

Desarrollo de Videojuegos

Pedro Yuri Marquez Solis



Universidad
Continental



unity

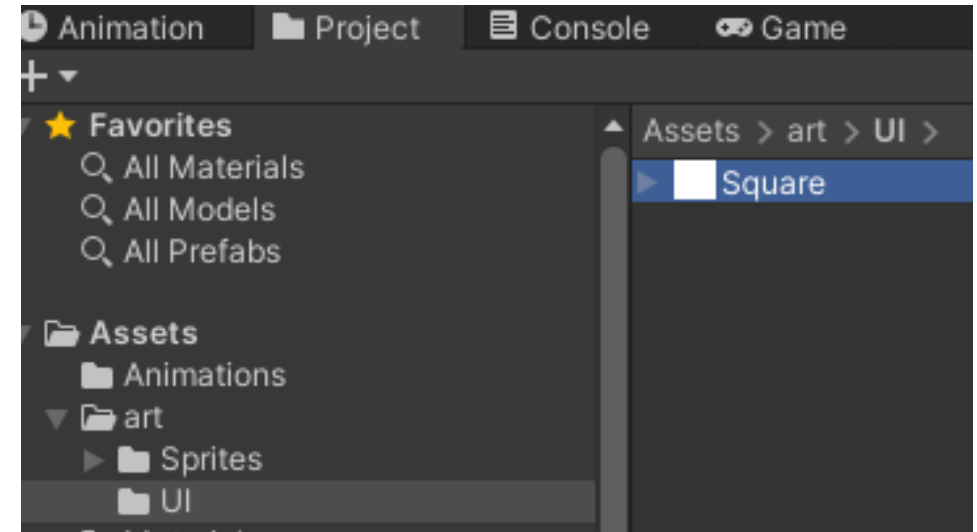


User Interface

- Crear un nuevo directorio:
Art → UI

Dentro de UI crear un nuevo sprite,
Nómbralo como **Square**.

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

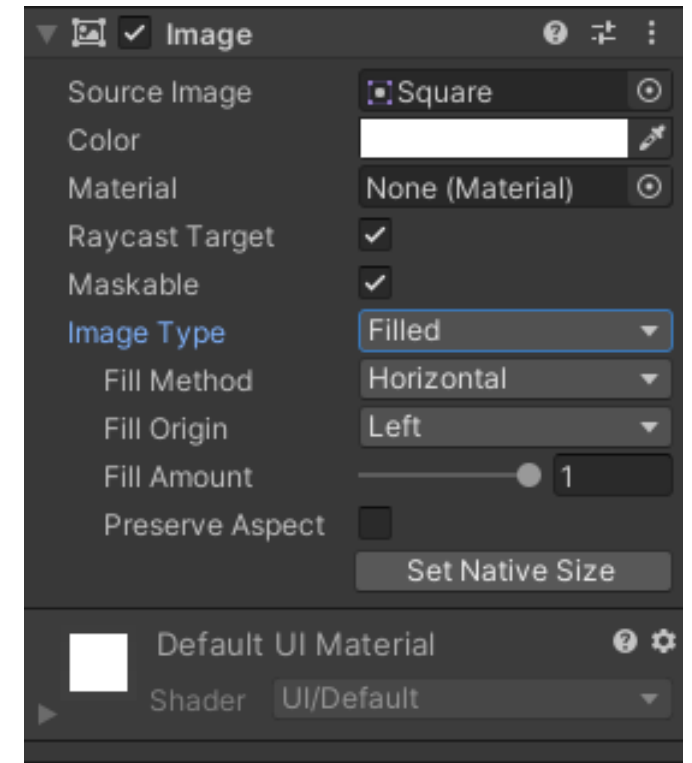


Pedro Yuri Marquez Solis

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**:

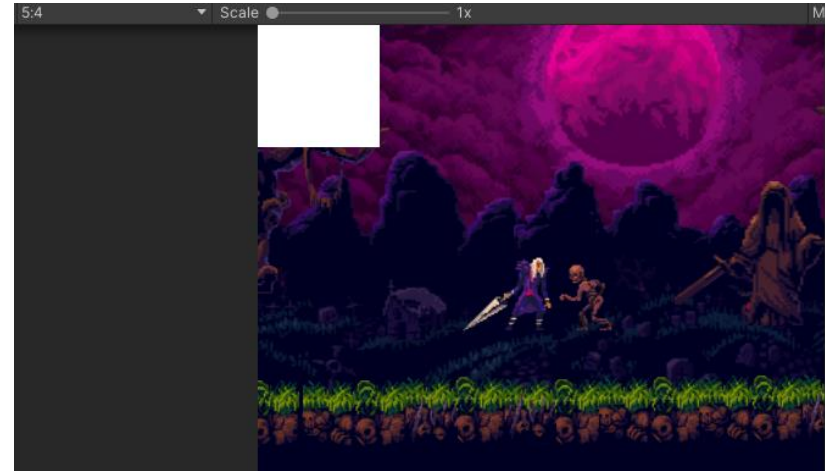
Finalmente dejar el ImageType como simple



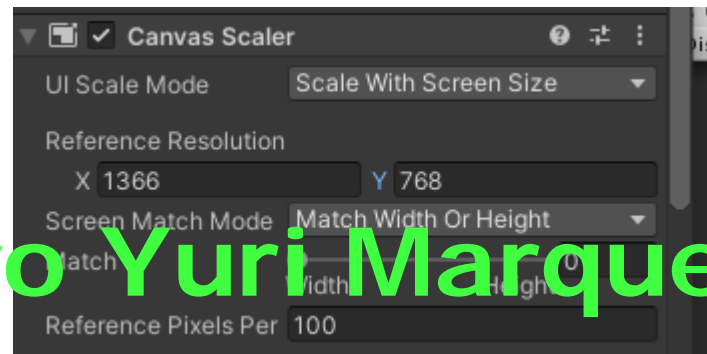
Pedro Yuri Marquez Solis

Probar el escalado

- Puedes probar considerando Otras resoluciones.



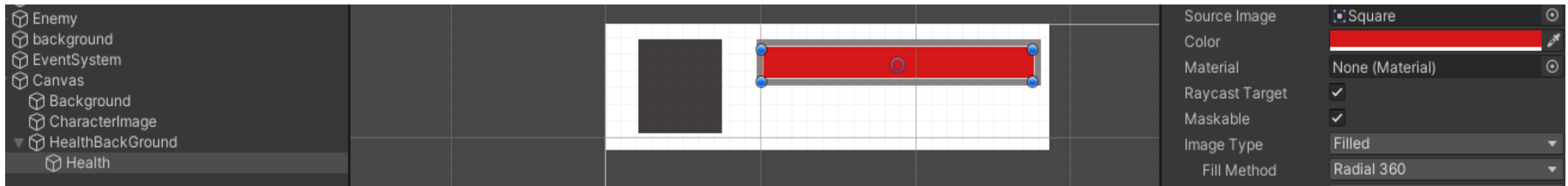
Efectúa los cambios



Pedro Yuri Marquez Solis

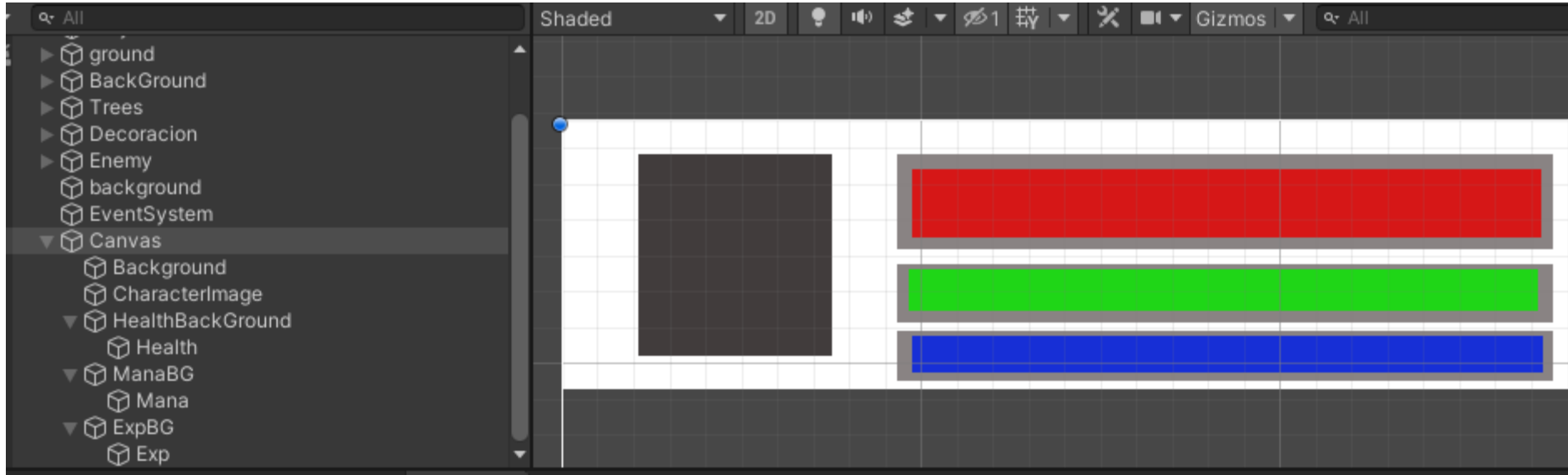
Ubicación de la imagen

- 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.

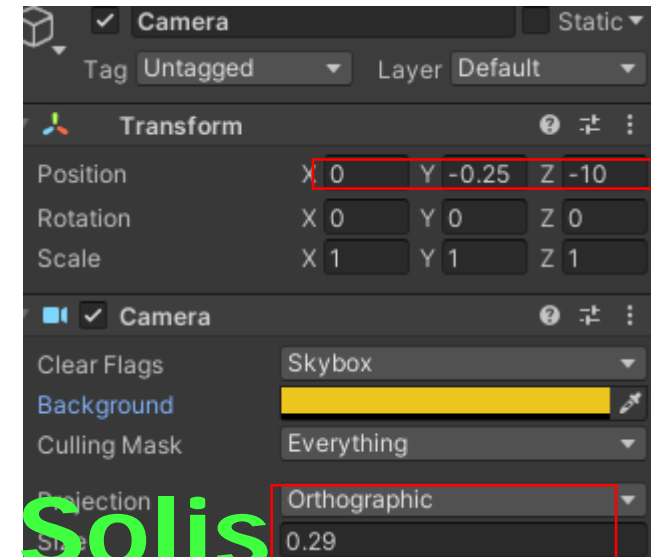
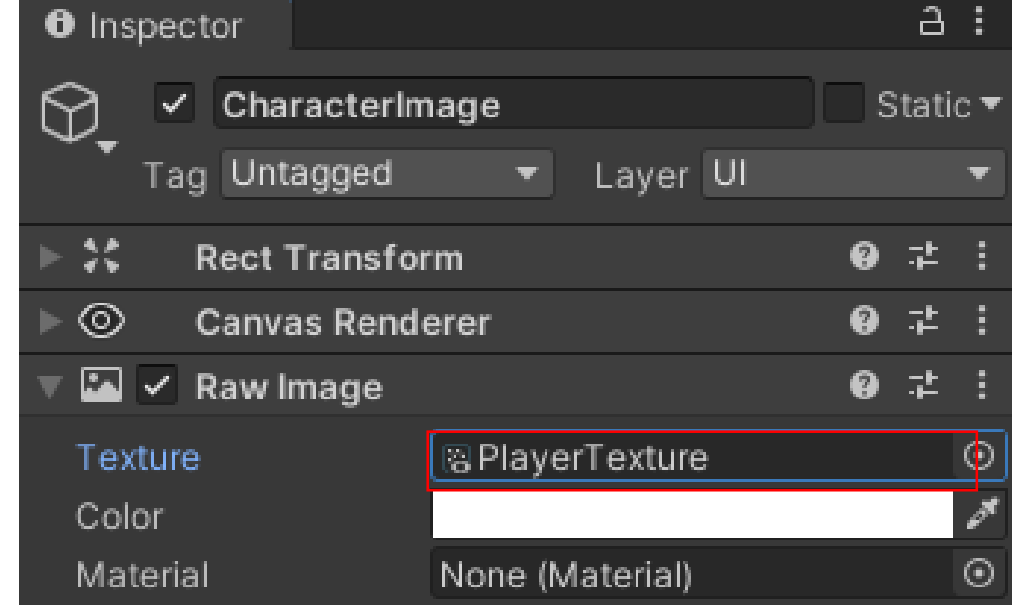
```
public Image HealthImage;  
void Start()  
{  
    salud = saludmax;  
    sprite = GetComponent<SpriteRenderer>();  
    material = GetComponent<Efectos>();  
    rb = GetComponent<Rigidbody2D>();  
}  
  
void Update()  
{  
    HealthImage.fillAmount = salud / 100;
```

- 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

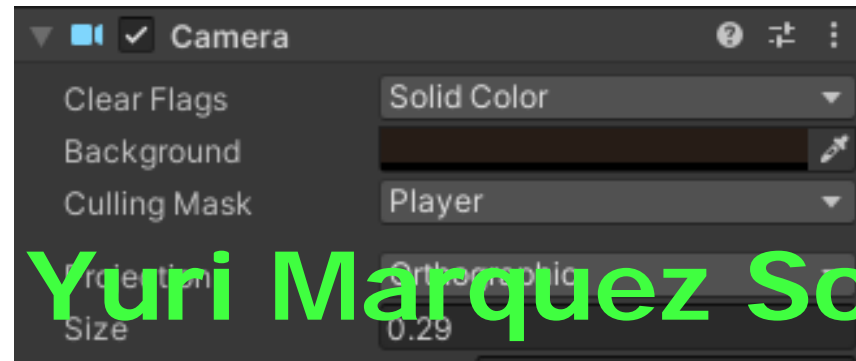
Crear el script UICamera

```
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);
    }
}
```

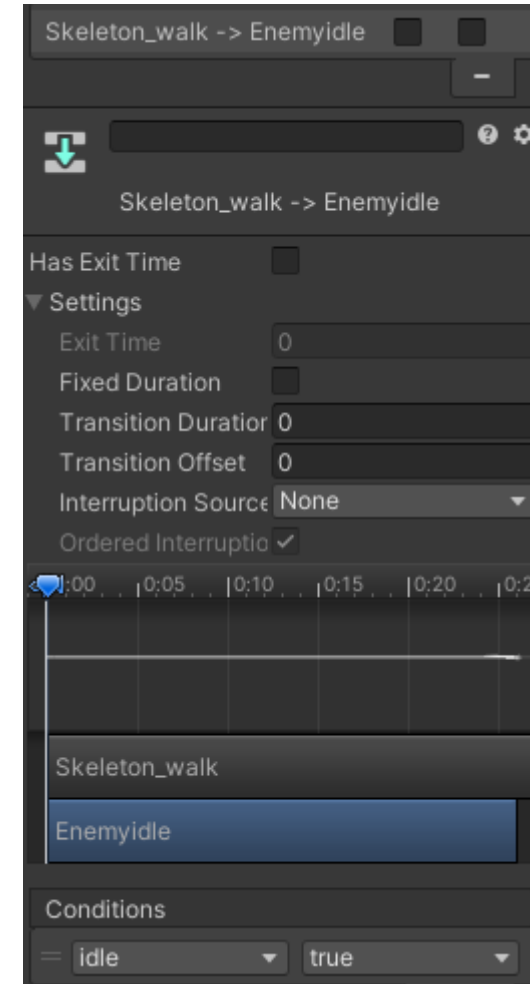
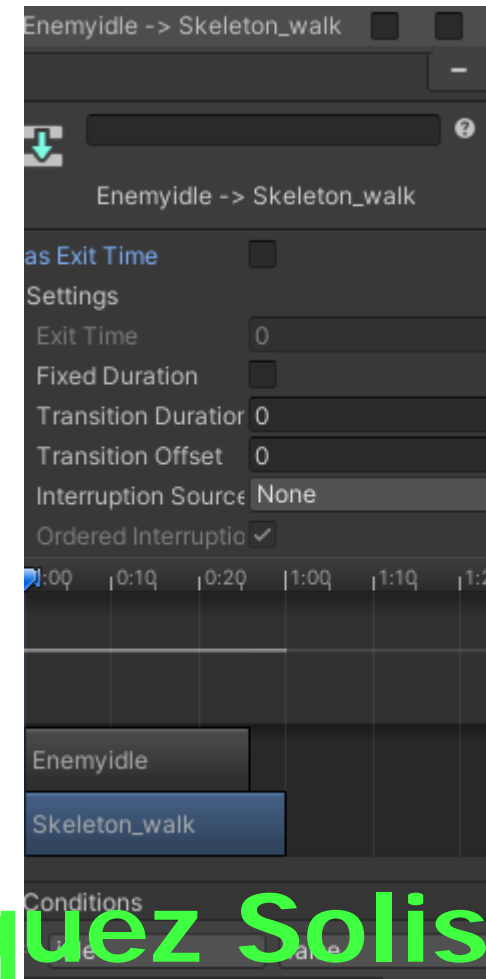
- Configurar la cámara



Pedro Yuri Marquez Solis

Movimiento del enemigo

- Agregar para Skeleton la animación Enemyidle



Pedro Yuri Marquez Solis

Script de movimiento

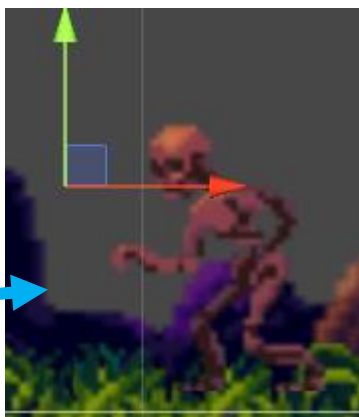
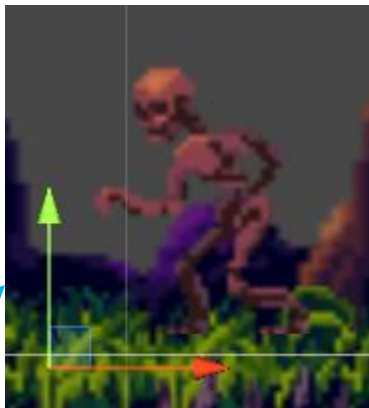
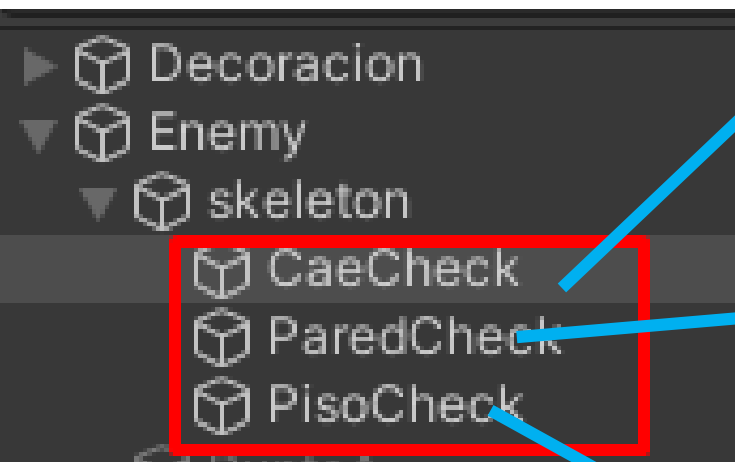
```
public class EnemyMov : MonoBehaviour
{
    float speed;
    Rigidbody2D rb;
    Animator anim_enemigo;

    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



Pedro Yuri Marquez Solis

Modificación 2

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

    public bool isStatic, isWalker, walksRight;

    public Transform paredCheck, caeCheck, pisoCheck;
    public bool paredDetectada, caeDetectado, estaPiso;
    public float radioDeteccion;
    public LayerMask quePisa;
```

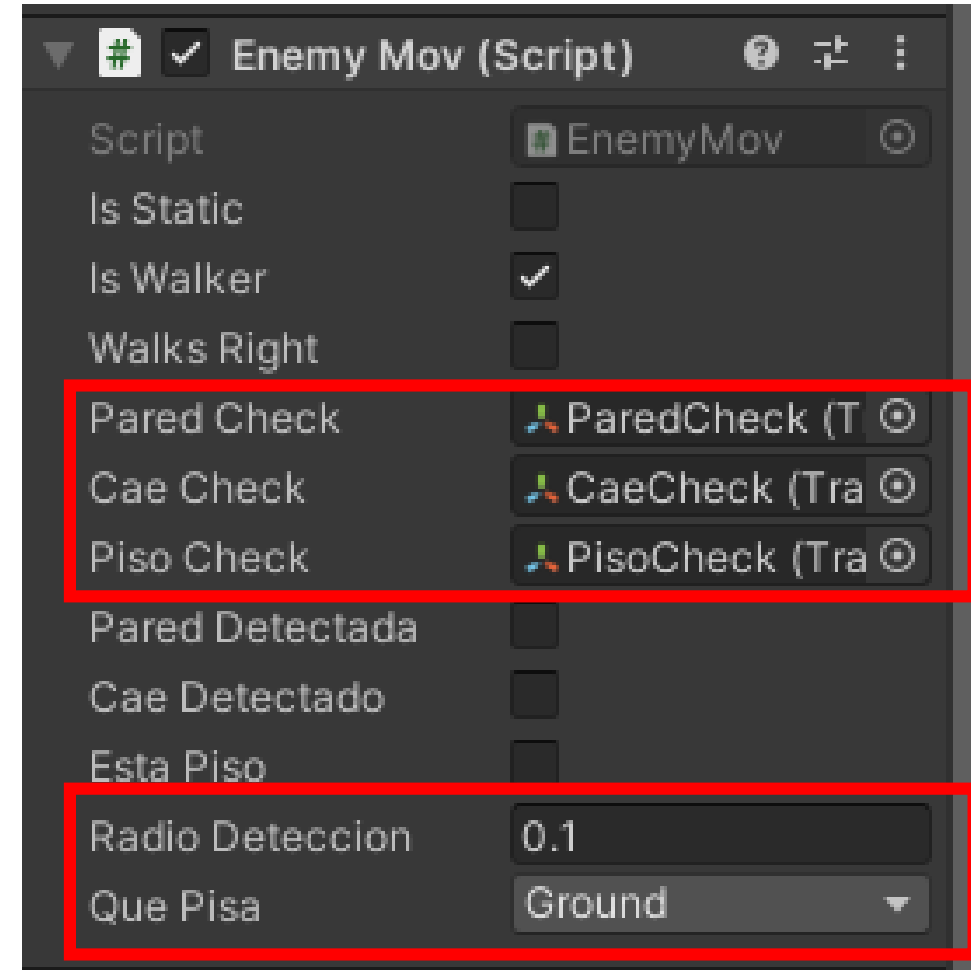
```
private void Update()
{
    caeDetectado = !Physics2D.OverlapCircle(caeCheck.position, radioDeteccion, quePisa);
    paredDetectada = Physics2D.OverlapCircle(paredCheck.position, radioDeteccion, quePisa);

    if (caeDetectado || paredDetectada)
    {
        flip();
    }
}

private void flip()
{
    walksRight = !walksRight;
    transform.localScale = new Vector2(transform.localScale.x * -1f, transform.localScale.y);
}
```

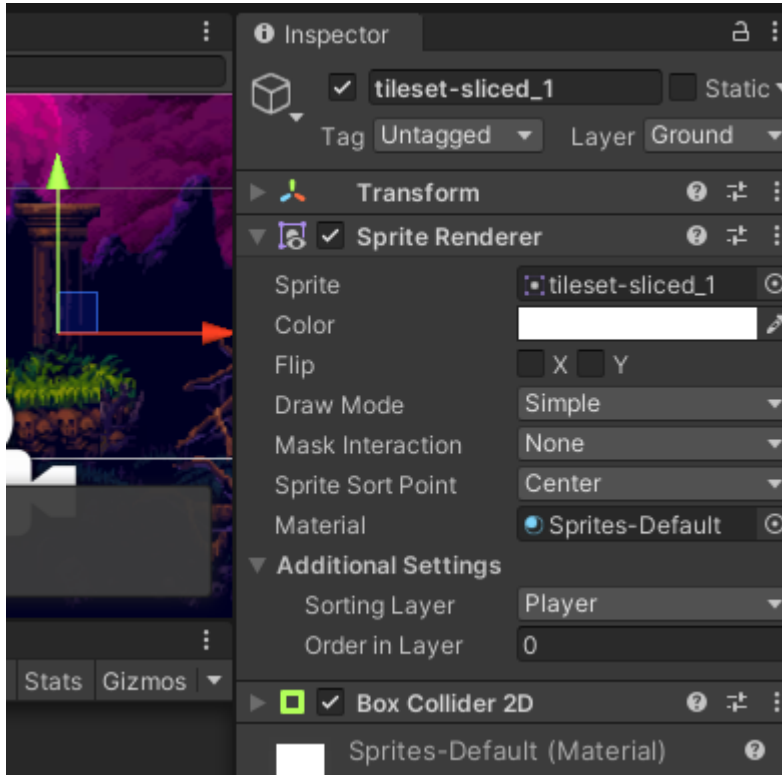
Pedro Yuri Marquez Solis

Asignar los objetos transform



Pedro Yuri Marquez Solis

Agregar un sprite para probar pared



Pedro Yuri Marquez Solis

Modificacion 3

```
estaPiso = Physics2D.OverlapCircle(pisoCheck.position, radioDeteccion, quePisa);
```

```
if (caeDetectado || paredDetectada && estaPiso )  
{  
    flip();  
}
```

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

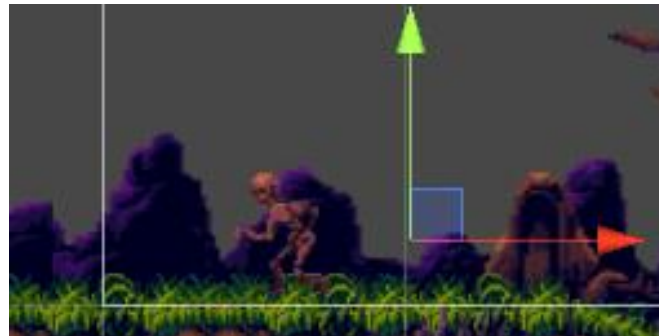
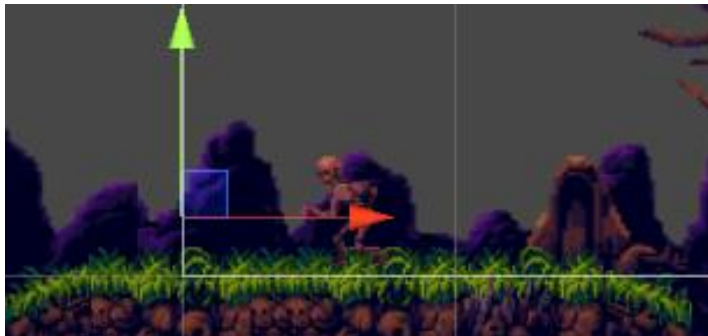
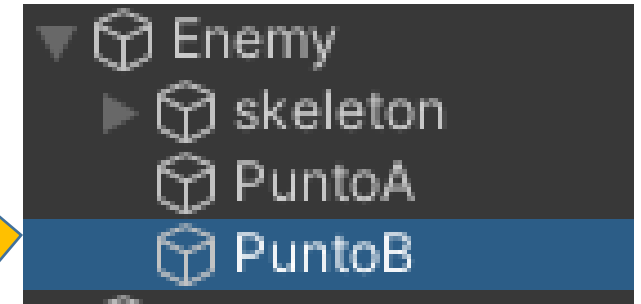
Pedro Yuri Marquez Solis

Movimiento de ataque del enemigo –patrullaje

Agregar dentro de Enemy dos **GameObject empty**.

Nombrarlos como PuntoA y PuntoB

Ubicarlos como se muestra



Puedes activar las etiqueta gráficas si deseas



Pedro Yuri Marquez Solis

Modificaciones en el script

```
public class EnemyMov : MonoBehaviour
{
    float speed;
    Rigidbody2D rb;
    Animator anim_enemigo;
    public bool isStatic, isWalker, walksRight;
    public Transform paredCheck, caeCheck, pisoCheck;
    public bool paredDetectada, caeDetectado, estaPiso;
    public float radioDeteccion;
    public LayerMask quePisa;

    public Transform puntoA, puntoB;
    bool gotoA, gotoB; //si gotoA es true ir a puntoA ~= en gotoB
    public bool isPatrol;

    void Start()
    {
        speed = GetComponent<Enemy>().speed;
        rb = GetComponent<Rigidbody2D>();
        anim_enemigo = GetComponent<Animator>();

        gotoA = true;
    }
}
```

```
private void FixedUpdate()
{
    if (isStatic) ...
    if (isWalker) ...
    if (isPatrol)
    {
        anim_enemigo.SetBool("Idle", false);
        if (gotoA)
        {
            rb.velocity = new Vector2(-speed * Time.deltaTime, rb.velocity.y);
            if (Vector2.Distance(transform.position, puntoA.position) < 0.2f)
            {
                gotoA = false;
                gotoB = true;
            }
        }
        if (gotoB) {
            rb.velocity = new Vector2(speed * Time.deltaTime, rb.velocity.y);
            if (Vector2.Distance(transform.position, puntoB.position) < 0.2f)
            {
                gotoA = true;
                gotoB = false;
            }
        }
    }
}
```

Pedro Yuri Marquez Solis

Probar el movimiento

- Por ahora no tiene rotación de dirección.



- Para que rote llamar a flip()

```
rb.velocity = new Vector2(-speed * Time.deltaTime, rb.velocity.y);  
if (Vector2.Distance(transform.position, puntoA.position) < 0.2f)  
{  
    gotoA = false;  
    gotoB = true; flip();  
}
```

Pedro Yuri Marquez Solis

Haciendo que enemigo espere

1ro. agregar las siguientes propiedades a EnemyMov:

```
public bool debeEsperar;  
public float tiempoEspera;  
public bool estaEsperando;
```

2do Modificar **isPatrol**

```
if (isPatrol)  
{  
    if (gotoA)  
    {  
        if (!estaEsperando)  
        {  
            rb.velocity = new Vector2(-speed * Time.deltaTime, rb.velocity.y);  
            anim_enemigo.SetBool("Idle", false);  
        }  
        if (Vector2.Distance(transform.position, puntoA.position) < 0.2f)  
        {  
            if (debeEsperar) StartCoroutine(Esperando());  
            gotoA = false;  
            gotoB = true;  
        }  
    }  
    if (gotoB) {  
        if (!estaEsperando)  
        {  
            rb.velocity = new Vector2(speed * Time.deltaTime, rb.velocity.y);  
            anim_enemigo.SetBool("Idle", false);  
        }  
        if (Vector2.Distance(transform.position, puntoB.position) < 0.2f)  
        {  
            if (debeEsperar) StartCoroutine(Esperando());  
            gotoA = true;  
            gotoB = false;  
        }  
    }  
}
```

Pedro Yuri Marquez Solis

Haciendo que enemigo espere 2

- Modificar Esperando:

```
IEnumerator Esperando() {  
    anim_enemigo.SetBool("Idle",true);// sólo espe  
    estaEsperando = true;  
    yield return new WaitForSeconds(tiempoEspera);  
    estaEsperando = false;  
    anim_enemigo.SetBool("Idle", false);  
    flip();  
}
```

- Para las pruebas →



Pedro Yuri Marquez Solis

Duplicando el enemigo

- Al probar se debe verificar que los enemigos se empujan entre si.

Solución:

En Edit → project settings → physics 2d:



Pedro Yuri Marquez Solis

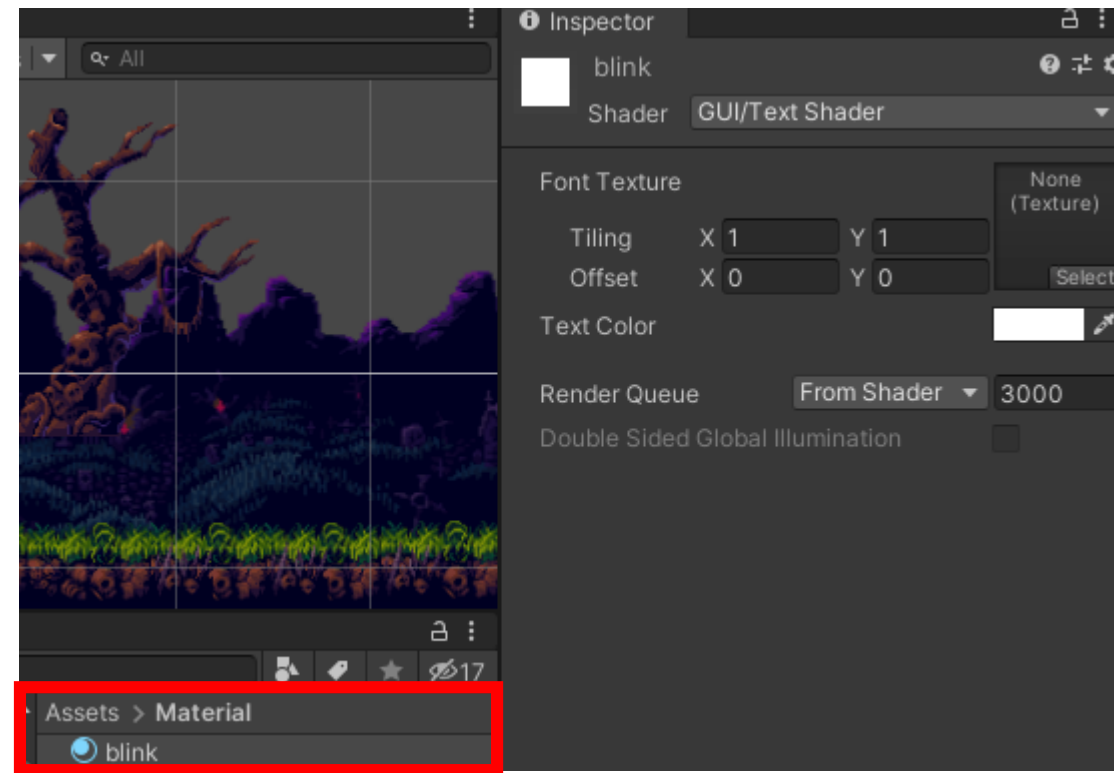
En el script PlayerHealth.

```
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

Corregir el error de no afectación del enemigo de espaldas.



Pedro Yuri Marquez Solis

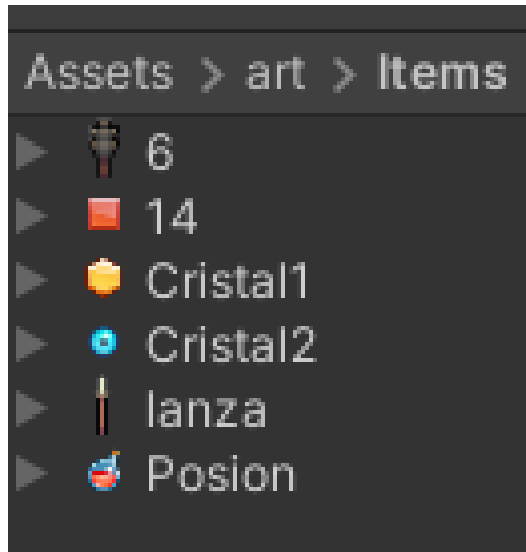
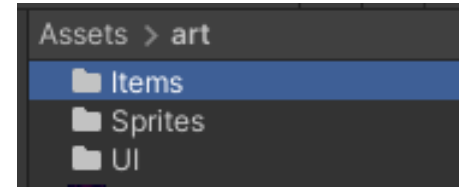
Agregando Items

Dinero, posiciones

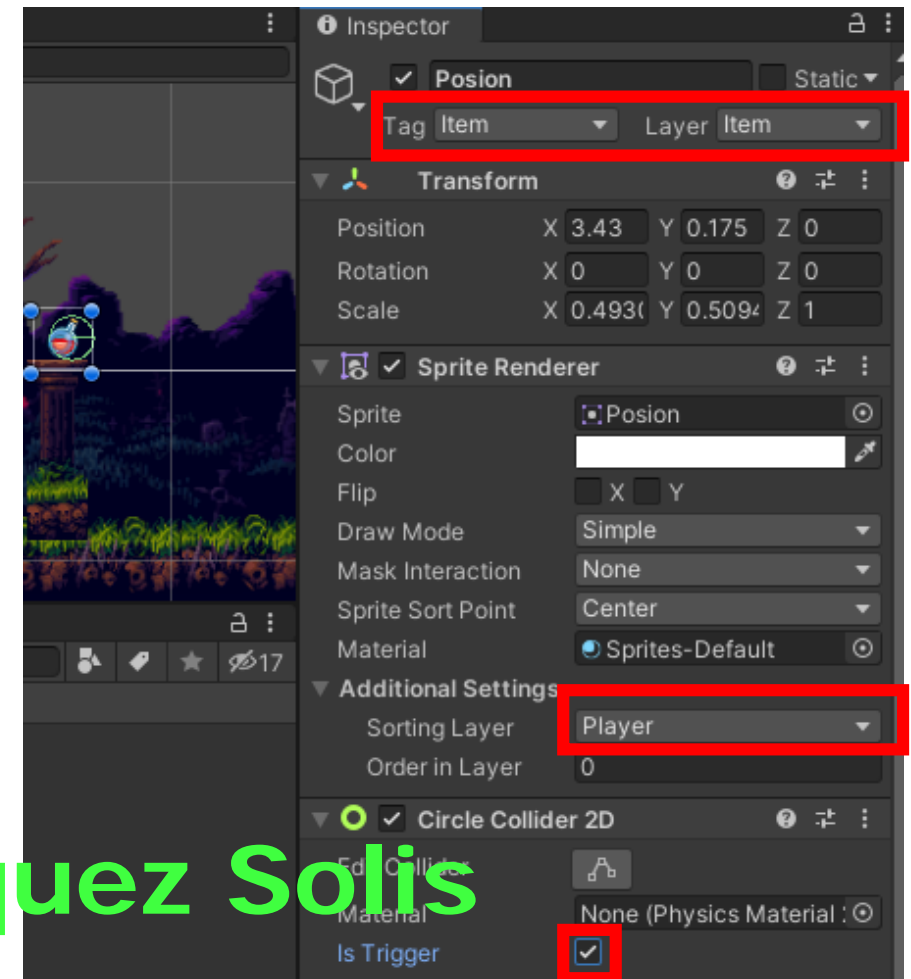
Pedro Yuri Marquez Solis

Creando Item de curación

- Agregar dentro del directorio Arte un directorio llamado Items.
- Agregar al proyecto dentro de Items, renombrar de ser necesario:



- Configurar Posion como se muestra

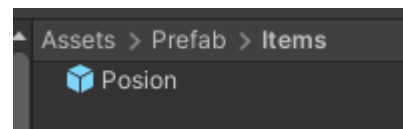
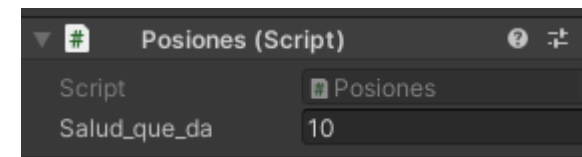


Pedro Yuri Marquez Solis

Crear el script Pociones

```
public class Posiones : MonoBehaviour
{
    public float salud_que_da;
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Player"))
        {
            collision.GetComponent<PlayerHealth>().salud += salud_que_da;
            Destroy(gameObject);
        }
    }
}
```

- Agregar el script a Posion y configurar
- Convertir a Prefab y utilizar



Pedro Yuri Marquez Solis