

Desarrollo de Videojuegos

Pedro Yuri Marquez Solis



Universidad
Continental



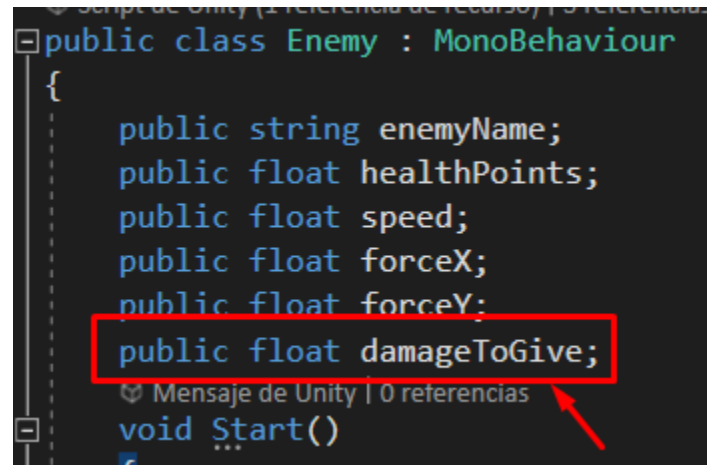
Construir el escenario



Agregar el
background y
las
plataformas



Modificar el enemigo agregando la propiedad : damageToGive



```
Script de Unity (1 referencia de recurso) | 0 referencias
public class Enemy : MonoBehaviour
{
    public string enemyName;
    public float healthPoints;
    public float speed;
    public float forceX;
    public float forceY;
    public float damageToGive;
    Mensaje de Unity | 0 referencias
    void Start()
    ...
}
```

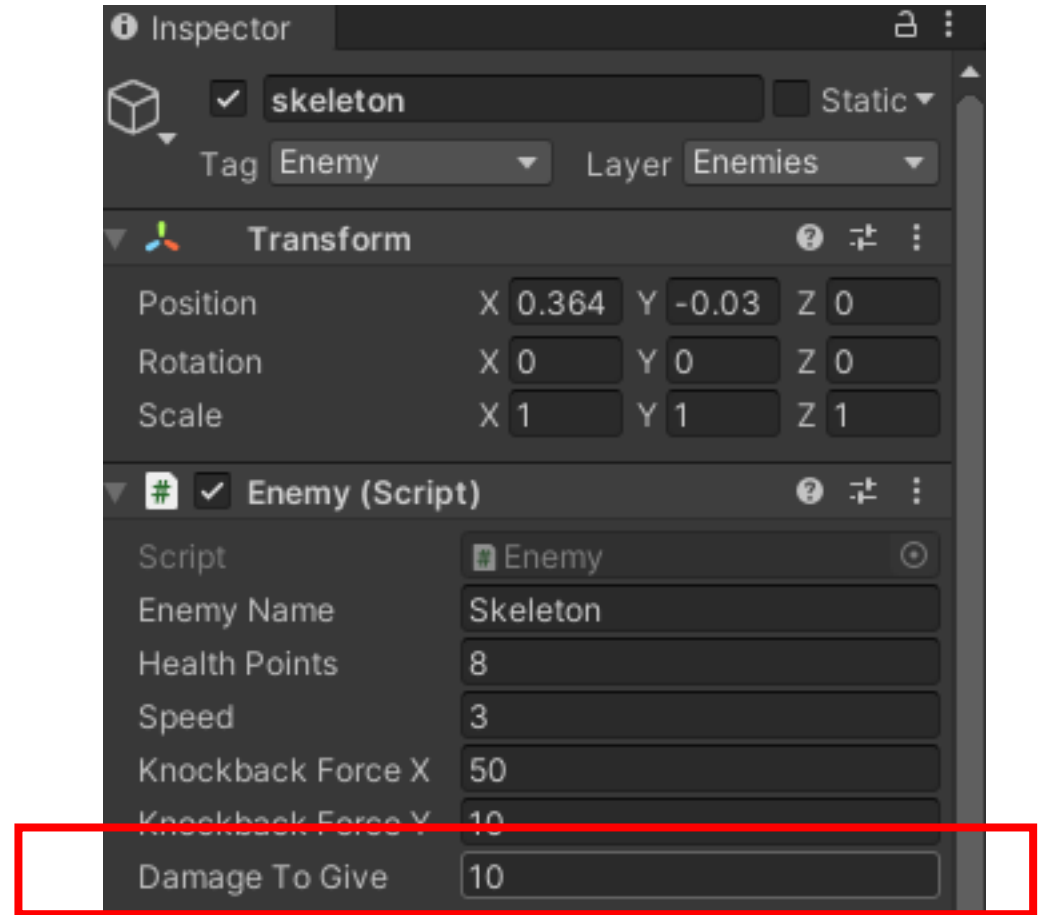
A screenshot of a code editor showing the C# code for an 'Enemy' class. The class inherits from 'MonoBehaviour'. It contains several public fields: 'enemyName' (string), 'healthPoints', 'speed', 'forceX', and 'forceY' (all floats). The line 'public float damageToGive;' is highlighted with a red rectangle, and a red arrow points to it from the right. Below the fields, there is a 'void Start()' method. The background is dark, and the text is light blue/green.

Pedro Yuri Marquez Solis

```
public class Enemy : MonoBehaviour
{
    public string enemyName;
    public float healthPoints;
    public float speed;
    public float knockbackForceX;
    public float knockbackForceY;
    public float damageToGive;

    // Start is called before the first frame update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
    }
}
```



Pedro Yuri Marquez Solis

Crear PlayerHealth

Asignar el script al player

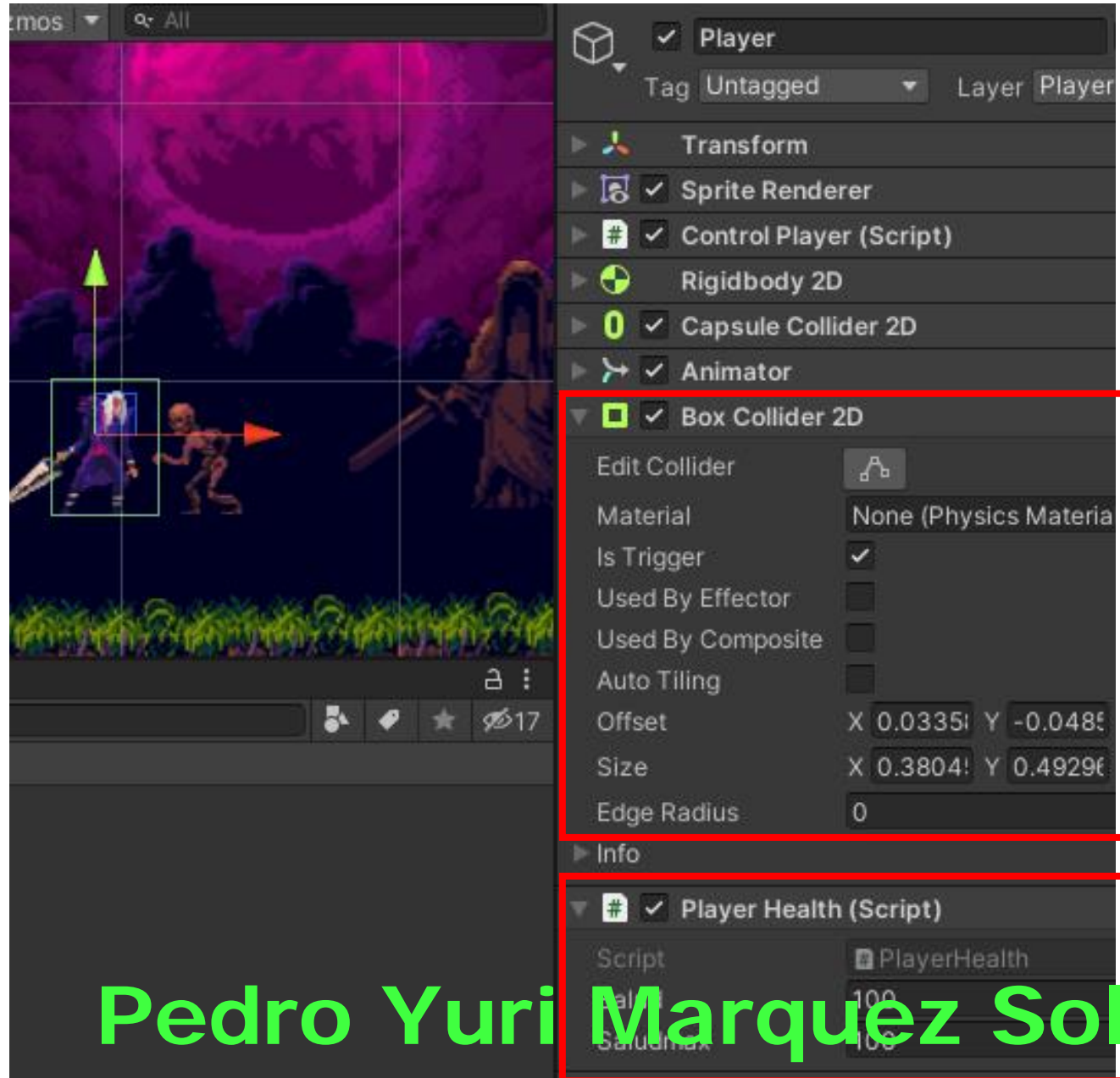
```
public class PlayerHealth : MonoBehaviour
{
    public float salud;
    public float saludmax;
    // Start is called before the first frame update
    void Start()
    {
        salud = saludmax;
    }

    // Update is called once per frame
    void Update()
    {
        if (salud > saludmax)
        {
            salud = saludmax;
        }
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Enemy"))
        {
            salud -= collision.GetComponent<Enemy>().damageToGive;
            if (salud < 0)
            {
                //aparecer pantalla de gameOver
                print("Player dead");
            }
        }
    }
}
```

Pedro Yuri Marquez Solis

Agregar al GameObject Player un trigger



Pedro Yuri Marquez Solis

Probar → Player se descuenta en puntos en forma descontrolada.

Pedro Yuri Marquez Solis

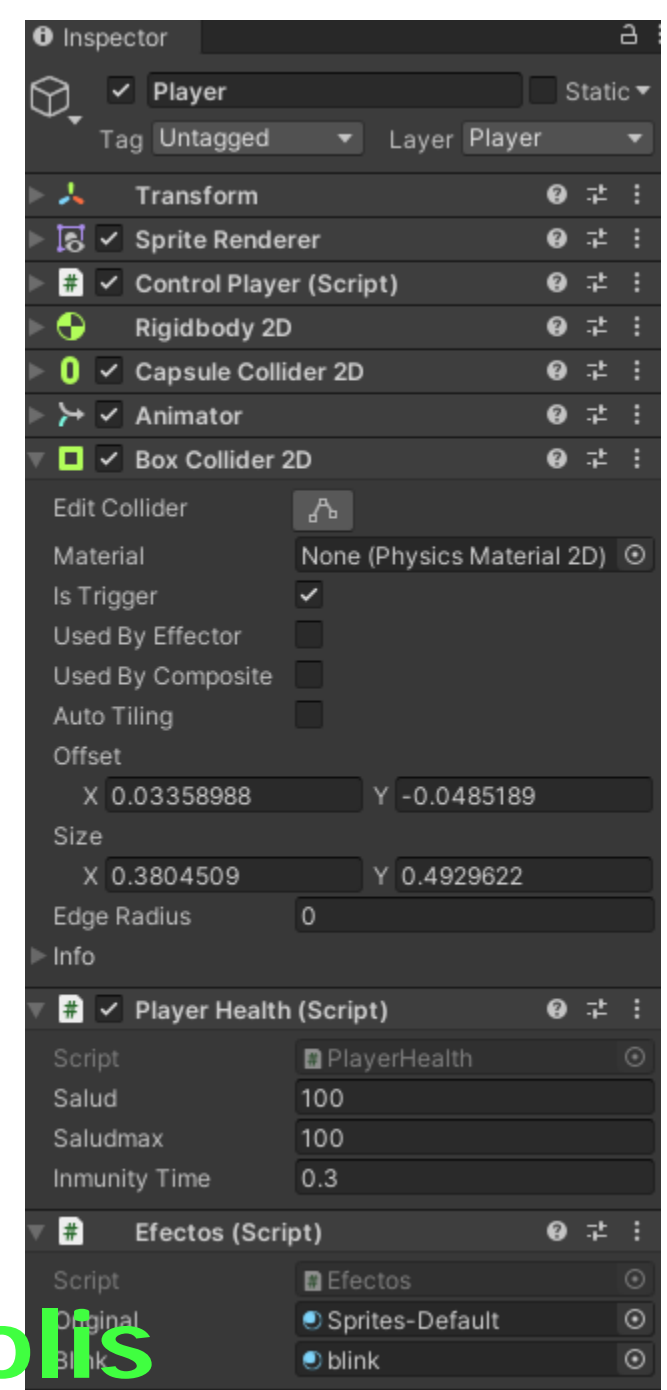
Corregir el descuento de salud a Player

```
public class PlayerHealth : MonoBehaviour
{
    public float salud;
    public float saludmax;
    bool isImmune;
    public float immunityTime;
    Efectos material;
    SpriteRenderer sprite;
    void Start()
    {
        salud = saludmax;
        sprite = GetComponent<SpriteRenderer>();
        material = GetComponent<Efectos>();
    }

    void Update()
    {
        if (salud > saludmax)
        {
            salud = saludmax;
        }
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Enemy") && !isImmune)
        {
            salud -= collision.GetComponent<Enemy>().damageToGive;
            StartCoroutine(Immunity());
            if (salud < 0)
            {
                //aparecer pantalla de gameOver
                print("Player dead");
            }
        }
    }

    IEnumerator Immunity() {
        isImmune = true;
        sprite.material = material.blink;
        yield return new WaitForSeconds(immunityTime);
        sprite.material = material.original;
        isImmune = false;
    }
}
```



Pedro Yuri Marquez Solis


```

public class PlayerHealth : MonoBehaviour
{
    public float salud;
    public float saludmax;
    bool isImmune;
    public float immunityTime;
    Efectos material;
    SpriteRenderer sprite;

    public float knockBackForceX;
    public float knockBackForceY;
    Rigidbody2D rb;
    void Start()
    {
        salud = saludmax;
        sprite = GetComponent<SpriteRenderer>();
        material = GetComponent<Efectos>();
        rb = GetComponent<Rigidbody2D>();
    }

    void Update()
    {
        if (salud > saludmax)
        {
            salud = saludmax;
        }
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Enemy") && !isImmune)
        {
            salud -= collision.GetComponent<Enemy>().damageToGive;
            StartCoroutine(Immunity());
            //3
            if (collision.transform.position.x > transform.position.x) {
                rb.AddForce(new Vector2(-knockBackForceX, knockBackForceY), ForceMode2D.Force);
            }
            else
            {
                rb.AddForce(new Vector2(knockBackForceX, knockBackForceY), ForceMode2D.Force);
            }
            if (salud < 0)
            {
                //aparecer pantalla de gameOver
                print("Player dead");
            }
        }
    }
}

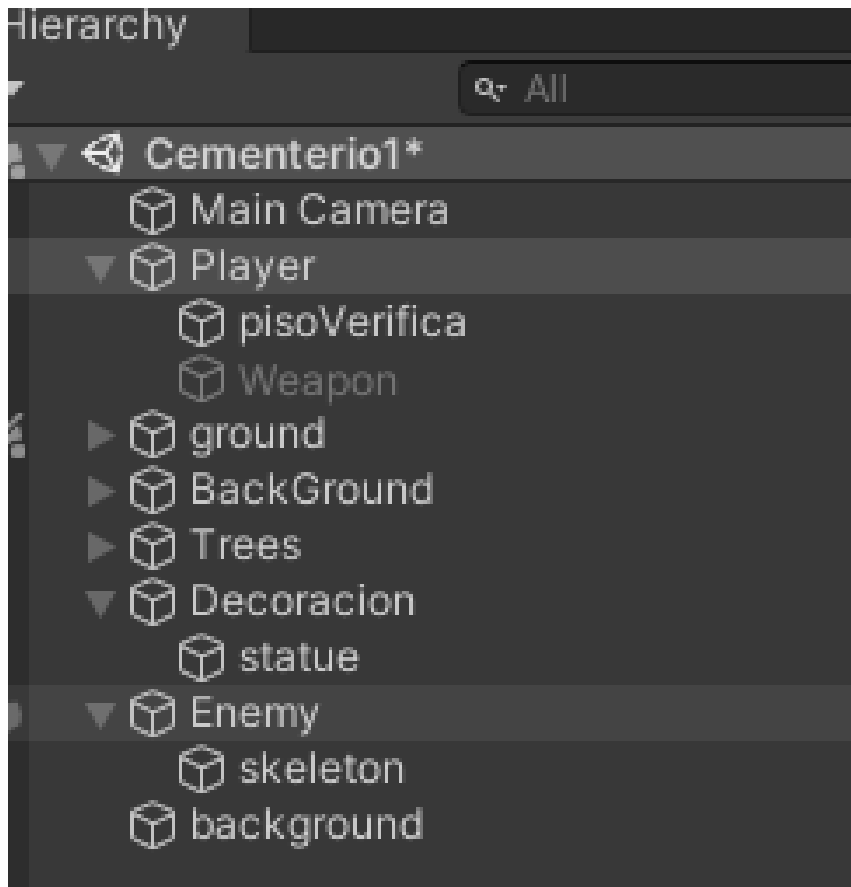
```

Agregando el efecto de Repulsión

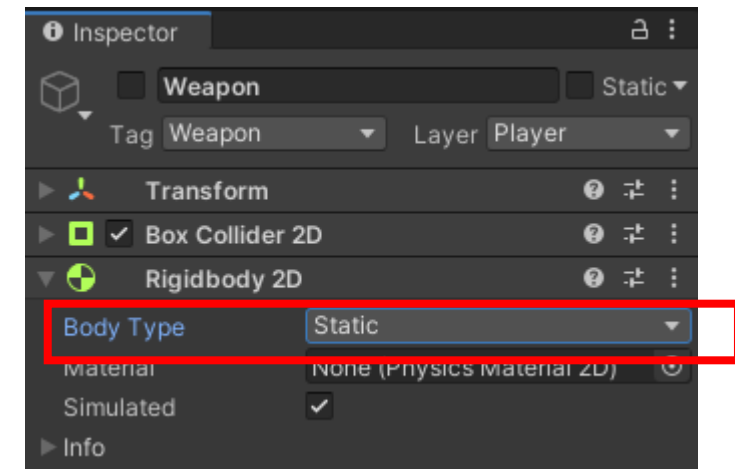


Pedro Yuri Marquez Solis

Corregir al disparar al enemigo



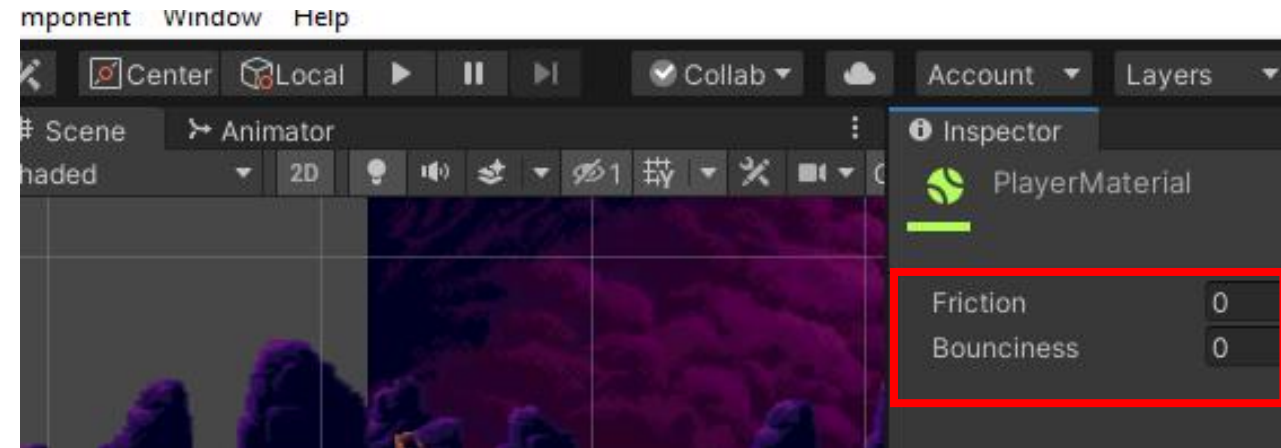
- Se busca al primer Rigidbody dentro de la jerarquía del objeto.
- Debido a que Weapon no tiene ningún **Rigidbody**
- Solucion: Agregar un Rigidbody2d al arma y su BodyType en static para que no vaya hacia abajo.



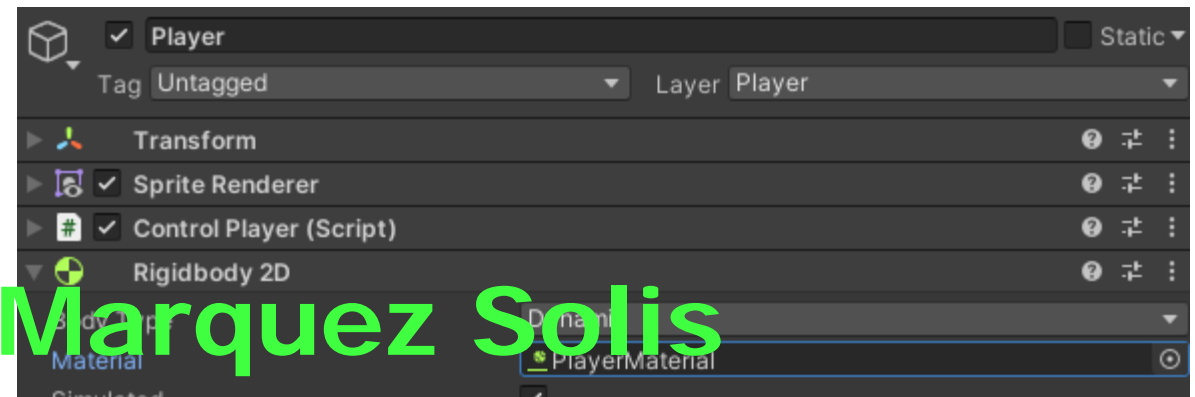
Pedro Yuri Marquez Solis

Si el personaje se pega al Background

- Crear un nuevo physics Material2d, llamado playerMaterial



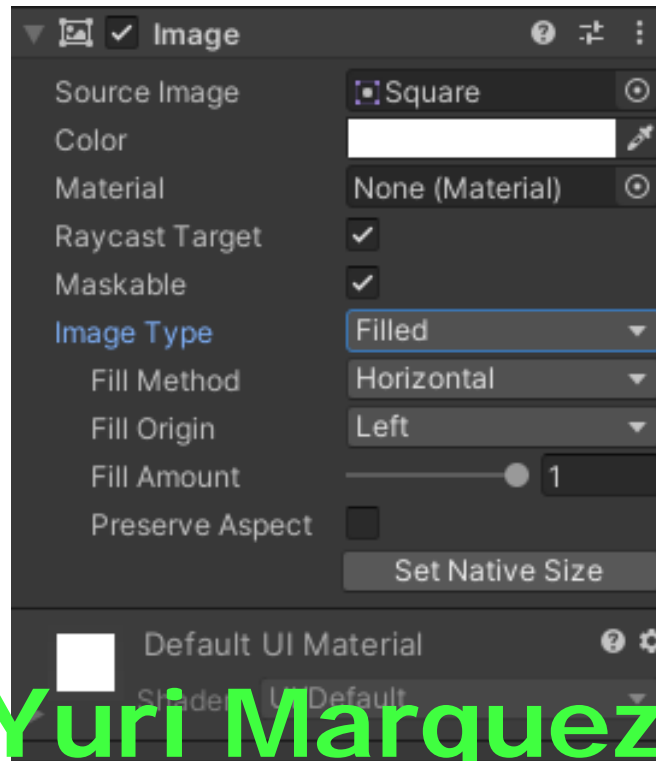
- Asignar al Rigidbody2d de Player



Pedro Yuri Marquez Solis

Interface

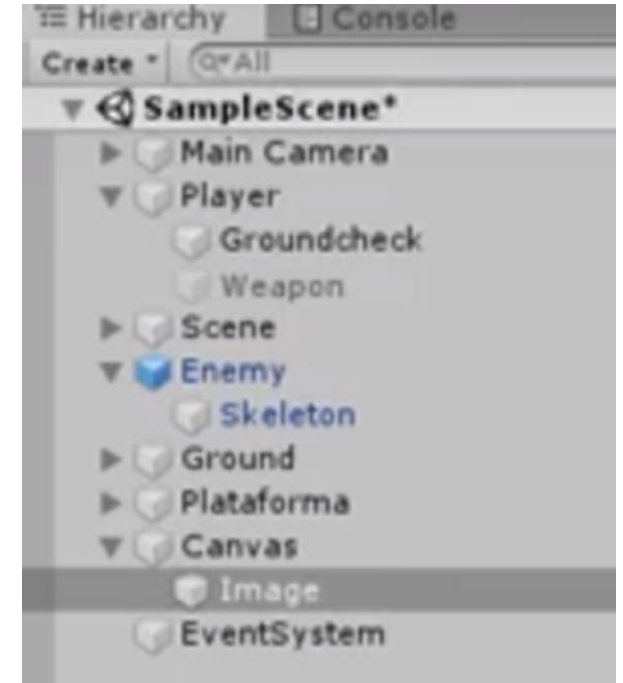
- Crear dentro de Art na nueva carpeta denominada **UI**.
- Crear un Sprite → Square
-



Pedro Yuri Marquez Solis

User Interface

- Crear un canvas en la jerarquía del proyecto
Y dentro del canvas un image, luego asignar al Image en Sourcelmage a Square.

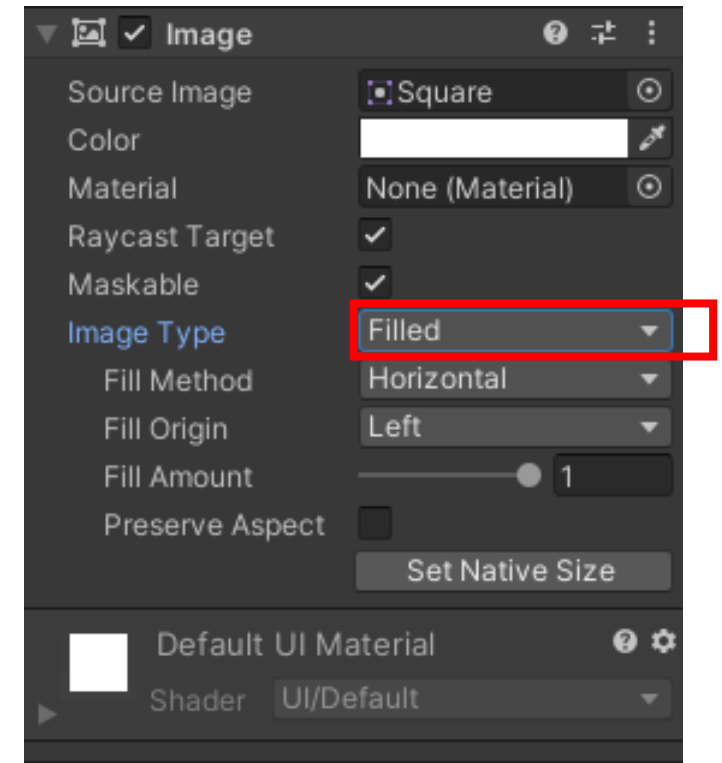


Pedro Yuri Marquez Solis

Interface

- Crear dentro de Art na nueva carpeta denominada UI.
- Crear un Sprite → Square
- Probar cambiando el **ImageType** a Filled y luego el **Fill Amount**:

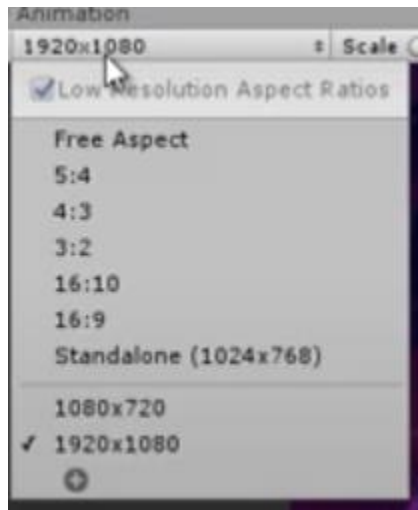
Finalmente dejar el ImageType como simple



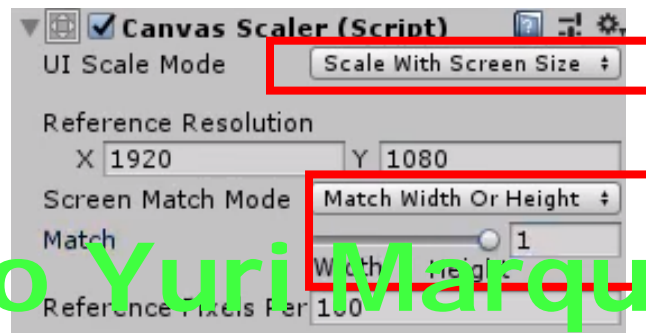
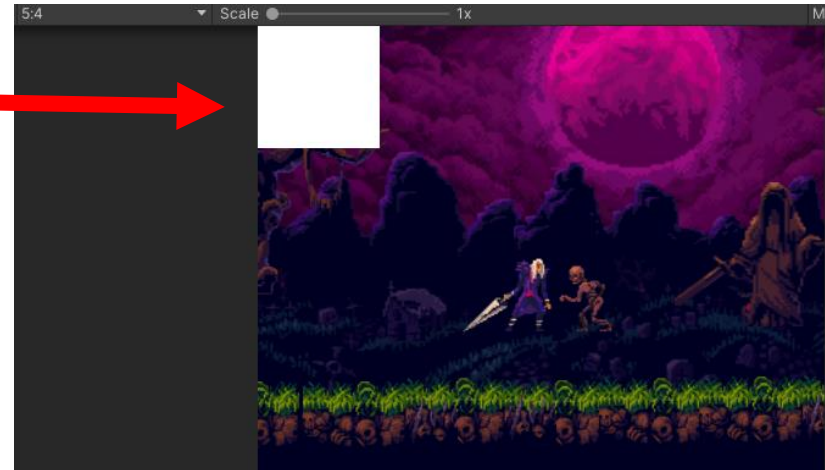
Pedro Yuri Marquez Solis

Probar el escalado

- Puedes probar considerando Otras resoluciones.



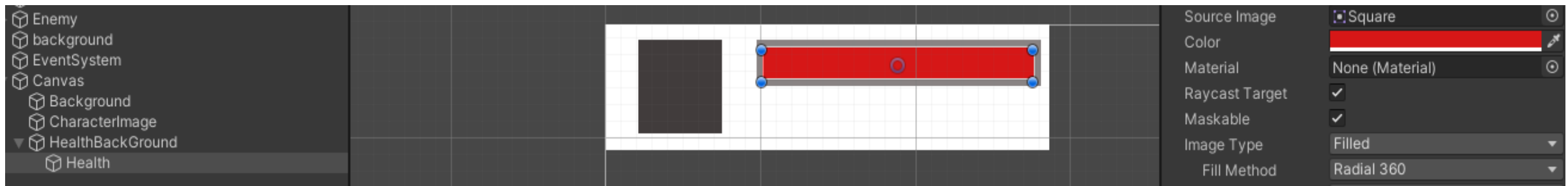
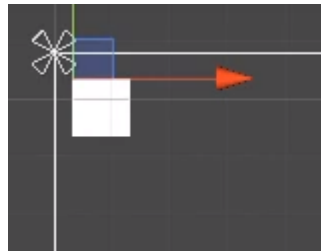
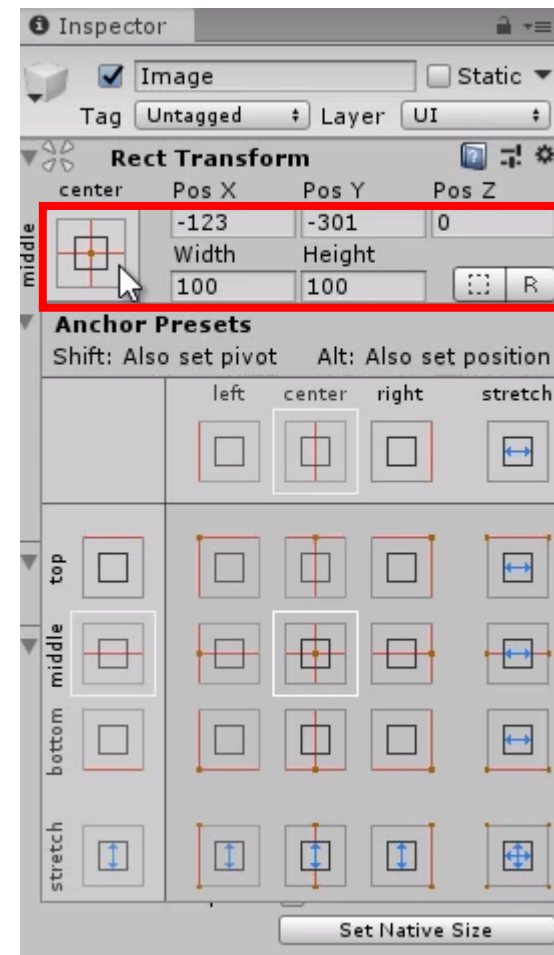
Efectúa los cambios



Pedro Yuri Marquez Solis

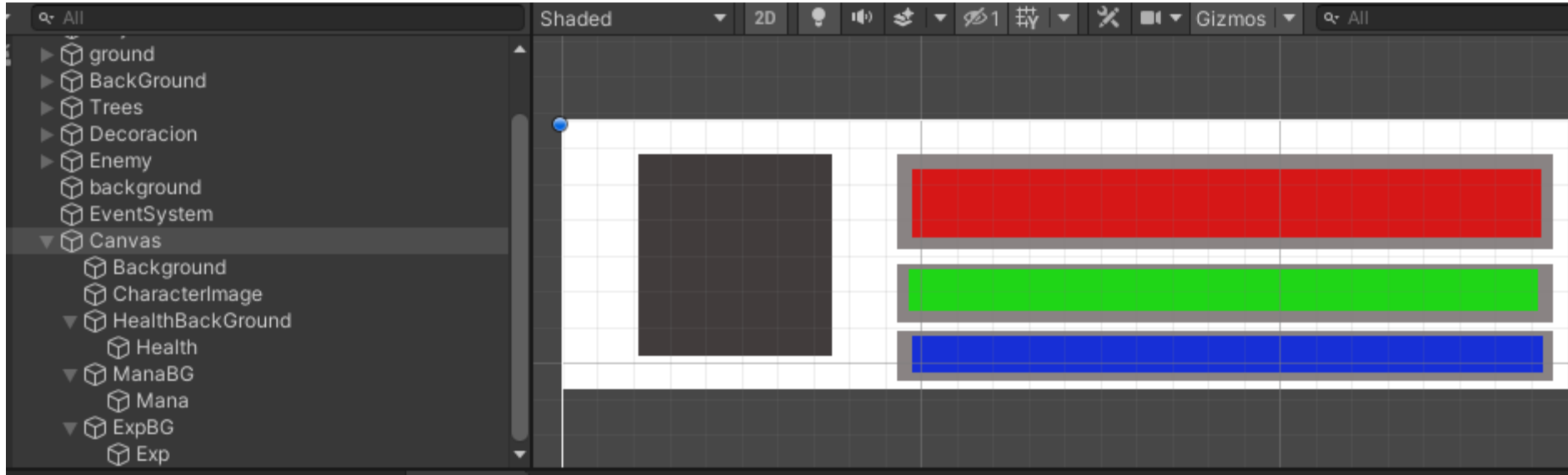
Ubicación de la imagen

- **Anclar** a la parte superior derecha, presionando alt + click.
- Duplica el GameObject y nombralos como:
 - Background (Simple)
 - characterImage (Simple)
 - HealthBackGround (Simple)
 - Health(Filled)



Pedro Yuri Marquez Solis

Duplicar los elementos



Pedro Yuri Marquez Solis

En el script PlayerHealth.

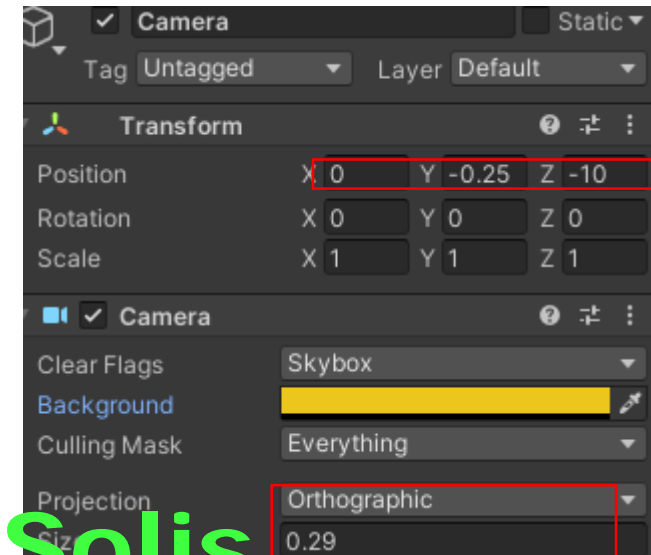
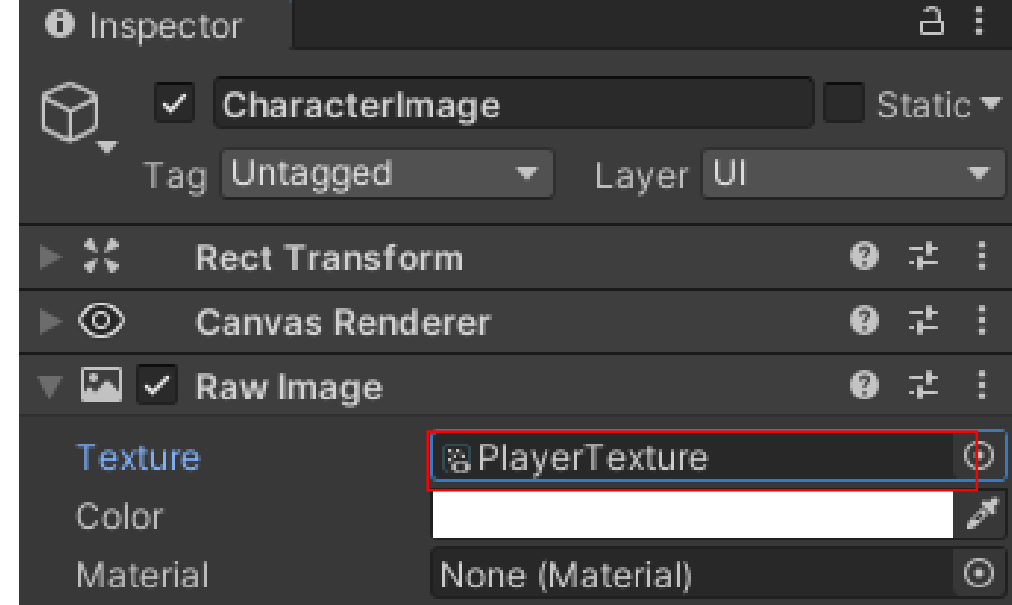
```
public Image HealthImage;  
void Start()  
{  
    salud = saludmax;  
    sprite = GetComponent<SpriteRenderer>();  
    material = GetComponent<Efectos>();  
    rb = GetComponent<Rigidbody2D>();  
}  
  
void Update()  
{  
    HealthImage.fillAmount = salud / 100;
```

- Al actualizar vincular en PlayerHealth en la propiedad HealthImage al image del GameObject **Health**

Pedro Yuri Marquez Solis

Imagen del Player

- Remover el componente Image.
- Agregar un RawImage.
- Crear un folder Texturas y dentro agregar una RenderTexture, llámala PlayerTexture y asignar a la propiedad Texture del RawImage del CharacterImage.
- Agrega una nueva cámara, reinicia todas las posiciones y asigna en el **TargetTexture** a Playertexture



Pedro Yuri Marquez Solis

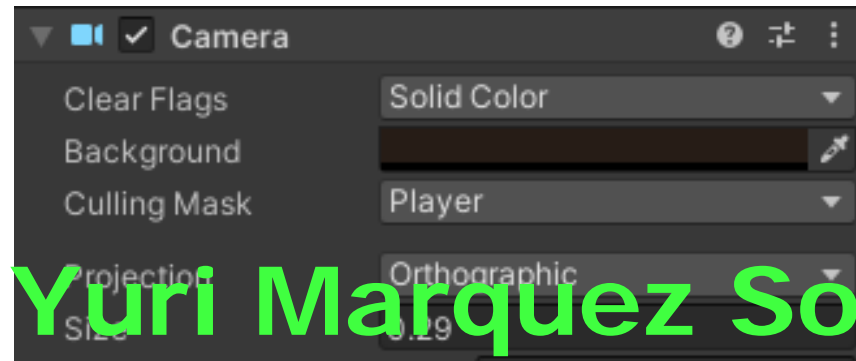
Crear el script UICamera

```
public class UICamera : MonoBehaviour
{
    public Transform player;
    public float xpos, ypos, zpos;

    void Start()
    {
        transform.position = new Vector3(player.position.x, player.position.y, player.position.z);
    }

    // Update is called once per frame
    void Update()
    {
        transform.position = new Vector3(player.position.x + xpos, player.position.y+ypos, zpos);
    }
}
```

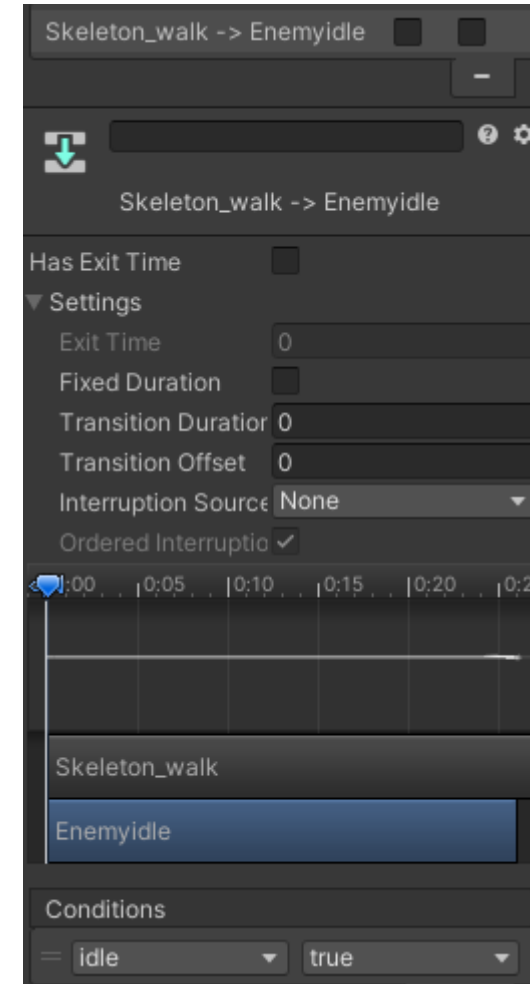
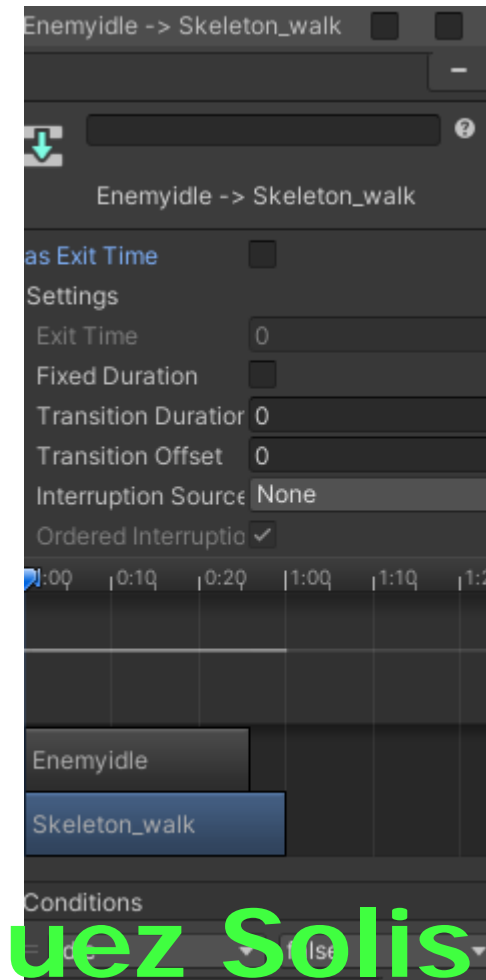
- Configurar la cámara



Pedro Yuri Marquez Solis

Movimiento del enemigo

- Agregar para Skeleton la animación Enemyidle



Pedro Yuri Marquez Solis

Script de movimiento

```
public class EnemyMov : MonoBehaviour
{
    float speed;
    Rigidbody2D rb;
    Animator anim_enemigo;

    public bool isStatic;

    public bool isWalker;
    public bool walksRight;
    void Start()
    {
        speed = GetComponent<Enemy>().speed;
        rb = GetComponent<Rigidbody2D>();
        anim_enemigo = GetComponent<Animator>();
    }
    void Update()
    {
    }

    private void FixedUpdate()
    {
        if (isStatic)
        {
            anim_enemigo.SetBool("idle", true);
            rb.constraints = RigidbodyConstraints2D.FreezeAll;
        }
        if (isWalker)
        {
            rb.constraints = RigidbodyConstraints2D.FreezeRotation;
            if (!walksRight)
            {
                rb.velocity = new Vector2(-speed * Time.deltaTime, rb.velocity.y);
            }
            else
            {
                rb.velocity = new Vector2(speed * Time.deltaTime, rb.velocity.y);
            }
        }
    }
}
```

Pedro Yuri Marquez Solis