

# Desarrollo de Videojuegos

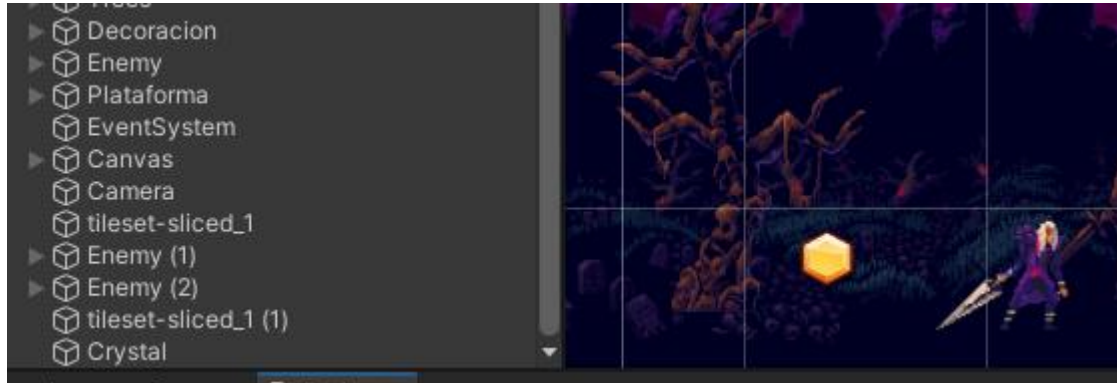
Pedro Yuri Marquez Solis



Universidad  
Continental



# Agregar el Item Crystal



- Agregar un CircleCollider y adaptar al tamaño de Crystal.
- Renombrar el Canvas por **GameManager**

**Pedro Yuri Marquez Solis**

# Agregar el scripts MonedaOro

```
public class MonedaOro : MonoBehaviour
{
    public float monedasOtorgar;

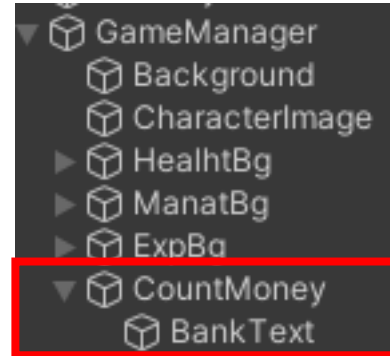
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Player"))
        {
            CuentaMonedas.instance.Money(monedasOtorgar);
            print("monedas");
            Destroy(gameObject);
        }
    }
}
```

- Asignar MonedaOro al Crystal

**Pedro Yuri Marquez Solis**

# El panel de puntos

- Agregar un image y un Text



- Anclar en la parte inferior izquierda el panel de monedas.
- Ajustar las propiedades



Pedro Yuri Marquez Solis

# CuentaMonedas

```
public class CuentaMonedas : MonoBehaviour
{
    public float monedas;
    public Text bankText;

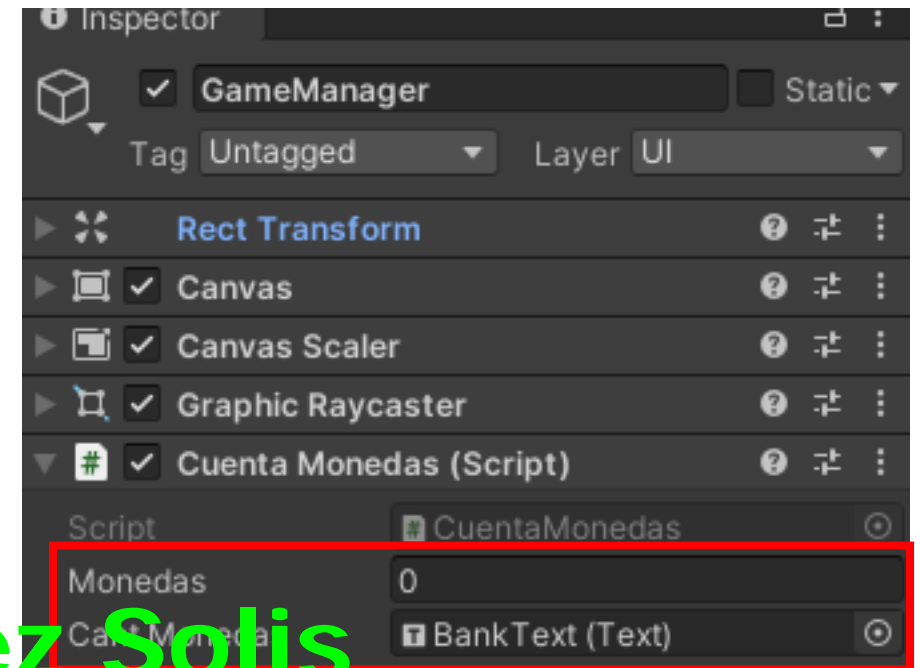
    //acceso inmediato desde otras clases
    public static CuentaMonedas instance;

    1 referencia
    public void Money(float monedasRecogidas)
    {
        monedas += monedasRecogidas;
        bankText.text = "x " + monedas.ToString();
    }

    Mensaje de Unity | 0 referencias
    private void Awake()
    {
        //singleton
        if (instance == null)
        {
            instance = this;
        }
    }
}
```

**Pedro Yuri Marquez Solis**

Asignar cuenta monedas al  
GameManager





# Reusar las monedas

- Convertir en Prefab.
- Agregar un GameObject, renombrar a Items.
- Insertar los prefabs de monedas dentro de Items
- También poción debería estar dentro de Items
- Los ítems deben tener tag: ítem y Layer: Item

**Pedro Yuri Marquez Solis**

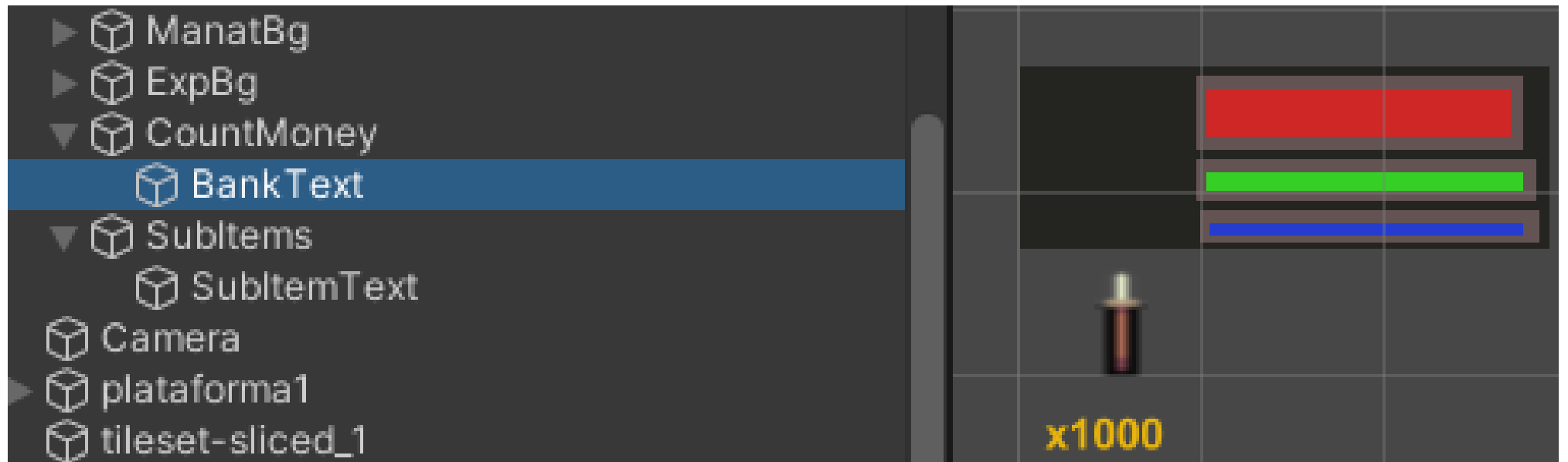
# Item de armas

- Insertar dentro del proyecto el Item Lanza, especificar tag: Item, Layer: Item, agrupar dentro del GameObject Items.
- Renombrar como ***LanzaRecolectable***
- Sorting Layer: Player
- Agregar un CircleCollider 2d con trigger activado.



**Pedro Yuri Marquez Solis**

# Agregar representación de Armas en el gameManager



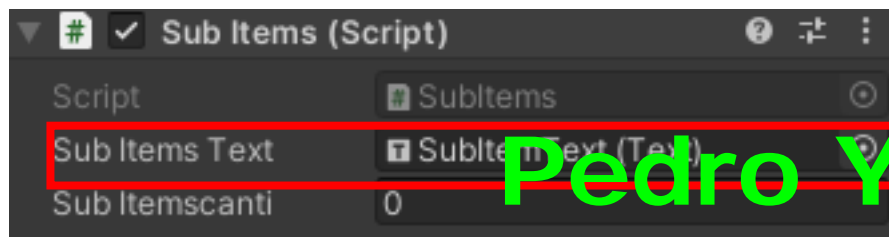
**Pedro Yuri Marquez Solis**



# Script subItems

Asignar SubItems al  
gameManager

Asignar el SubItemText



```
using UnityEngine.UI;

public class SubItems : MonoBehaviour
{
    public Text subItemsText;
    public float subItemsScanti;

    public static SubItems instance;

    public void SubItem(int subItemcanti) {
        subItemsScanti += subItemcanti;
        subItemsText.text = "x " + subItemsScanti.ToString();
    }

    public void Awake()
    {
        if (instance == null)
        {
            instance = this;
        }
    }

    public void Start()
    {
        subItemsText.text = "x " + subItemsScanti.ToString();
    }
}
```

Pedro Yuri Marquez Solis

# Script Coleccionables.cs

```
public class Coleccionables : MonoBehaviour
{
    public int lanzasBrindar;
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Player"))
        {
            SubItems.instance.SubItem(lanzasBrindar);
            Destroy(gameObject);
        }
    }
}
```

- Agregar Coleccionables.cs al GameObject ***LanzaRecolectable***

**Pedro Yuri Marquez Solis**

# Disparo de proyectiles - player

- Agregar el grafico del proyectil(Lanza) a la escena.
- Configurar Tag: Weapon, Layer: Player, Sorting Layer: Player.
- Agregar un Polygon Collider2d, activar Istrigger.
- Agregar un Rigidbody2d, establecer Gravity Scale: 0
- Guardar como Prefab Lanza

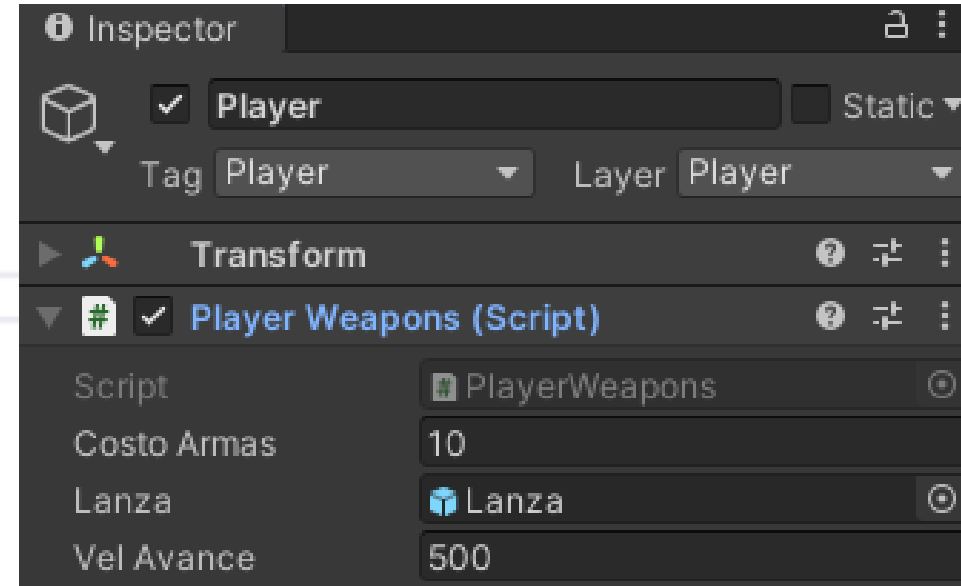
Assets > Prefab > armas

 Lanza

**Pedro Yuri Marquez Solis**

# Script PlayerWeapons

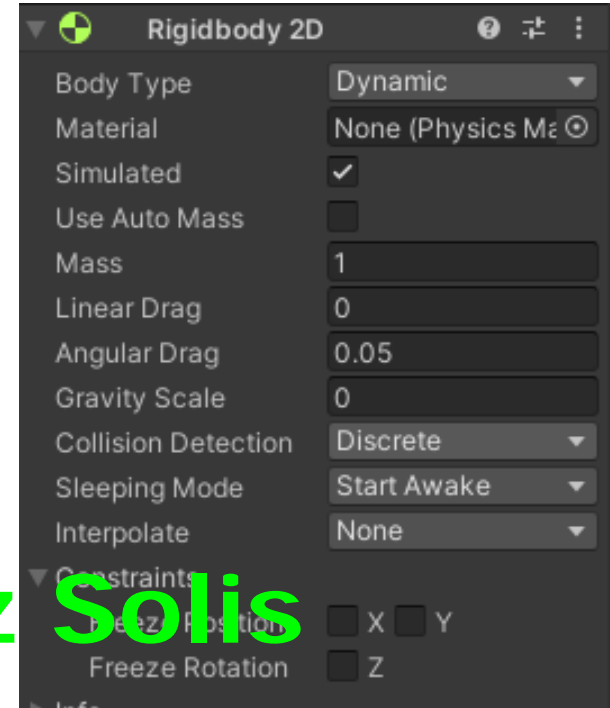
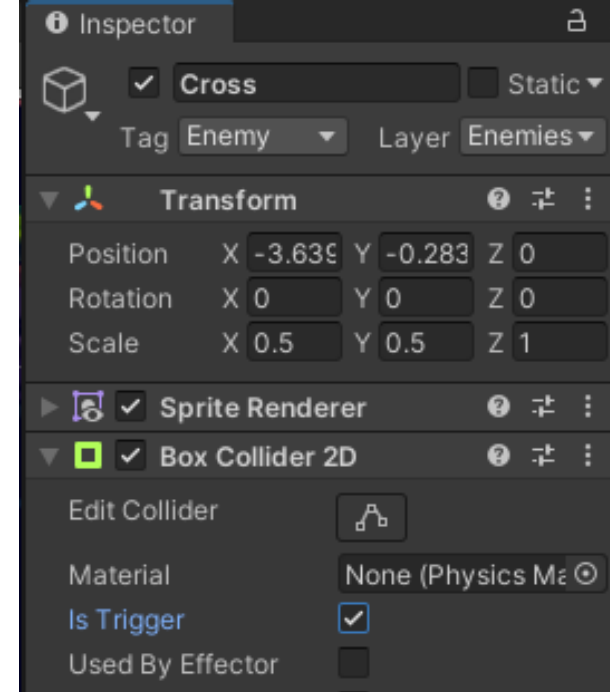
```
public class PlayerWeapons : MonoBehaviour
{
    public int CostoArmas;
    public GameObject lanza;
    public float velAvance;
    void Update()
    {
        UsarArma();
    }
    public void UsarArma() {
        if (Input.GetButtonDown("Fire2") && CostoArmas <= SubItems.instance.subItemscant)
        {
            SubItems.instance.SubItem(-CostoArmas);
            GameObject subItem = Instantiate(lanza, transform.position, Quaternion.Euler(0,0,-90));
            if (transform.localScale.x < 0) {
                subItem.GetComponent<Rigidbody2D>().AddForce(new Vector2(-velAvance, 0f), ForceMode2D.Force);
                subItem.transform.localScale = new Vector2(-1,-1);
            }
            else
                subItem.GetComponent<Rigidbody2D>().AddForce(new Vector2(velAvance, 0f), ForceMode2D.Force);
        }
    }
}
```



**Pedro Yuri Marquez Solis**

# Disparo - Enemigo

- Configurar Tag: Projectile, Layer: Enemies, Sorting Layer: Player.
- Agregar un BoxCollider2d, activar IsTrigger.
- Agregar un Rigidbody2d, establecer Gravity Scale: 0
- Guardar como Prefab Cross



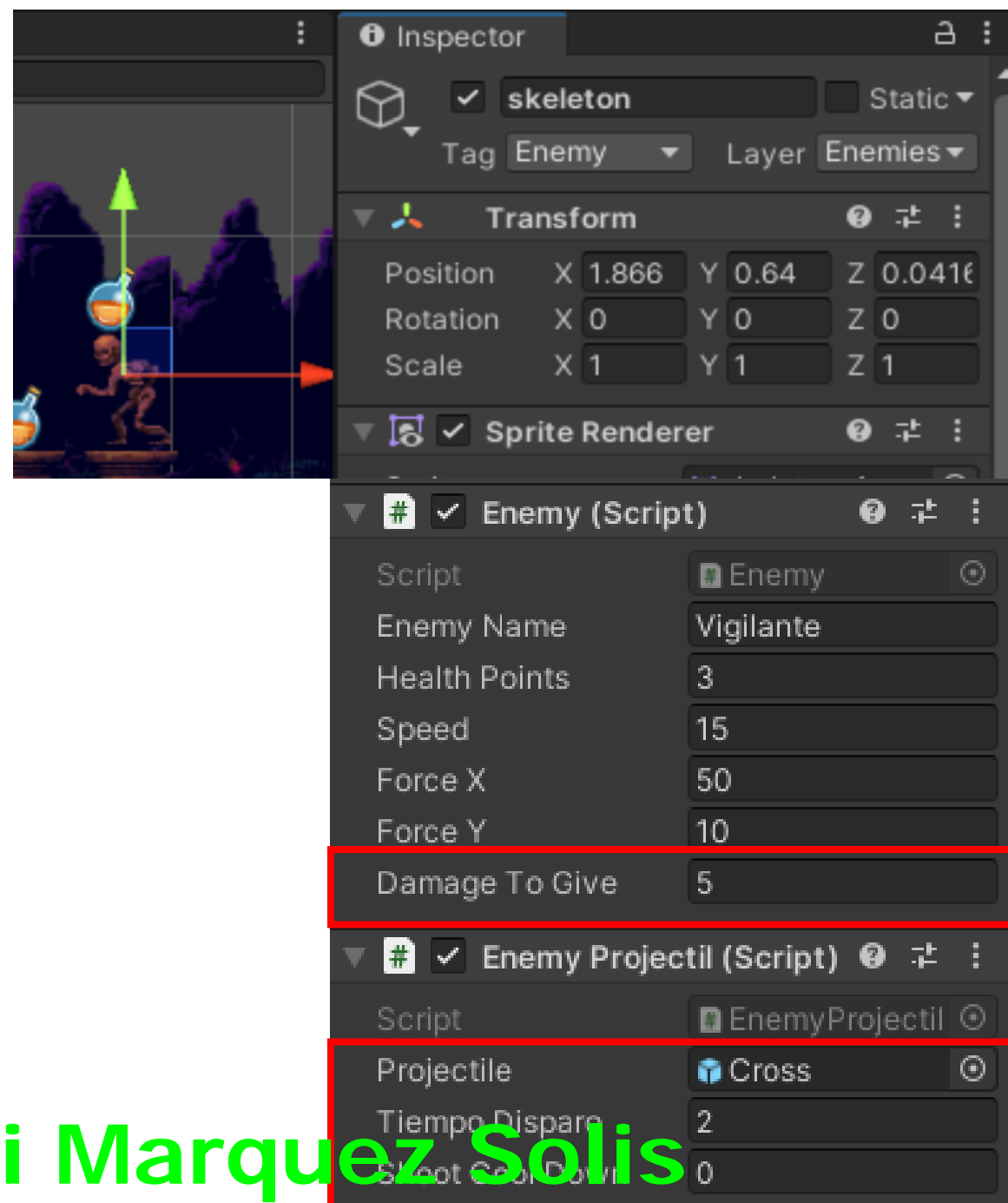
Pedro Yuri Marquez Solis

# Script:

```
public class EnemyProjectil : MonoBehaviour
{
    public GameObject projectile;
    public float tiempoDisparo, shootCoolDown;
    void Start()
    {
        shootCoolDown = tiempoDisparo;
    }
    void Update()
    {
        shootCoolDown -= Time.deltaTime;
        if (shootCoolDown<0)
        {
            GameObject cross= Instantiate(projectile, transform.position, Quaternion.identity);
            if (transform.localScale.x < 0)
            {
                cross.GetComponent<Rigidbody2D>().AddForce(new Vector2(500f, 0f), ForceMode2D.Force);
            }
            else
            {
                cross.GetComponent<Rigidbody2D>().AddForce(new Vector2(-500f, 0f), ForceMode2D.Force);
            }
            shootCoolDown = tiempoDisparo;
        }
    }
}
```

**Pedro Yuri Marquez Solis**

# Configurar Skeleton



Pedro Yuri Marquez Solis