

Trabajo de animación digital.

Entrega unity

ALEJANDRO GÓMEZ CURTO

70914886k

[Main Menu](#)

[Transición Ganar](#)

[Funcion OnColission](#)

[Escena ganar](#)

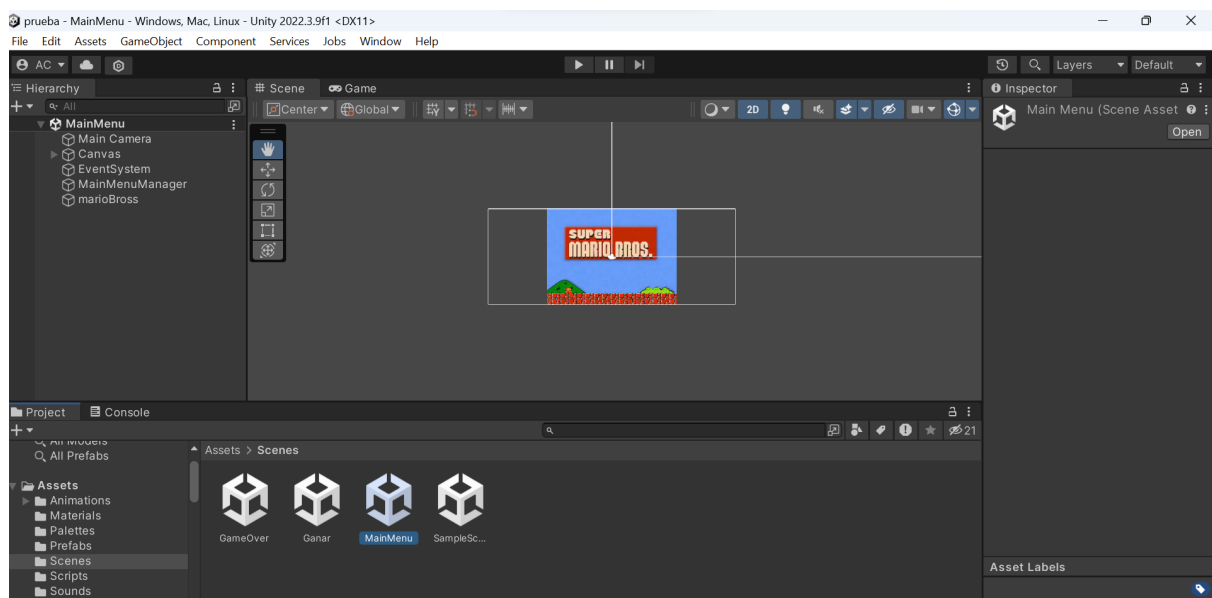
[Escena gameOver](#)

[Perder Vida](#)

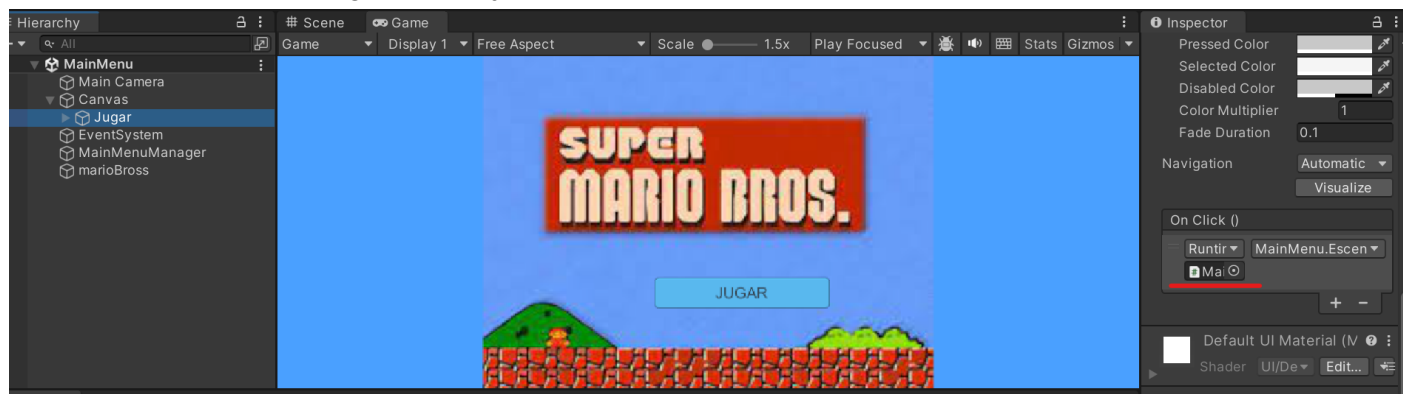
[Main Menu Script](#)

Main Menu

En esta imagen podemos ver la escena MainMenu la cual es la primera que debemos de abrir nada mas ejecutar el proyecto y en ella tenemos una imagen de fondo y un botón jugar el cual si lo pulsamos nos llevará a SampleScene la cual contiene nuestro mapa.

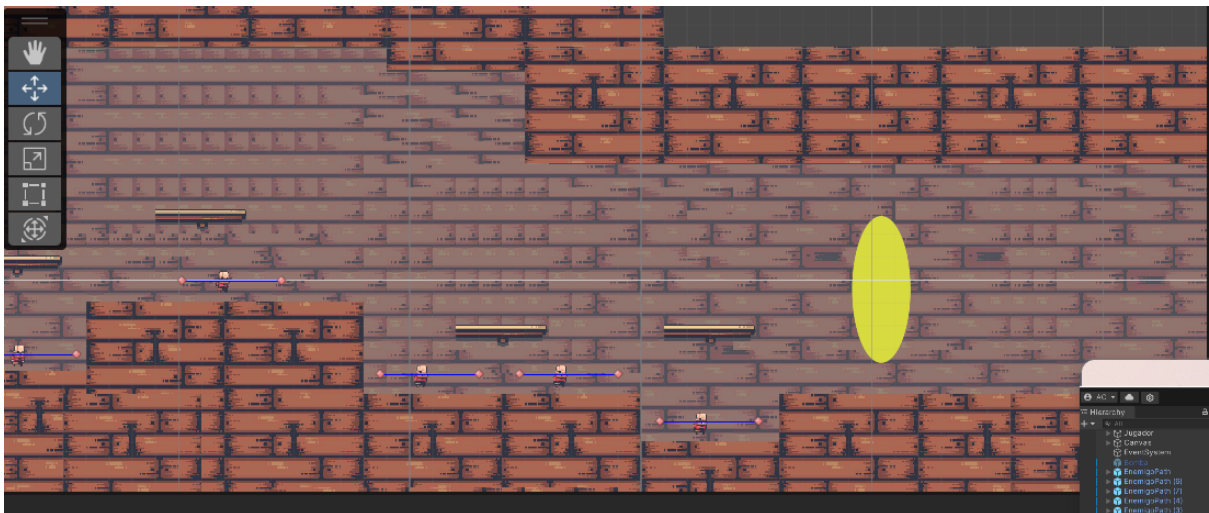


En el botón jugar que hemos añadido le hemos asociado un emptyObject que es MainMenuManager el cual tiene un Script asociado que es MainMenu.cs. Este script que he creado yo tiene una función que es EscenaJuego la cual la llamamos utilizando el evento onClick que sale en la imagen de abajo.

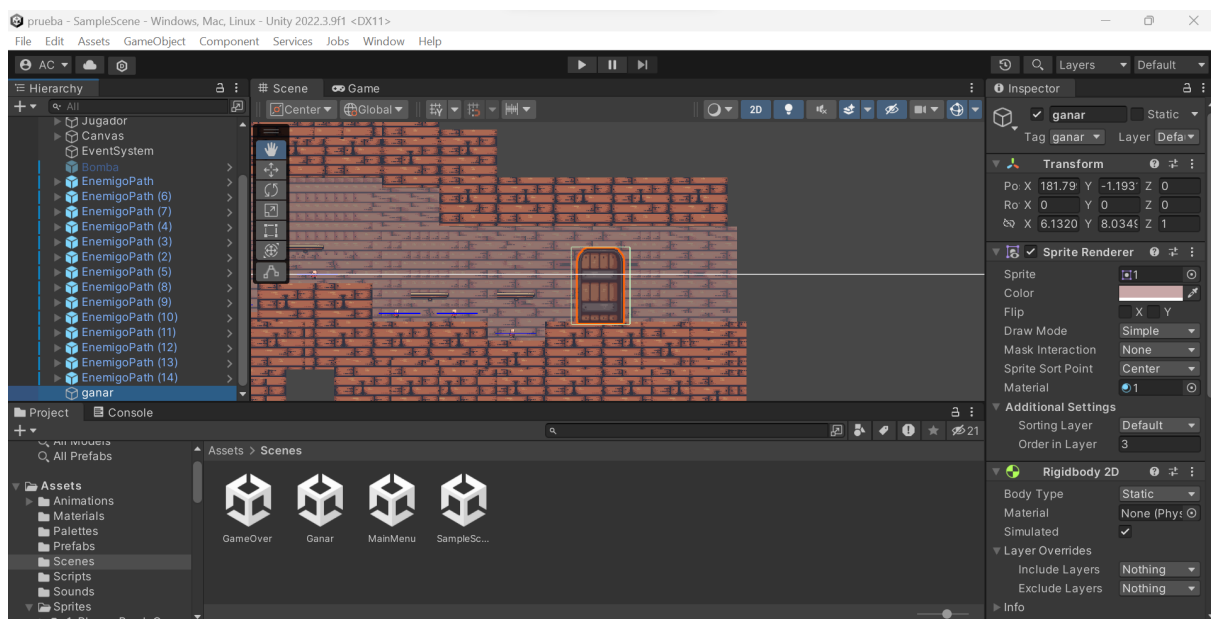


El círculo amarillo es un objeto que tiene un rigidbody y un colider que hace que cuando el jugador colisione con el objeto se cambie a la escena ganar y se muestre la interfaz de la victoria lo que significaria que hemos ganado al videojuego.

Transición Ganar

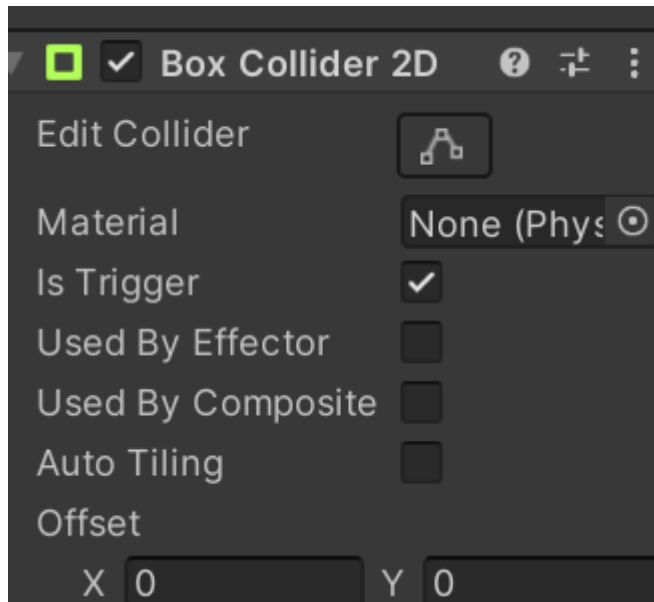


En la siguiente imagen muestro la evolución del círculo amarillo al que le hemos dado una forma de puerta. añadiendo a su sprite render un sprite de puerta que hemos cogido de nuestros assets



Funcion OnCollision

Este es el código que se ejecuta cuando el jugador choca con la puerta ya que tenemos habilitada la opción de Is Trigger



```
void OnTriggerEnter2D(Collider2D other)
{
    // Verifica si el jugador choca con un objeto llamado "ganar"
    if (other.CompareTag("ganar"))
    {
        Debug.Log("Ganar");
        GanarPartida();
    }
}
```

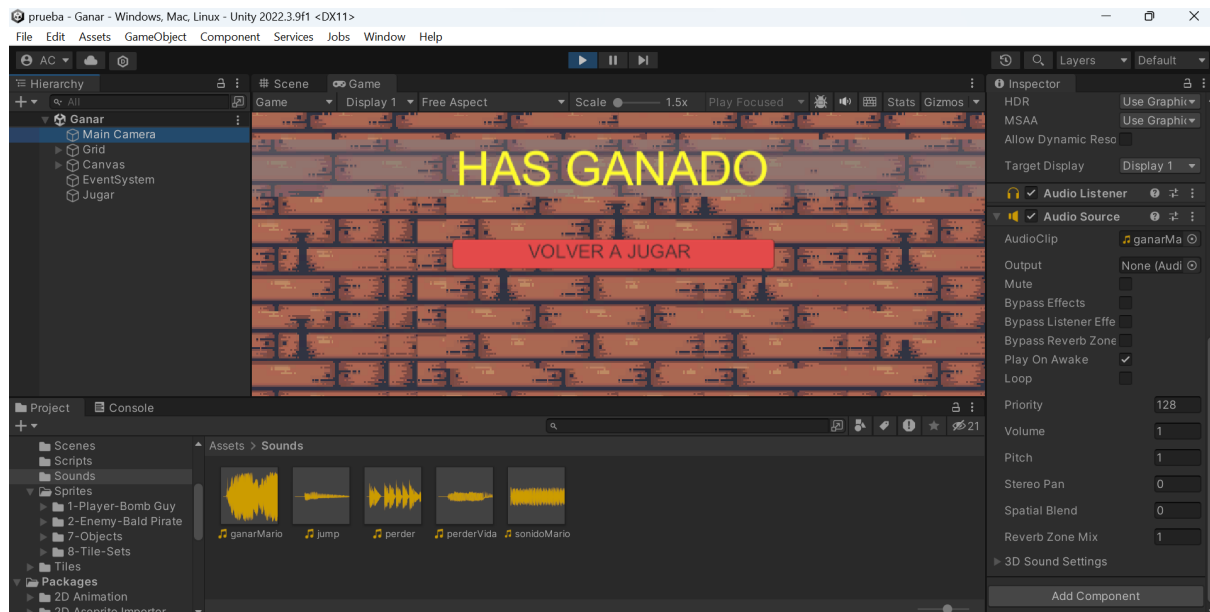
Escena ganar

Aquí tenemos la escena Ganar la cual informa al usuario de que la partida ha acabado y si desearse volver a jugar tiene un botón para poder iniciar otra partida. He usado el tile Palette para crear el escenario y he añadido un texto que informa de que el usuario ha ganado



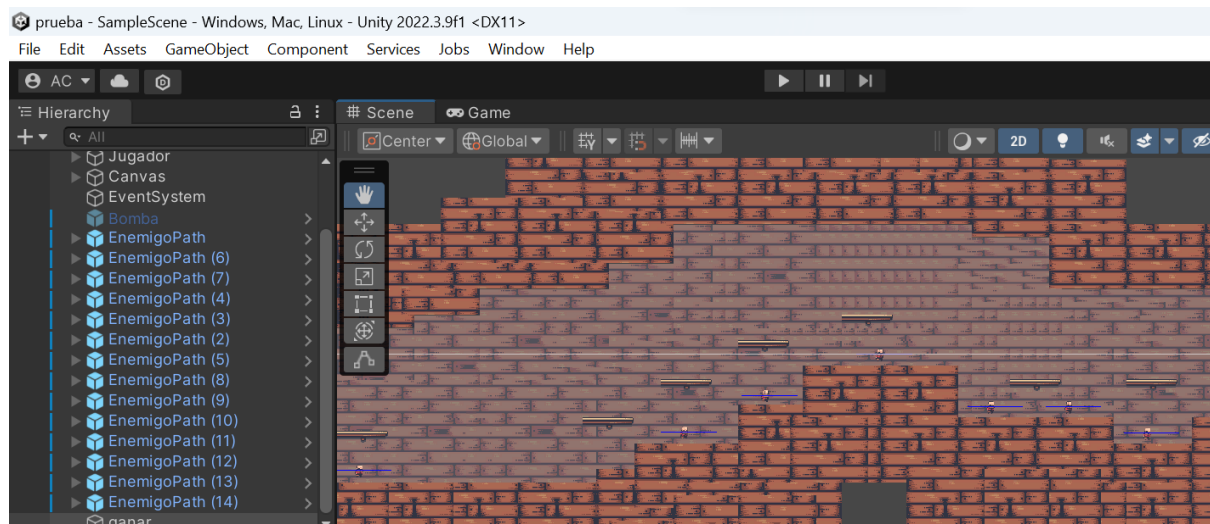
Sonido Ganar

Cuando llegamos a esta escena he añadido un sonido que es el sonido de cuando se gana la partida y que está configurado con play On Awake lo que significa que cada vez que se ejecute la escena va a sonar el sonido nada más se cargue la escena “ganar”



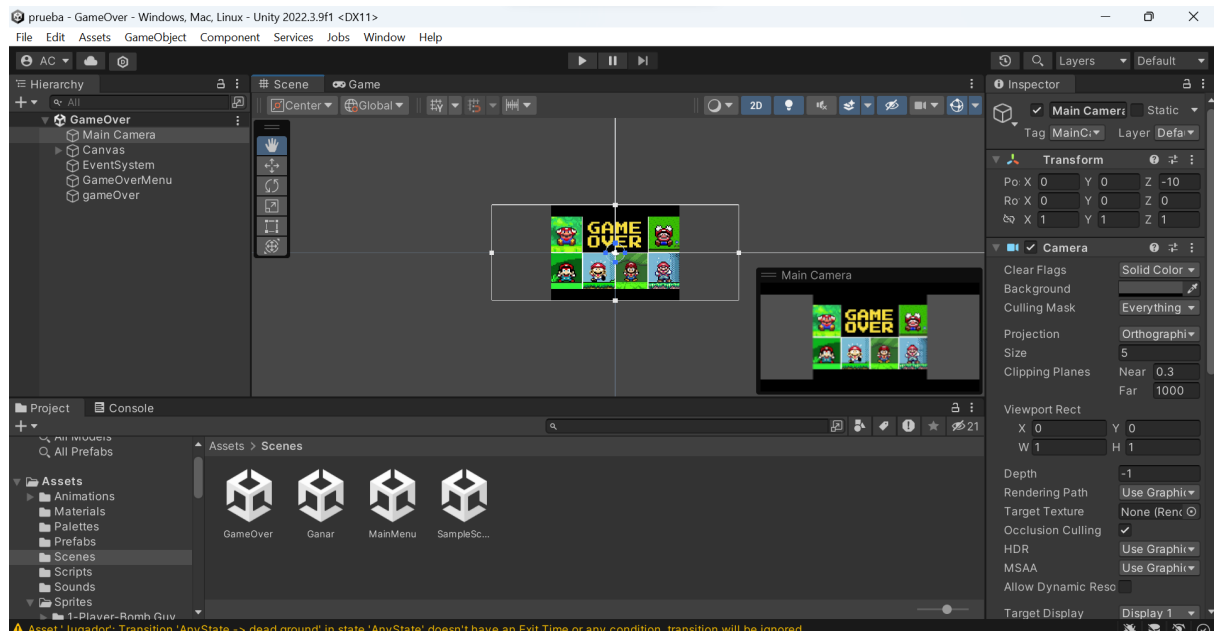
Diseño nivel

En esta imagen se puede observar los distintos enemigos que hemos ido añadiendo para completar el nivel a los cuales será necesario esquivar y/o matar para poder llegar al final. Lo que hemos modificado de los enemigos es el tamaño del path (camino) dependiendo de lo largo que nos interesaba que fuese para que no diese errores en nuestro videojuego el enemigo al colisionar con algo.



Escena gameOver

La siguiente captura es de la escena GameOver la cual tiene una imagen de gameOver y también tiene un botón de volver a jugar por si el usuario después de perder desea seguir jugando. El botón tiene asociado un emptyObject que es GameOverMenu que funciona de la misma manera que he explicado anteriormente en la escena de MainMenu.



Perder Vida

Con este código de abajo. En la primera imagen que está en `jugador.cs` se ejecuta cuando el jugador desde el código de la segunda imagen que está en `enemigoPirata.cs` colisiona con el jugador por lo que se invoca a la función `perderVida` que a su vez invoca a la función `desactivarVida` que está en `HUD.cs` y hace que el array de vidas desactive una vida

```
1 referencia
public void perderVida()
{
    vidas -= 1;
    if (vidas == 0)
    {
        SceneManager.LoadScene("GameOver");
    }
    hud.desactivarVida(vidas);
    vida.Play();
}
```

```

private void OnCollisionEnter2D(Collision2D other)
{
    if (other.gameObject.CompareTag("Player"))
    {
        // Obtiene una referencia al componente Jugador
        Jugador jugador = other.gameObject.GetComponent<Jugador>();

        // Asegúrate de que el componente Jugador existe
        if (jugador != null)
        {
            if (!GetComponent<Animator>().GetBool("morir")) // compruebo si el enemigo esta muerto
            {
                // Llama a la función perderVida del jugador
                jugador.perderVida();
            }
        }
    }
}

```

Main Menu Script

En el script mainMenu.cs tengo esta función que se ejecuta cuando pulso cualquiera de los botones de mi interfaz para volver a mi escena principal. He tenido que importar la librería using UnityEngine.SceneManagement. Gracias a esta librería puedo cargar las escenas pasándole como argumento a la función LoadScene el nombre de la escena que quiera cargar, la cual previamente he tenido que hacer build para que no me de error.

```

}
0 referencias
public void EscenaJuego()
{
    SceneManager.LoadScene("SampleScene");
}

```