

MEMORIA REPENTANCE



María López Ausín

11/06/21

Diseño y desarrollo de videojuegos

2020 - 2021

ESNE

ÍNDICE

1. Descripción general	2
2. Descripción técnica	2

DESCRIPCIÓN GENERAL

El juego está inspirado en la nueva expansión de The Binding of Isaac: Repentance. Este juego es un top-down roguelike en el que te mueves con WASD y disparas con las arrows.

Lo que le diferencia en cuanto a mecánicas con el juego original es que en esta expansión hay una fase final, en la que en vez de por salas te mueves en un espacio abierto 2D scroll lateral, volando por el mismo y matando a los enemigos, que vuelan también.

En mi juego, tendrás que derrotar a un boss que spawna monstruos sin que estos te maten.

DESCRIPCIÓN TÉCNICA

En este apartado explicaré los componentes técnicos del proyecto.

Para comenzar, tenemos un index con el canvas, una hoja de estilos, y tres carpetas: source, assets y una para el icono de la página.

En el canvas se podrá jugar al juego. Aquí se dibuja el menú de inicio, compuesto por dos botones: Jugar y Créditos. Al darle a jugar, se creará una nueva escena de juego.

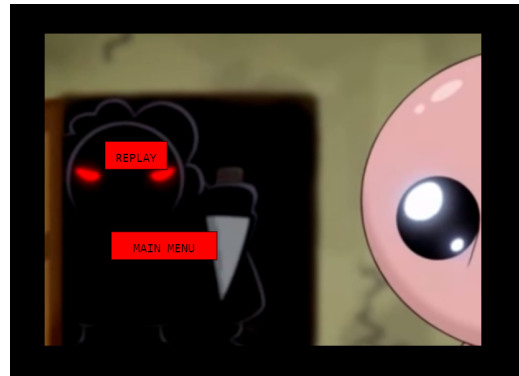
Esta nueva escena hereda de una clase Scene, la cual dibuja todos los gameObjects en la escena y tiene Update y Start. En la gameScene nos encontramos variables globales como la vida del boss y la vida del pj, además de tener sus variables internas como mundo, pj, o boss.

Además, gestiona los propios límites, contiene la actualización del menú de pausa y dibuja y gestiona todos los elementos de UI.

El player es un objeto con un cuerpo box2d que puede disparar con una cadencia y colisionar con enemigos. Si esto último ocurre, bajará la vida del persona y podrás llegar a morir.

```
this.body = CreateBox(world, this.position.x / scale, this.position.y / scale, 0.5, 0.45,
    {fixedRotation: true, restitution: 0.5, linearDamping: 8});
this.body.SetUserData('player') ;
```

Cabe destacar que el menú de muerte es diferente al menú de win.



Por otro lado contamos con los enemigos, gameObjects que al entrar en contacto con nuestro personaje le harán daño o recibirán daño si entran en contacto con una de nuestras bullets. Tenemos tres: lilfly, fly, keeper y el boss

Comenzando por el boss, es un objeto creado en la gameScene que realiza un movimiento circular continuo, spawna enemigos y tiene un montón de vida.

```
this.angle +=deltaTime;  
  
let movementVector = new b2Vec2(2.2*Math.cos(this.angle) - 2.2*Math.sin(this.angle),  
                                2.2*Math.sin(this.angle)+ 2.2*Math.cos(this.angle));
```

El ratio de spawn depende de su vida y de unos timers aleatorios, spawnando también un enemigo aleatorio.

Las dos moscas realizan un movimiento en el eje y de coseno, aunque la mosca grande tiene un offset aleatorio y mayor velocidad.

El keeper funciona de forma diferente ya que no lo spawna el boss, sino que es un enemigo con más vida que las moscas que cada vez que le matas vuelve a spawnear, dificultándote la partida.

Al morir todos estos, se llama a la función world.DestroyBody(), cosa interesante a saber puesto que yo me tiré horas buscando cómo destruir el cuerpo; y se desactiva como gameObject.

Por último, cabe destacar también el uso de librerías externas como box2d o scripts con funcionalidades proporcionadas por el profe que han sido bastante útiles para el desarrollo del juego.

El juego, finalmente, quedó así:

