

# Cosmic Heir

**Francisco Moreno Bonnin**

Grado de ingeniería Informática  
Área de Videojuegos

**Joel Servitja Feu**

**Joan Arnedo Moreno**

04/06/2021



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Cosmic Heir</i>
<b>Nombre del autor:</b>	<i>Francisco Moreno Bonnin</i>
<b>Nombre del consultor/a:</b>	<i>Joel Servitja Feu</i>
<b>Nombre del PRA:</b>	<i>Joan Arnedo Moreno</i>
<b>Fecha de entrega (mm/aaaa):</b>	06/2021
<b>Titulación::</b>	<i>Grado ingeniería Informática</i>
<b>Área del Trabajo Final:</b>	<i>Videojuegos</i>
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave</b>	<i>Shooter, Pixel-Art, Videojuego</i>

**Resumen del Trabajo (máximo 250 palabras):** *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

El objetivo principal de este proyecto poner en práctica todos los conocimientos adquiridos durante la creación de un videojuego (gráficos incluidos) para la realización del TFG. Siguiendo una metodología ágil, se ha diseñado, desarrollado y publicado Cosmic Heir, un twin-stick shooter pixelart para Windows y HTML5, basado es un clásico de la SNES, Super Smash T.V. con el motor de videojuegos Godot.

El desarrollo esta dividido en cuatro fases, la de análisis y 3 de desarrollo, donde cada una de estas contiene la parte de desarrollo propiamente dicha, pruebas con personas externas y corrección de bugs/mejoras. Aunque el desarrollo de un videojuego completo a manos de una sola persona puede llegar a ser una tarea titánica, con constancia, buena organización y unos objetivos bien definidos y asequibles, es posible.

Lo mas complicado fue la selección de una idea, que fuese interesante y divertida de desarrollar, pero a su vez que fuese realista y pudiese completarse en el tiempo establecido.

**Abstract (in English, 250 words or less):**

The main objective of this project is to put into practice all the knowledge acquired during the creation of a video game (graphics included) for the completion of the TFG. Following agile methodology, Cosmic Heir, is a twin-stick pixelart shooter for Windows and HTML5, has been designed, developed and published, based on an SNES classic, Super Smash T.V. with the Godot Game Engine.

The development is divided into four phases, the analysis phase and three development phases, where each of these contains the development part, tests with external people and correction of bugs / improvements. Although the development of a complete video game by a single person can be a daunting task, with perseverance, good organization and well-defined and affordable objectives, it is possible.

The most complicated thing was the selection of an idea, which was interesting and fun to develop, but at the same time that was realistic and could be completed in the established time.

# Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido.....	1
1.4 Planificación del Trabajo.....	2
1.5 Breve resumen de productos obtenidos.....	3
1.6 Breve descripción de los otros capítulos de la memoria.....	3
2. Estado del arte.....	4
2.1 Género.....	4
2.2 Tecnología.....	5
3. Definición del juego.....	7
3.1 Historia.....	7
3.2 Personajes y elementos del juego.....	7
3.3 Interacción entre los actores del juego.....	8
3.4 Interacción del jugador con el personaje.....	9
3.5 Objetivos del jugador.....	9
3.6 Arte.....	10
3.7 Plataformas.....	17
4. Diseño técnico.....	18
4.1 Entorno.....	18
4.2 Requerimientos técnicos.....	18
4.3 Herramientas utilizadas.....	18
4.4 Recursos del juego.....	20
4.5 Esquema de arquitectura del juego.....	21
4.6 I.A.....	21
5. Diseño de niveles.....	22
6. Pruebas con usuarios.....	25
7. Manual de usuario.....	28
7.1 Requisitos mínimos.....	28
7.2 Menús.....	28
7.3 Instrucciones de juego.....	28
8. Conclusiones.....	29
8.1 Logros de los objetivos planteados.....	29
8.2 Seguimiento de la planificación y metodología.....	29
8.3 líneas de trabajo futuro.....	29
9. Glosario.....	30
10. Bibliografía.....	31

## Lista de figuras

Ilustración 1 Diagrama de Gantt	2
Ilustración 2 Super Smash T.V. (SNES)	4
Ilustración 3 Galaxy Champions T.V. (Steam)	4
Ilustración 4 The Binding of Isaac (Steam)	5
Ilustración 5 Controles por defecto de Jugador 1	9
Ilustración 6 Controles por defecto de Jugador 2	9
Ilustración 7 Pantalla de introducción	10
Ilustración 8 Menú principal	11
Ilustración 9 Pantalla de colecciones	11
Ilustración 10 Pantalla de puntuaciones	12
Ilustración 11 Pantalla de créditos	12
Ilustración 12 Menú de opciones	13
Ilustración 13 Menú de controles	13
Ilustración 14 Pantalla de selección de nivel	14
Ilustración 15 Pantalla de personalización de personaje	14
Ilustración 16 Pantalla de juego	15
Ilustración 17 Pantalla de fin de partida	15
Ilustración 18 Hoja de sprites del personaje original	16
Ilustración 19 Hoja de sprites del personaje para la personalización	16
Ilustración 20 Hojas de sprites de enemigos	16
Ilustración 21 Hoja de sprites de efectos y objetos	16
Ilustración 22 Nivel torre 1	17
Ilustración 23 Mapa de la torre	17
Ilustración 24 Nivel jefe torre 1 fase 1	17
Ilustración 25 Nivel jefe torre 1 fase 2	17
Ilustración 26 Mapa	22
Ilustración 27 Nivel	22
Ilustración 28 Nivel con jugador muerto	23
Ilustración 29 Nivel jefe fase 1	23
Ilustración 30 Nivel jefe fase 2	24
Ilustración 31 Mejoras y errores fase pruebas 1	25
Ilustración 32 Mejoras y errores fase pruebas 2	26
Ilustración 33 Mejoras y errores fase pruebas 3	27

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

Los videojuegos están muy presentes en la sociedad actual, y cada vez más, siendo una industria bastante rentable. Por esta razón, se ha decidido desarrollar un videojuego.

Lo que se pretende con este proyecto, es realizar un videojuego entretenido y que las partidas no fuesen muy largas. Aunque el producto aquí obtenido está incompleto, y con una calidad visual no muy alta, el objetivo final sería completarlo y publicarlo de forma pública.

## 1.2 Objetivos del Trabajo

El objetivo de este trabajo es la realización de un videojuego completamente jugable, aunque no con todos los niveles desarrollados. Para llegar a completar este objetivo, podemos desglosarlo en objetivos más pequeños.

- Definir la idea básica del videojuego
- Diseñar prototipos de las diferentes pantallas del videojuego
- Aprender a utilizar y desenvolverse adecuadamente con el motor Godot[1]
- Creación de los sprites a utilizar con la herramienta Aseprite[2]
- Añadir control de código fuente
- Integración continua para la publicación del juego en itch.io[3]
- Desarrollar el videojuego
- Documentar el proceso

## 1.3 Enfoque y método seguido

Para este proyecto he desarrollado un videojuego desde cero con el motor Godot[1]. De esta manera se ha podido aprender la complejidad de todos los aspectos de la creación de un videojuego, desde la idea inicial hasta la publicación. Este desarrollo estaba dividido en tres entregas, donde en cada una de estas estaba dividida en 4 fases.

- Análisis y diseño del trabajo a realizar
- Implementación de las tareas
- Fase de pruebas incluyendo gente externa
- Corrección de errores y mejoras detectados en la fase de pruebas.

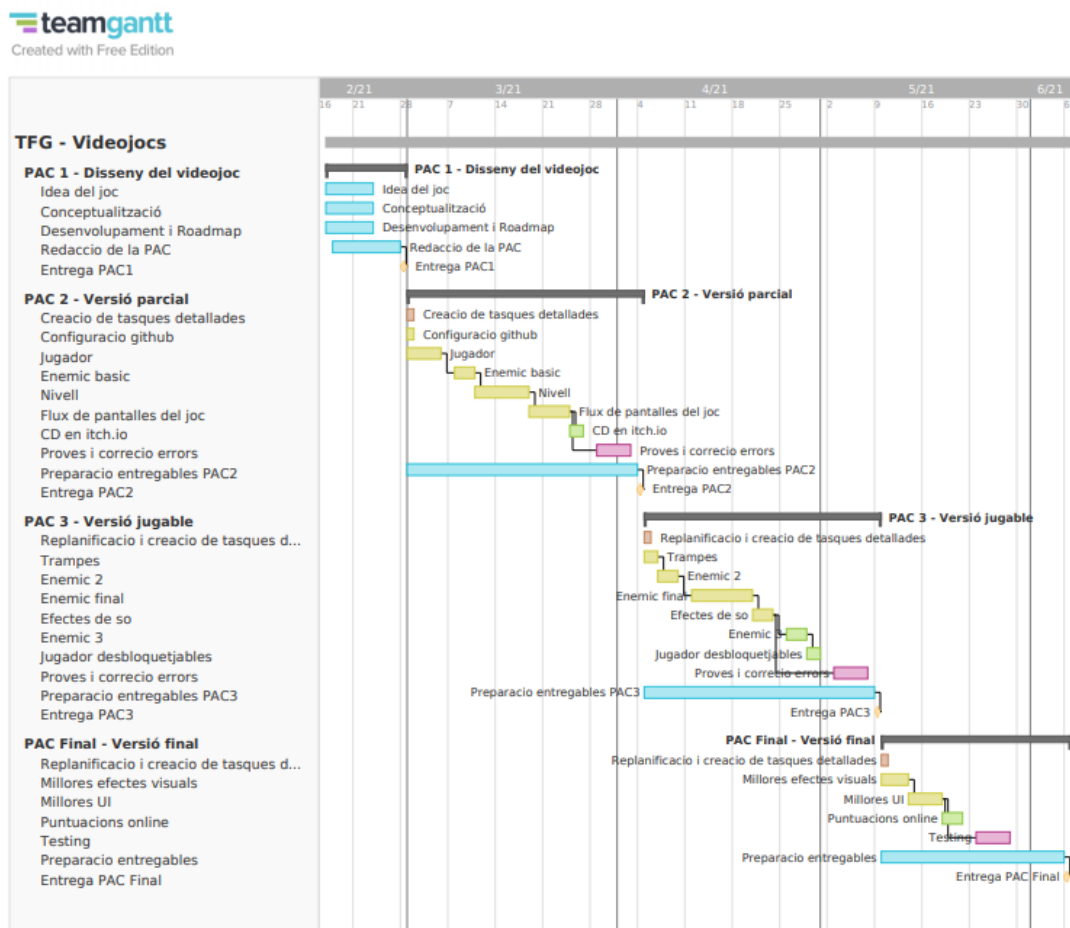
De esta manera, desde la primera entrega, ya se tenía algo jugable, y con gente externa para validarlo, lo que es muy importante y como en este caso debería hacerse cuanto antes.

Cabe destacar, que los únicos recursos de terceros utilizados son los audios, habiendo sido descargados desde la web de sonidos libres freesound [4]. Todos los sprites se han desarrollado para este proyecto.

## 1.4 Planificación del Trabajo

La planificación inicial de este trabajo, ha sido realizada con la herramienta TeamGantt[5], podemos ver en la siguiente figura el diagrama con todos los objetivos realizados durante las 16 semanas de desarrollo a lo largo de cuatro fases.

1. Definir la idea y planificar su desarrollo
2. Creación de la estructura base del videojuego, pero ya jugable
3. En esta fase se añadió toda la funcionalidad restante
4. Últimos retoques visuales



Il·lustració 1 Diagrama de Gantt



Aunque han surgido imprevistos, se han cumplido con éxito todos los plazos marcados, incluso se han llevado a cabo algunas tareas extras que no se habían contemplado inicialmente, como el sistema de misiones, multiidioma, modificación de controles y la inclusión de un segundo jugador. En contrapartida, de las objetivos opcionales (en verde), descarte el desarrollo de puntuaciones online, no por complejidad, si no que no lo encontraba necesario para el proyecto actual, siendo mucho más interesante dedicar ese tiempo a otras tareas.

### 1.5 Breve resumen de productos obtenidos

A lo largo del desarrollo del proyecto, se han producido los siguientes productos:

- Repositorio Github, con una rama por entrega
- Ejecutable versión parcial
- Video PAC2
- Ejecutable versión jugable
- Video PAC3
- Ejecutable versión final
- Video presentación
- Tráiler
- Memoria

### 1.6 Breve descripción de los otros capítulos de la memoria

- Capítulo 2: Se hablará del estado del arte del género, así como de la tecnología utilizada
- Capítulo 3: En este capítulo se detallará la historia del videojuego, así como el detalle de sus diferentes actores
- Capítulo 4: Detalle técnico del proceso de desarrollo
- Capítulo 5: Guía de los niveles e interfaz del videojuego
- Capítulo 6: Detalles de las pruebas con usuarios
- Capítulo 7: Incluye el manual de usuario y los requisitos técnicos
- Capítulo 8: Conclusiones obtenidas tras el desarrollo de un videojuego desde cero

## 2. Estado del arte

### 2.1 Género

Este videojuego estaría clasificado en el género shooter, concretamente sería un twin-stick shooter con vista superior. Aunque es bastante antiguo, una de las bases seguidas es la del juego de snes Super Smash T.V.[6].



Ilustración 2 Super Smash T.V. (SNES)

Más recientes he visto juegos parecidos con una calidad muy superior, como son Galaxy Champions TV[7] y el más famoso The Binding of Isaac[8].



Ilustración 3 Galaxy Champions T.V. (Steam)

Mientras que el primero es básicamente una copia del original con algunas mejoras, The binding of Isaac es una clara evolución del género, eliminando la cámara fija con niveles siempre del mismo tamaño, y un gran número de personajes y poderes a obtener. Debido a esta extra complejidad este proyecto está más basado en el original, con su “zona cerrada”, aunque si que se le han añadido algunas mejoras.



Ilustración 4 The Binding of Isaac (Steam)

## 2.2 Tecnología

Hay una gran variedad de motores disponibles (incluso se podría haber creado uno). Pero debido a los plazos de entre se decidió estudiar la posibilidad de utilizar Unity[9] y Godot[1].



Ilustración 5 Logo Unity

Unity es uno de los motores más populares (antes estaba más orientado al 3D, pero actualmente funciona bastante bien en 2D) y cuenta con una gran comunidad de desarrolladores y tutoriales para casi cualquier cosa que quieras hacer. Dispone de una versión gratuita y otras de pago, y se programa en c#.



**Ilustración 6 Logo Godot**

Por otro lado, Godot es un motor más joven, pero cada vez va ganando más popularidad. En un principio el punto fuerte del motor es el 2D, pero actualmente es bastante bueno también con el 3D. Este motor de código abierto pudiendo hacerte las modificaciones que quieras si no te gusta como esta algo. Al contrario que Unity puedes vender todo lo que hagas libremente sin tener nunca que pagar ni un euro. El lenguaje de programación es propio, se llama GDScript y está basado en Python, por lo que son muy parecidos.





## 3. Definición del juego

### 3.1 Historia

El videojuego, esta ambientado en un universo, donde el Demiurgo, un ser cósmico que vigila el universo, esta cansado, y quiere encontrar un sucesor digno al que traspasarle sus poderes. Para ello, crea siete torres que el jugador deberá superar para poder acceder a su reino, enfrentarse a él y derrotarlo para ocupar su puesto como el nuevo Demiurgo.






### 3.2 Personajes y elementos del juego

#### Personajes






- Demiurgo: Enemigo final del juego (actualmente solo se nombra y no tiene diseño)
- Personaje 1,2:  Los dos personajes iniciales del jugador pudiendo personalizar el diseño seleccionando entre dos tipos de pelo y cambiando el color de pelo, ojos, piel, cuerpo y pantalones.
- Ghostmming:  Lemming fantasma que persigue al jugador y le ataca cuerpo a cuerpo.
- Ghostworm:  Gusano fantasma que persigue al jugador y le dispara misiles.
- Suiciball:  Forma inferior del jefe del nivel, que persigue al jugador y se suicida contra él, ralentizando su movimiento.
- Jefe1: Enemigo final de la primera torre, va abriendo la boca hasta comerse al jugador, donde atacara al jugador con su corazón, el cual esta protegido por un escudo conectado a cuatro huevos.



#### Objetos

- Corazón:  Recupera medio corazón
- Trozo de alma:  El jugador obtiene 10 puntos
- Moneda:  Vida extra
- Bomba:  Daño de área (Aunque genera un agujero)
- Ataque doble:  De cada disparo salen 2 balas

## Escenario

- Trampa de pinchos: El jugador recibe daño al salir los pinchos  

- Tapa de lava: El jugador recibe daño constante si permanece en la lava  

- Laser: El jugador recibe daño si es alcanzado
- Maquina recreativa: Si quedan monedas, el jugador resucita al acercarse  

- Agujeros: El jugador cae por ellos y recibe daño  

- Huevos: El jugador puede romperlos, lo que hará que salgan enemigos de dentro.  


### 3.3 Interacción entre los actores del juego

#### 1. Interacciones del jugador

- Puede moverse por el escenario, chocar con las paredes,
- Puede dañarse al caerse por los agujeros, al chocar con las trampas y al recibir disparos
- Puede disparar a los enemigos y recolectar los objetos soltados por estos.
- Cuando muere, si quedan monedas, puede interactuar con la maquina recreativa de la parte superior del nivel para volver a la vida.

#### 2. Interacciones de los enemigos

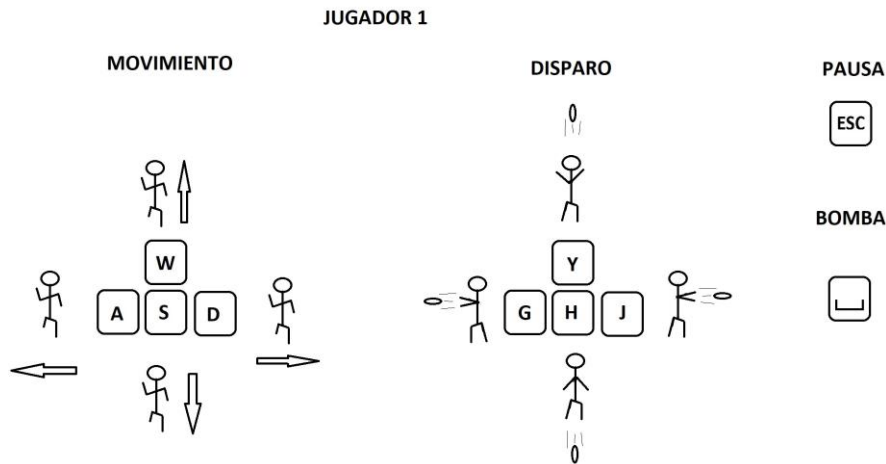
- Persiguen y atacan al jugador
- Reciben daño al ser atacados por el jugador
- Al morir pueden soltar un objeto

#### 3. Interacciones del mundo

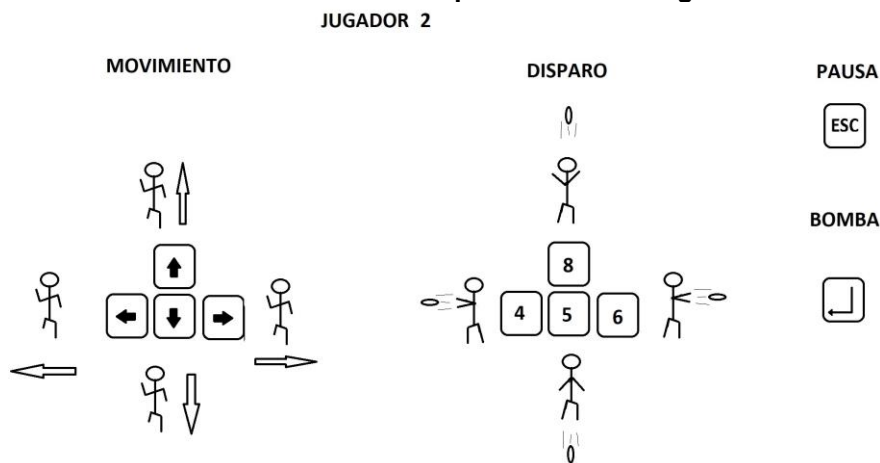
- Genera las oleadas de enemigos y trampas

### 3.4 Interacción del jugador con el personaje

Como se puede jugar a uno o dos jugadores, cada jugador controlara a un personaje. Los jugadores solo necesitan el teclado para interactuar con el juego. A continuación, se muestran los controles por defecto, pudiéndose modificar desde el menú de opciones.



**Ilustración 7 Controles por defecto de Jugador 1**



**Ilustración 8 Controles por defecto de Jugador 2**

### 3.5 Objetivos del jugador

El objetivo del jugador es derrotar al Demiurgo, para ello necesitara completar siete niveles (torres). Para completar cada nivel, el jugador tiene que superar seis pisos en cada torre eliminando a todos los enemigos que aparezcan, para posteriormente enfrentarse al jefe del nivel en el séptimo piso.

En cada piso que no sea de jefe, hay una probabilidad de que aparezca una misión opcional, que al completarse desbloqueas las bombas para todas las partidas y una medalla. Actualmente se han desarrollado las siguientes:

1. Misión destruir huevos  
Destruir cuatro huevos dentro del límite de tiempo  
Medalla de oro:30s  
Medalla de plata:35s  
Medalla de bronce:40s  
Recompensa: 15/10/5 bombas
2. Misión protección  
Proteger al enemigo amistoso  
Medalla de oro: no muere  
Medalla de plata: sobrevive al menos 40s  
Medalla de bronce: sobrevive al menos 30s  
Recompensa: 15/10/5 bombas

### 3.6 Arte

Menús / pantallas del videojuego

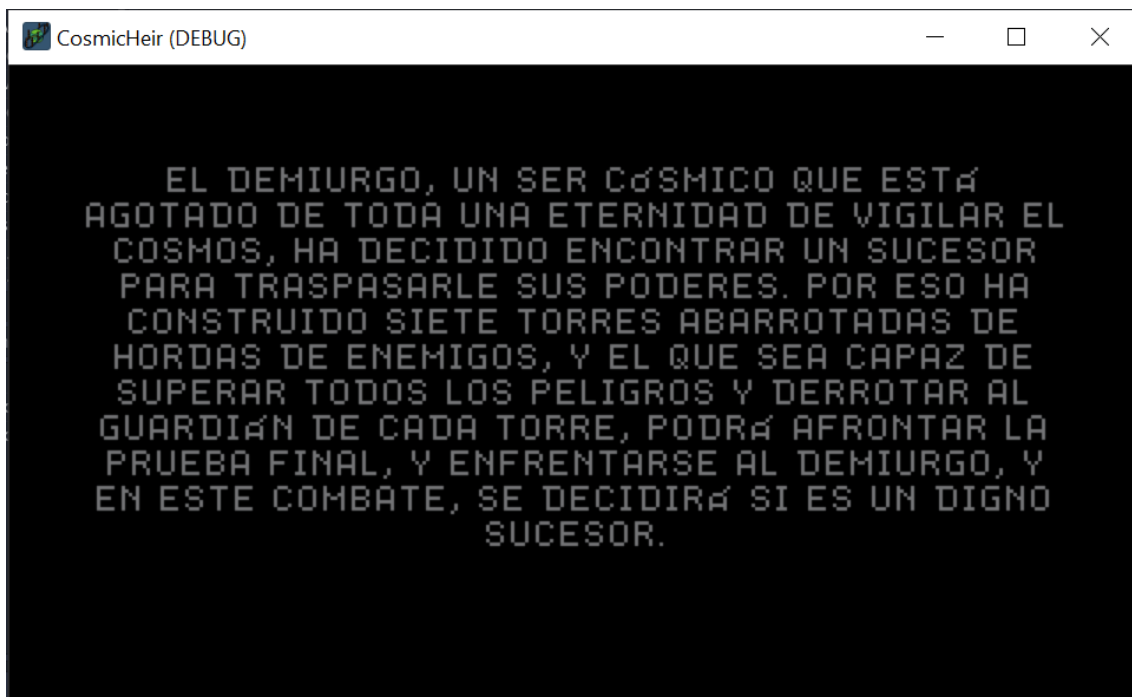


Ilustración 9 Pantalla de introducción





Ilustración 10 Menú principal



Ilustración 11 Pantalla de colecciones

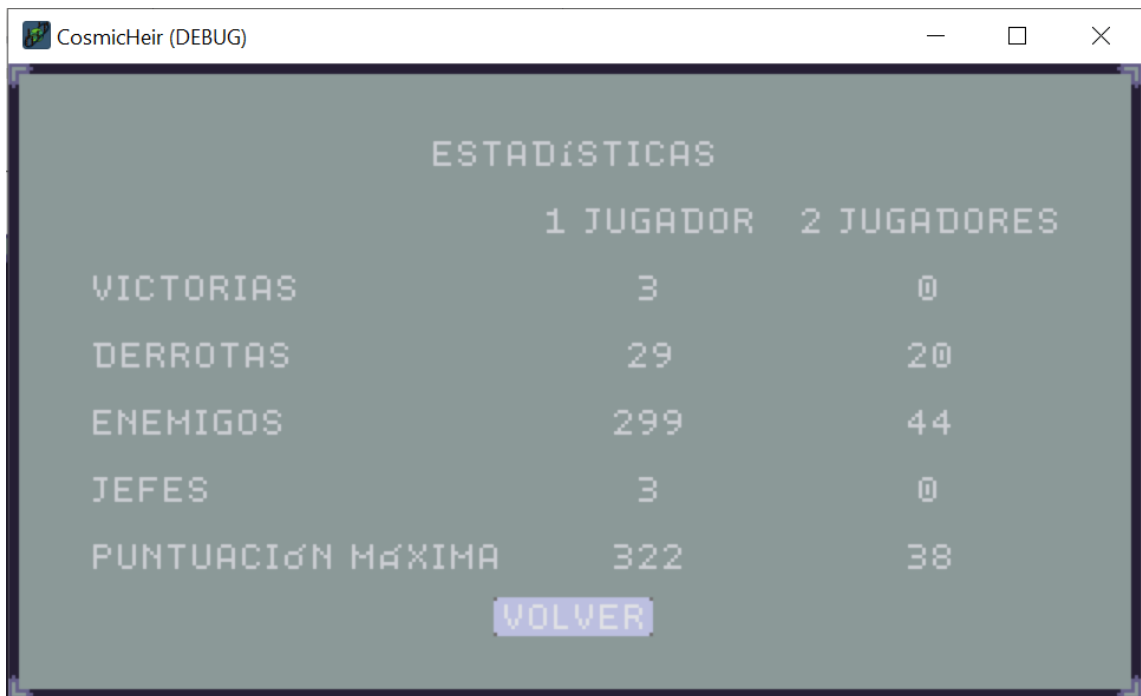


Ilustración 12 Pantall de puntuaciones

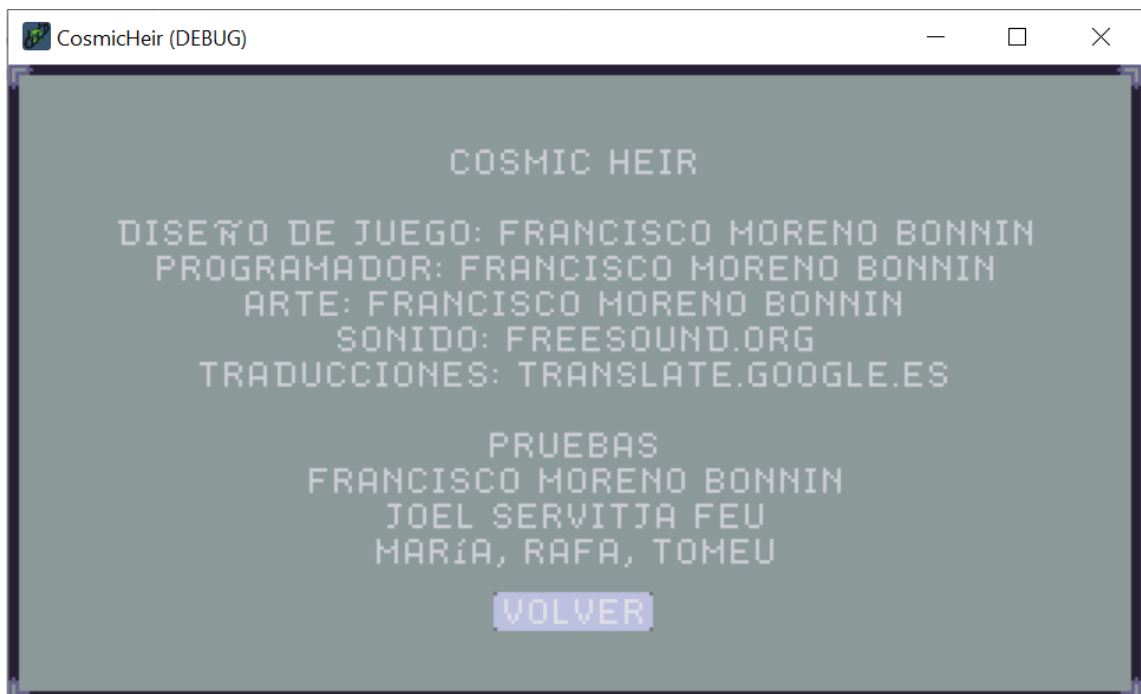


Ilustración 13 Pantalla de créditos



Ilustración 14 Menú de opciones



Ilustración 15 Menú de controles



Ilustración 16 Pantalla de selección de nivel



Ilustración 17 Pantalla de personalización de personaje



Ilustración 18 Pantalla de juego

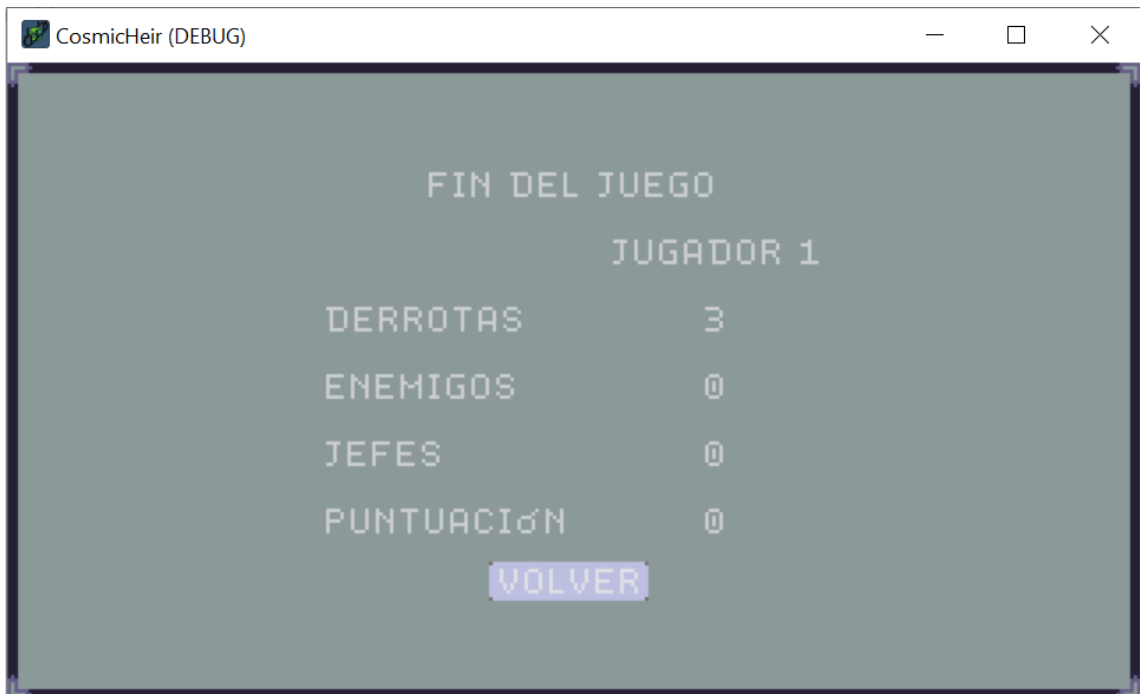


Ilustración 19 Pantalla de fin de partida

## Personajes y enemigos



Ilustración 20 Hoja de sprites del personaje original

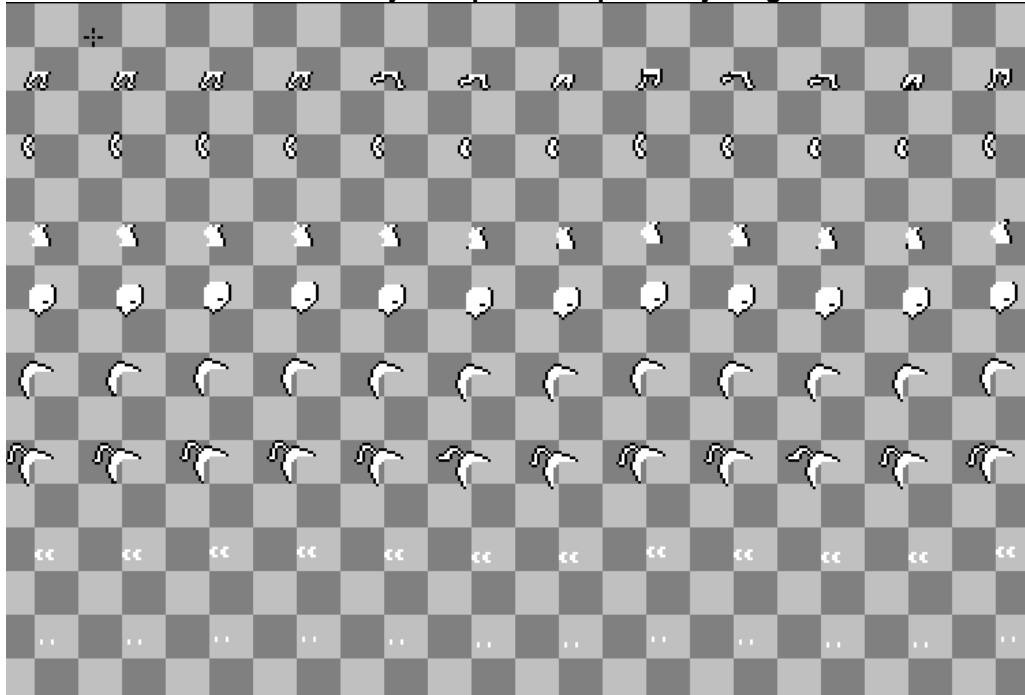


Ilustración 21 Hoja de sprites del personaje para la personalización

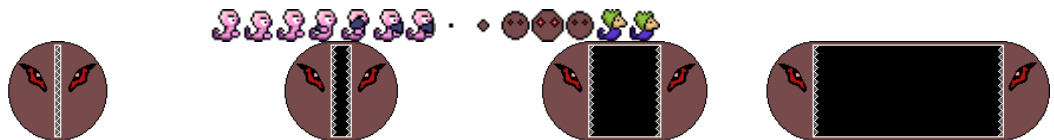


Ilustración 22 Hojas de sprites de enemigos

## Efectos y objetos

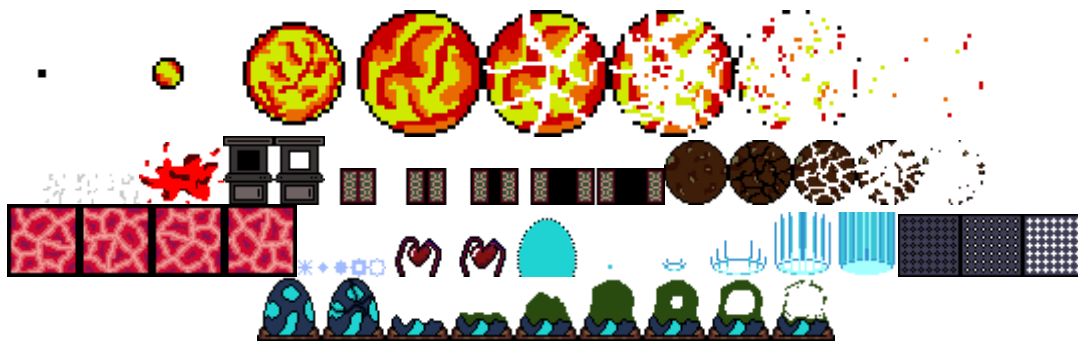
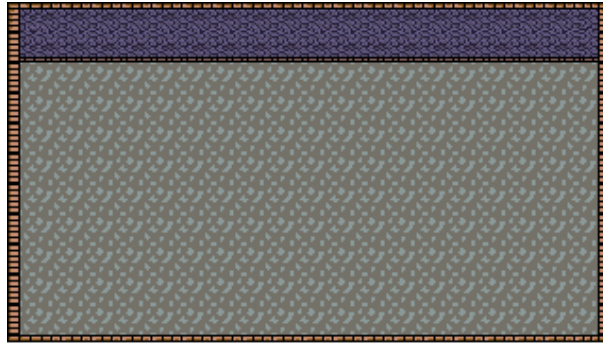
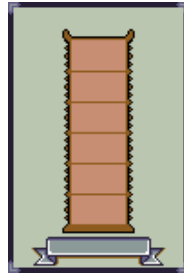


Ilustración 23 Hoja de sprites de efectos y objetos

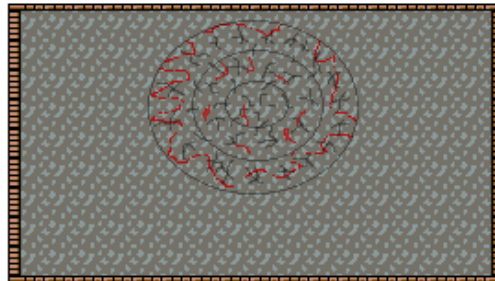
## Niveles



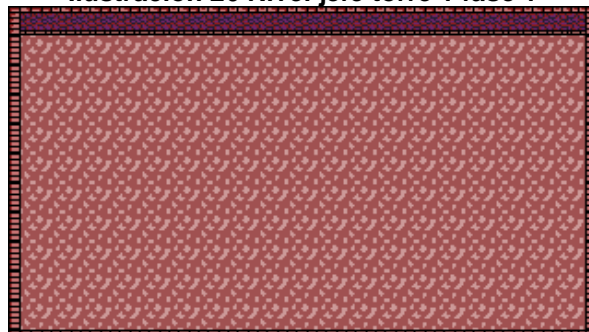
**Ilustración 24 Nivel torre 1**



**Ilustración 25 Mapa de la torre**



**Ilustración 26 Nivel jefe torre 1 fase 1**



**Ilustración 27 Nivel jefe torre 1 fase 2**

### 3.7 Plataformas

Las plataforma de destino principal es Windows, aunque se ha publicado una versión HTML5 en [itch.io](https://itch.io)[3] que se puede ejecutar desde el navegador.

## 4. Diseño técnico

### 4.1 Entorno

Para este proyecto se ha decidido usar Godot v3.3, por su especialización del 2d, facilidad de uso, es muy ligero, es software de código abierto, y un poco, por probar algo no tan famoso como lo es Unity.

### 4.2 Requerimientos técnicos

Los requisitos mínimos para utilizar todas las herramientas indicadas en el punto siguiente son básicamente las del OBS Studio, herramienta utilizada para la creación de los videos, ya que el resto necesita menos.

- Windows 8
- Directx 10.1
- OpenGL 2.1
- CPU dual core
- 4GB ram

### 4.3 Herramientas utilizadas

- Godot: Motor utilizado para la creación del videojuego. La versión utilizada es la 3.3



Ilustración 28 Logo Godot

- Github: Control de código fuente utilizado para guardar el código



Ilustración 29 Logo GitHub



- Aseprite: Programa de píxel art utilizado para la creación de todos los sprites y hojas de sprites (animaciones)



Ilustración 30 Logo Aseprite

- Taiga: Herramienta utilizada para llevar el control de las tareas a realizar, así como para que los QA anotaras errores y mejoras



Ilustración 31 Logo Taiga

- Itch.io: Web donde se hizo que github publicara el videojuego para que los testers tuviesen un forma rápida y sencilla de probar el videojuego.



Ilustración 32 Logo itch.io

- OBS Studio: Para la creación de los videos



Ilustración 33 Logo OBS Studio

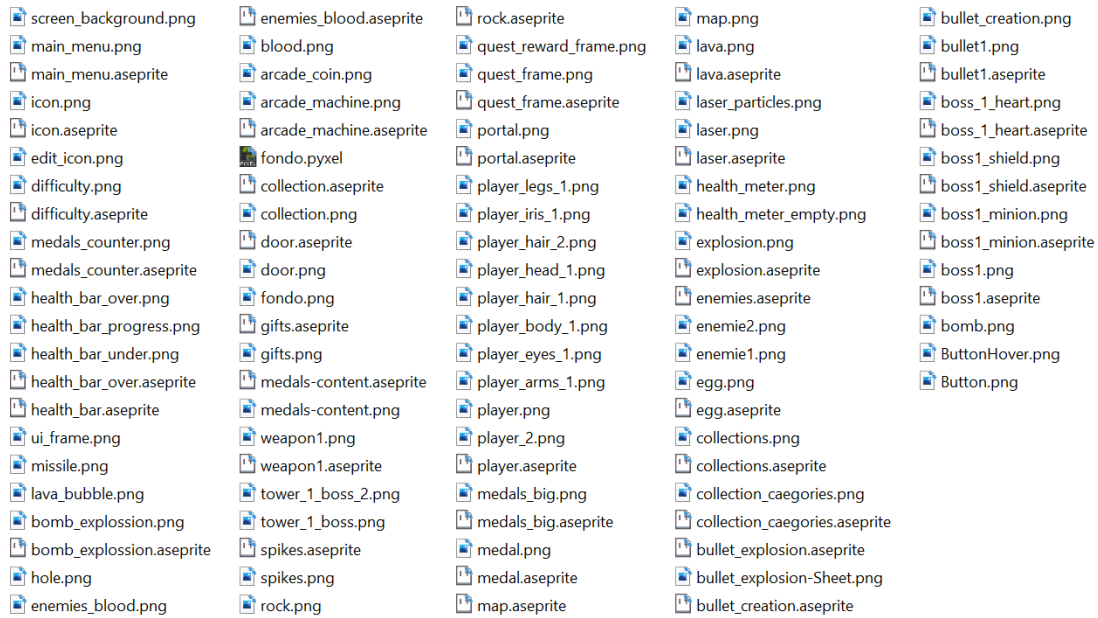
- Draw.io: Para generar los diagramas



Ilustración 34 Logo draw.io

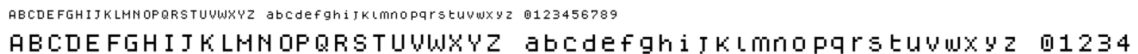
## 4.4 Recursos del juego

Todos los recursos gráficos (que ya hemos visto anteriormente), los he realizado para este proyecto con la herramienta Aseprite.



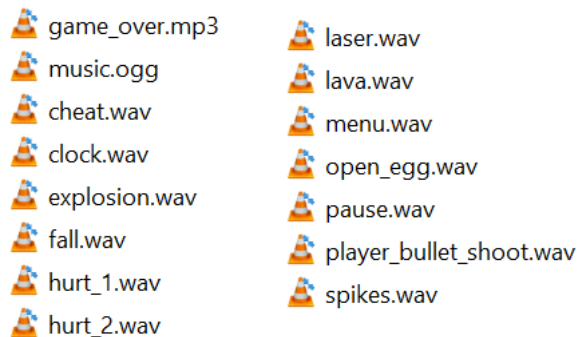
**Ilustración 35 Listado de recursos gráficos**

A parte de estos recursos, he utilizado la fuente dogica pixel descargada desde dafont [16].



**Ilustración 36 Fuente**

Y los recursos de audio descargador desde freesound [4]



**Ilustración 37 Sonidos**

## 4.5 Esquema de arquitectura del juego

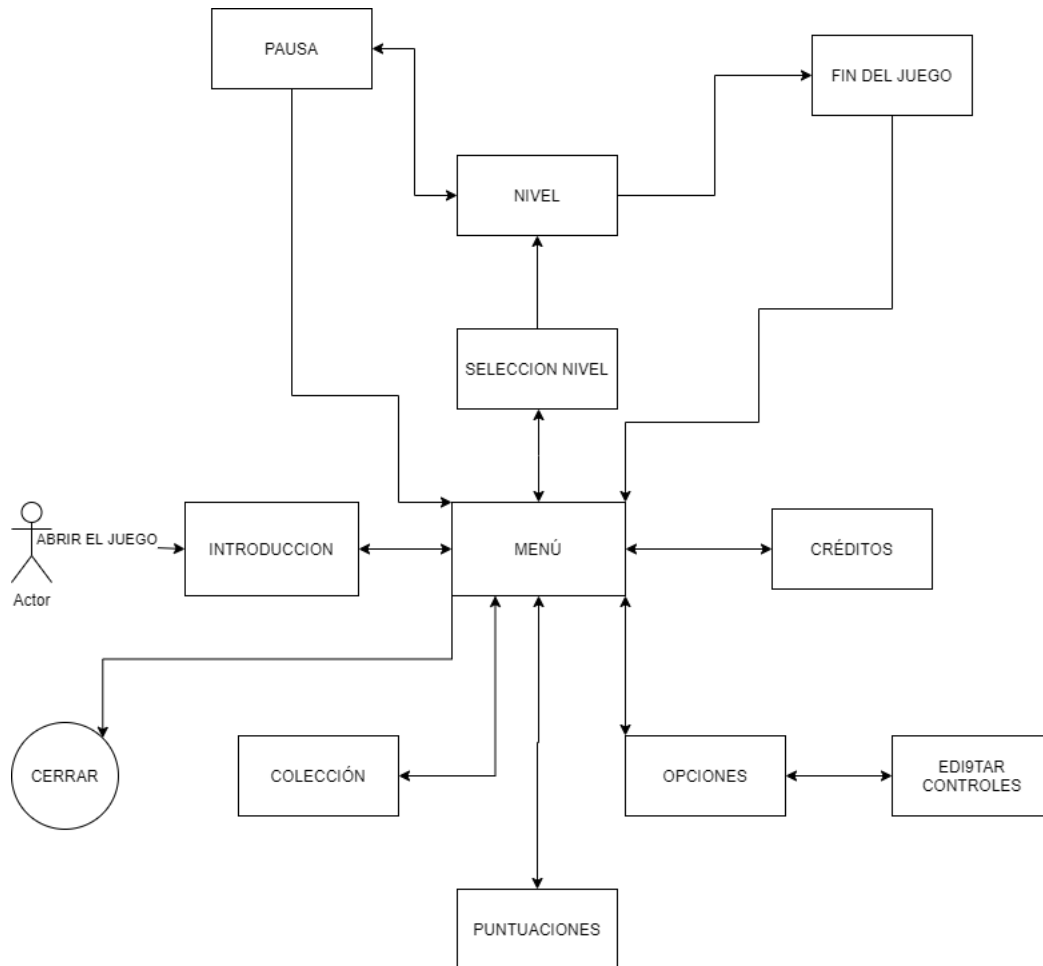


Ilustración 38 Árbol de navegación

## 4.6 I.A.

En el estado actual del videojuego se pueden encontrar un total de tres enemigos y un jefe cada uno con sus diferentes comportamientos.

Para ello he creado una máquina de estados basándome en la documentación de Godot, así como de diferentes tutoriales. Cada tipo de enemigo tiene el nodo de máquina de estado, y dentro de este los estados que utiliza.

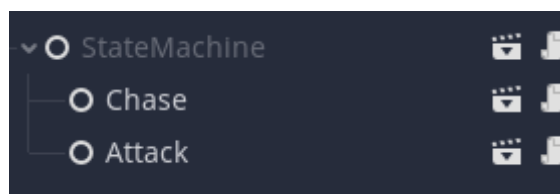


Ilustración 39 Máquina de estados de un enemigo

El nodo maquina de estados, tiene un parámetro que es estado inicial, en el caso de los enemigos es el de perseguir. Cuando el enemigo tiene al jugador a cierta distancia, pegado para cuerpo a cuerpo, y mas alejado para distancia, cambia al estado de atacar. Hay un tercer enemigo que en lugar del estado atacar, tiene un estado suicidarse, lo que produce daños y ralentiza al jugador. El comportamiento del jefe esta programado directamente, ya que la primera fase es solo una animación, y la segunda es un ataque cada x segundos.

## 5. Diseño de niveles

A continuación, un detalle de los niveles del juego, indicando sus elementos.

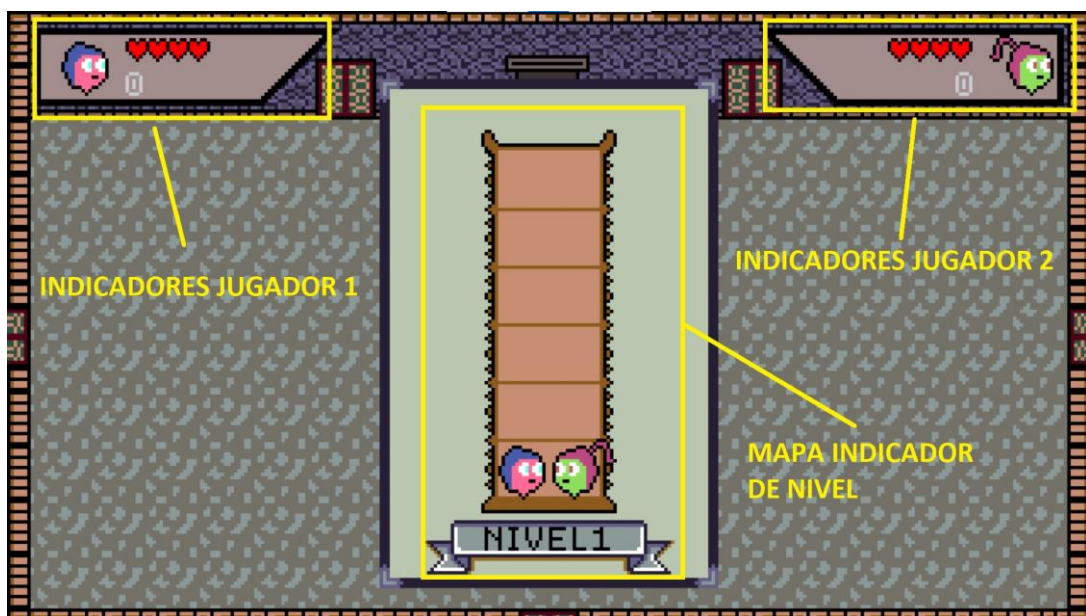


Ilustración 40 Mapa



Ilustración 41 Nivel



Ilustración 42 Nivel con jugador muerto

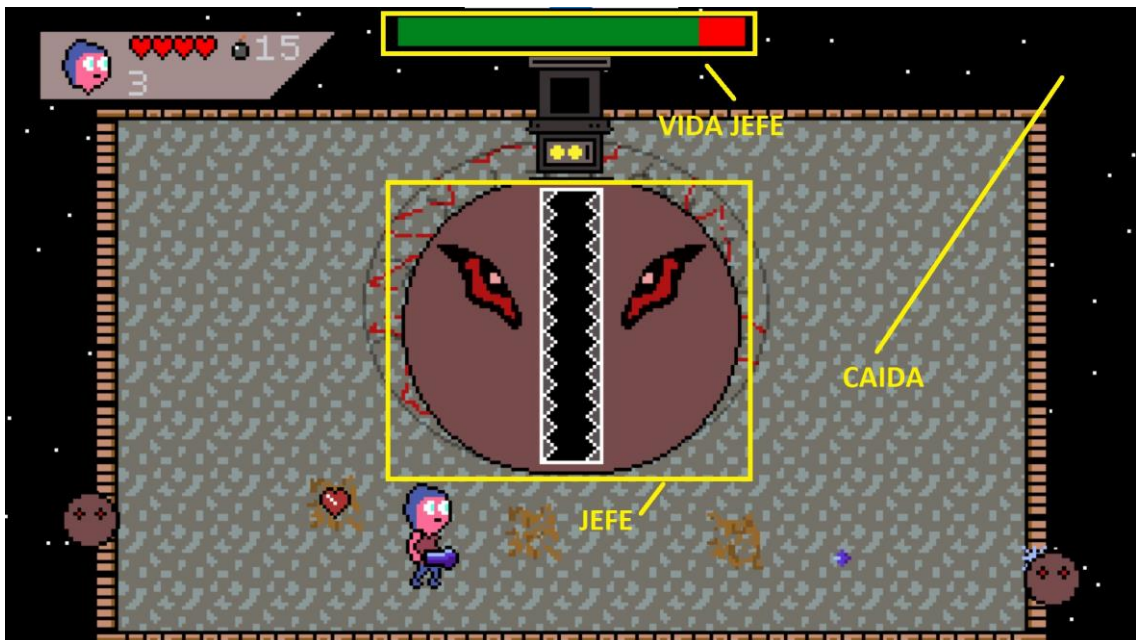


Ilustración 43 Nivel jefe fase 1

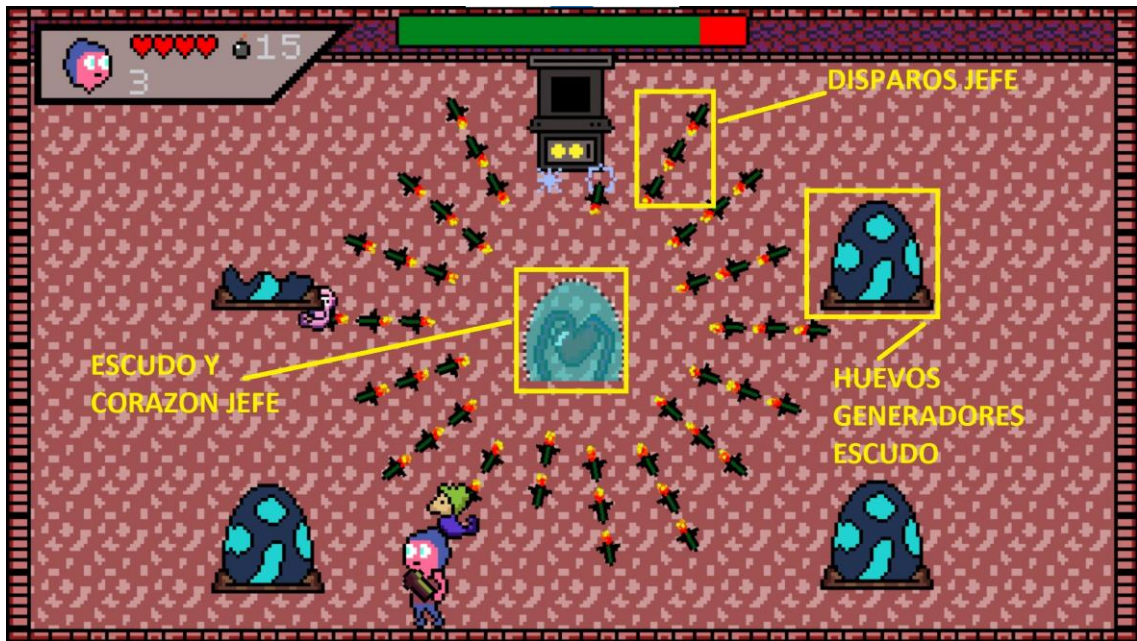


Ilustración 44 Nivel jefe fase 2

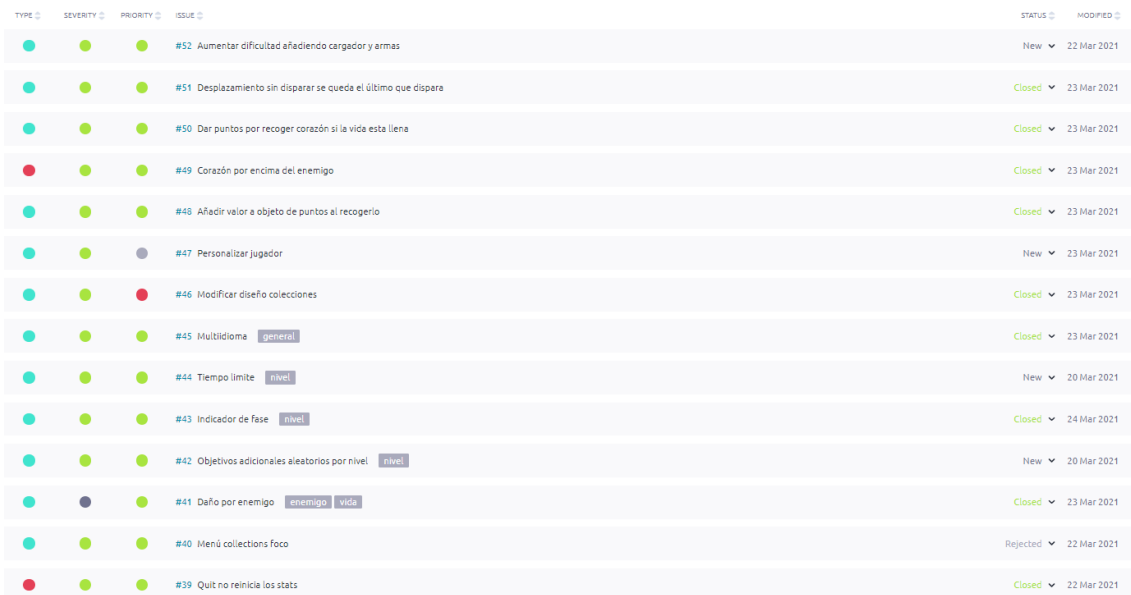
## 6. Pruebas con usuarios

Como se ha comentado anteriormente, a lo largo del desarrollo del videojuego, se han tenido 3 periodos de pruebas, una durante cada una de las 3 entregas. Estas pruebas han sido realizadas tanto por el desarrollador y el tutor (al realizar la entrega) como por tres personas externas.

Estos tres participantes son:

- Rafa, informático de 34 años
- María, informática de 30 años
- Tomeu, estudiante de 11 años

Los dos adultos tienen experiencia con todo tipo de juegos, lo que aporta bastantes ideas de mejoras, así como la detección de errores. Gracias al niño que es el hijo de uno de los probadores adulto, se pudo probar la parte de dos jugadores. A continuación, mostrare todos los tiques creados durante estas pruebas en la herramienta Taiga[10].



TYPE	SEVERITY	PRIORITY	ISSUE	STATUS	MODIFIED
●	●	●	#52 Aumentar dificultad añadiendo cargador y armas	New	22 Mar 2021
●	●	●	#51 Desplazamiento sin disparar se queda el último que dispara	Closed	23 Mar 2021
●	●	●	#50 Dar puntos por recoger corazón si la vida está llena	Closed	23 Mar 2021
●	●	●	#49 Corazón por encima del enemigo	Closed	23 Mar 2021
●	●	●	#48 Añadir valor a objeto de puntos al recogerlo	Closed	23 Mar 2021
●	●	●	#47 Personalizar jugador	New	23 Mar 2021
●	●	●	#46 Modificar diseño colecciones	Closed	23 Mar 2021
●	●	●	#45 Multidioma <span>general</span>	Closed	23 Mar 2021
●	●	●	#44 Tiempo limite <span>nivel</span>	New	20 Mar 2021
●	●	●	#43 Indicador de fase <span>nivel</span>	Closed	24 Mar 2021
●	●	●	#42 Objetivos adicionales aleatorios por nivel <span>nivel</span>	New	20 Mar 2021
●	●	●	#41 Daño por enemigo <span>enemigo</span> <span>vida</span>	Closed	23 Mar 2021
●	●	●	#40 Menú collections foco	Rejected	22 Mar 2021
●	●	●	#39 Quit no reinicia los stats	Closed	22 Mar 2021

**Ilustración 45 Mejoras y errores fase pruebas 1**

TYPE	SEVERITY	PRIORITY	ISSUE	STATUS	MODIFIED
<span style="color: cyan;">●</span>	<span style="color: orange;">●</span>	<span style="color: green;">●</span>	#69 Mensaje de recompensa misión	Closed	05 May 2021
<span style="color: red;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#70 No te asigna la victoria	Closed	05 May 2021
<span style="color: cyan;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#75 Nuevo objeto súper corazón	Closed	05 May 2021
<span style="color: red;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#77 La roca en las misiones	Closed	05 May 2021
<span style="color: red;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#76 Error visualización medalla dificultad	Closed	05 May 2021
<span style="color: cyan;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#67 Añadir dificultad del juego	Closed	05 May 2021
<span style="color: cyan;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#72 Cambiar estética del boss final	Closed	04 May 2021
<span style="color: cyan;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#66 Apartado medallas	Closed	04 May 2021
<span style="color: cyan;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#71 Animación atacar enemigos	Closed	04 May 2021
<span style="color: cyan;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#68 Solo un monstruo por huevo	Closed	04 May 2021
<span style="color: cyan;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#42 Objetos adicionales aleatorios <span style="background-color: #ccc; padding: 2px;">nivel</span>	Closed	04 May 2021
<span style="color: cyan;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#44 Tiempo limite <span style="background-color: #ccc; padding: 2px;">nivel</span>	Closed	04 May 2021
<span style="color: cyan;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#73 Laser por turnos	Closed	04 May 2021
<span style="color: red;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#65 Eliminar huevos rotos misión	Closed	04 May 2021
<span style="color: red;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#74 2 jugadores se pisan la cabeza	Closed	04 May 2021
<span style="color: cyan;">●</span>	<span style="color: green;">●</span>	<span style="color: gray;">●</span>	#47 Personalizar jugador	Closed	28 Apr 2021
<span style="color: cyan;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	#52 Aumentar dificultad añadiendo cargador y armas	Closed	28 Apr 2021

**Ilustración 46 Mejoras y errores fase pruebas 2**



TYPE	SEVERITY	PRIORITY	ISSUE	STATUS	MODIFIED
●	●	●	#93 Jugador al caer sale fuera de las paredes	Closed	25 May 2021
●	●	●	#95 Cambiar nodo audio2d por audio	Closed	25 May 2021
●	●	●	#94 sangre se queda al llegar al jefe	Closed	25 May 2021
●	●	●	#91 Desplazamiento foco en menú editor falla editor personaje	Closed	25 May 2021
●	●	●	#92 Traducciones pantalla colecciones	Closed	25 May 2021
●	●	●	#89 Radiobuttons	Closed	25 May 2021
●	●	●	#88 Estadísticas	Closed	25 May 2021
●	●	●	#90 Eliminar botón volver por icono info Esc	Postponed	24 May 2021
●	●	●	#87 Bomba control remoto	Closed	22 May 2021

### Ilustración 47 Mejoras y errores fase pruebas 3

En la fecha de generación de esta memoria, todas las tareas reportadas has sido realizadas excepto la 90, que se ha pospuesto debido a falta de tiempo.

## 7. Manual de usuario

### 7.1 Requisitos mínimos

- 4Gb Memoria ram
- NVIDIA GeForce 6200 o AMD Radeon 9550
- Intel Core2 Duo 2.83GHz o AMD A10
- Windows 7
- 100 megabytes de espacio en disco

Cabe destacar, que el juego puede ejecutarse en cualquier sistema operativo al utilizar la plataforma web itch.io, siempre y cuando dispongamos de teclado.

### 7.2 Menús

Al arrancar el juego se vera la introducción, la cual podemos saltar pulsando una tecla. A continuación, nos muestra el menú principal. Todas las opciones del menú son auto explicativas, pero las detallamos a continuación:

- Jugar: SE abre el menú de selección de nivel
  - Permite seleccionar el nivel de dificultad, si va a jugar uno o dos jugadores, así como editar sus diseños y comenzar la partida
- Colección: Permite ver imágenes y descripciones de los personajes, torres, enemigos y medallas desbloqueadas
- Estadísticas: Muestra las estadísticas en los modos de uno y dos jugadores
- Créditos: Muestra las personas/entidades implicadas en el proyecto
- Opciones: Permite la modificación del volumen, el cambio de idioma y la modificación de los controles
- Salir: Cierra el juego (solo en la versión de escritorio)

### 7.3 Instrucciones de juego

- Derrota a todos los enemigos para pasar al siguiente piso
- Completa las misiones para desbloquear bombas (que luego te podrán aparecer en cualquier partida)
- Ten cuidado las trampas como la lava, los pinchos o los temidos láseres.
- Cuidado con tus propias bombas, eres inmune a su explosión, pero no al agujero que deja en el suelo
- Aunque tengas 4 corazones, el todopoderoso Demiurgo te facilita un poco la batalla dándote dos monedas mágicas que al meterlas en la máquina recreativa de volverán a la vida
- Durante la partida podemos entrar y salir del menú de pausa pulsando la tecla ESC.
- Y, sobre todo, no olvides recoger todos lo que suelten los enemigos, ¡podrían salvarte la vida!

## 8. Conclusiones

La principal conclusión que he sacado de este proyecto es que hacer un juego solo no es fácil. No importa la experiencia que tengas en alguna de las áreas, si quieres hacerlo tú, tendrás que tocarlas todas, lo que necesita una dedicación y esfuerzo considerable. Pero, nada es imposible, con tiempo y perseverancia, y planteándote objetivos a corta plazo que estén dentro de tus capacidades, se puede llegar a hacer (que el juego tenga éxito ya es otra historia).

La otra conclusión que he sacado, aunque ya la tenía medio aprendida es la importancia de mostrar tu trabajo. Cuando un desarrollador prueba su propio juego, inconscientemente evita las acciones que podrían provocar algún error, por lo que, abrirlo al mundo (o a una pequeña parte), te proveerá de unos comentarios muy valiosos.

### 8.1 Logros de los objetivos planteados

Estoy muy contento con mi trabajo, he cumplido todos los objetivos principales establecidos en la planificación, de los que tenía como extras he quitado uno (puntuaciones online), pero en cambio he añadido bastantes cosas que no estaban planificadas, y que me han gustado bastante, como el sistema de misiones, el segundo jugador y la personalización.

### 8.2 Seguimiento de la planificación y metodología

La planificación la he seguido en la medida de lo posible, quitando la última entrega que he tenido problemas en el trabajo, y he perdido 2 días de trabajar en este proyecto, el resto del calendario lo he seguido como tenía propuesto. Si que es verdad, que, sobre todo al principio, altere un poco el orden de las tareas, ya que era difícil trabajar en algo concreto antes de tener la base del juego para poder ir viendo como un conjunto.

En cuanto a la metodología, he seguido la metodología ágil en la medida de lo posible entregando un producto funcional desde la primera entrega, y en cada entrega posterior añadiéndole cada vez más. Para el seguimiento de las tareas a realizar cree en la herramienta taiga[10] tres paneles (equivalente a tres sprints) con las tareas a realizar.

### 8.3 líneas de trabajo futuro

De cara a finalizar este juego, he dejado las siguientes cosas pendiente:

- Creación de música y sonidos personalizada
- Diseño de las 6 torres restantes
- Diseño de 3 enemigos, un jefe por cada torre y al Demiurgo
- Añadir más misiones opcionales y armas
- Añadir una secuencia animada en la introducción y un final
- Añadir controles táctiles para jugar en móvil
- Publicarlo en la tienda de aplicaciones de Android

## 9. Glosario

- Godot: Motor de videojuegos de código libre.
- Hoja de sprites: Conjunto de sprites utilizados, generalmente para realizar una animación.
- Repositorio: Donde está alojado el código fuente.
- Shooter: Genero de videojuegos de disparos.
- Sprite: Mapa de bits que se utiliza para “pintar” un elemento ya sea personaje, objeto, fondo, etc.
- Twin-stick shooter: Subgénero dentro del shooter, donde generalmente se usa un joystick para mover el personaje y el otro para disparar.

## 10. Bibliografía

- [1] G. Engine, «Godot Engine», *Godot Engine*. <https://godotengine.org/> (accedido feb. 19, 2021).
- [2] «Aseprite - Animated sprite editor & pixel art tool». <https://www.aseprite.org/> (accedido mar. 29, 2021).
- [3] «Itch.io», *itch.io*. <https://itch.io/> (accedido feb. 19, 2021).
- [4] «Freesound - Freesound». <https://freesound.org/> (accedido may 06, 2021).
- [5] «TeamGantt». <https://app.teamgantt.com/> (accedido feb. 19, 2021).
- [6] «Smash T.V. for SNES (1992)», *MobyGames*. <https://www.mobygames.com/game/snes/smash-tv> (accedido feb. 19, 2021).
- [7] «Galaxy Champions T.V. (2018)», *MobyGames*. <https://www.mobygames.com/game/galaxy-champions-tv> (accedido feb. 19, 2021).
- [8] «The Binding of Isaac (2011)», *MobyGames*. <https://www.mobygames.com/game/binding-of-isaac> (accedido feb. 19, 2021).
- [9] U. Technologies, «Unity Real-Time Development Platform | 3D, 2D VR & AR Engine». <https://unity.com/> (accedido feb. 22, 2021).
- [10] «Taiga». <https://www.taiga.io/> (accedido feb. 19, 2021).
- [11] «Zotero». <https://www.zotero.org/> (accedido feb. 19, 2021).
- [12] «Draw.io». <https://app.diagrams.net/> (accedido feb. 19, 2021).
- [13] «Organización del proyecto — Documentación de Godot Engine (stable) en español». [https://docs.godotengine.org/es/stable/getting\\_started/workflow/project\\_setup/project\\_organization.html](https://docs.godotengine.org/es/stable/getting_started/workflow/project_setup/project_organization.html) (accedido mar. 02, 2021).
- [14] «GDScript Style Guide — Godot Engine latest documentation». [https://docs.godotengine.org/en/3.2/getting\\_started/scripting/gdscript/gdscript\\_styleguide.html](https://docs.godotengine.org/en/3.2/getting_started/scripting/gdscript/gdscript_styleguide.html) (accedido mar. 02, 2021).
- [15] «Continuous Delivery pipeline for Godot and itch.io · David Saltares». <https://saltares.com/continuous-delivery-pipeline-for-godot-and-itch.io/> (accedido mar. 12, 2021).
- [16] «Dogica Font | dafont.com». <https://www.dafont.com/dogica.font> (accedido mar. 29, 2021).