# 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);

}

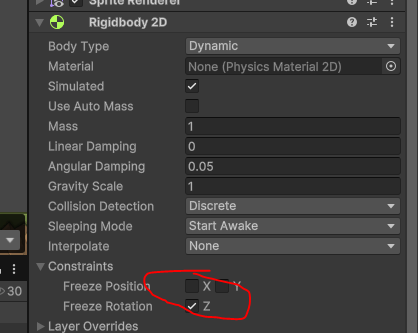
private void FixedUpdate()

{

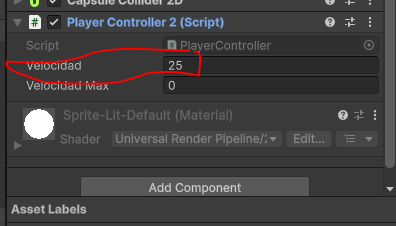
}

}

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.



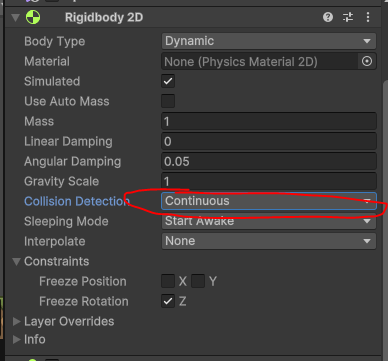
3ro representando las variables publicas que declaramos, nos va a salir en la capa de Player en La sección que representa a nuestro controlador, la velocidad, le damos un valor para que nuestro personaje se pueda mover



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);

rPlayer.linearVelocity=new Vector2(limiteVelocidad,rPlayer.linearVelocityY);

}

public void girarPlater(float horizontal) {

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

miraDerecha = !miraDerecha;

Vector3 escalaGiro = transform.localScale;

escalaGiro.x \*= -1;

transform.localScale = escalaGiro;

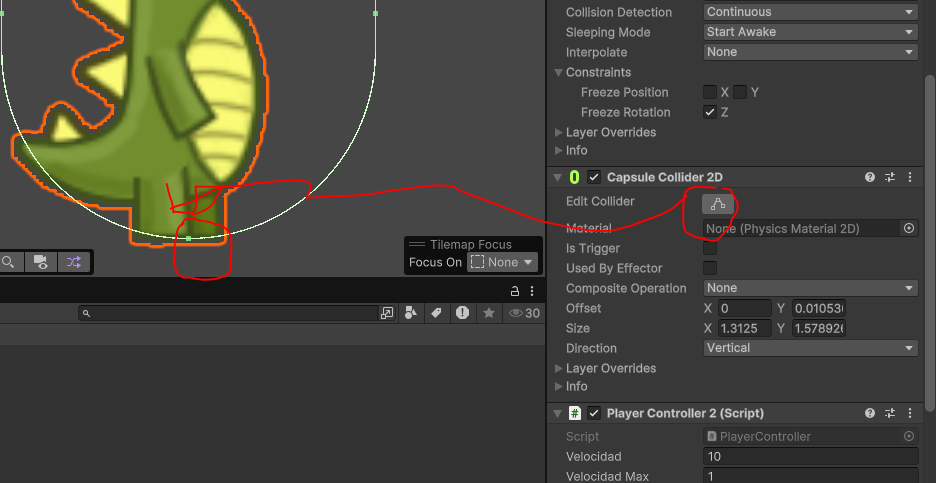
}

}

}

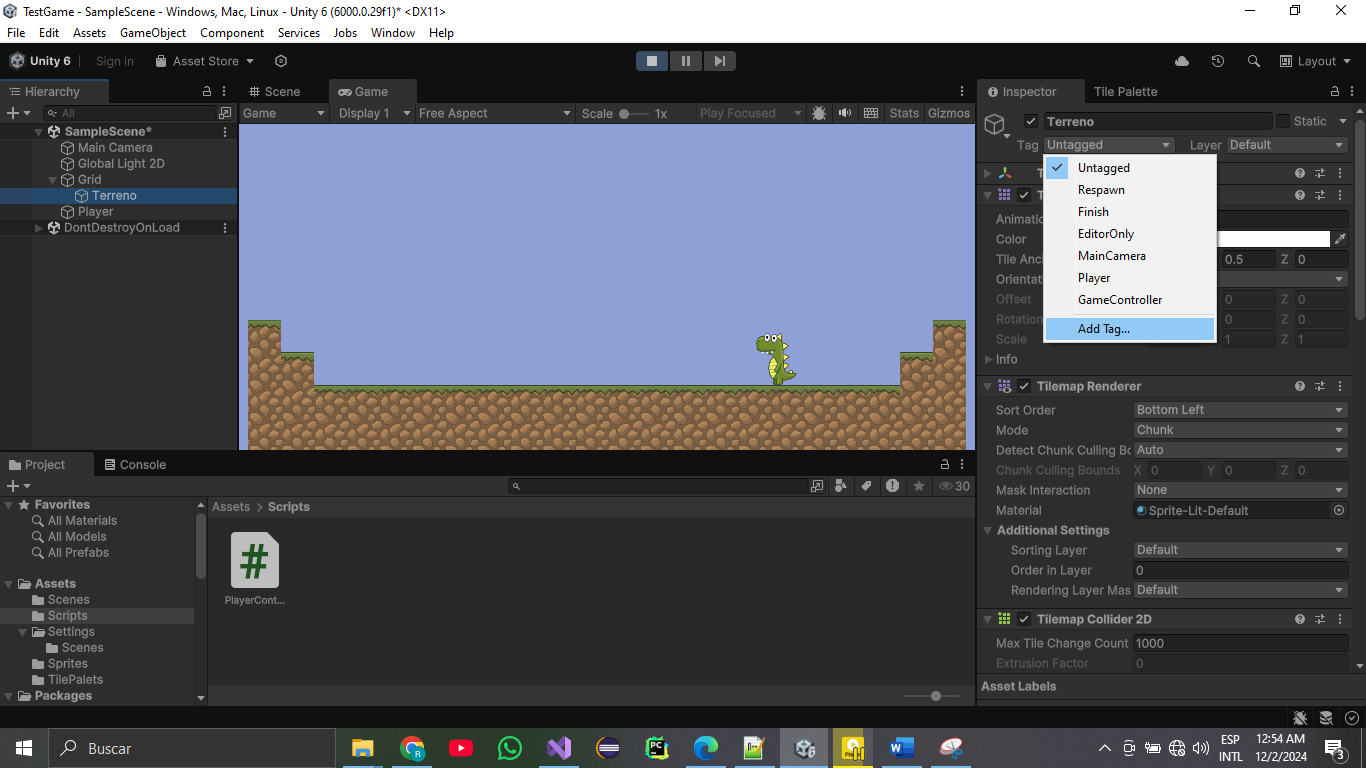
# Alinear capsuleColaider2d

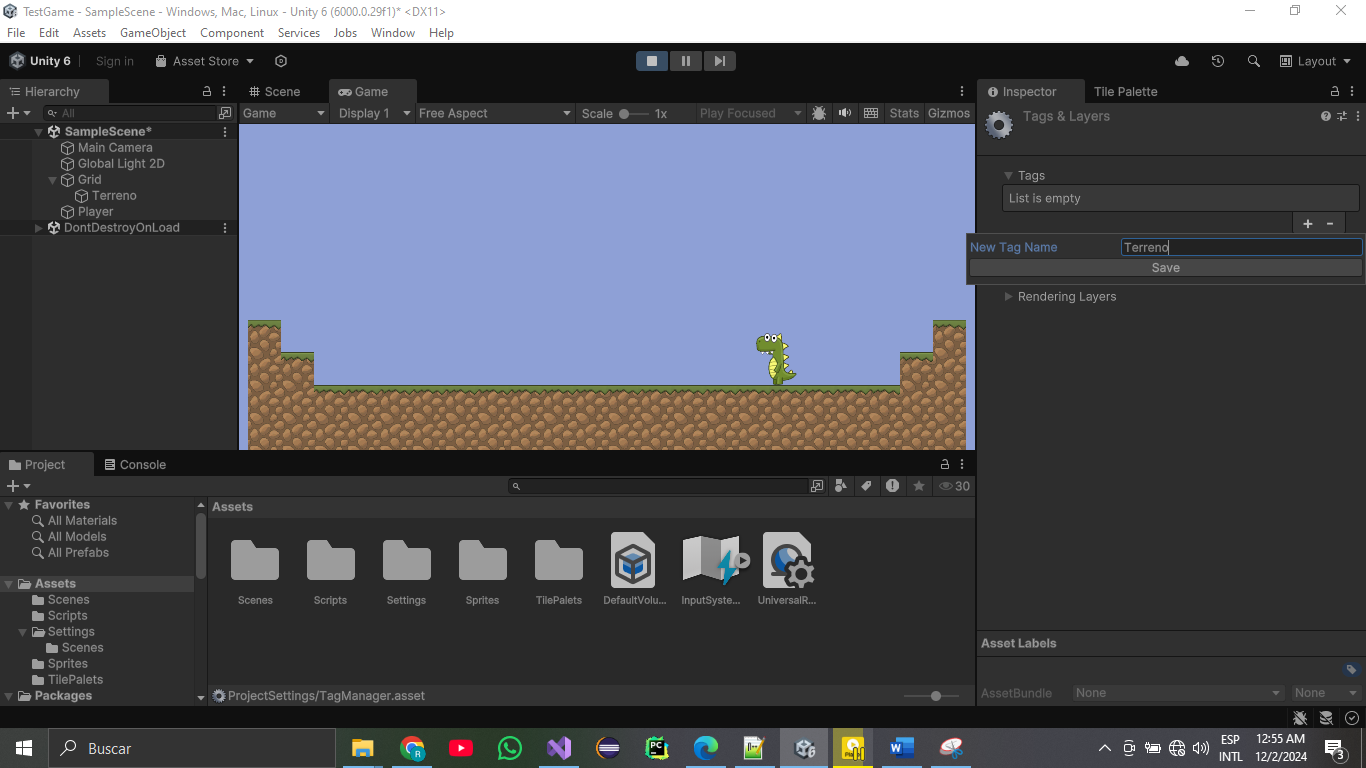
Subir un poco la colisión

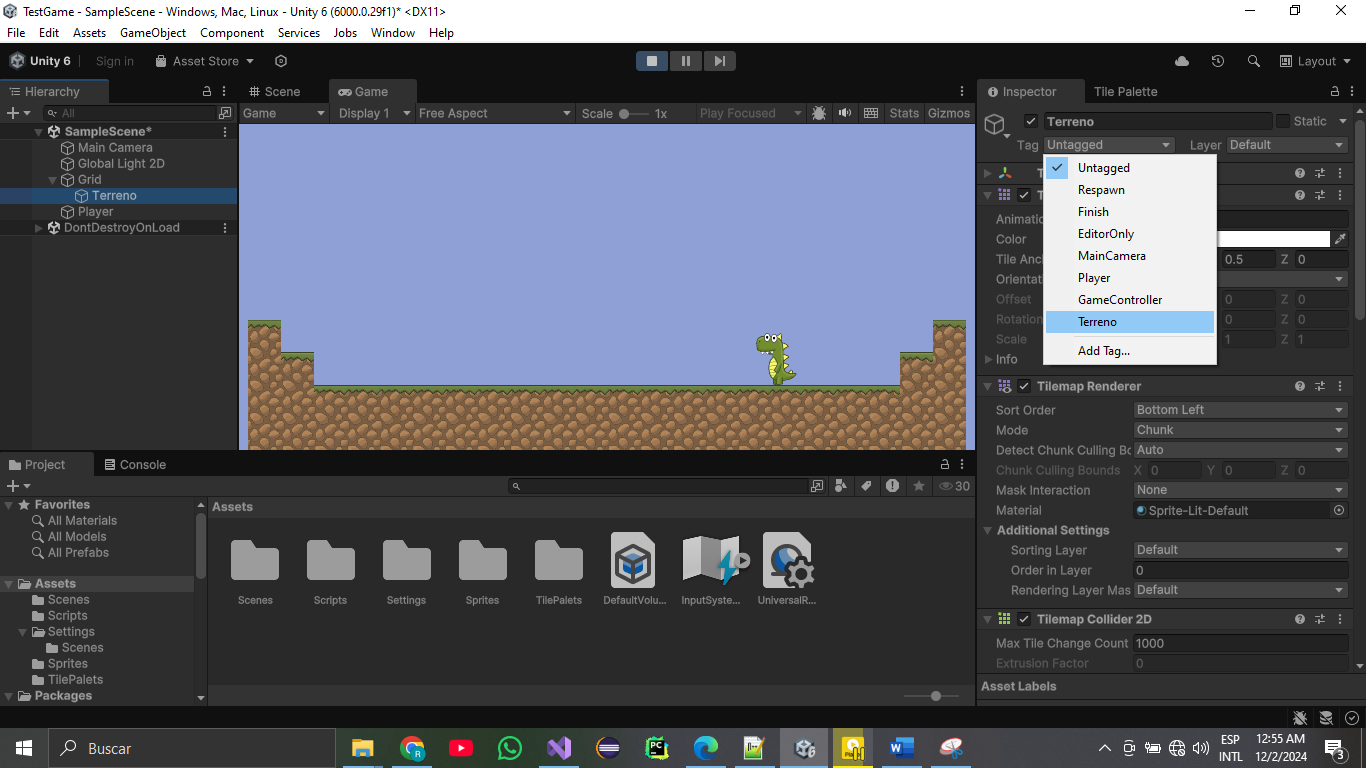


# Ajustar roze

## 1ro Agregar etiqueta al terreno

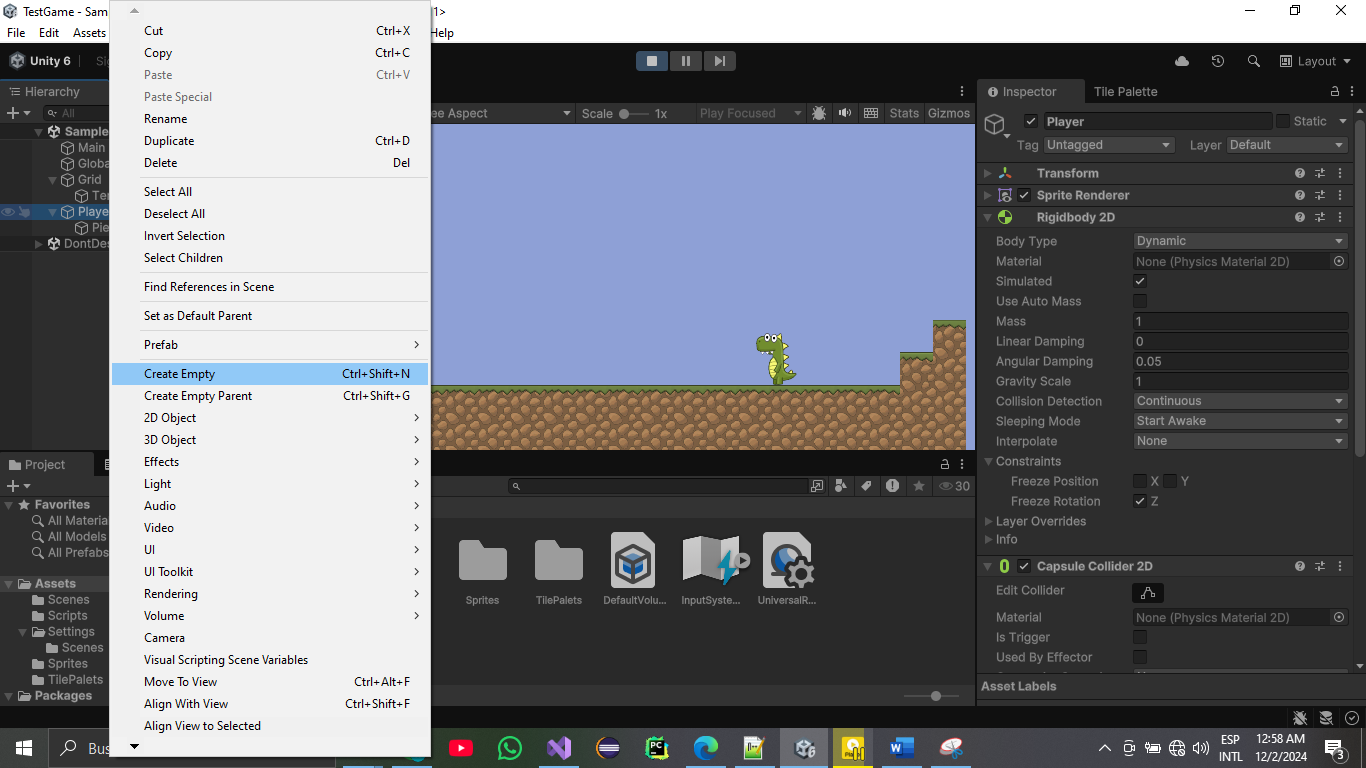




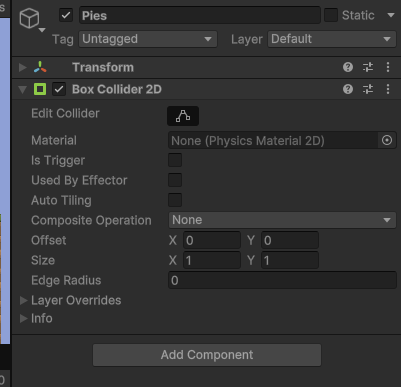


## 2do Agregar pies

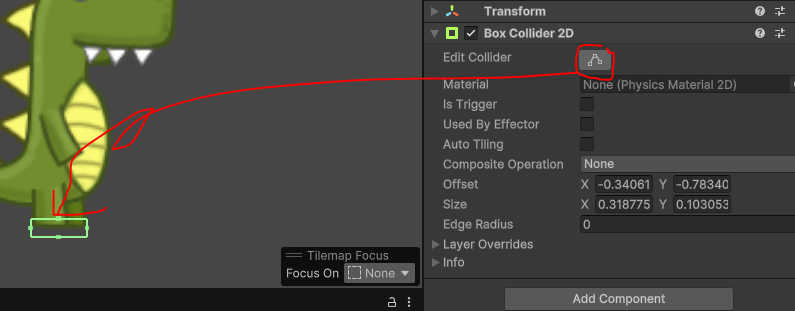
Seleccionar el Player y mandar a crear un objeto vacío llamado pies



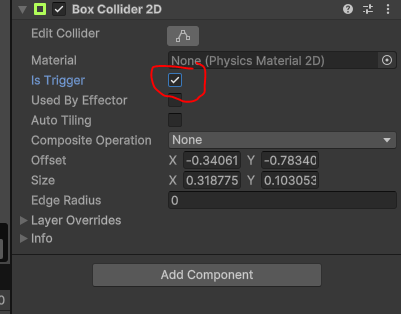
## 3ro Agregarle a esto objeto un box Colaider 2D



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



## 5to marcarlo como trigger



# Saltar

Luego de ajustar el rose y por tanto tener el rectángulo de los pies con la etiqueta de terreno

-1ro Crear el scrit CheckGround.cs

using UnityEngine;

public class CheckGround : MonoBehaviour

{

public static bool colPies;

private void OnTriggerEnter2D(Collider2D collision)

{

if (collision.gameObject.tag=="Terreno") {

colPies = true;

}

}

private void OnTriggerExit2D(Collider2D collision)

{

if (collision.gameObject.tag == "Terreno")

{

colPies = false;

}

}

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

void Start()

{

}

// Update is called once per frame

void Update()

{

}

}

-2do usar esta variable en el script principal, recordar que es estatica y esto facilita acceder a ella

public static bool colPies;

// Update is called once per frame

void Update()

{

colPies = CheckGround.colPies;

girarPlater(h);

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

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

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

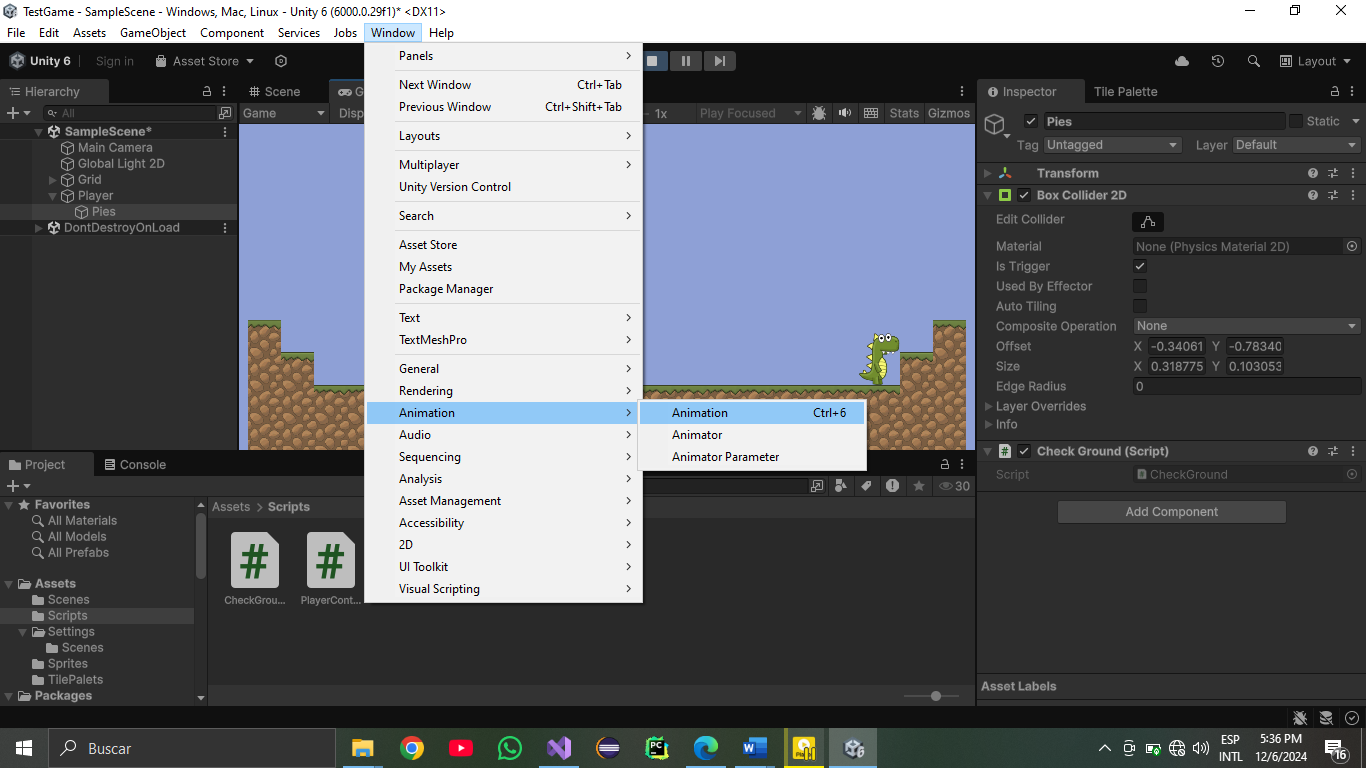
}

}

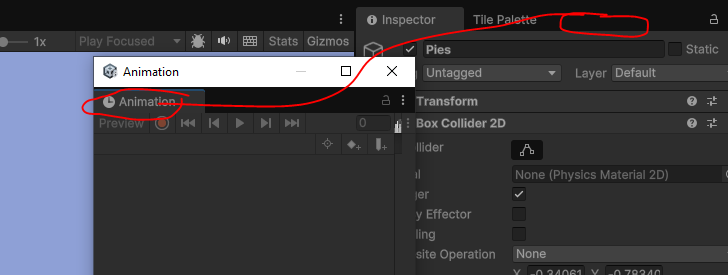
-3do Unirlo al layer Pies (arrastrar el scritp al agregar componente)

# Animación

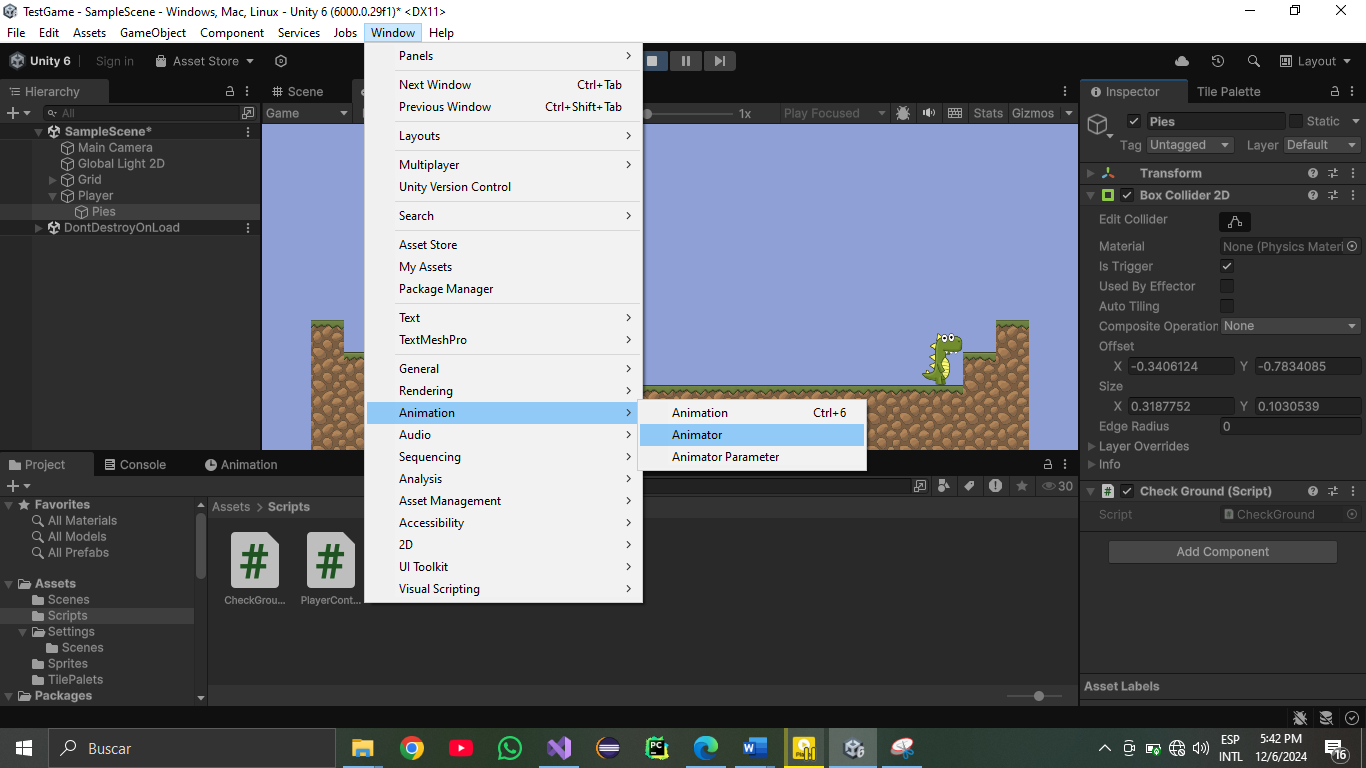
-1ro Buscar la ventana de animación



-2do Arrastrar esta ventana a un sitio mas cómodo

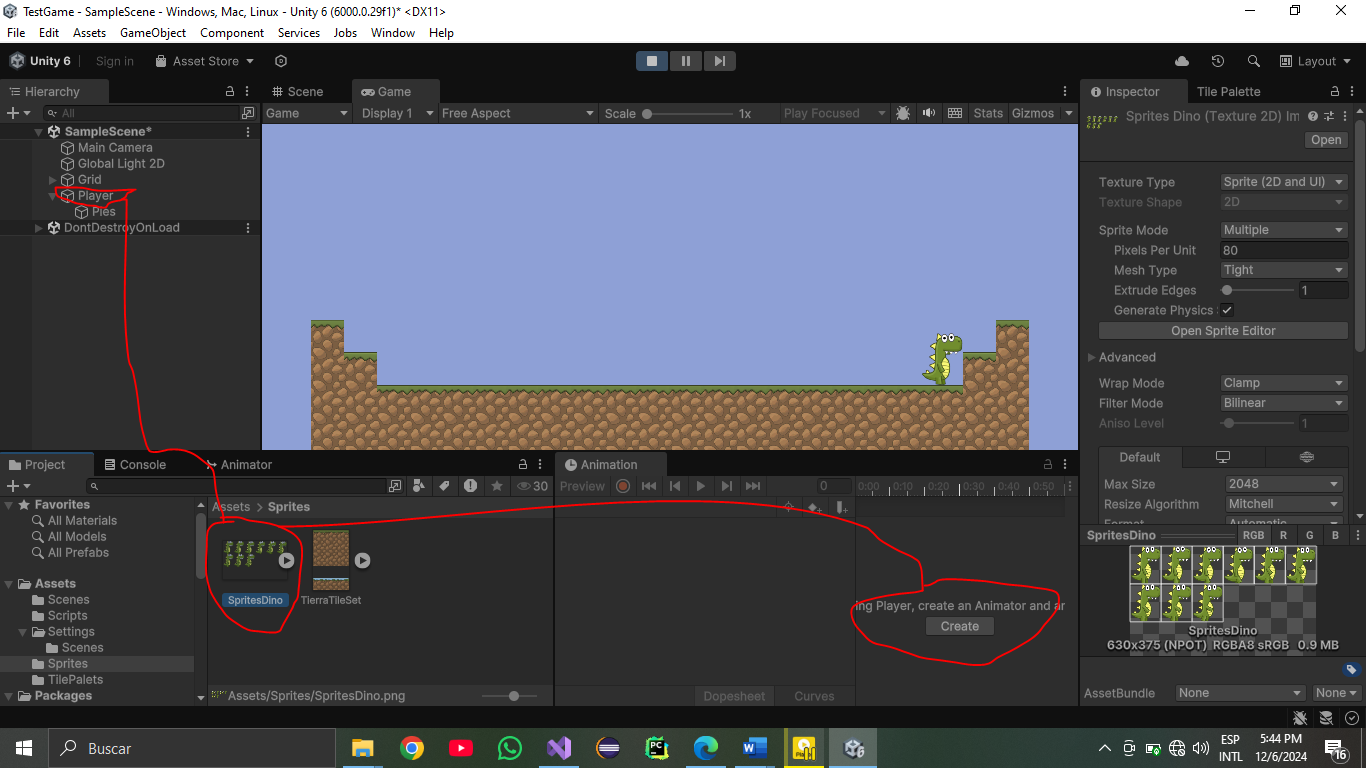


-3ro Buscar la ventana de animator



-4to arrastrar la ventan a un sitio mas cómodo

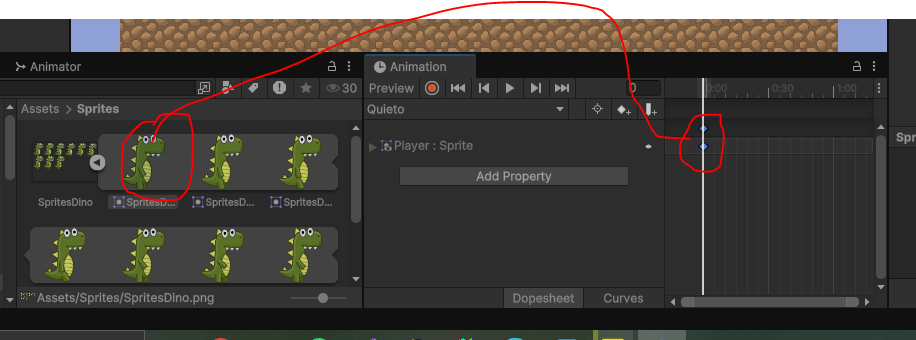
-5to Con el player selccionado, seleccionar el Sprite correspondiente y dar en crear animacion en la ventana de animacion



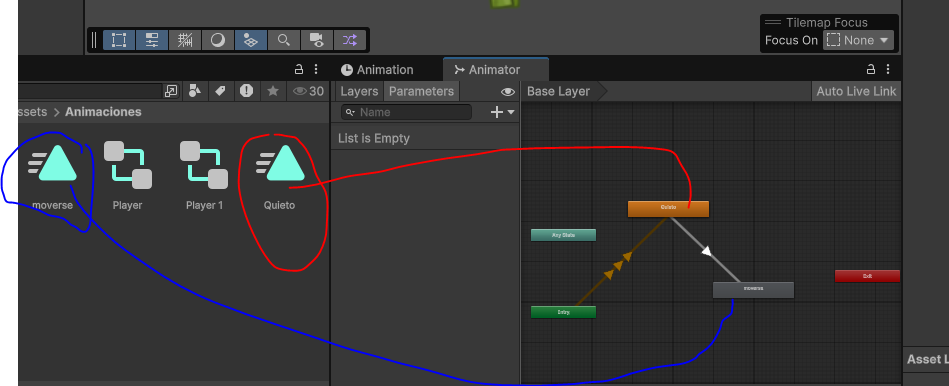
-6to guardarla en una carpeta Animaciones



-7mo se arrastra el sprit (la sección o secciones a la ventana de animacion)

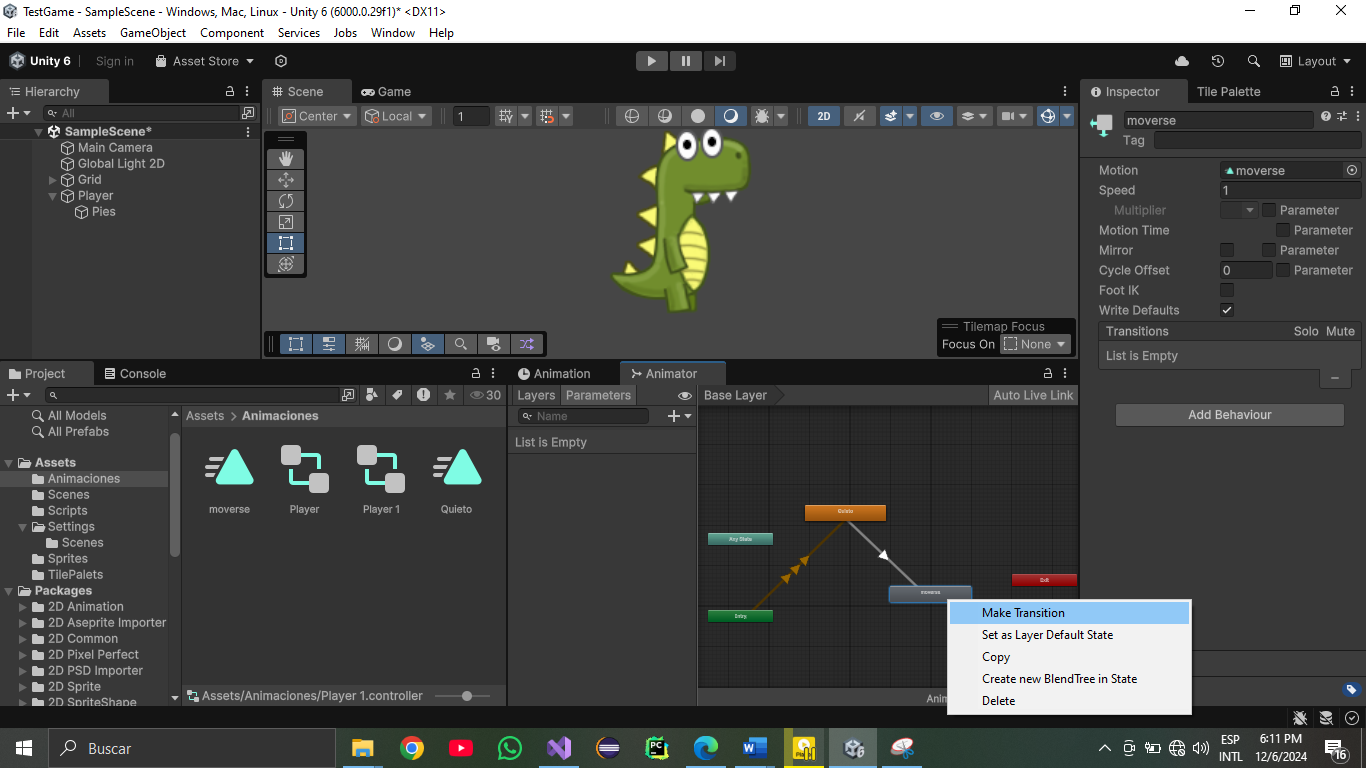


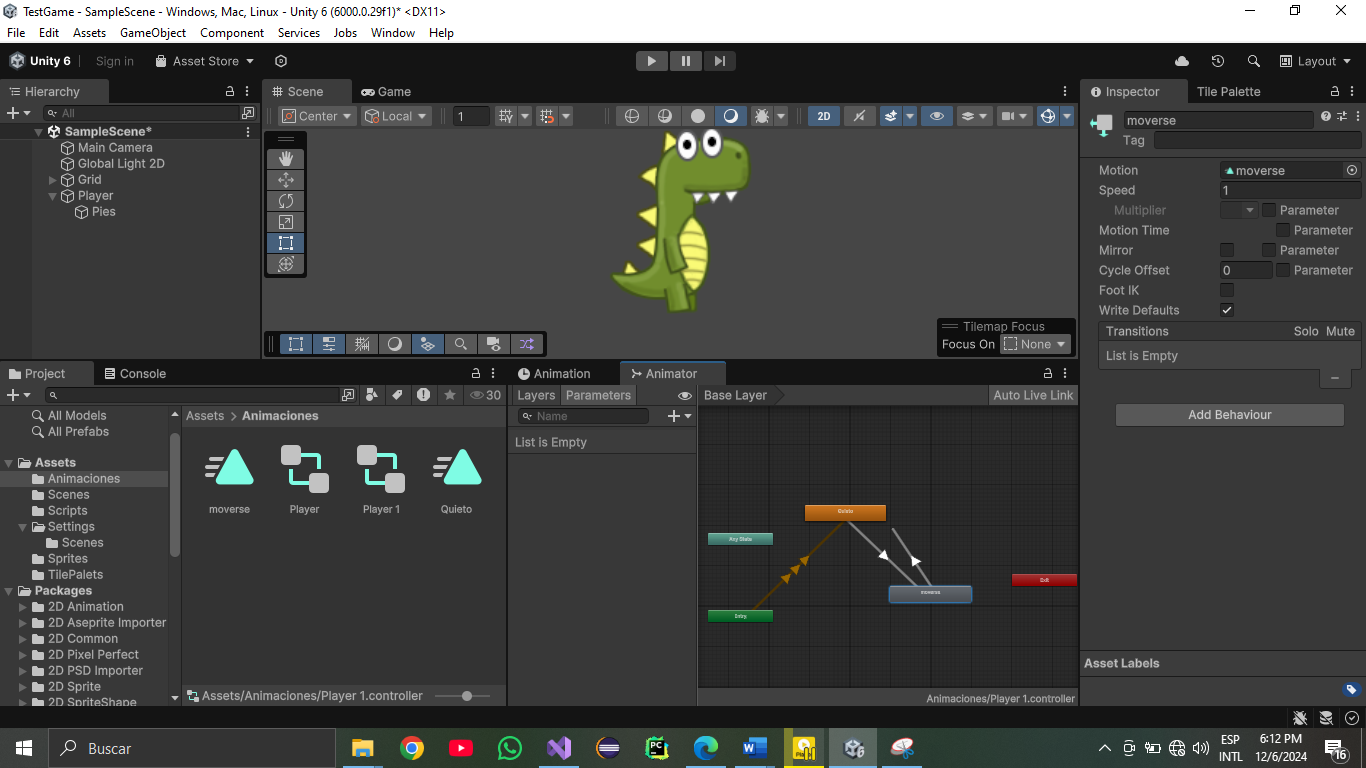
-8vo Con las animaciones creadas, estas se arrastran a la ventana de animator



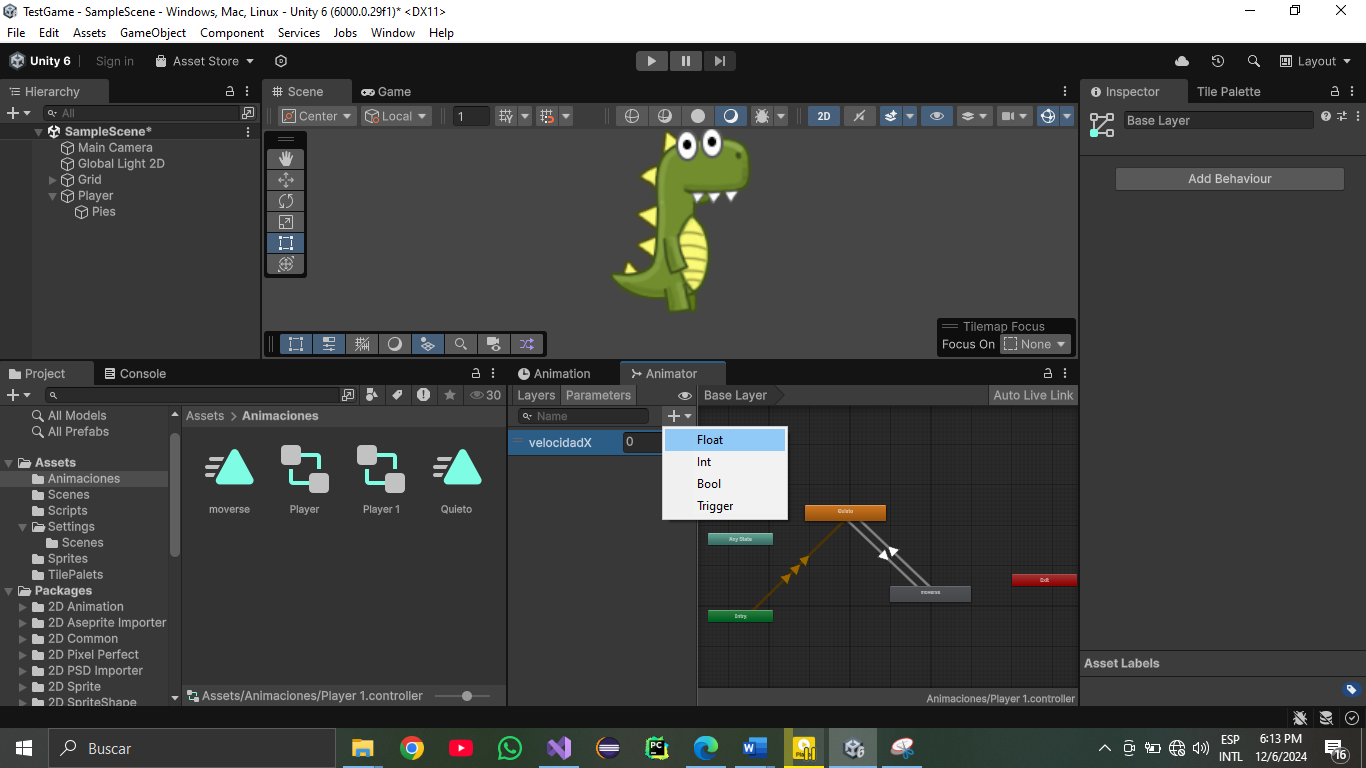
-9no Si la conecion no es correcto entre dos animaciones hay que borrar la animacion de animator, (apretar delete sobre su rectangulo)

-10mo Para crear una transiccion entre dos animaciones, se selecciona el rectángulo de la primera y se da click derecho crear transiccion y luego se apunta al destino

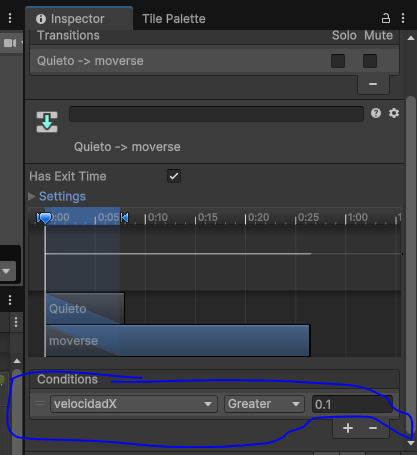




-11no se crearn variables que van a servir para activar o no una transición [float velocidadX]



-12 agregar condiciones en base a las variables, primero seleccionar la flecha de la transición adecuada



-13 Agregar esta variable al player controller

private Animator aPlayer;

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

void Start()

{

rPlayer = GetComponent<Rigidbody2D>();

aPlayer = GetComponent<Animator>();

}

// Update is called once per frame

void Update()

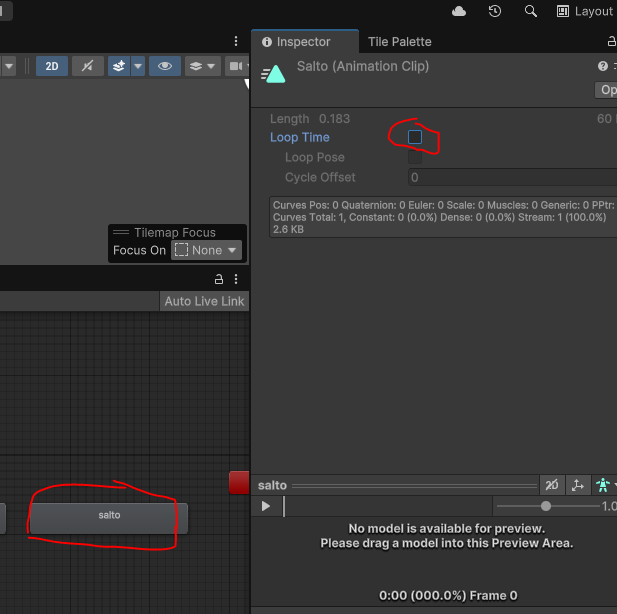
{

colPies = CheckGround.colPies;

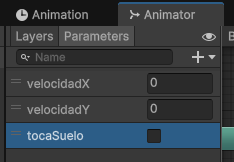
aPlayer.SetFloat("velocidadX",Mathf.Abs(rPlayer.linearVelocityX));

## Salto

-1ro Darle doble click en su animacion y desactivar el loop



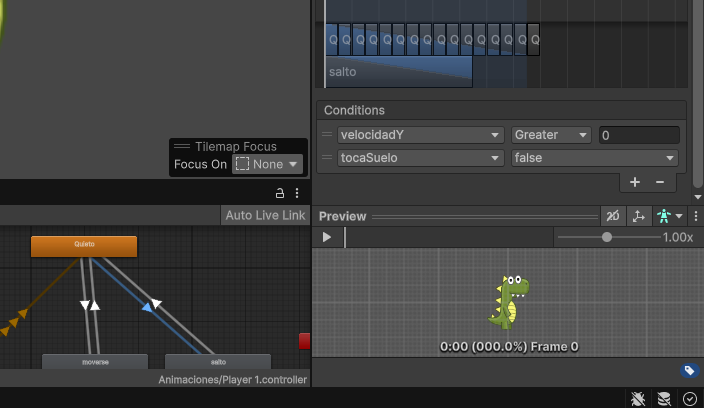
-2do crear la variable float velocidadY, bool tocaSuelo



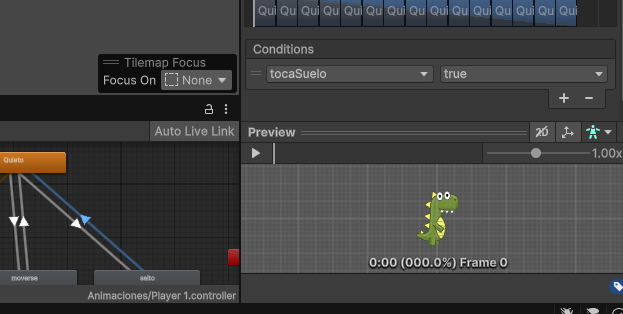
aPlayer.SetFloat("velocidadY", Mathf.Abs(rPlayer.linearVelocityY));

aPlayer.SetBool("tocaSuelo", colPies);

-3ro crear transición [quieto-salto] con las condiciones de velocidadY>0 y tocaSuelo=false



4- quieto-salto tocaSuelo=true



Es exactamente lo mismo entre salto y moverse

-5 crear transición caída y sus condiciones

Salto-caida velocidadY<0

Caida-quieto velodidadX<0 tocaSuelo=false

Caida-moverse velodidadX>0 tocaSuelo=false

### Salto Ligeramente mejorado

Para que halla una pequeña diferencia entre presionar rápido el salto y dejarlo precionado un poco

1ro crear un script que va a modificar la velocidad del player, por tanto asociarlo a el

using UnityEngine;

public class MejoraSalto : MonoBehaviour

{

public float saltoLArgo = 1.5f;

public float saltoCorto = 1.0f;

Rigidbody2D rb;

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

void Start()

{

rb= GetComponent<Rigidbody2D>();

}

// Update is called once per frame

void Update()

{

if (rb.linearVelocityY < 0 ) {

rb.linearVelocity += Vector2.up \* Physics2D.gravity.y \* saltoLArgo \* Time.deltaTime;

} else if (rb.linearVelocityY > 0 && !Input.GetButton("Jump")){

rb.linearVelocity += Vector2.up \* Physics2D.gravity.y \* saltoCorto \* Time.deltaTime;

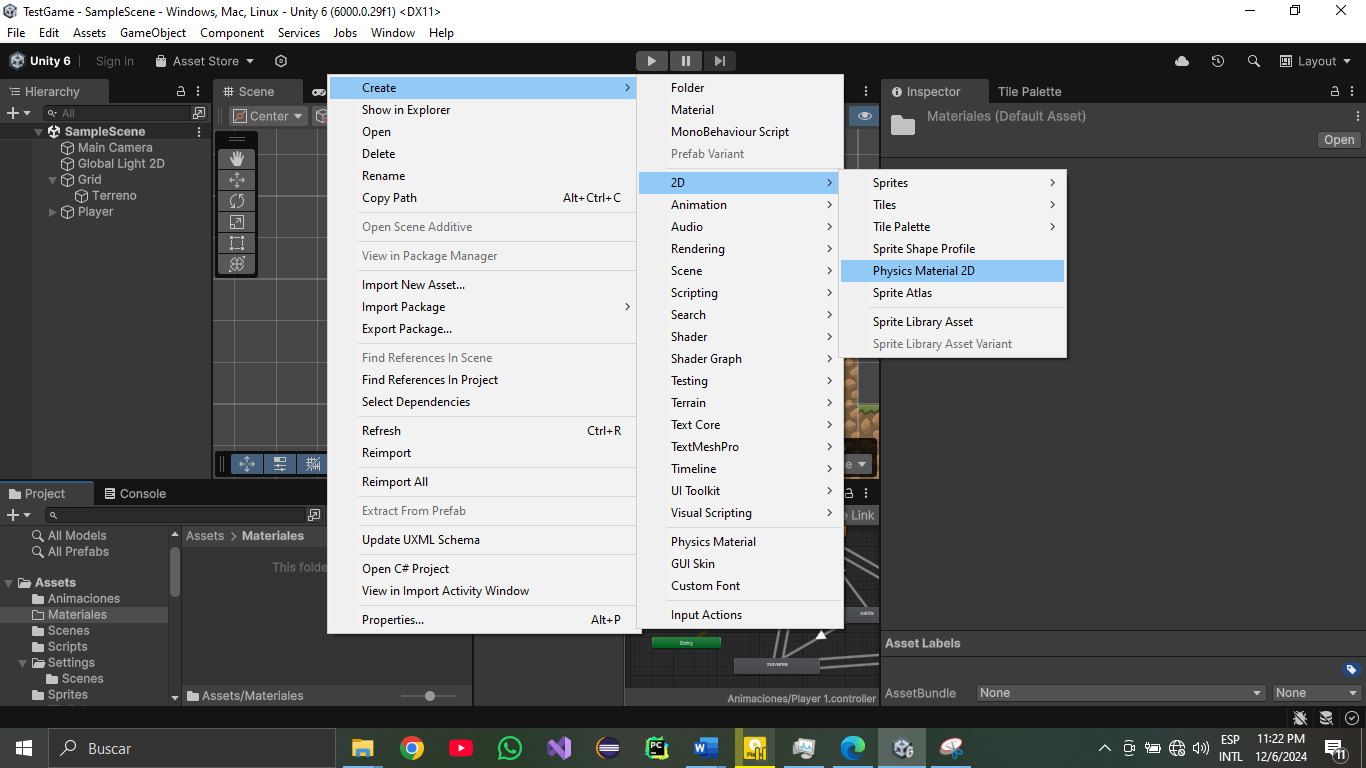
}

}

}

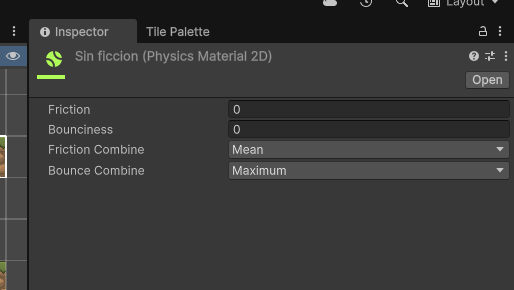
# Crear Materiales 2d

Crea un carpeta para materiales, y luego da click derecho crear material fisico 2d



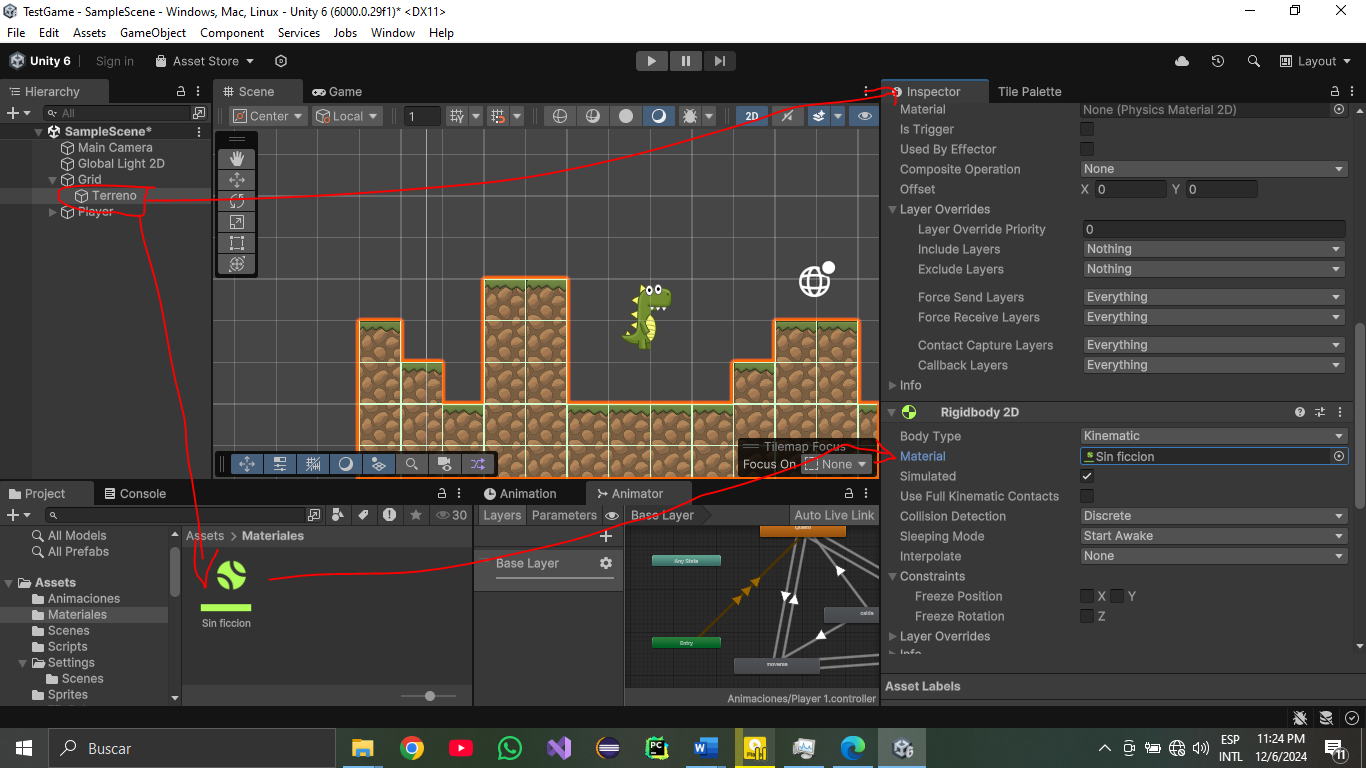
## Sin fricción

Se pone esta variable a 0

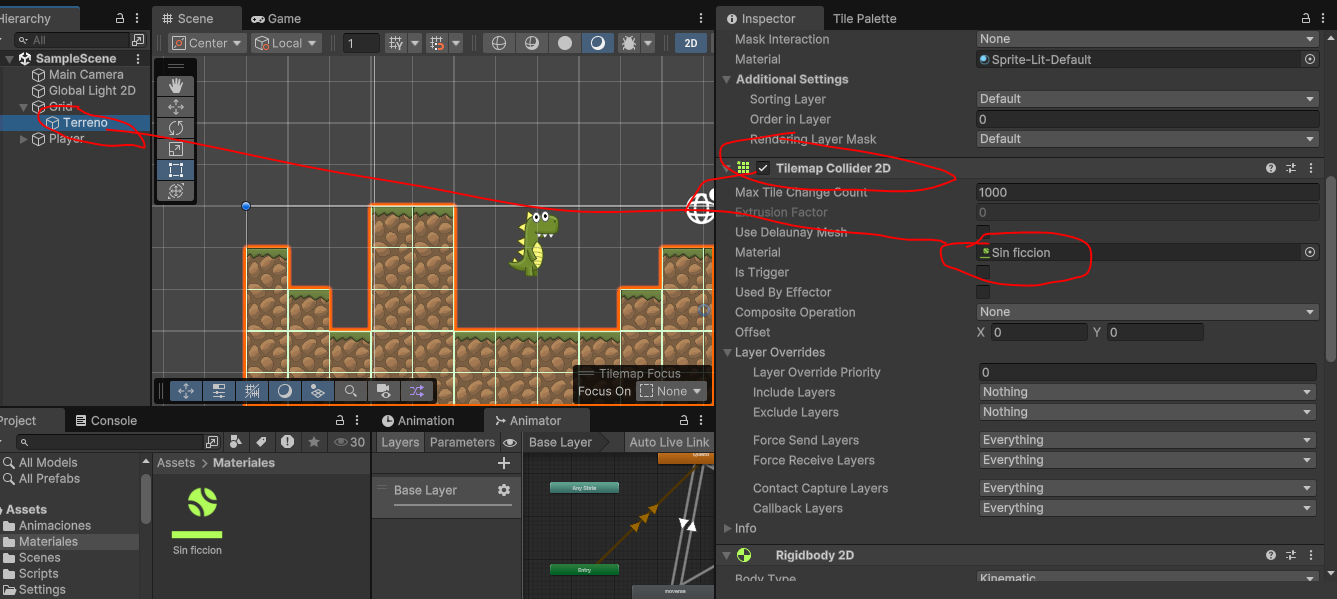


## Agregarlo a alguna capa

Luego se le agrega a alguna campa



A mi me funciono mejor en el tile colaider



## Ajustar fricción de forma ficticia

public float friccionSuelo;

private void FixedUpdate()

{

h = Input.GetAxisRaw("Horizontal");

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

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

rPlayer.linearVelocity=new Vector2(limiteVelocidad,rPlayer.linearVelocityY);

if (h==0&&colPies) {

Vector3 velocidadArreglada=rPlayer.linearVelocity;

velocidadArreglada.x \*= friccionSuelo;

rPlayer.linearVelocity = velocidadArreglada;

}

}

# Seguir Cámara

Crear un script y agregárselo a la camara

using UnityEngine;

public class SeguimentoCamara : MonoBehaviour

{

public Vector2 minCamPos,maxCamPos;

public GameObject seguir;

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

void Start()

{

}

// Update is called once per frame

void Update()

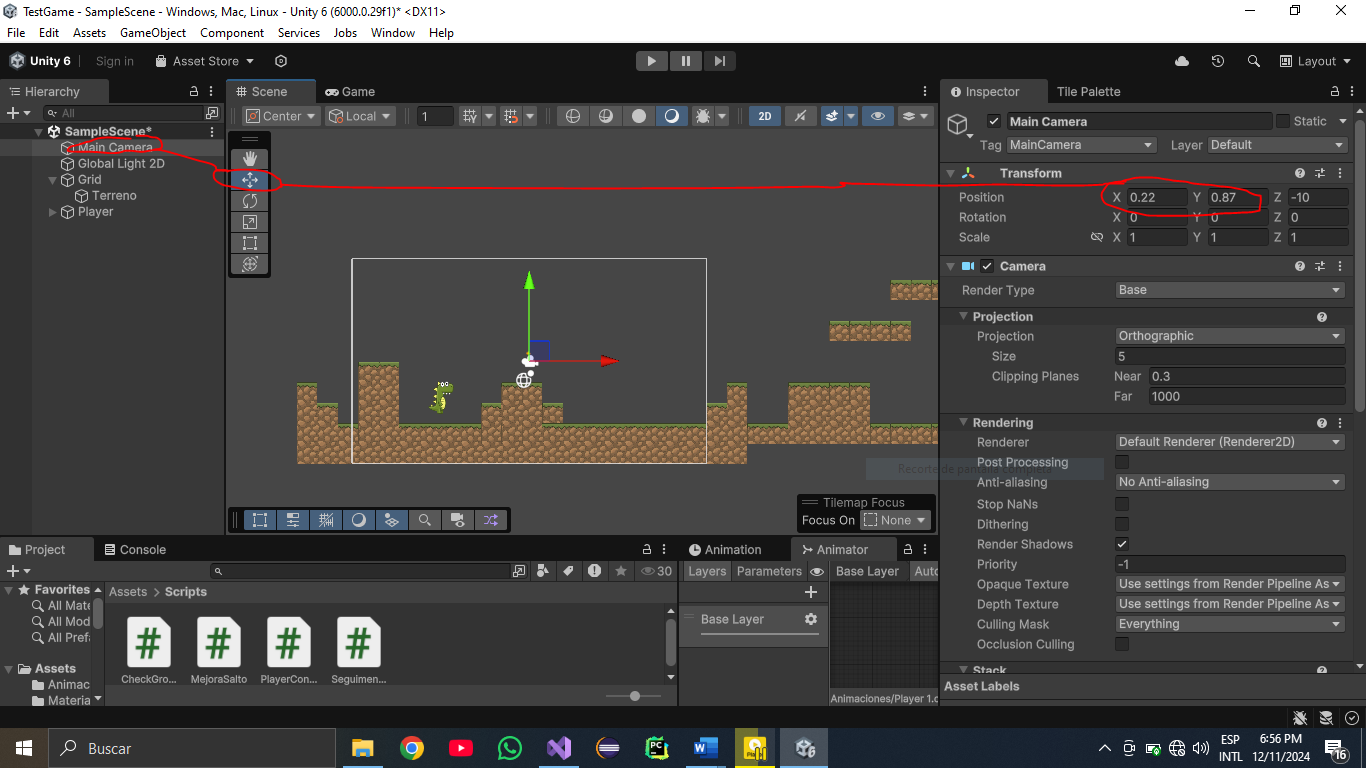
{

}

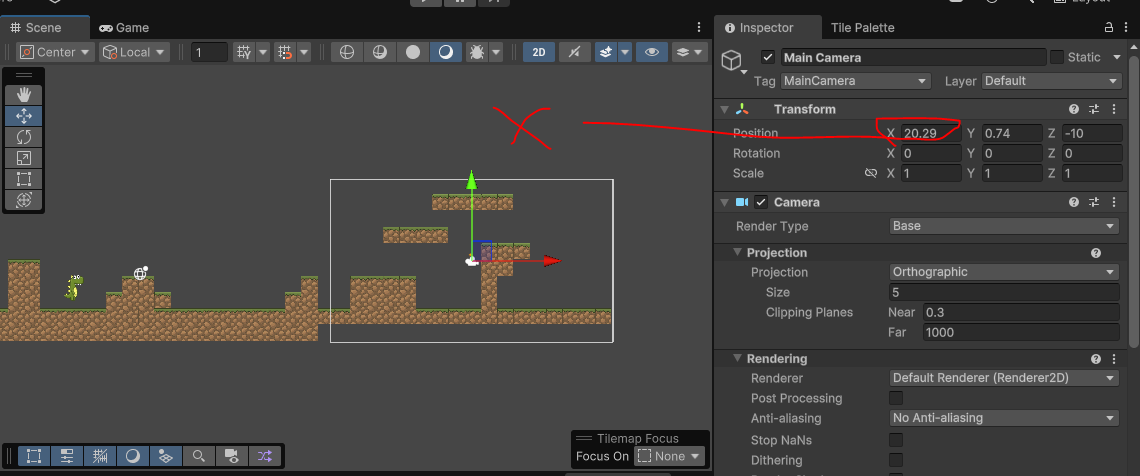
}

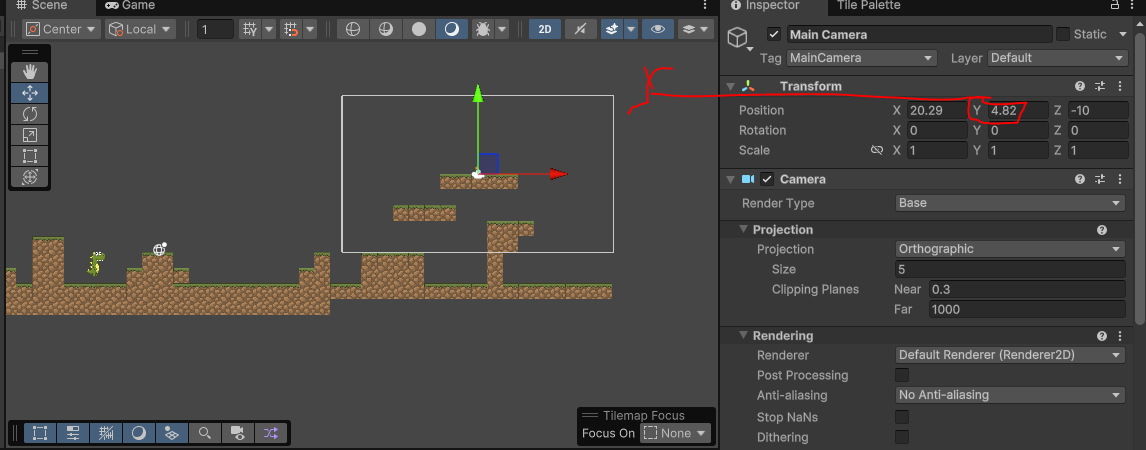
## Ver la posición de la cámara y ubicar la posición inicial

Seleccionar la capa de la cámara y ver las coordenadas de ubicación



Si mueves la cámara puedes ver los valores exactos que sirven para las variables de maxmia posición, entonces mueve la cámara hasta lo mas lejos posible en X y luego en Y





## Seguir

-1ro luego de los pasos anteriores, en el script creamos un objeto que va a contener a nuestro personaje y del cual vamos a utilizar sus coordenadas para reubicar la cámara

using UnityEngine;

public class SeguimentoCamara : MonoBehaviour

{

public Vector2 minCamPos,maxCamPos;

public GameObject seguir;

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

void Start()

{

}

// Update is called once per frame

void Update()

{

float posX=seguir.transform.position.x;

float posY = seguir.transform.position.y;

transform.position = new Vector3(posX, posY,transform.position.z);

}

}

2do seleccionar la cámara y luego arrastrar la capa del jugador al objeto “seguir”

