# Movimiento

Si tienes el personaje y terreno configurado, para lograr un movimiento laterar

1ro hay que crear un script de movimiento (PlayerController.cs)

using UnityEngine;

public class PlayerController2 : MonoBehaviour

{

public float velocidad;

public float velocidadMax;

private Rigidbody2D rPlayer;

private float h;

// Start is called once before the first execution of Update after the MonoBehaviour is created

void Start()

{

rPlayer = GetComponent<Rigidbody2D>();

}

// Update is called once per frame

void Update()

{

h = Input.GetAxisRaw("Horizontal");

rPlayer.AddForce(Vector2.right \* velocidad \* h);

}

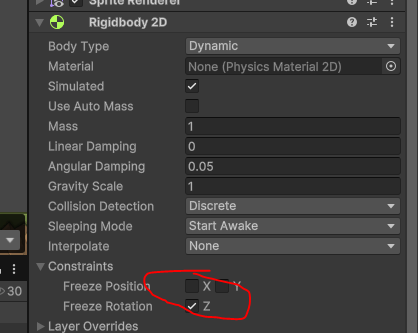
private void FixedUpdate()

{

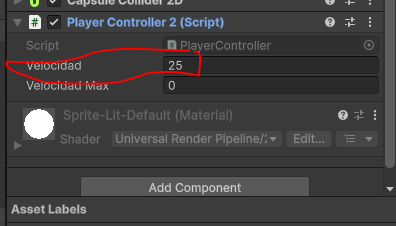
}

}

2do en la capa del Player (personaje que vamos a animar) buscar su rigidbody2d y en la pestaña de Constrains marcar el eje z para que solo se mueva en X y Y.



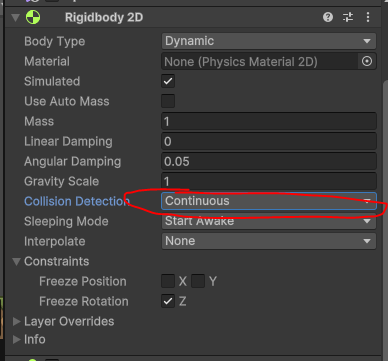
3ro representando las variables publicas que declaramos, nos va a salir en la capa de Player en La sección que representa a nuestro controlador, la velocidad, le damos un valor para que nuestro personaje se pueda mover



4to evitar efecto revote

ha beses pasa que si tiene mucha velocidad como que entra y sale de los contornos, para evitar esto hay que ajustar el comportamiento de las colisiones

En la pestaña del Player en la sección de rigidbody2d buscar el combobox de Collision Detection y cambiar su valor a Continuos



# Girar personaje según la dirección del movimiento

using UnityEngine;

public class PlayerController2 : MonoBehaviour

{

public float velocidad;

public float velocidadMax;

private Rigidbody2D rPlayer;

private float h;

private bool miraDerecha=true;

// Start is called once before the first execution of Update after the MonoBehaviour is created

void Start()

{

rPlayer = GetComponent<Rigidbody2D>();

}

// Update is called once per frame

void Update()

{

girarPlater(h);

}

private void FixedUpdate()

{

h = Input.GetAxisRaw("Horizontal");

rPlayer.AddForce(Vector2.right \* velocidad \* h);

}

public void girarPlater(float horizontal) {

if (horizontal > 0 && !miraDerecha || horizontal < 0 && miraDerecha) {

miraDerecha = !miraDerecha;

Vector3 escalaGiro = transform.localScale;

escalaGiro.x \*= -1;

transform.localScale = escalaGiro;

}

}

}

# Salto

using UnityEngine;

public class PlayerController2 : MonoBehaviour

{

public float velocidad;

public float velocidadMax;

public float fuerzaSalto;

private Rigidbody2D rPlayer;

private float h;

private bool miraDerecha=true;

// Start is called once before the first execution of Update after the MonoBehaviour is created

void Start()

{

rPlayer = GetComponent<Rigidbody2D>();

}

// Update is called once per frame

void Update()

{

girarPlater(h);

if (Input.GetButtonDown("Jump")) {

rPlayer.linearVelocity = new Vector2(rPlayer.linearVelocityX, 0f);

rPlayer.AddForce(new Vector2(0, fuerzaSalto), ForceMode2D.Impulse);

}

}

private void FixedUpdate()

{

h = Input.GetAxisRaw("Horizontal");

rPlayer.AddForce(Vector2.right \* velocidad \* h);

float limiteVelocidad=Mathf.Clamp(rPlayer.linearVelocity.x, -velocidadMax, velocidadMax);

}

public void girarPlater(float horizontal) {

if (horizontal > 0 && !miraDerecha || horizontal < 0 && miraDerecha) {

miraDerecha = !miraDerecha;

Vector3 escalaGiro = transform.localScale;

escalaGiro.x \*= -1;

transform.localScale = escalaGiro;

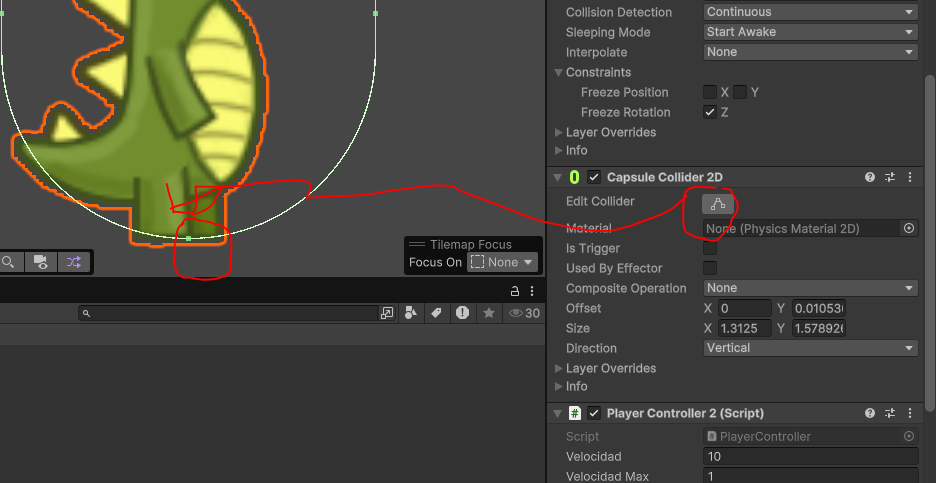
}

}

}

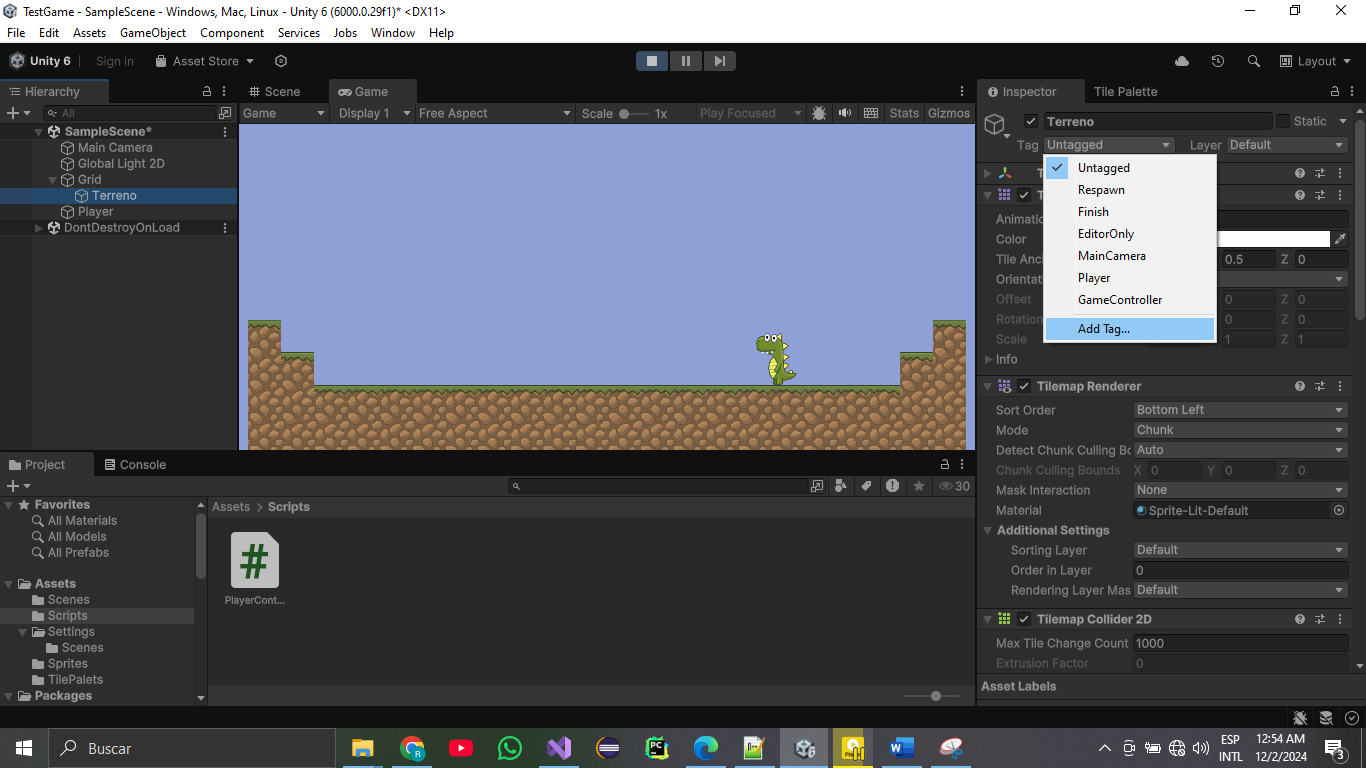
# Alinear capsuleColaider2d

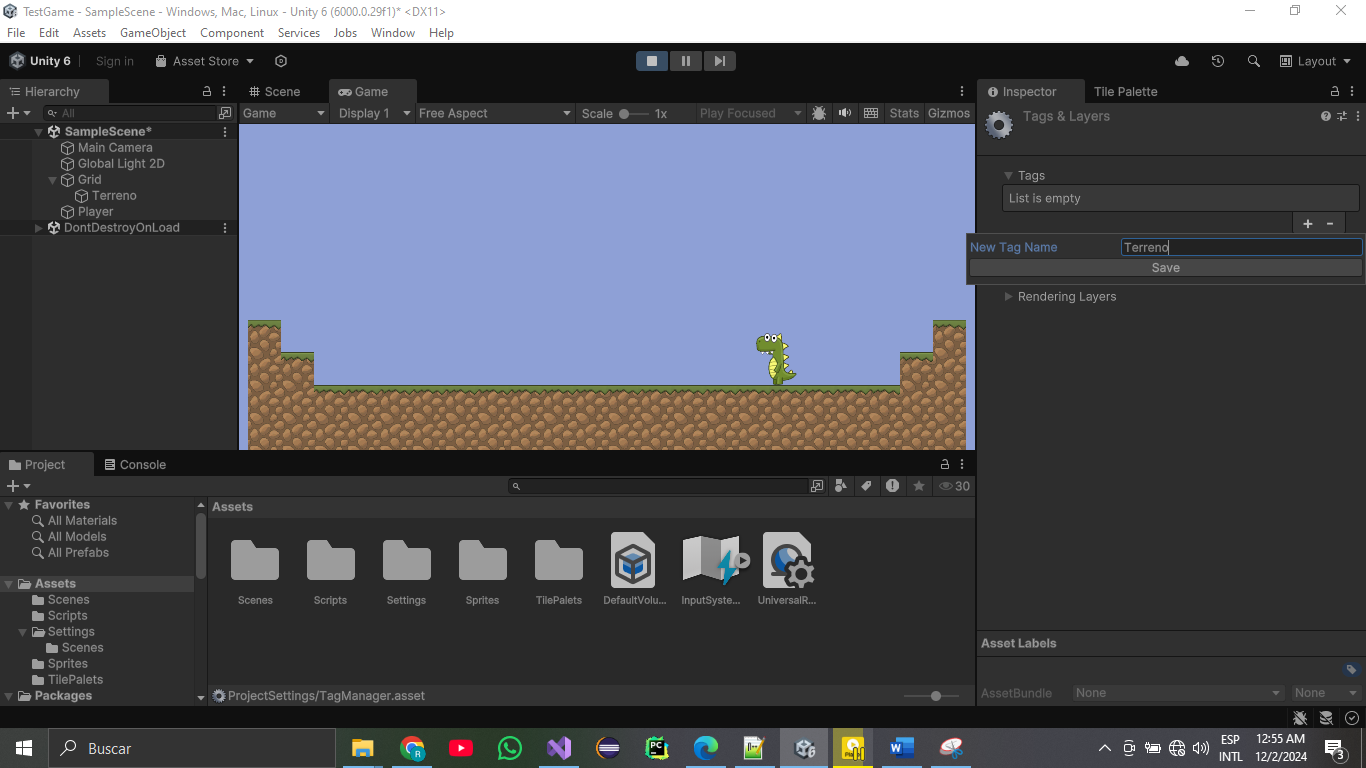
Subir un poco la colisión

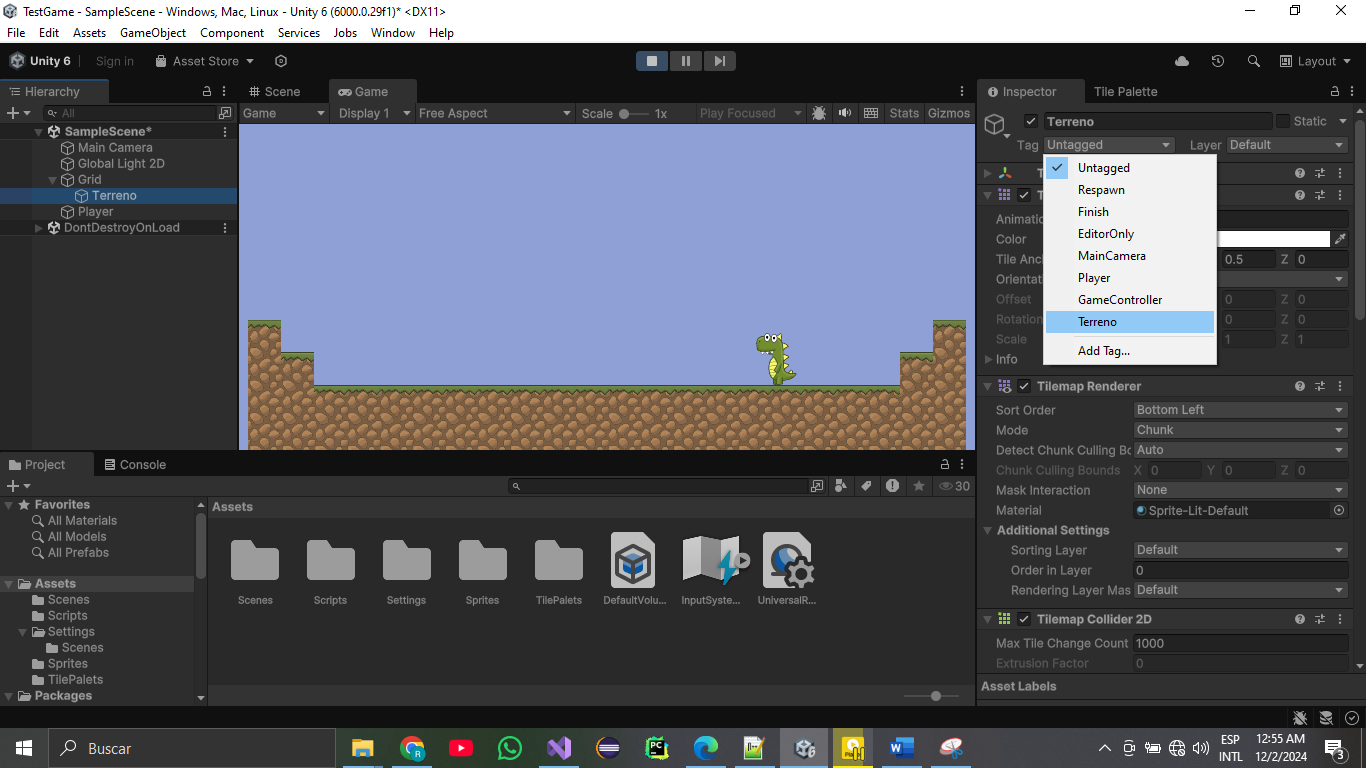


# Ajustar roze

## 1ro Agregar etiqueta al terreno

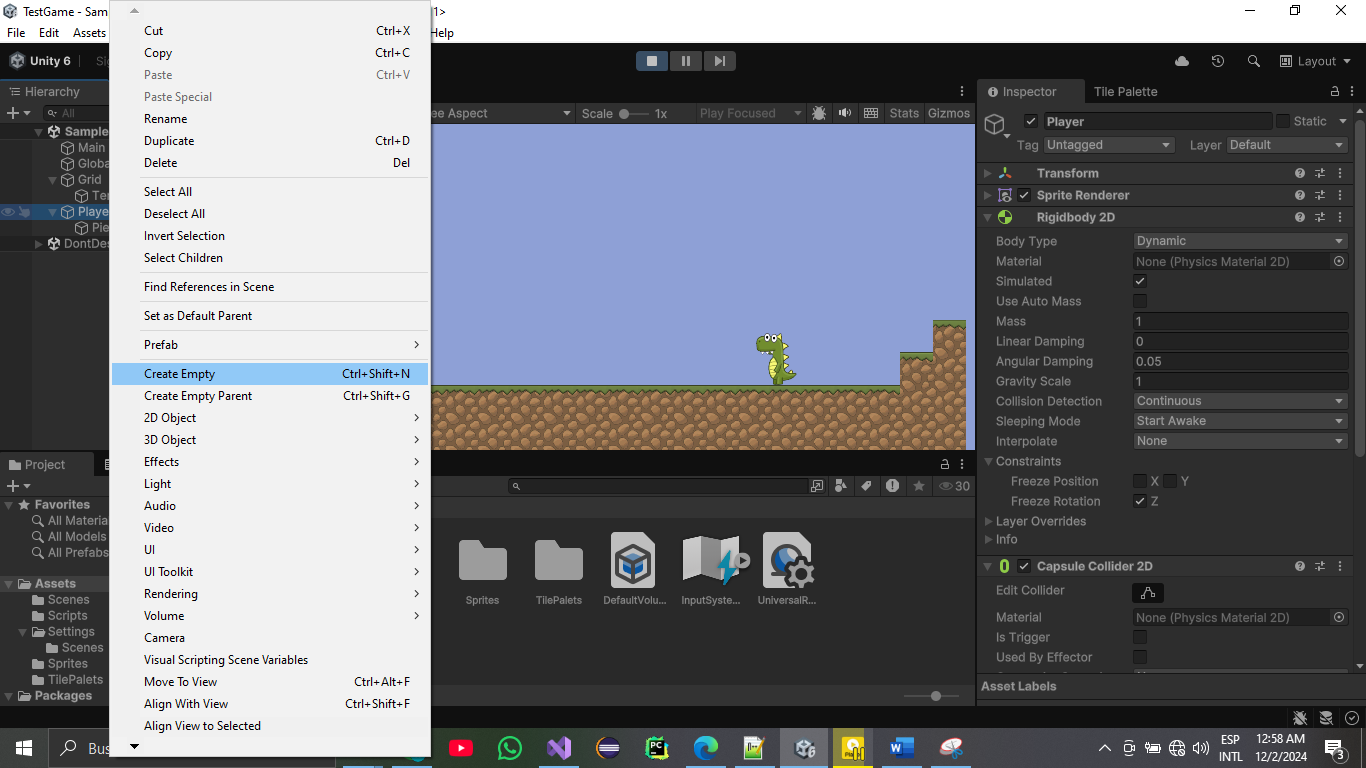




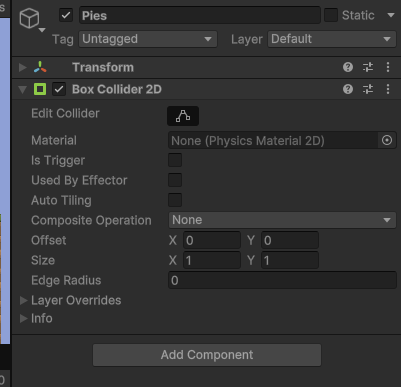


## 2do Agregar pies

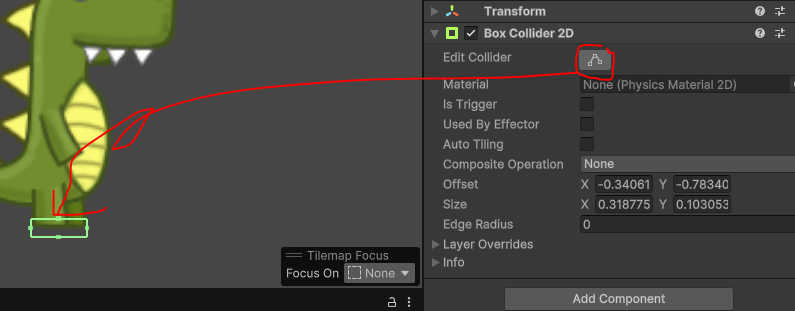
Seleccionar el Player y mandar a crear un objeto vacío llamado pies



## 3ro Agregarle a esto objeto un box Colaider 2D



## 4to Ubicar el box colaider 2d en los pies



## 5to marcarlo como trigger

