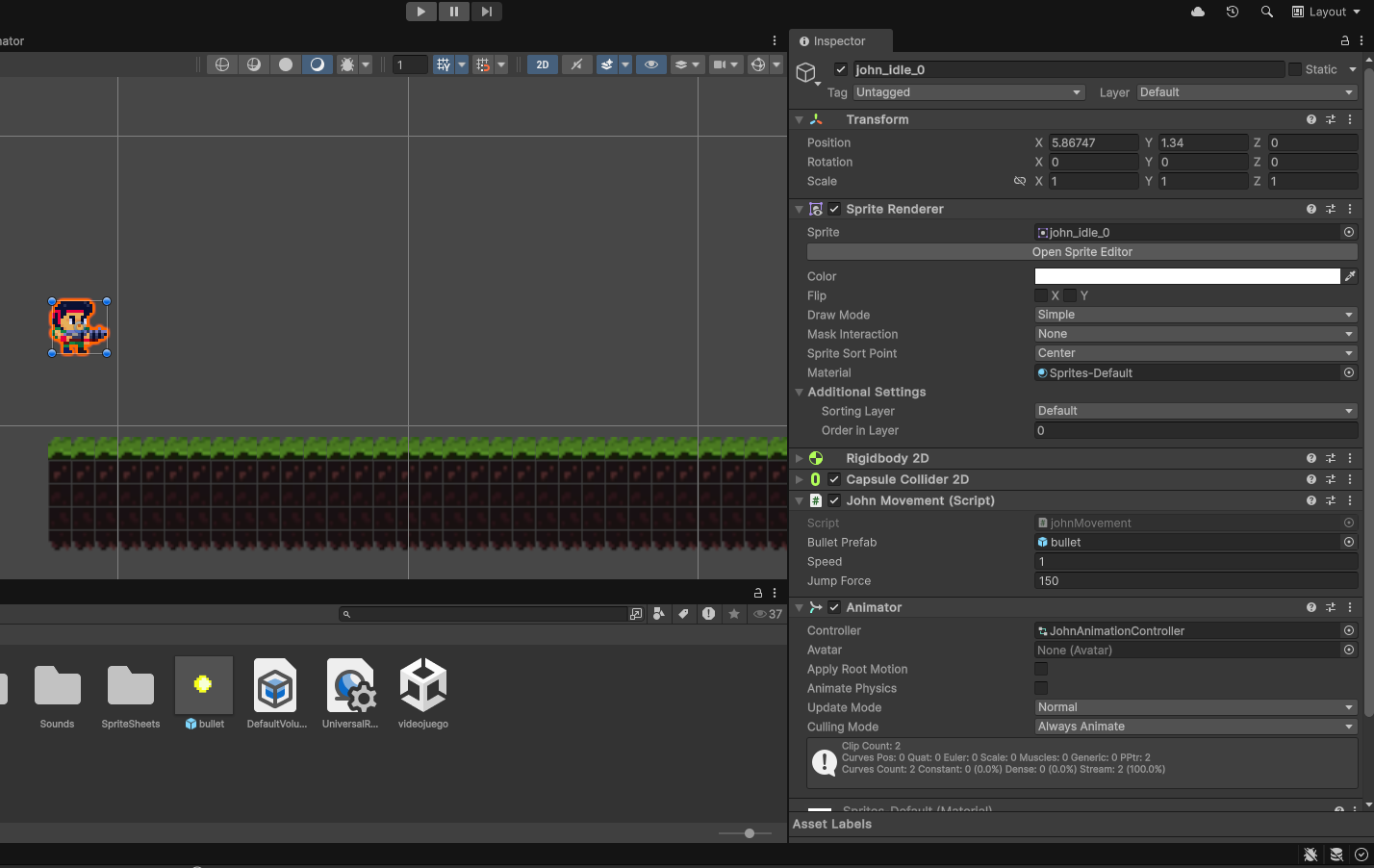
**Explicación del demo del juego Guardianes del Planeta**

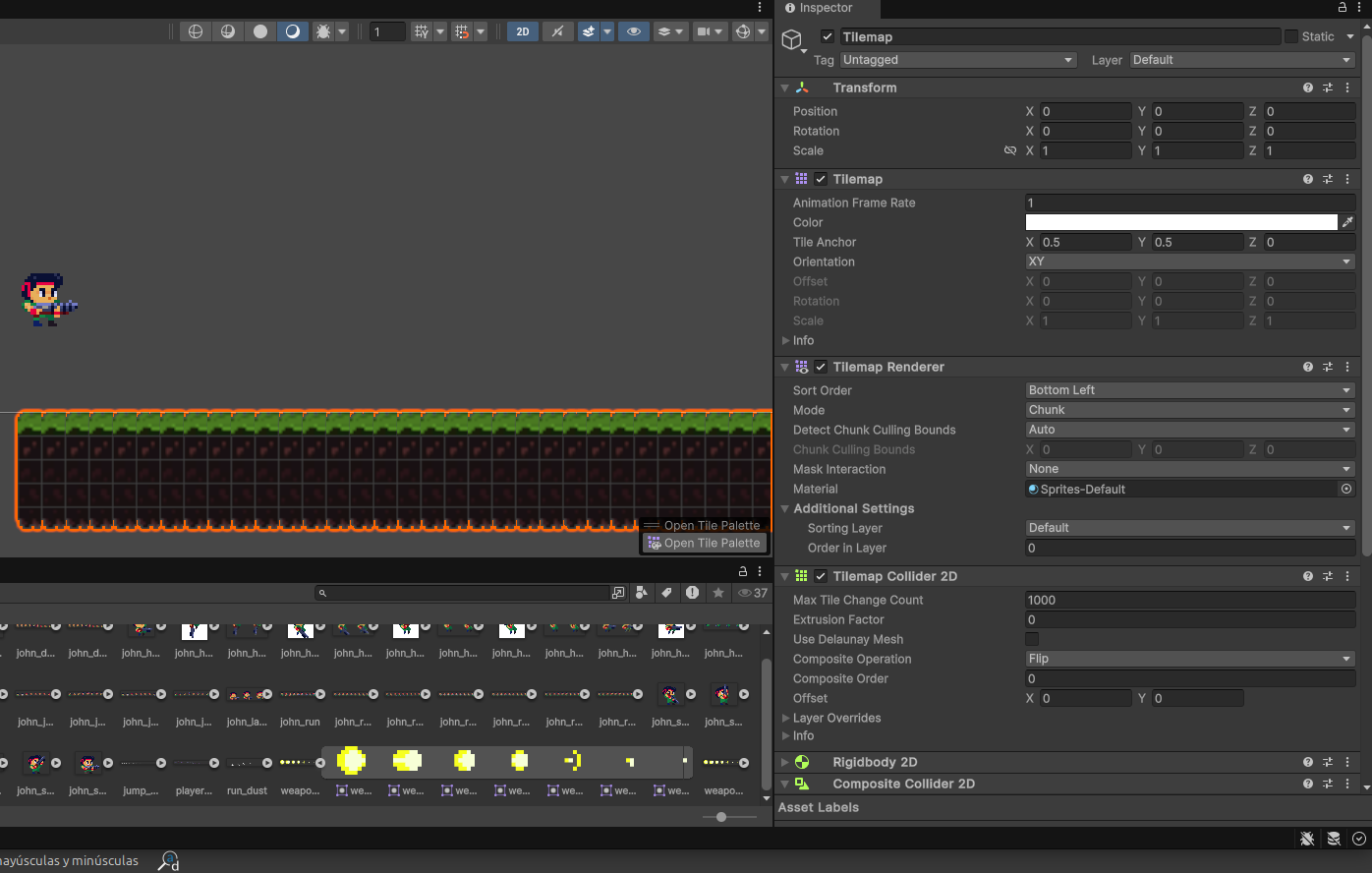
Este juego fue creado en la plataforma de creación de video juegos llamada Unity, en donde es posible realizar la programación del entorno del video juego, esta programación se realiza con el lenguaje C#, a continuación se muestra la explicación general del entorno de programación en la plataforma UNITY.



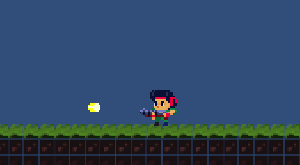
En la figura se puede observar al personaje en el entorno de edición de la plataforma Unity, en donde se modificaron algunas propiedades de este para que se acoplara al entorno de juego.



En la figura anterior se pueden observar las propiedades del personaje, en donde se le adiciona la propiedad de Rigidbody 2D, en donde se le asigna la gravedad del personaje y lo reconoce como un cuerpo rigido en el sistema, ademas de esto se le asigna el script de movimiento del personaje.



En la figura anterior se pueden ver las propiedades del terreno de juego, en donde tiene algunas especificaciones, entre las cuales que es un cuerpo rigido, tambien se le asigna una composición de tipo rectangular y se juntan cada una de las cuadriculas.



En la figura anterior se ve al personaje disparando en el entorno creado, este disparo también fue animado y se le dio un scripts especifico para su funcionamiento.

Script del personaje en C#

using UnityEngine;

using UnityEngine.Rendering;

public class johnMovement : MonoBehaviour

{

public GameObject BulletPrefab;

public float Speed;

public float JumpForce;

private Rigidbody2D Rigidbody2D;

private Animator Animator;

private float Horizontal;

private bool Grounded;

private float LastShoot;

void Start()

{

Rigidbody2D = GetComponent<Rigidbody2D>();

Animator = GetComponent<Animator>();

}

void Update()

{

Horizontal = Input.GetAxisRaw("Horizontal");

if (Horizontal < 0.0f) transform.localScale = new Vector3(-1.0f, 1.0f, 1.0f);

else if (Horizontal > 0.0f) transform.localScale = new Vector3(1.0f, 1.0f, 1.0f);

Animator.SetBool("running",Horizontal != 0.0f);

Debug.DrawRay(transform.position, Vector3.down \* 0.1f, Color.red);

if (Physics2D.Raycast(transform.position, Vector3.down, 0.1f))

{

Grounded = true;

}

else Grounded = false;

if (Input.GetKeyDown(KeyCode.UpArrow)&& Grounded)

{

Jump();

}

if (Input.GetKey(KeyCode.Space) && Time.time > LastShoot + 0.25f)

{

Shoot();

LastShoot = Time.time;

}

}

private void Jump()

{

Rigidbody2D.AddForce(Vector2.up\*JumpForce);

}

private void Shoot()

{

Vector3 direction;

if (transform.localScale.x == 1.0f) direction = Vector2.right;

else direction = Vector2.left;

GameObject bullet = Instantiate(BulletPrefab, transform.position + direction \* 0.1f, Quaternion.identity);

bullet.GetComponent<bulletscript>().SetDirection(direction);

}

private void FixedUpdate()

{

Rigidbody2D.linearVelocity = new Vector2(Horizontal, Rigidbody2D.linearVelocityY);

}

}

Script Bola de fuego

using System;

using UnityEngine;

using UnityEngine.Rendering;

public class bulletscript : MonoBehaviour

{

public float Speed;

private Rigidbody2D Rigidbody2D;

private Vector2 Direction;

void Start()

{

Rigidbody2D = GetComponent<Rigidbody2D>();

}

private void FixedUpdate()

{

Rigidbody2D.linearVelocity = Direction\*Speed;

}

public void SetDirection(Vector2 direction)

{

Direction = direction;

}

}