

ANGELBANE

PROGRAMACIÓN DE JUEGOS

Proyecto realizado por:

Alejandro Pérez Domínguez y Cristian González Borrero

ÍNDICE

- [1. Introducción](#)
- [2. Menú Principal](#)
- [3. Escena 1 \(Bosque\)](#)
 - [1. Entorno](#)
 - [1. Pinchos del suelo:](#)
 - [1. Agua venenosa:](#)
 - [2. Plataformas móviles](#)
 - [3. Puerta del castillo](#)
 - [2. Enemigos](#)
- [4. Escena 2 \(Interior del castillo\)](#)
- [5. Pantalla de récords](#)
- [6. Game Over](#)
- [7. Referencias](#)

1. *Introducción*

AngelBane es un juego de tipo plataforma en estilo 2D desarrollado en la versión 6000.0.40f1 de Unity en el cual, Wizard, nuestro personaje principal, deberá ir avanzando a través del mundo principal hasta dar con la guarida del malvado ángel y derrotarlo para liberar al pueblo, el boss que ha sembrado el caos en todo el reino con su poderío y sus espadas.

Para ello, Wizard deberá enfrentarse a los peligrosos esqueletos deambulantes, a las aguas venenosas, a un jefe final que representa un ángel corrupto, y a otros peligros establecidos.

El juego se divide en 5 escenas, el menú principal donde establecemos la configuración de la partida, 2 escenas donde transcurre la acción del juego, y 2 escenas para controlar el sistema de records.

2. Menú Principal

Nada más cargar el juego, en la pantalla de título, podemos ir al menú de configuración para elegir algunas configuraciones a nuestro gusto. Basta con pulsar el icono de engranaje que vemos en la esquina superior izquierda.

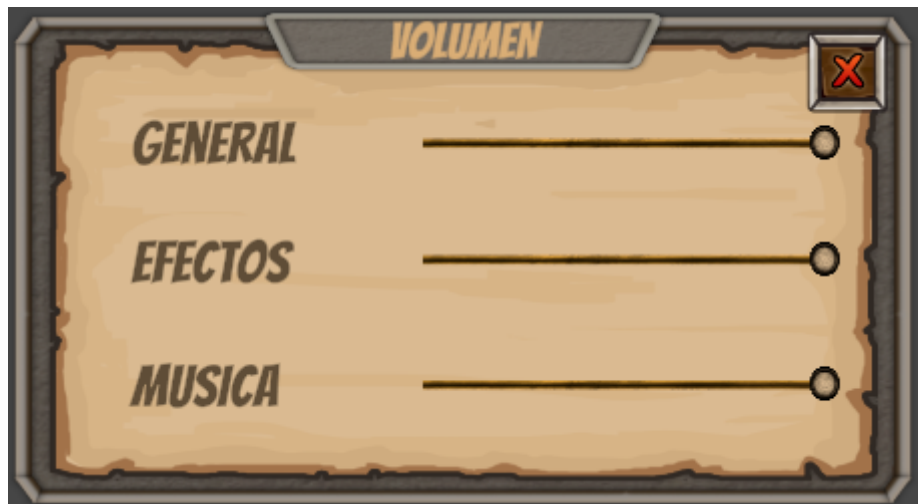


Se nos abrirá el siguiente menú:



En él, podremos:

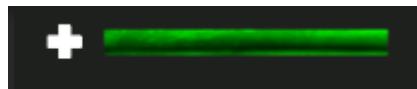
-Ajustar el volumen del juego: volumen general, la música y efectos de sonidos por separado



Esta configuración se guardará hasta que la cambiemos aunque salgamos del juego y entremos de nuevo.

-Cambiar de pantalla completa a modo ventana con el botón “modo ventana”. Al activarse, se modificará el texto del botón a “pantalla completa” y se podrá pulsar para realizar la acción contraria.

-Establecer la dificultad del juego: Wizard tendrá 400 puntos de vida en modo fácil, 200 puntos de vida en modo normal y 100 puntos de vida en modo difícil. La longitud de la barra de vida no se modifica, pero al ser la cantidad de daño fija, los golpes a Wizard harán mayor daño porcentual.



Esta configuración se guardará hasta que la cambiemos aunque salgamos del juego y entremos de nuevo.



-Echar un vistazo a los controles del juego. Dichos controles son:

A: Moverse hacia la izquierda.



D: Moverse hacia la derecha.



W: Saltar



C: Disparar



Espacio: Interactuar (con objetos interactuables).

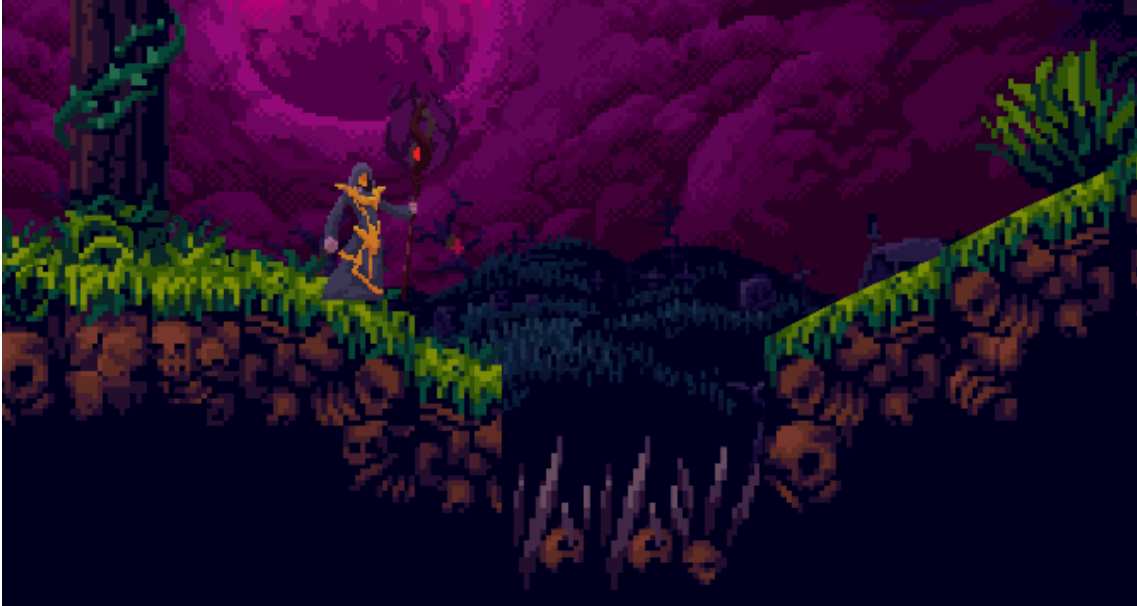


3. *Escena 1 (Bosque)*

1. Entorno

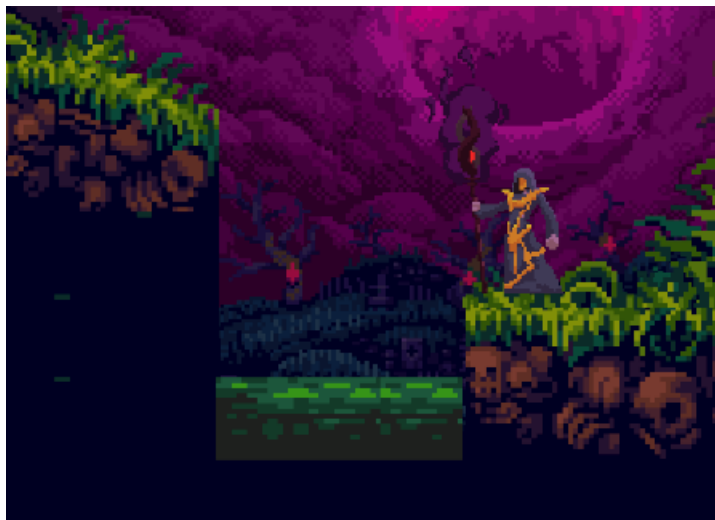
El bosque está cargado de peligros que nuestro protagonista deberá sortear para llegar hacia su objetivo, de lo contrario recibirá daño; y también de elementos con los que deberá interactuar para poder avanzar.

1. Pinchos del suelo:



Si el jugador cae en este elemento del escenario, recibirá daño y saltará automáticamente para evitar seguir recibiendo daño y poder reposicionarse.

2. Agua venenosa:



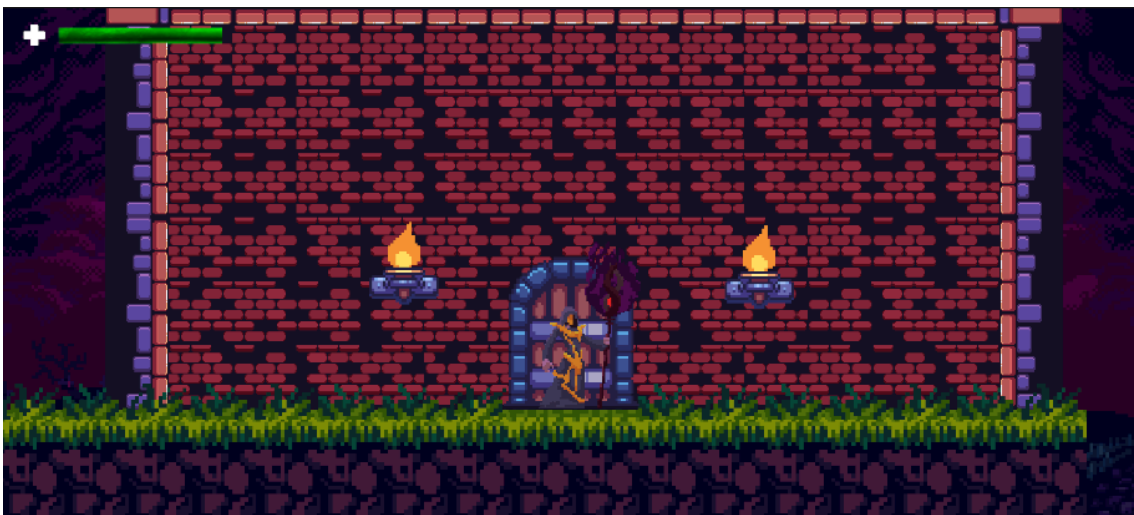
Si el jugador cae en este elemento, empezará a recibir daño lentamente, hasta que salte y escape de la superficie.

3. Plataformas móviles



Estas plataformas suben y bajan de forma automáticamente a diferente velocidad, moviendo al jugador con ellas.

4. Puerta del castillo



Este elemento contiene una puerta con la que el jugador puede interactuar pulsando el botón “espacio” para abrirla y luego pulsar el botón de nuevo para atravesarla y pasar a la siguiente escena.

2. Enemigos

En el mundo, los enemigos principales son **los esqueletos**, que permanecen alerta esparcidos por el mundo esperando el momento para atacar con sus espadas afiladas.



Aunque **no tienen un algoritmo específico de inteligencia artificial** implementado, los esqueletos **son lo suficientemente inteligentes** como para dirigirse hacia el personaje principal dada una determinada distancia, subir y bajar rampas, detenerse ante un precipicio o ante un peligro como pinchos o agua venenosa y atacar cuerpo a cuerpo.

Con un **prefab** se instanciarán todos los esqueletos que hay en el mapa. Dicho prefab tiene asignado un script (que se asigna automáticamente a cada instancia), para controlar los comportamientos de los esqueletos.

El siguiente método es el método Update, el cuál se encarga de actualizar en todo momento en qué situación se encuentra cada esqueleto.

```

void Update()
{
    if (Wizard == null) return;
    Vector3 direccion = Wizard.transform.position - transform.position;
    float distancia = direccion.magnitude;
    float diferenciaAltura = Mathf.Abs(Wizard.transform.position.y - transform.position.y);
    anim.SetFloat("Distance", distancia);

    if (direccion.x >= 0.0f) //Mirar hacia el mago
    {
        transform.localScale = new Vector3(3.0f, 3.0f, 3.0f);
    }
    else
    {
        transform.localScale = new Vector3(-3.0f, 3.0f, 3.0f);
    }

    if (distancia < 15.0f && HaySueloDelante()) //Sólo actúa si está dentro del rango de detección
    {
        //Ataque
        if (distancia < 1.8f && Time.time - tiempoUltimoAtaque > tiempoEntreAtaques && diferenciaAltura <= alturaPermitida)
        {
            anim.SetFloat("Speed", 0);
            anim.SetTrigger("Attack");
            tiempoUltimoAtaque = Time.time;

            if (!audioManager.SFXSource.isPlaying)
            {
                audioManager.reproducirEfecto(audioManager.espada);
            }
        }
        else if (diferenciaAltura <= alturaPermitida && distancia < 15.0f) /*PERSECUCIÓN*/ {
            //Solo se mueve si está a la misma altura que el mago
            if (Mathf.Abs(Wizard.transform.position.y - transform.position.y) <= alturaPermitida) {
                anim.SetFloat("Speed", speed);
            }
            else {
                anim.SetFloat("Speed", 0);
            }
            // Moverse hacia el mago
            direccion.y = 0;
            transform.position += direccion.normalized * speed * Time.deltaTime;
        }
        else if (diferenciaAltura > alturaPermitida && distancia < 15.0f) {
            direccion.y = 0;
            transform.position += direccion.normalized * speed * Time.deltaTime;
            anim.SetFloat("Speed", speed);
        }
        else {
            anim.SetFloat("Speed", 0);
        }
    }
    else {
        anim.SetFloat("Speed", 0); //Fuera de rango de detección
    }
}

```

Repasemos lo que puede hacer cada esqueleto:

-Dirigirse hacia el personaje principal:

De eso se encarga el bucle siguiente dentro del update:

```

} else if (diferenciaAltura <= alturaPermitida && distancia < 15.0f) /*PERSECUCIÓN*/ {
    //Solo se mueve si está a la misma altura que el mago
    if (Mathf.Abs(Wizard.transform.position.y - transform.position.y) <= alturaPermitida) {
        anim.SetFloat("Speed", speed);
    }
    else {
        anim.SetFloat("Speed", 0);
    }
    // Moverse hacia el mago
    direccion.y = 0;
    transform.position += direccion.normalized * speed * Time.deltaTime;
}

```

En el cuál, si la distancia entre nuestro personaje principal y la instancia de esqueleto es menor de 15 y no hay un vacío delante, este nos detectará y empezará a venir a por nosotros.

-Atacar cuerpo a cuerpo:

```
//Ataque
if (distancia < 1.8f && Time.time - tiempoUltimoAtaque > tiempoEntreAtaques && diferenciaAltura <= alturaPermitida)
{
    anim.SetFloat("Speed", 0);
    anim.SetTrigger("Attack");
    tiempoUltimoAtaque = Time.time;

    if (!audioManager.SFXSource.isPlaying)
    {
        audioManager.reproducirEfecto(audioManager.espadazo);
    }
}
```

Una vez esté a una distancia inferior a 1.8 de nosotros (y esté a nuestra misma altura), el esqueleto atacará. Se lanzará un trigger que hace que se produzca la animación de ataque asignada y que si nos da, nos reste vida. Esto también se encuentra dentro del Update, como decíamos.

También tenemos los métodos hacerDanio() y recibirDanio(), según el esqueleto nos esté atacando o le estemos atacando nosotros. Son ambos perfectamente configurables, pudiendo modificar el daño recibido o provocado.

```
public void HacerDanio()
{
    if (Wizard != null)
    {
        WizardMovement wizardScript = Wizard.GetComponent<WizardMovement>();
        if (wizardScript != null)
        {
            wizardScript.RecibirDanio(8);
        }
    }
}

1 reference
public void RecibirDanio(int cantidad)
{
    vida -= cantidad;
    if (vida <= 0)
    {
        Morir();
    }
    else
    {
        audioManager.reproducirEfecto(audioManager.esqueletoRecibirDanio);
    }
}
```

4. *Escena 2 (Interior del castillo)*

Al entrar en la escena se nos presenta un pasillo, a través del cual, si avanzamos, llegaremos a una sala. Al entrar en la sala, la cámara se aleja y sube de posición para reflejar mejor la batalla. Además, también suben unas vallas para evitar que el jugador pueda escapar de la sala hasta que derrote al jefe final.

A partir de aquí, todo lo que ocurre en esta escena está relacionado con la pelea de jefe (bossfight) del ángel corrupto o caído.



El ángel se mantiene inmóvil flotando en el aire en todo momento, y al esperar 2.5 segundos ejecutará un ataque en el que invoca espadas en el aire, formando un patrón aleatorio entre 3 patrones, y tras 0.5 segundos, caen hacia abajo, atravesando el suelo y haciendo daño al jugador si entran en contacto con él.

```
private IEnumerator BossFightSequence()
{
    corrutinaActiva = true;

    if (!esperaInicialCompletada)
    {
        yield return new WaitForSeconds(2.5f);
        esperaInicialCompletada = true;
    }

    int randomIndex = Random.Range(0, swordPrefabs.Length);
    GameObject swordPattern = Instantiate(swordPrefabs[randomIndex], new Vector3(24.6f, -1, -1), Quaternion.identity);

    yield return new WaitForSeconds(0.5f);

    anim.SetTrigger("Attack");

    Rigidbody2D[] swords = swordPattern.GetComponentInChildren<Rigidbody2D>();
    foreach (Rigidbody2D sword in swords)
    {
        sword.gravityScale = 2.5f;
    }

    yield return new WaitForSeconds(2f);

    Destroy(swordPattern);
    contador++;
    if (contador == 4)
    {
        plataformasActivas = true;
        contador = 0;
        StartCoroutine(MoverPlataformas());
    }

    corrutinaActiva = false;
}
```

Tras ejecutar este ataque un total de 4 veces, el ataque de las espadas se detendrá y aparecerán desde el suelo una serie de plataformas, que subirán lentamente para que nuestro protagonista pueda llegar a la misma altura del ángel y poder golpearle. Estas plataformas se mantendrán en el aire durante 6 segundos y luego volverán a su estado original.

```
private IEnumerator MoverPlataformas()
{
    plataformasActivas = true;

    float duracion = duracionMovimiento;
    float elapsed = 0f;

    // Subida
    while (elapsed < duracion)
    {
        float t = elapsed / duracion;
        plataformas.transform.position = Vector3.Lerp(plataformasAbajo, plataformasArriba, t);
        elapsed += Time.deltaTime;
        yield return null;
    }
    plataformas.transform.position = plataformasArriba;

    yield return new WaitForSeconds(6);

    // Bajada
    elapsed = 0f;
    while (elapsed < duracion)
    {
        float t = elapsed / duracion;
        plataformas.transform.position = Vector3.Lerp(plataformasArriba, plataformasAbajo, t);
        elapsed += Time.deltaTime;
        yield return null;
    }
    plataformas.transform.position = plataformasAbajo;

    plataformasActivas = false;
}
```

Esto se repetirá indefinidamente hasta que el jugador mate al jefe final, tras lo cual la cámara volverá a su posición original, las vallas bajarán, y se levantarán desde el suelo unas escaleras ocultas para poder acceder al pasillo siguiente, tras lo cual se accede a la pantalla de récords.

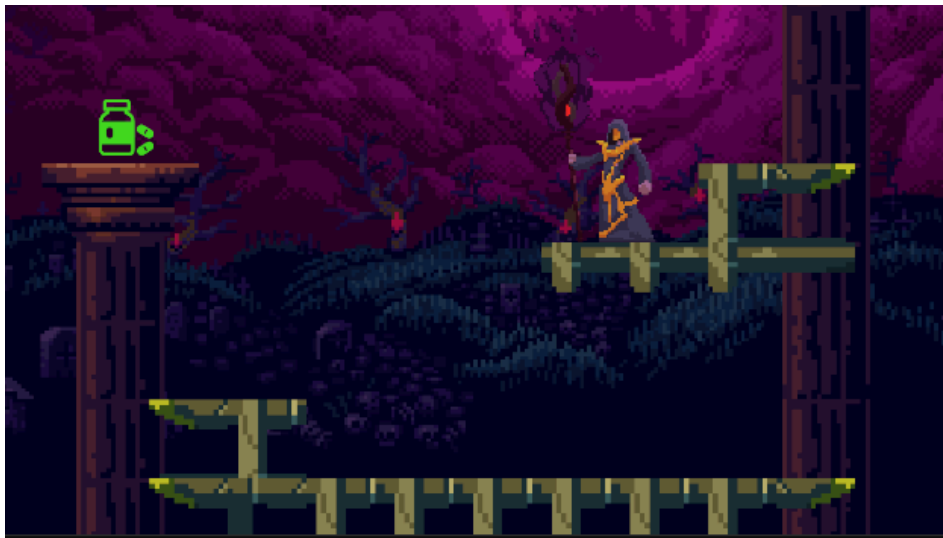
```
referencia
public void PostMuerte()
{
    StartCoroutine(MoverObjeto(vallas, vallas.transform.position, posicionObjetivoVallas, 0.5f));
    StartCoroutine(MoverObjeto(escaleras, escaleras.transform.position, posicionObjetivoEscaleras, 0.5f));

    if (camara != null)
    {
        camara.AjustarAlturaYZoom(0f, 0f, 0.5f);
    }
    else
    {
        Debug.LogWarning("Camara no asignada en el script Angel.");
    }
}
```

5. Extras: Objetos de salud

En distintas zonas del mapa podemos hallar objetos que consisten en un sprite que representa un bote con pastillas. Dichos objetos flotan en bucle arriba y abajo lentamente, animándonos a cogerlos. Son instancias de un prefab.

Si entramos en contacto con cualquiera de ellos (algunos están a simple vista, otros más complicados de ver), estos desaparecerán y nos regenerarán 30 puntos de vida (aunque es fácilmente configurable).



Son útiles, es muy recomendable llegar a la puerta que nos lleva a la batalla contra el ángel corrupto con un nivel de vida considerable, porque de lo contrario se nos puede complicar la escena final.

Estos objetos se manejan con el siguiente Script:

```
using UnityEngine;

0 references
public class curacion : MonoBehaviour
{
    1 reference
    public int cantidadCura = 20;
    1 reference
    public float floatAmplitude = 0.2f; // altura del movimiento
    1 reference
    public float floatSpeed = 2f; // velocidad del movimiento

    4 references
    private Vector3 startPos;

    0 references
    void Start()
    {
        startPos = transform.position;
    }

    0 references
    void Update()
    {
        // Movimiento flotante vertical
        float y = Mathf.Sin(Time.time * floatSpeed) * floatAmplitude;
        transform.position = new Vector3(startPos.x, startPos.y + y, startPos.z);
    }

    0 references
    void OnTriggerEnter2D(Collider2D other)
    {
        WizardMovement jugador = other.GetComponent<WizardMovement>();

        if (jugador != null && !jugador.isDead)
        {
            jugador.CurarVida(cantidadCura);
            //audioManager.reproducirEfecto(audioManager.recogerCura);
            Destroy(gameObject);
        }
    }
}
```


6. Pantalla de récords

Al derrotar al Ángel Corrupto y finalizar el juego, si avanzamos por el *pasillo* de salida se nos abrirá la siguiente pantalla:



Al empezar el juego, comienza un cronómetro que registra cuánto tiempo tardamos desde que empezamos hasta que llegamos al final del pasillo tras derrotar al Ángel. Dicho tiempo se muestra en la pantalla de récords, y nos pide un nombre para entrar en la Tabla de Récords.

Si pulsamos "Guardar", se nos abre la siguiente pantalla:



En dicha pantalla aparecerán los mejores tiempos registrados. Únicamente se mostrarán los 10 mejores tiempos.

7. *Game Over*

Si en cualquier momento, la vida de Wizard llega a 0, se activará una animación de muerte, su correspondiente audio, y se mostrará un menú interactivo en pantalla, con 2 botones, uno para volver a la escena del menú principal y otro para salir de la aplicación:



8. Referencias

La enorme mayoría de recursos visuales y sonoros del juego han sido extraídos de [la tienda de assets de Unity](#):

- [Modelo de Wizard](#)
- [Entorno de Escena 1 y enemigo esqueleto](#)
- [Entorno de Escena 2 enemigo Ángel y proyectil de bola de fuego personaje principal](#)
- [Elementos de Interfaz](#)
- [Límite a la izquierda de Escena 1 y agua venenosa](#)
- [Puerta de castillo final de escena 1](#)
- [Colección de efectos de sonido 1](#)
- [Colección de efectos de sonido 2](#)

También hay algunos efectos de sonido que no se han extraído de la asset store de Unity, sino de [Freesound](#).

- [Espadazo de esqueleto](#)
- [Muerte Wizard](#)
- [Muerte esqueleto](#)
- [Esqueleto al recibir daño](#)
- [Explosión de bola de fuego](#)

También hay algunos sonidos extraídos de [Youtube](#):

- [Muerte ángel](#)
- [Música Escena 2](#)

Y la apariencia de las espadas de fuego en la Escena2 se extrajo de [itch.io](#):

- [Espadas de fuego](#)

Y algunas imágenes han sido generadas por [ChatGPT](#), como:

- La portada de la escena del menú principal
- La pantalla de Game Over