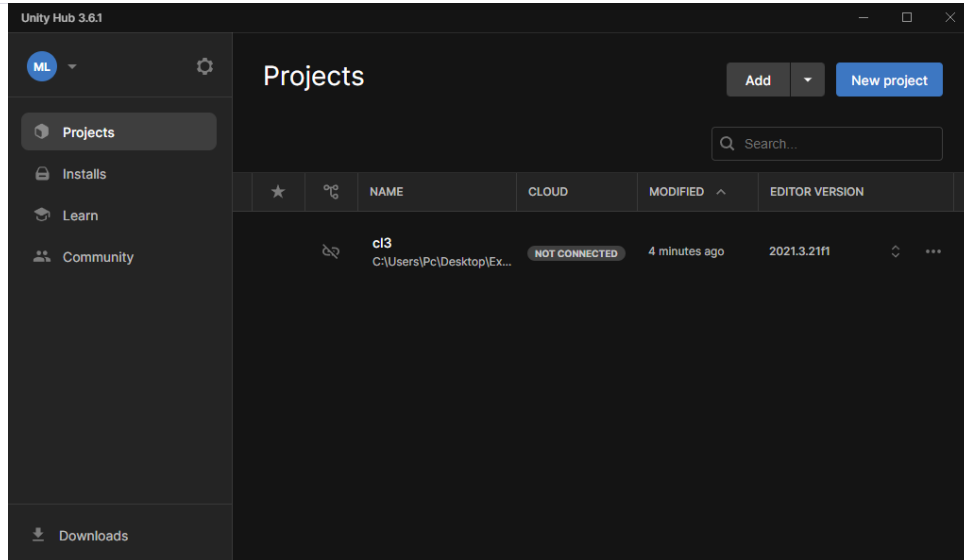


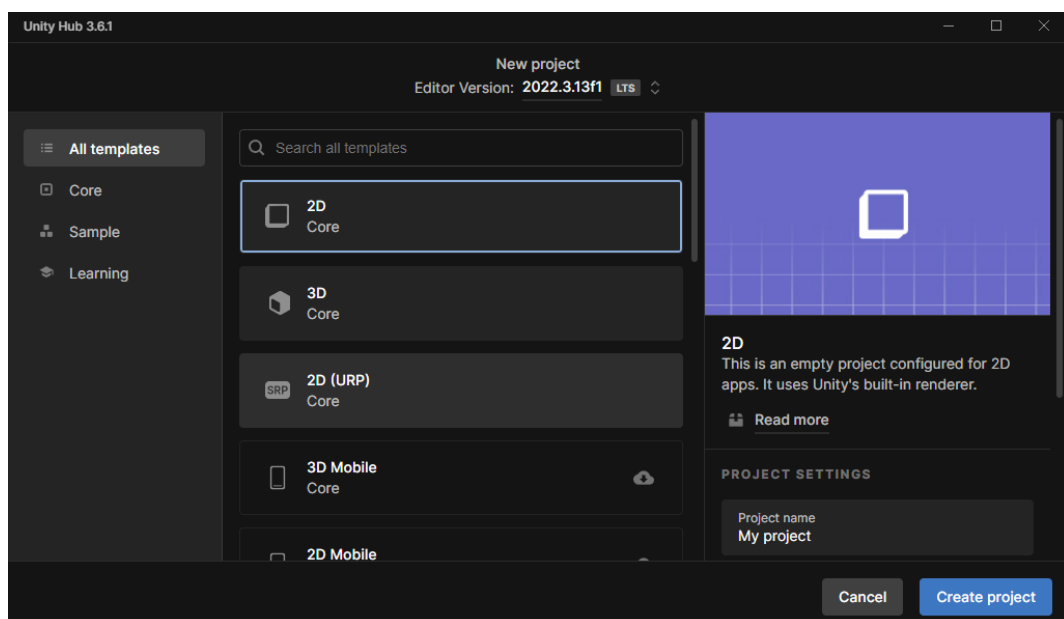
1. Instalación de Unity:

- Descargamos e instalamos Unity Hub desde el sitio web oficial de Unity.
- Abre Unity Hub y usa la pestaña "Installs" para instalar una versión de Unity compatible. (Version 3.6.1)



2. Creación de un Nuevo Proyecto:

- Abrimos Unity Hub y selecciona "New" para crear un nuevo proyecto.
- Elige la plantilla de 2D según el juego tilin.

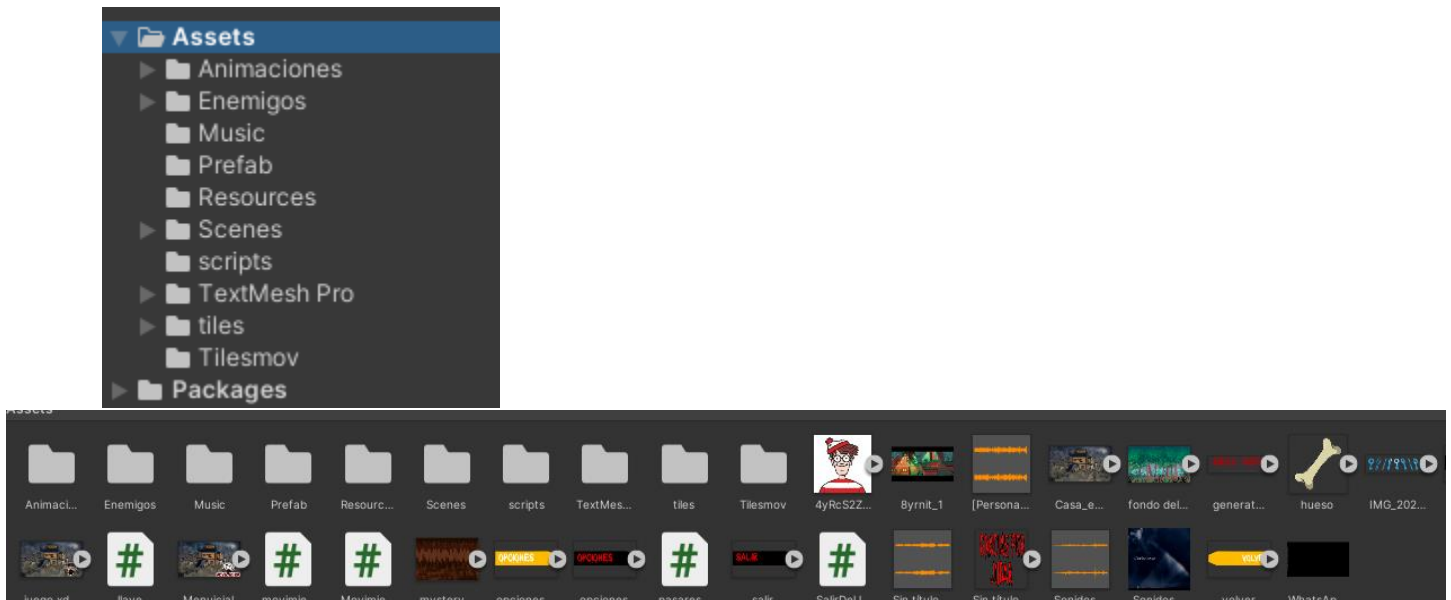


3. Organización de nuestro proyecto

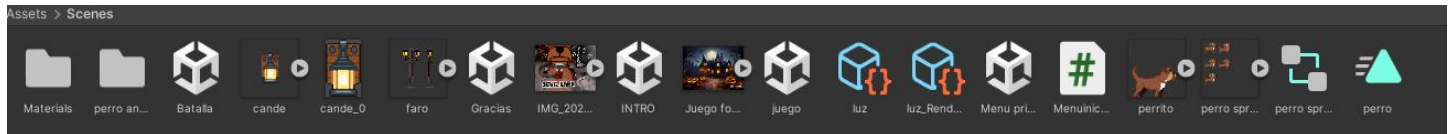
- Estructura tu proyecto en carpetas. Por ejemplo:
 - **Assets/** para nuestros recursos
 - **Scenes/** para las escenas que teníamos dentro del juego de igual forma almacenamos todas las imágenes y previsualización que tenemos.
 - **Scripts/** para los scripts de C#.

Assets	5/12/2023 19:05	Carpeta de archivos	
Library	5/12/2023 19:04	Carpeta de archivos	
Logs	5/12/2023 18:59	Carpeta de archivos	
obj	7/7/2023 16:08	Carpeta de archivos	
Packages	7/7/2023 16:08	Carpeta de archivos	
ProjectSettings	5/12/2023 19:05	Carpeta de archivos	
Temp	5/12/2023 19:04	Carpeta de archivos	
UserSettings	2/12/2023 02:12	Carpeta de archivos	
.vsconfig	16/6/2023 16:48	Archivo VSCONFIG	1 KB
Assembly-CSharp	20/6/2023 14:00	Archivo de origen ...	66 KB
cl3	16/6/2023 16:48	Microsoft Visual S...	1 KB

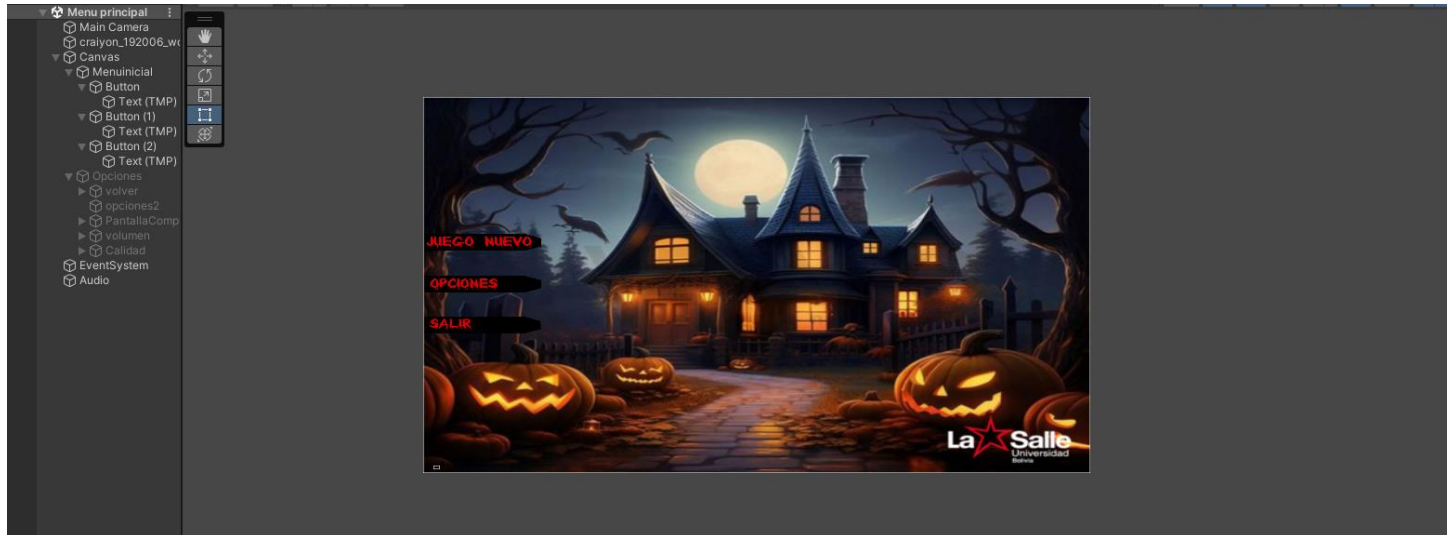
Assets.



Scenes.



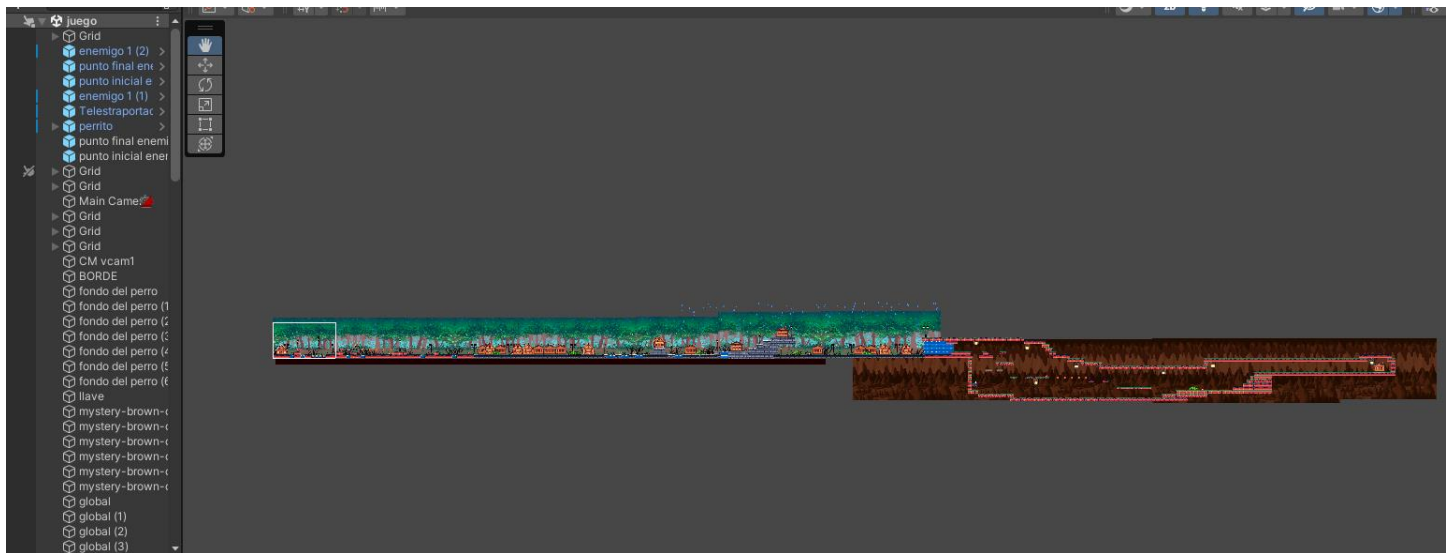
Menu Principal



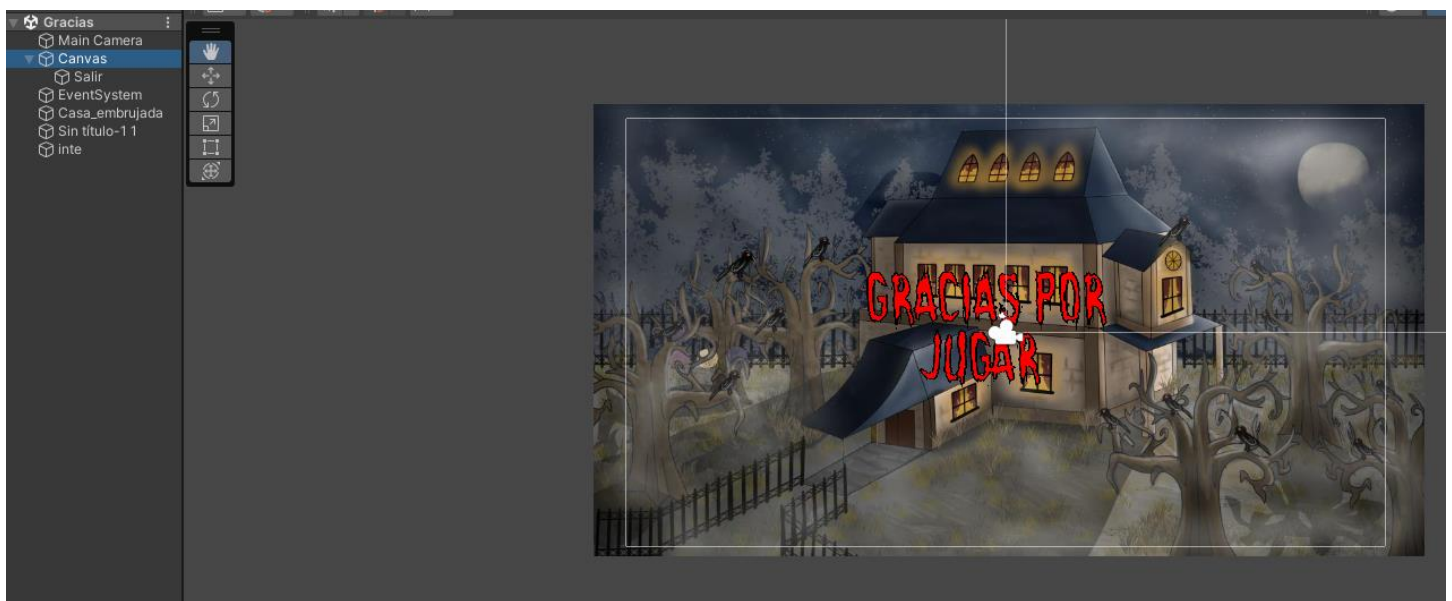
INTRO



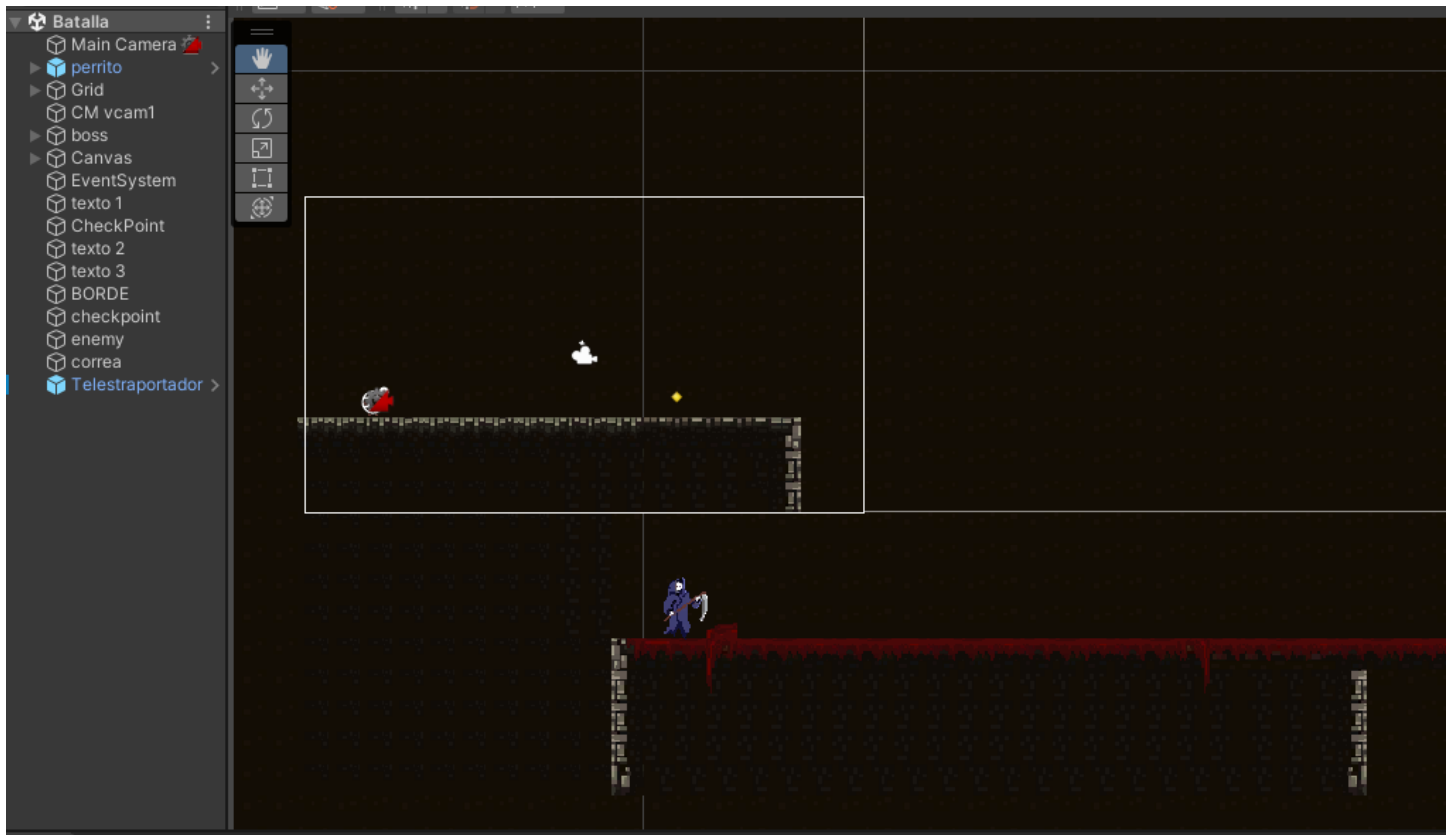
JUEGO



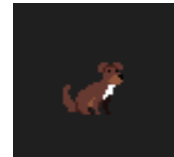
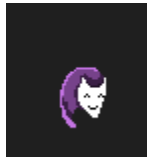
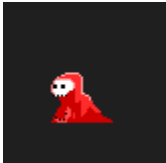
GRACIAS



BATALLA



Nuestros personajes



SCRIPTS

Animacion inicial Jefe

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class AnimacionInicialJefe : MonoBehaviour
{
    public Animator animator; // Referencia al componente Animator del objeto que contiene la animacion

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Player"))
        {
            // Comprobar si el objeto que colisiona tiene la etiqueta "Player" (puedes ajustar la etiqueta segun tu juego)

            // Activar la animacion
            bool sube = true;
        }
    }
}
```

Boss

```
using UnityEngine;
using UnityEngine.Tilemaps;

public class Boss : MonoBehaviour
{
    public Transform playerTransform; // Referencia al transform del jugador
    public float moveSpeed = 3f; // Velocidad de movimiento del enemigo
    public float jumpForce = 5f; // Fuerza de salto del enemigo
    public float jumpInterval = 2f; // Intervalo de tiempo entre saltos del enemigo
    public Tilemap groundTilemap; // Referencia al Tilemap del suelo
    private Rigidbody2D rb;
    private bool isJumping = false;
    private bool isGrounded = false;

    private void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    private void Update()
    {
        // Mover al enemigo horizontalmente hacia el jugador
        float moveDirection = (playerTransform.position - transform.position).normalized.x;
        rb.velocity = new Vector2(moveDirection * moveSpeed, rb.velocity.y);

        // Saltar si está en el suelo y ha pasado el intervalo de tiempo
        if (isGrounded && !isJumping && Time.timeSinceLevelLoad >= jumpInterval)
        {
            rb.AddForce(new Vector2(0f, jumpForce), ForceMode2D.Impulse);
            isJumping = true;
            Invoke(nameof(ResetJump), jumpInterval);
        }
    }

    private void OnCollisionEnter2D(Collision2D collision)
    {
        // Comprobar si ha colisionado con el Tilemap del suelo
        if (collision.gameObject.CompareTag("Suelo"))
        {
            Vector3 hitPosition = Vector3.zero;
            foreach (ContactPoint2D hit in collision.contacts)
            {
                hitPosition.x = hit.point.x - 0.01f * hit.normal.x; // Ajustar posición para evitar problemas de detección
                hitPosition.y = hit.point.y - 0.01f * hit.normal.y;
                Vector3Int cellPosition = groundTilemap.WorldToCell(hitPosition);
                if (groundTilemap.HasTile(cellPosition))
                {
                    isGrounded = true;
                    isJumping = false;
                    break;
                }
            }
        }
    }

    private void OnCollisionExit2D(Collision2D collision)
    {
        // Comprobar si ha dejado de colisionar con el Tilemap del suelo
        if (collision.gameObject.CompareTag("Suelo"))
        {
            isGrounded = false;
        }
    }

    private void ResetJump()
    {
        isJumping = false;
    }
}
```

Conversacion

```
using UnityEngine;

public class Conversacion : MonoBehaviour
{
    public GameObject textObject; // Objeto que contiene el componente TextMeshPro o TextMesh
    public string textToShow = "♦Hola, jugador!"; // Texto que se mostrar en el juego
    public float textDuration = 3f; // Duración en segundos que se mostrar el texto

    private bool hasTriggered = false;

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Player") && !hasTriggered)
        {
            // Comprobar si el objeto que colisiona tiene la etiqueta "Player" (ajstala segn tus necesidades)

            // Mostrar el texto
            ShowText();
            Invoke(nameof(HideText), textDuration);
        }
    }

    private void ShowText()
    {
        textObject.SetActive(true);

        // Asignar el texto que se mostrar ♦
        // Asegurate de que el objeto que contiene el texto tenga un componente TextMeshPro o TextMesh
        if (textObject.TryGetComponent(out TMPPro.TextMeshProUGUI textMeshPro))
        {
            textMeshPro.text = textToShow;
        }
        else if (textObject.TryGetComponent(out UnityEngine.UI.Text textMesh))
        {
            textMesh.text = textToShow;
        }

        hasTriggered = true;
    }

    private void HideText()
    {
        textObject.SetActive(false);
    }
}
```

Enemigo

```
using System.Collections;
using UnityEngine;
using UnityEngine.SceneManagement;

public class Enemy : MonoBehaviour
{
    public GameObject player;
    public Transform respawnPoint;
    public float transitionTime = 1f; // Duración de la transición

    private void OnCollisionEnter2D(Collision2D collision)
    {
        if (collision.gameObject.CompareTag("Player"))
        {
            // DisableCollision(); // Deshabilita la colisión del enemigo

            StartCoroutine(TransitionAndTeleportPlayer()); // Inicia la rutina para la transición y teletransportación
        }
    }

    private IEnumerator TransitionAndTeleportPlayer()
    {
        var transitionObject = new GameObject("TransitionObject"); // Crea un objeto para la transición
        var transitionSpriteRenderer = transitionObject.AddComponent<SpriteRenderer>(); // Añade un SpriteRenderer al objeto

        // Configura el SpriteRenderer con un sprite negro y ajusta su tamaño y posición para que cubra toda la pantalla
        transitionSpriteRenderer.sprite = Sprite.Create(Texture2D.blackTexture, new Rect(0, 0, 1, 1), Vector2.one * 0.5f);
        transitionSpriteRenderer.transform.position = new Vector3(0f, 0f, 1f); // Asegura que esté en frente de todos los demás objetos
        transitionSpriteRenderer.transform.localScale = new Vector3(Screen.width, Screen.height, 1f);

        yield return new WaitForSeconds(transitionTime); // Espera el tiempo de transición

        player.transform.position = respawnPoint.position; // Realiza la teletransportación del jugador

        Destroy(transitionObject); // Destruye el objeto de transición

        EnableCollision(); // Vuelve a habilitar la colisión del enemigo
    }

    public void DisableCollision()
    {
        GetComponent<Collider2D>().enabled = false;
    }

    public void EnableCollision()
    {
        GetComponent<Collider2D>().enabled = true;
    }
}
```


Escondite

```
using UnityEngine;

public class Escondite : MonoBehaviour
{
    public Enemy EnemyScript;

    private void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Player"))
        {
            // Desactiva la colisión del enemigo
            EnemyScript.DisableCollision();
        }
    }

    private void OnTriggerExit2D(Collider2D other)
    {
        if (other.CompareTag("Player"))
        {
            // Activa la colisión del enemigo
            EnemyScript.EnableCollision();
        }
    }
}
```

Menu principal

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class MenuPrincipal : MonoBehaviour
{
    public void juego()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }

    public void Salir()
    {
        Debug.Log("salir...");
        Application.Quit();
    }
}
```

Movimiento enemigo

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MovimientoEnemigo : MonoBehaviour
{
    public Transform puntoInicial;
    public Transform puntoFinal;
    public float velocidad = 2f;

    private Transform objetivo;
    private bool moverDerecha = true;

    private void Start()
    {
        objetivo = puntoFinal;
    }

    private void Update()
    {
        // Mueve el enemigo hacia el objetivo actual
        transform.position = Vector2.MoveTowards(transform.position, objetivo.position, velocidad * Time.deltaTime);

        // Si el enemigo llega al objetivo, cambia de dirección
        if (Vector2.Distance(transform.position, objetivo.position) < 0.1f)
        {
            CambiarDireccion();
        }
    }

    private void CambiarDireccion()
    {
        // Cambia el objetivo y la dirección de movimiento
        if (objetivo == puntoFinal)
        {
            objetivo = puntoInicial;
            moverDerecha = false;
        }
        else
        {
            objetivo = puntoFinal;
            moverDerecha = true;
        }

        // Voltea el sprite del enemigo en función de la dirección
        transform.localScale = new Vector3(moverDerecha ? 1f : -1f, 1f, 1f);
    }
}
```

Muerte Reiniciar Escena

```
using UnityEngine;
using UnityEngine.SceneManagement;

public class MuerteReiniciarEscena : MonoBehaviour
{
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Player"))
        {
            // Comprobar si el objeto que colisiona tiene la etiqueta "Player" (ajústala según tus necesidades)

            // Reiniciar la escena
            SceneManager.LoadScene(SceneManager.GetActiveScene().name);
        }
    }
}
```

Siguiente escena

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class NextSceneTrigger : MonoBehaviour
{
    public string escena; // Nombre de la escena a cargar

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.CompareTag("Player"))
        {
            SceneManager.LoadScene(escena); // Carga la siguiente escena
        }
    }
}
```

Opciones

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Audio;

public class Opciones : MonoBehaviour
{
    [SerializeField] private AudioMixer audioMixer;

    public void PantallaCompleta(bool PantallaCompleta)
    {
        Screen.fullScreen = PantallaCompleta;
    }

    public void CambiarVolumen(float volumen)
    {
        audioMixer.SetFloat("Volumen", volumen);
    }

    public void CambiarCalidad(int index)
    {
        QualitySettings.SetQualityLevel(index);
    }
}
```

Pausa

```
Users > PC > Desktop > Exa > Iphone > CL3ARREGLO2 > c
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PauseMenu : MonoBehaviour
{
    public GameObject PausaPanel;

    public void Pause()
    {
        PausaPanel.SetActive(true);
        Time.timeScale = 0;
    }

    public void Continue()
    {
        PausaPanel.SetActive(false);
        Time.timeScale = 1;
    }
}
```

Teletransportador

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Teletransportador : MonoBehaviour
{
    public Transform destino; // La posición de destino a la que se llevará al personaje

    private void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Player"))
        {
            // Teletransporta al personaje a la posición de destino
            other.transform.position = destino.position;
        }
    }
}
```