

Desarrollo de Videojuegos

Pedro Yuri Marquez Solis



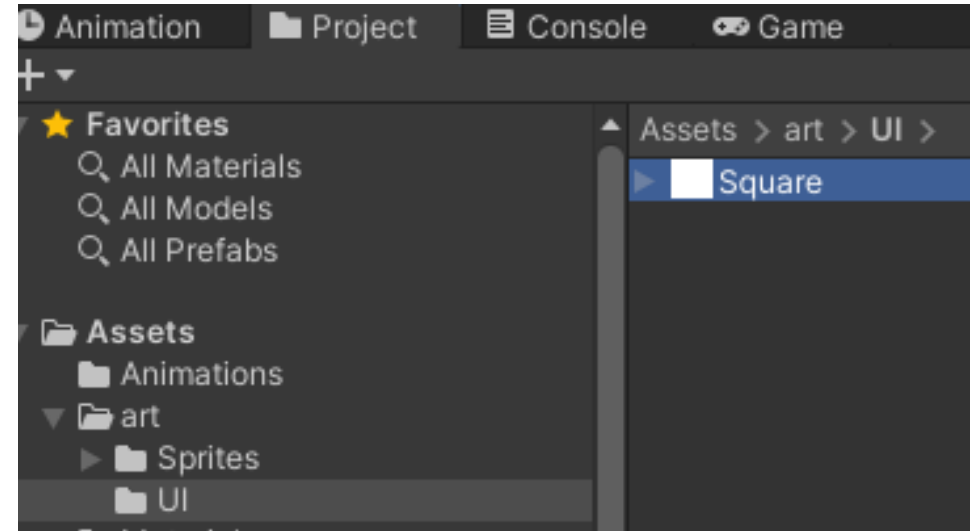
Universidad
Continental



User Interface

- Crear un nuevo directorio:
Art → UI

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



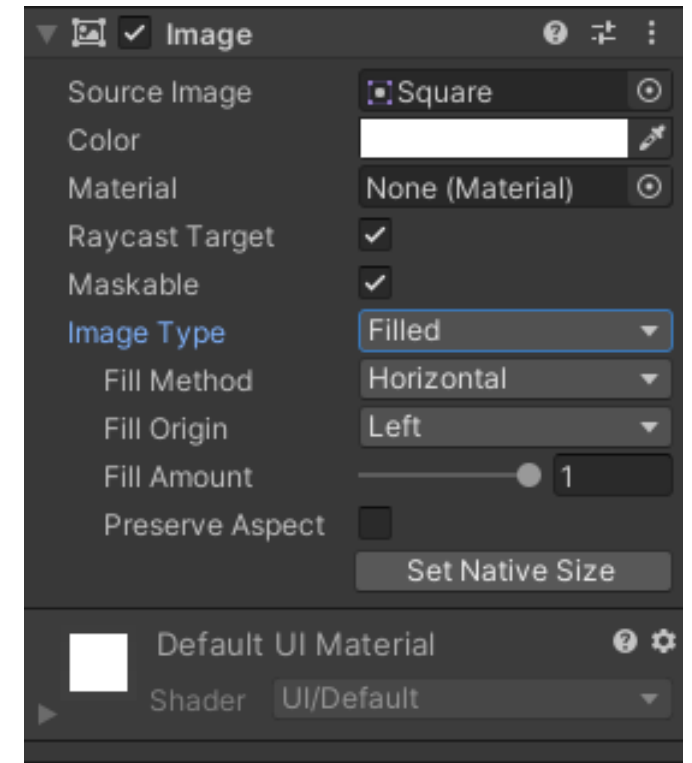
- Crear un canvas en la jerarquía del proyecto
Y dentro del canvas un image, luego asignar al
Image en Sourcelmage a Square.

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**:

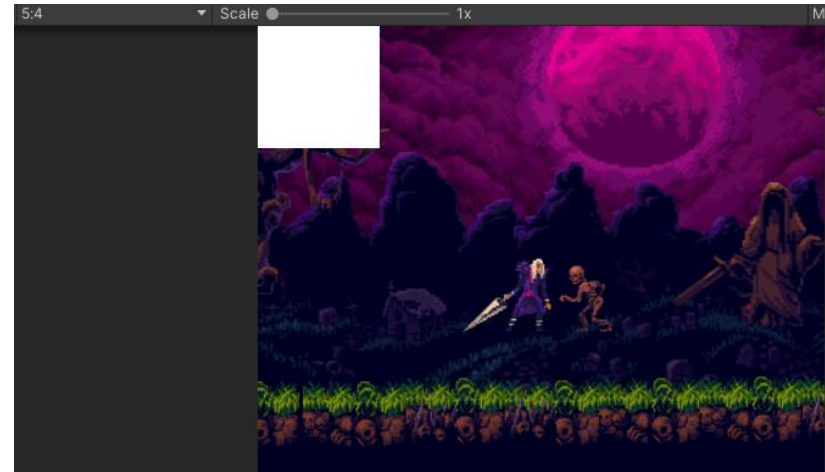
Finalmente dejar el ImageType como simple



Pedro Yuri Marquez Solis

Probar el escalado

- Puedes probar considerando Otras resoluciones.



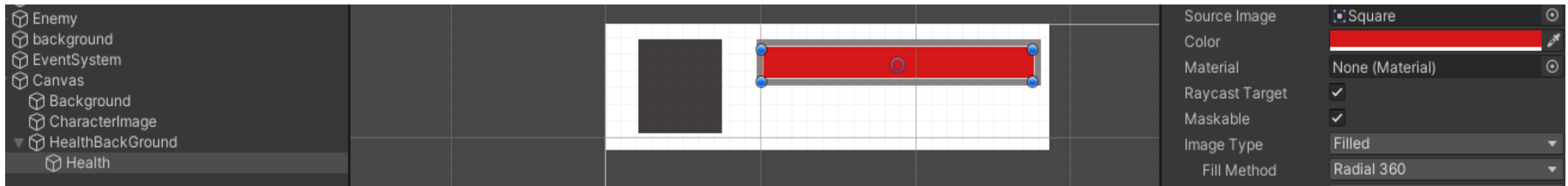
Efectúa los cambios



Pedro Yuri Marquez Solis

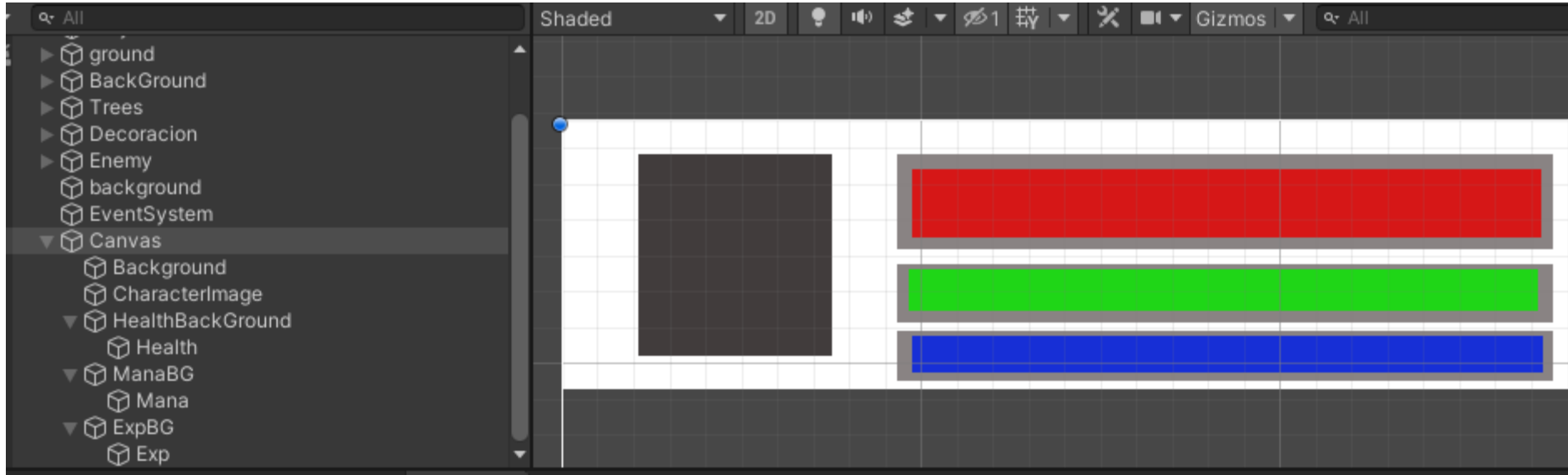
Ubicación de la imagen

- 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.

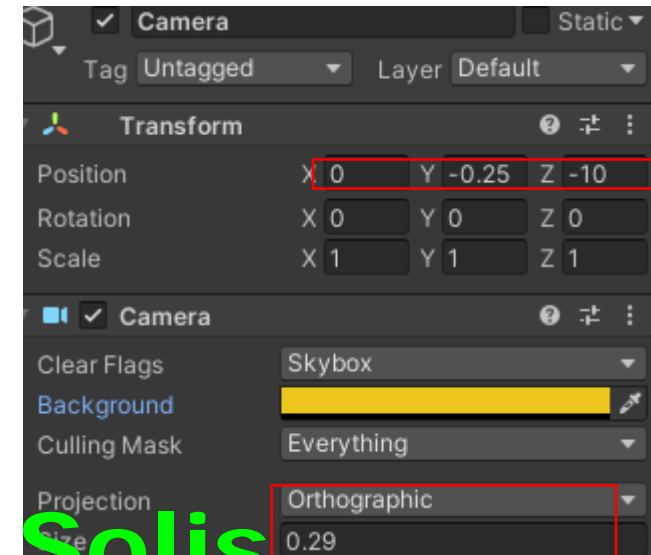
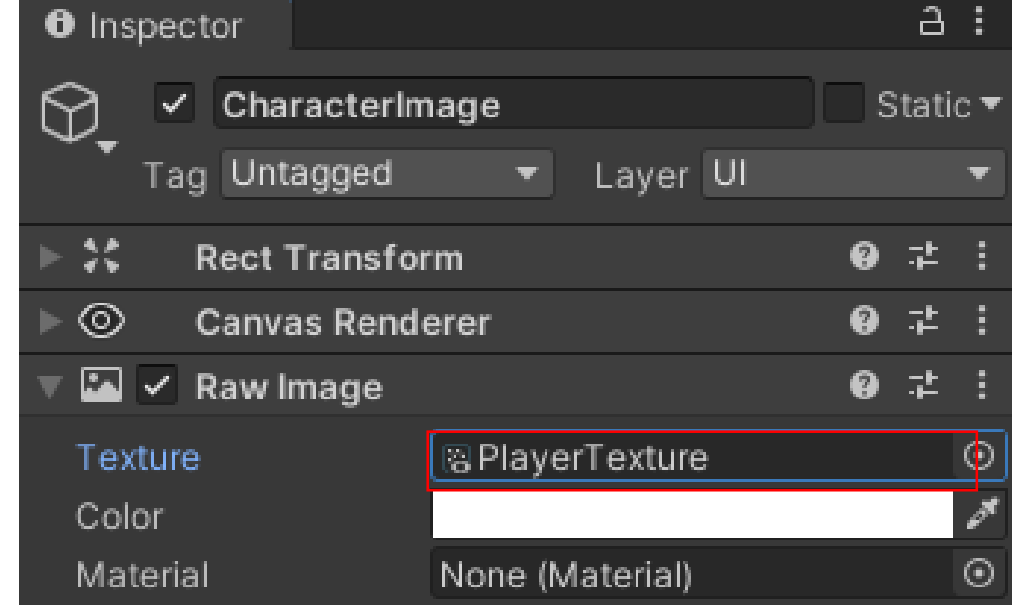
```
public Image HealthImage;  
void Start()  
{  
    salud = saludmax;  
    sprite = GetComponent<SpriteRenderer>();  
    material = GetComponent<Efectos>();  
    rb = GetComponent<Rigidbody2D>();  
}  
  
void Update()  
{  
    HealthImage.fillAmount = salud / 100;
```

- 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);
    }
}
```

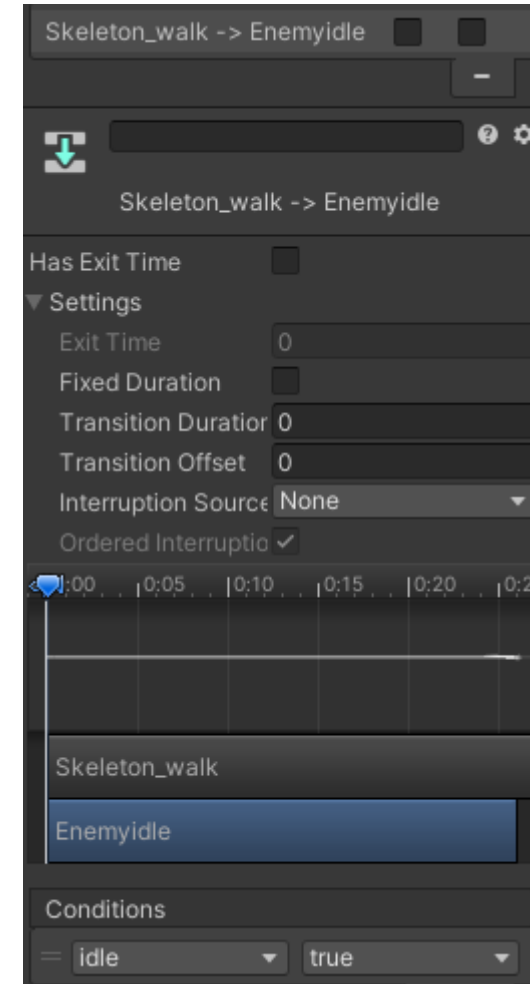
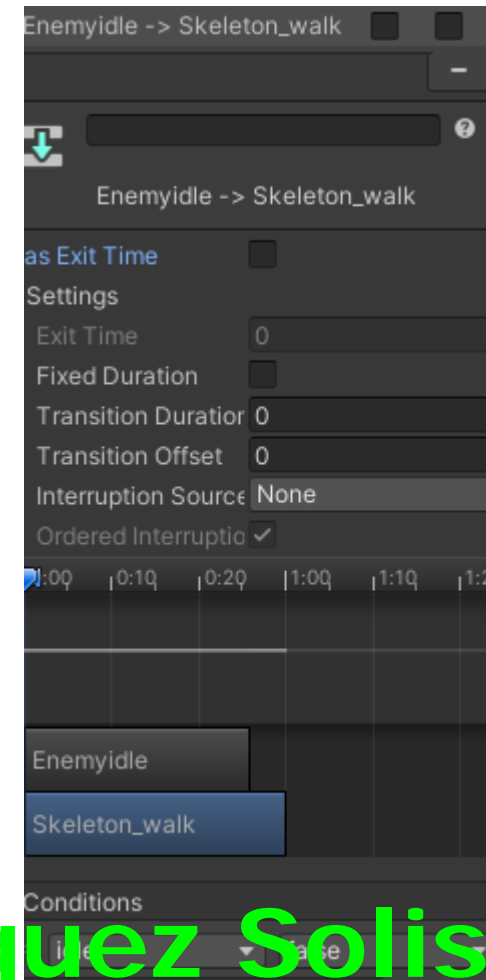
- Configurar la cámara



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