

# Memoria

# Trabajo de Fin

# de Máster

**Samuel Valcárcel Arce**

Máster Universitario en Diseño y Programación de videojuegos

Trabajo final de Máster

**Joan Arnedo Moreno**

**Helio Tejedor Navarro**

7 de junio de 2020

## Licencia



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

## Ficha del trabajo final

<b>Título del trabajo:</b>	Videojuego realizado para el Trabajo de fin de Máster para la plataforma de PC
<b>Nombre del autor:</b>	Samuel Valcárcel Arce
<b>Nombre del consultor:</b>	Joan Arnedo Moreno
<b>Nombre del PRA:</b>	Helio Tejedor Navarro
<b>Fecha de entrega:</b>	7 de junio de 2020
<b>Titulación:</b>	Máster Universitario en Diseño y Programación de Videojuegos
<b>Área del trabajo final:</b>	Trabajo final de Máster
<b>Idioma del trabajo:</b>	Español
<b>Palabras clave:</b>	Máster, Videojuegos, Unity
<b>Resumen del trabajo:</b>	<p>Desde el inicio de los videojuegos, estos se han entendido como un programa electrónico, normalmente destinado a entretener, en los que una o más personas pueden interactuar con los mismos a través de un dispositivo.</p> <p>Estos programas se pueden ejecutar en diferentes plataformas, ya sea en consolas (portátiles o de sobremesa) o en ordenadores, posibilitando que cualquier persona pueda acceder a este mundo virtual.</p> <p>Este proyecto se ha desarrollado para el trabajo de Fin de Máster, utilizando el motor de videojuegos Unity, con la finalidad de aprovechar y aplicar los conocimientos aprendidos a lo largo del curso, cumpliendo así con los objetivos propuestos para el mismo.</p> <p>Con ello, se ha conseguido finalizar el desarrollo de un videojuego que cumple con la idea inicial propuesta y podrá ser mejorado en líneas futuras.</p>

**Abstract:** Since the beginning of video games, these have been understood as an electronic program, usually intended to entertain, in which one or more people can interact with them through a device.

These programs can be run on different platforms, either on consoles (portable or desktop) or also on computers, making it possible for anyone to access to this virtual world.

This project has been developed for the Final Master's project, using the Unity video game engine, in order to take advantage of the knowledge learned and apply it throughout the course, thus fulfilling the objectives proposed for it.

With all of this, the development of a video game that complies with the initial idea proposed has been achieved and may be improved in future lines.

# Índice

<b>Introducción</b>	<b>8</b>
Contexto y justificación del trabajo	8
Objetivos del trabajo	9
Enfoque y método seguido	10
Planificación del trabajo	11
Breve resumen de productos obtenidos	12
Breve descripción de los otros capítulos de la memoria	13
<b>Estado del arte</b>	<b>13</b>
Géneros del videojuego	13
Plataformas de desarrollo	14
Licencias	15
Documentación	15
Disponibilidad de plataformas de exportación	15
Lenguajes de programación	16
Dificultad	16
Recursos para el desarrollo	16
<b>Definición del juego</b>	<b>17</b>
Idea del juego	17
Descripción del juego	17
Subgénero y referencias a videojuegos existentes	17
Tipo de interacción juego - jugador	18
Plataforma de destino	18
Conceptualización	18
Historia, ambientación y/o trama	18
Definición de los personajes/elementos	19
Interacción entre los actores del juego	19
Objetivos planteados al jugador	19
Concept Art	20
Feedback acerca de la idea inicial	20
<b>Diseño técnico</b>	<b>21</b>
Entornos utilizados	21
Plataforma Unity	21
Piskel	22
Microsoft Visual Studio 2019	23
Sprites	24
Sonidos	26
Arquitectura del juego	26
Inteligencia artificial de enemigos	30

<b>Diseño de niveles</b>	<b>31</b>
<b>Manual de usuario</b>	<b>33</b>
Requisitos	33
Instrucciones de uso	34
<b>Tests realizados</b>	<b>36</b>
<b>Conclusiones</b>	<b>38</b>
<b>Glosario</b>	<b>40</b>
<b>Bibliografía</b>	<b>41</b>

## Lista de figuras

- [Imagen 1. Diagrama de Gantt](#)
- [Imagen 2. Máquina recreativa](#)
- [Imagen 3. Motores de videojuegos \(Unreal, Unity, Godot\)](#)
- [Imagen 4. Referencia al Metal Slug](#)
- [Imagen 5. BoxHead](#)
- [Imagen 6. Enter The Gungeon](#)
- [Imagen 7. Logo de Piskel](#)
- [Imagen 8. Logo de Microsoft Visual Studio](#)
- [Imagen 9. Página principal de perfil en Piskel](#)
- [Imagen 10. Ejemplo de animación de enemigo](#)
- [Imagen 11. Jerarquía de la escena inicial y controles](#)
- [Imagen 12. Jerarquía de la escena del tutorial](#)
- [Imagen 13. Jerarquía de la escena de niveles](#)
- [Imagen 14. Jerarquía de la escena final](#)
- [Imagen 15. Diseño del nivel 1](#)
- [Imagen 16. Diseño del nivel 2](#)
- [Imagen 17. Diseño del nivel 3](#)
- [Imagen 18. Pantalla inicial de The Power of Fire](#)
- [Imagen 19. Controles de The Power of Fire](#)
- [Imagen 20. Configuración de The Power of Fire](#)

# Introducción

## Contexto y justificación del trabajo

Durante muchos años, el mundo de los videojuegos ha sufrido una serie de cambios que lo han hecho modernizarse hasta el punto de rozar la realidad.

Lo habitual, muchos años atrás, consistía en reunirse con un grupo de conocidos o amigos en un local que tuviera máquinas recreativas y disfrutar la tarde jugando con dichas máquinas.

Poco a poco este concepto ha ido evolucionando hasta la actualidad, donde cada uno desde su casa puede tener su consola o PC de sobremesa, permitiendo la posibilidad de jugar con personas de todo el mundo sin tener que trasladarse a ningún lugar adicional.

Existen videojuegos de todo tipo, desde el género de acción más convencional, hasta la actual RV (Realidad virtual), permitiendo éste último un grado de inmersión e interacción con el mundo virtual mucho más cercano a la realidad. Con el paso de los años, las empresas han variado su foco principal de jugadores en la finalidad buscada en el desarrollo de los videojuegos.

Este cambio pasa desde mantener al jugador durante horas en su campaña/historia, a juegos donde lo que predomina es que se invierta en el mismo, ya sea con objetos comprables dentro del juego o con ampliaciones a su posible modo historia, que en muchos casos, se pueden obtener con dinero real traducido a la moneda del juego correspondiente.

Para este proyecto, debido a las limitaciones de tiempo y de equipo de trabajo (al tener que realizarlo una sola persona), se ha optado por un juego sencillo, que entretenga al usuario, similar a los juegos arcade que permiten ser re jugables, con el propósito de conseguir la máxima puntuación al final de la partida. Este género de videojuegos hacen que el jugador fomente sus ganas de superar su puntuación constante.



## Objetivos del trabajo

Para este proyecto, se han marcado una serie de objetivos, de cara a finalizar en los plazos propuestos el videojuego en su totalidad:

### **Apartado artístico:**

- Mapa del nivel, que se utiliza en los 3 disponibles del juego y que servirá para delimitar el espacio que puede utilizar el jugador para moverse durante la partida.
- Sprites de personaje y enemigos, de manera que todos sean diferentes entre sí y dote de entretenimiento a la jugabilidad del videojuego.
- Una interfaz usable y accesible, con la que el usuario pueda moverse entre las diferentes escenas y menús disponibles, consiguiendo introducir al jugador de manera gradual.

### **Apartado técnico:**

- La gestión de oleadas de los enemigos, de forma que los niveles no resulten tan sencillos de superar.
- El comportamiento de los enemigos, ya sea por su movimiento, como por su manera de atacar al personaje principal.
- Puntuación obtenida durante la partida, de modo que suponga un reto a superar cada vez que se vuelvan a intentar los niveles disponibles.

## Enfoque y método seguido

Un proyecto de este estilo, como es el de un videojuego, requiere de mucho esfuerzo y tiempo a invertir de cara a presentar un producto que esté a la altura de la demanda actual en este tipo de mercado.

Esto incluye un equipo al mando, ya sea de diseñadores, programadores, artistas, etc., donde cada uno utiliza su tiempo para dar lo mejor de sí en el desarrollo del videojuego.

En mi caso particular, al no disponer de un amplio grupo de recursos humanos para llevar a cabo el proyecto, he optado por desarrollar un videojuego que a la par de sencillo, puede llegar a resultar entretenido al permitir su re jugabilidad, atrayendo así al jugador casual que quiera pasar un buen rato delante de un ordenador o consola.

Por ello, en cuanto a la parte artística, en este caso con estilo pixel-art (que predomina entre el mundo indie de videojuegos), he utilizado assets gratuitos, que me han permitido centrarme en otros aspectos que no estuvieran tan focalizados en el arte, así como en el apartado sonoro que también juega un papel fundamental en cualquier proyecto de este estilo.

Durante el desarrollo, se han pasado por diferentes fases, entre las cuáles destacan:

- Elegir una idea inicial de videojuego, que fuera posible realizar en el período aproximado de tres meses, que es la duración que se dispone para realizar el Trabajo de Fin de Máster.
- Una vez pensada la idea, se llevaría a cabo con el Motor Unity, para poder aprovechar los conocimientos adquiridos a lo largo del curso, no teniendo que aprender un lenguaje ni motor nuevo, que supondría una inversión de tiempo adicional.
- Diseñar un prototipo con las mecánicas básicas planteadas en la idea inicial con *sprites* básicos como enemigos o el entorno del nivel.
- Posteriormente, se utilizaron los *assets* gratuitos obtenidos, con el único defecto de falta de animaciones, por lo que parte del tiempo se tuvo que utilizar para dicho fin.
- Por último, se finalizaron detalles correspondientes al *gameplay* y se arreglaron fallos que pudieran surgir durante la partida.

## Planificación del trabajo

Para la realización de este videojuego, se estructuró de manera que se cumplieran ciertos objetivos básicos, de cara a ir entregando poco a poco los puntos esperados en cada seguimiento, de manera que se viera un progreso real en el proceso.

Durante este tiempo, se han replanteado ciertas tareas con sus respectivos tiempos, debido a la pandemia que sufre todo el mundo a fecha de hoy, el coronavirus. Muchas de estas tareas estaban relacionadas con el apartado artístico, por lo que no han afectado de manera significativa al desarrollo.

De manera resumida, los seguimientos consistieron en tres entregas principales:

- En primer lugar, un primer documento inicial, de forma que se encapsulara la idea principal y se explicará con detalle cuál iba a ser el proceso que conlleva

un proyecto real y se estipularan fechas correspondientes a todas las tareas, teniendo así el proyecto en constante progreso.

- Por otro lado, una versión inicial del videojuego, en la que se pudiera observar un primer mockup de cómo sería el producto final, obteniendo así un primer acercamiento de lo que se iba a desarrollar en próximas entregas.
- Por último, una entrega próxima a la definitiva, en la que se mostrara un videojuego interactuable y que pudiera ofrecer unos minutos de juego, a falta de añadir posibles detalles o corregir pequeños fallos.

Diagrama de Gantt (The Power Of Fire)

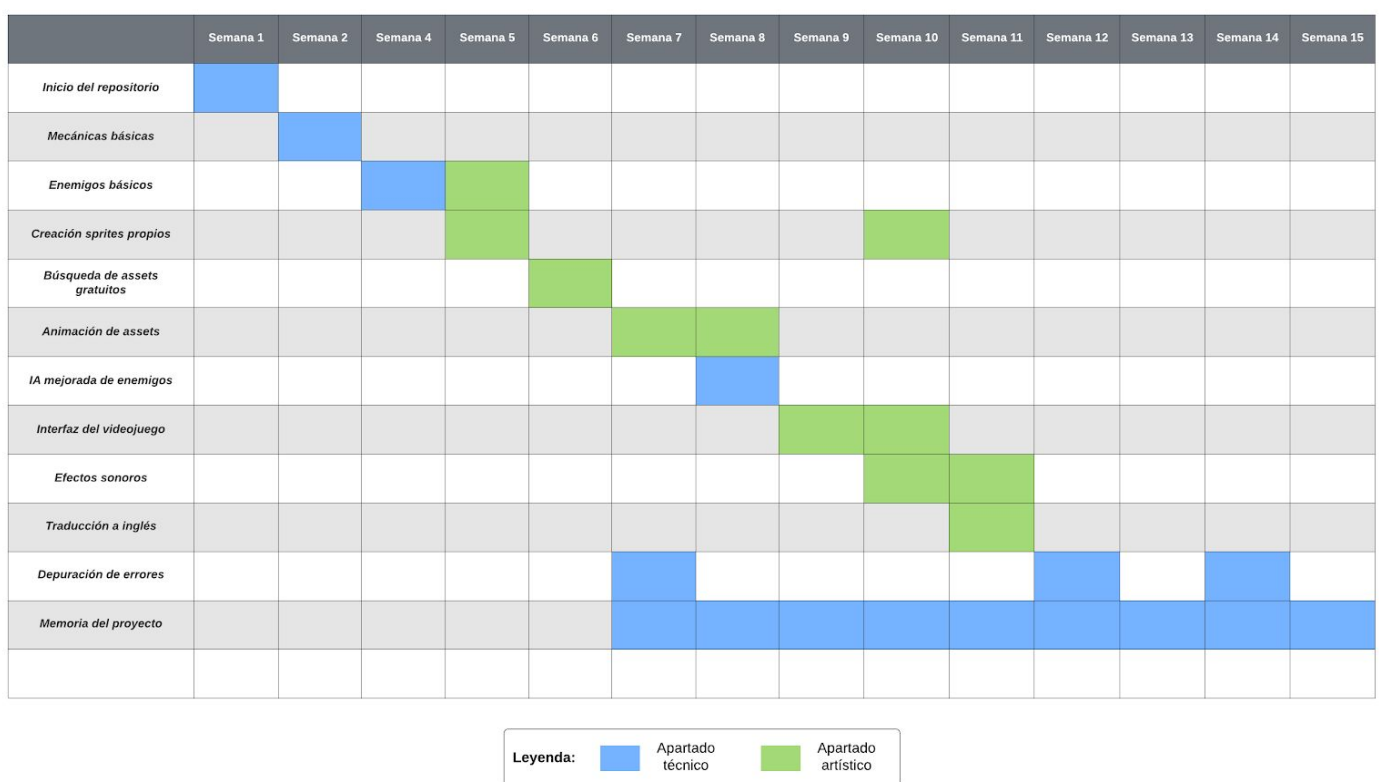


Imagen 1. Diagrama de Gantt

## Breve resumen de productos obtenidos

Durante el desarrollo de este proyecto, se ha utilizado una herramienta básica, que ha permitido un control de versiones del videojuego y así tenerlo actualizado a medida que se realizaba cualquier cambio.

Los productos obtenidos al finalizar esta planificación mencionada anteriormente han sido:

- Un archivo comprimido, que contiene el ejecutable del juego, a su vez en una carpeta, así como todo lo necesario para que éste se inicie sin problema.
- Varios vídeos que se han ido grabando a lo largo del desarrollo, con la intención de ir plasmando las diferentes modificaciones realizadas durante todo el tiempo de creación.
- Un vídeo final, a modo de trailer demostrativo, en el que se muestra un pequeño corto del videojuego, que se podría utilizar para su posterior publicación en alguna plataforma disponible.
- Memoria del proyecto, en el que se aúna todo el proceso en cuanto a la creación desde cero hasta su estado final.

## Breve descripción de los otros capítulos de la memoria

En los siguientes capítulos explicados a continuación, se abordarán los siguientes temas relacionados con el videojuego:

- Qué se entiende por videojuego, cómo influye actualmente entre la población y como ha sabido aprovechar el avance de la tecnología para irse actualizando constantemente.
- Una descripción detallada del proyecto realizado, desde la idea inicial pasando por todas las fases de su desarrollo, con la intención de mostrar el recorrido que ha llevado.
- Elección de entorno y motor para crear el videojuego, así como detallar la jerarquía de carpetas y ficheros referentes al proyecto, de manera que se expliquen assets y recursos del juego.
- Los niveles utilizados para la partida, con una justificación de su elección y cómo se adaptan al resto de elementos del videojuego.
- Un manual de usuario que resuma algunos aspectos de su jugabilidad, de cara a un nuevo usuario que se quiera introducir en la historia.

## Estado del arte

### Géneros del videojuego

Los videojuegos, a medida que pasa el tiempo, se han ido reinventando con la finalidad de atraer a mayor número de personas para que se interesen por esta variedad de mundos virtuales.

Para ello, se han desarrollado multitud de proyectos, que se han categorizado desde lo más simple, como pueden ser los que se encontraban en máquinas recreativas a

finales del siglo pasado, hasta la actualidad, donde ya cualquier persona puede tener acceso a la realidad virtual, permitiendo así una interacción casi real con el mundo del videojuego.

Para este proyecto, como se menciona en puntos anteriores, se ha diseñado un producto sencillo que a la vez enganche al jugador, de modo que fomente que realice muchos intentos hasta conseguir la máxima puntuación. Estos se categorizan como arcade, y su intención es provocar ese estado de frenetismo y velocidad en la partida, de manera que no resulte en una experiencia tosca y pesada, no suscitando así interés por parte del usuario.



Imagen 2. Máquina recreativa

Los juegos destinados a estas máquinas (Imagen 2) suelen carecer de una historia elaborada y consisten en alcanzar la máxima puntuación, ya sea en una carrera de coches, en un *shooter* eliminando oleadas de enemigos o simplemente un juego de peleas con otro jugador que se pueda unir a la misma partida.

Debido a que estos proyectos no resultan complejos a la hora de desarrollarlos, ya que no necesitan de una idea que resulte en una trama que se alargue durante muchas horas, se utilizan motores y entornos no muy pesados, puesto que la carga de trabajo no supone un gran esfuerzo, como puede ser con videojuegos del tipo AAA [1].

En mi caso, aún habiendo muchos motores que están más optimizados para este estilo de videojuegos en 2D, he optado por utilizar Unity, ya que es con el que me siento más familiarizado y conozco con más detalle, puesto que empezar a aprender

otro lenguaje que corresponda a otro motor suponía una inversión de tiempo que no me permitiría pulir el juego en su totalidad.

## Plataformas de desarrollo

Teniendo en cuenta la existencia de diferentes motores para el diseño y desarrollo de videojuegos, podemos analizar las ventajas y desventajas de cada uno de ellos.



Imagen 3. Motores de videojuegos (Unreal, Unity, Godot)

En este caso, se van a valorar tres de los motores actuales más utilizados: **Unreal**, **Unity** y **Godot**, respectivamente.

### Licencias

Unreal es uno de los motores más longevos en el tiempo, que surgió con el famoso videojuego desarrollado en 1994, “Unreal” y más tarde con proyectos como “Borderlands 2”, “Specs Ops: The Line”.

Su precio se diferencia con motores como Unity o Godot, en la manera que se financian, puesto que es gratuito, hasta que el proyecto realizado alcanza cierto margen de beneficios, donde la empresa se lleva un 5% de los beneficios.

Unity, por el contrario, tiene diferentes grados de suscripción que varían en función de los beneficios obtenidos por la empresa, y Godot es un software open source, por lo que supone un llamativo en cuanto a este aspecto.

### Documentación

La documentación es un aspecto importante, en tanto en cuanto es necesaria para el desarrollo de cualquier proyecto. Durante cualquier desarrollo, pueden surgir dudas en cuanto a la utilización del motor, por lo que éste debe de disponer de un lugar donde se ubiquen todas las funcionalidades y cómo utilizarlas.

En cuanto a este aspecto, los 3 motores cumplen con una documentación amplia y variada, disponible en varios idiomas, lo que facilita el trabajo en las empresas o particulares dedicados al desarrollo de videojuegos.

### Disponibilidad de plataformas de exportación

Estos motores proveen la capacidad de exportar los videojuegos a diferentes plataformas, según la necesidad del proyecto y su mejor viabilidad.

En este sentido, Godot es el motor que menos plataformas aúna en cuanto a su posibilidad de exportación de videojuegos, mientras que motores como Unity o Unreal permiten hasta 25 en el mejor de los casos.

### Lenguajes de programación

En cuanto a los lenguajes de programación, todos los motores tienen sus ventajas y desventajas.

Unity permite únicamente la utilización de un lenguaje, el cual es C#, lo que implica que los usuarios que quieran adentrarse con este motor tengan que aprender su codificación.

Unreal da la posibilidad tanto de utilizar C++ como lenguaje principal o, de manera alternativa, para principiantes que no estén habituados a los motores de videojuegos, los llamados blue prints, que es un lenguaje visual propio del motor que hace que sea más sencillo.

Godot es, en comparación con el resto de motores, el más versátil en cuanto a la posibilidad de la utilización de diferentes lenguajes, incluso posibilitando el uso de un lenguaje propio de scripting, llamado GDScript que, al igual que con los blue prints, está destinado a los nuevos usuarios en éste ámbito.

### Dificultad

Si bien es cierto que para adentrarse en el desarrollo de los videojuegos se recomienda tener una mínima base en programación (aún teniendo dos de estos motores un lenguaje propio para facilitar el progreso), puede suponer un reto dependiendo de la persona que se decida a empezar a utilizar un motor por primera vez.

Por lo tanto, se podría concluir que entre todos los analizados, el que más dificultad supone sería Unreal, aunque apoya al desarrollador con muchos de los tutoriales disponibles en su plataforma web.

### Recursos para el desarrollo

Por último, muchos desarrolladores indie que se lanzan a la aventura en el mundo de los videojuegos, suelen recurrir a recursos gratuitos para el modelado o diseño de su proyecto, normalmente por falta de artistas en el grupo de desarrollo.

En este sentido, dos de las plataformas (Unity y Unreal), en mayor o menor medida, disponen de una gran lista de recursos, muchos de ellos creados por los propios del sector, que ponen a disposición de los equipos de trabajo, pudiendo obtenerlos a diferentes precios, incluso algunos de ellos de manera gratuita.

Godot, por el contrario, no dispone aún de una lista de recursos disponibles para su uso.

## Definición del juego

### Idea del juego

#### Descripción del juego

El proyecto propuesto para esta práctica se podría caracterizar como un videojuego de acción. Este se compondrá de un personaje principal, que encarnará el poder del fuego, y que tendrá que hacer frente a oleadas de enemigos con poderes opuestos al protagonista, lo que harán que se debilite.

#### Subgénero y referencias a videojuegos existentes

Este videojuego se caracteriza por pertenecer principalmente al género arcade, concretamente al subgénero de Run and Gun, el cual comprende a los juegos en los que el objetivo principal consiste en esquivar a los enemigos que se aproximen al jugador, mientras se dispara.

Algunas referencias que se conocen por éste género o subgénero comprende videojuegos como el “Metal Slug”.





#### Imagen 4. Referencia al Metal Slug

Este tipo de videojuegos buscan la acción frenética por parte del jugador, que se suelen asociar a los utilizados en las máquinas arcade, intentando lograr que se desarrollen partidas de corta duración y que enganchan al jugador.

Se asemejan al estilo de los clásicos Shoot'em up, en el que se encarna a un personaje que va por tierra, diferenciándose de los clásicos en naves o algún vehículo volador.

#### Tipo de interacción juego - jugador

Muchos de los videojuegos que pertenecen al género de acción actualmente implican el manejo de un personaje principal que tendrá un objetivo que cumplir. En este caso, para finalizar los diferentes niveles propuestos en The Power of Fire, se harán uso de los controles básicos con el teclado y ratón, al ser destinado a la versión de escritorio, de forma que:

- Las teclas de control serán las encargadas del movimiento principal del personaje a lo largo del nivel, que permitirá esquivar los ataques de los enemigos.
- El ratón cumplirá la función de disparo, con lo que se eliminarán las distintas oleadas que se aproximen al jugador.

Estos controles permiten que, cualquier jugador casual habituado a jugar a juegos del estilo, sea capaz de desenvolverse de igual manera que lo hace con el resto de videojuegos, permitiendo así que la curva de aprendizaje no suponga una barrera a la hora de equilibrar la habilidad del jugador, con la dificultad propuesta.

#### Plataforma de destino

Los videojuegos que están diseñados en base al género mencionado en puntos anteriores, suele estar destinado a dispositivos móviles, por su rapidez en movimientos y dinamismo en su jugabilidad.

Por el contrario, al haber trabajado anteriormente con proyectos destinados para la versión de escritorio, se ha decidido que esta será la plataforma de destino.

De igual manera, posteriormente se podría valorar una portabilidad a dispositivos móviles, con la intención de integrarlo en el mercado móvil.

## Conceptualización

### Historia, ambientación y/o trama

Desde siempre, han convivido diversos poderes elementales, siempre intentando mantener la paz entre ellos. Durante años, han intentado mantener la armonía entre los tres, pero un día todo cambió. Ahí es donde entra nuestro protagonista, que encarnará el poder del fuego en su interior e intentará mantenerlo a toda costa, evitando que los enemigos que posean su “kryptonita” (resto de poderes) se lo debiliten.

La ambientación se caracterizará por utilizar mapas y personajes en dos dimensiones y con un estilo sencillo, que priorice la experiencia del jugador con el videojuego.

### Definición de los personajes/elementos

**Tipo:** Player character (FireWing).

**Descripción:** Este personaje será el que lleve el jugador durante toda la partida y su apariencia quedará definida por su aspecto ígneo.

Su particularidad lo diferencia de los demás en la utilización del poder de fuego para mantener alejados a los enemigos, intentando que su salud no disminuya.

**Función:** Impedir que los otros poderes, encarnados en enemigos, debiliten al personaje principal.

**Habilidades:** Su habilidad principal radica en su poder elemental, el cual podrá ser potenciado con algunos objetos que estén disponibles en el nivel.

**Tipo:** Cast character (Enemigos).

**Descripción:** Los enemigos serán los encargados de intentar que FireWing, nuestro protagonista, se debilite, hasta el punto que el poder de fuego desaparezca.

Estos variarán en su aspecto en función del poder al que representen.

**Función:** Conseguir, a toda costa, que el poder del fuego desaparezca, es decir, que el protagonista deje de existir.

**Habilidades:** Dependiendo del enemigo, tendrán comportamientos diferentes, con la intención de aportar variedad.

### Interacción entre los actores del juego

Al no haber personajes neutrales (NPC) que, como en otros videojuegos, provean al jugador de misiones u objetivos específicos a completar, la única interacción que se lleva a cabo será entre el personaje principal y los enemigos, siendo su única misión derrotarlos, para así no apagar la llama que ilumina el nivel.

## Objetivos planteados al jugador

FireWing, el protagonista, posee una llama a su alrededor, que será la que ilumine el nivel por completo. Esta llama podrá ser debilitada a medida que los enemigos de diferentes tipos de poderes ataquen al jugador, lo que aumentará la complejidad a la hora de finalizar el nivel.

Durante las partidas, se podrán obtener mejoras y objetos que permitirán al jugador mejorar su poder (aumento de cadencia, daño aumentado, etc.) o de la misma manera, impedir que los enemigos derroten al jugador, con objetos en los que se ralentice el paso de los mismos o se debilite su poder.

## Concept Art

Como se mencionaba en el apartado de “Conceptualización”, el mapa y los personajes estarán diseñados en dos dimensiones, permitiendo focalizar en mayor medida las mecánicas del juego.



Imagen 5. BoxHead

BoxHead es un juego, del estilo Shoot'em up, que se asemeja al estilo que se pretende conseguir con este videojuego, en el que los personajes se mueven sobre un plano 2D y su objetivo será sobrevivir a las oleadas.



Imagen 6. Enter The Gungeon

Enter the Gungeon cumple de igual forma con el estilo artístico que se pretende conseguir con este videojuego, personajes y mapeado plano, simulando en algunos casos el movimiento en tres dimensiones adaptado a este apartado artístico.

### Feedback acerca de la idea inicial

Durante la fase inicial, se planteó como idea inicial este concepto de videojuego, que resultaría sencillo, en el que tendría que centrarme en las mecánicas.

Como aporte a dicha idea, una de las sugerencias que se me informó fue acerca de realizar un nivel a modo de tutorial, de manera que el jugador que entre por primera vez en el juego tenga claro el concepto y cuál es la meta a conseguir durante el juego.

Para ello, se introdujo la opción tanto de un apartado de controles y de objetivo, así como un tutorial introductorio, que podrá ser omitido una vez que se haya completado por primera vez.

## Diseño técnico

### Entornos utilizados

#### Plataforma Unity

Al haber analizado todas las plataformas disponibles para el desarrollo de un videojuego en 2D del estilo arcade, como se planteó en la idea inicial, se valoró el uso de Unity [5], al haber adquirido suficientes conocimientos en el motor y poder desarrollar el proyecto sin una inversión de tiempo adicional, al no tener que aprender otro motor diferente.

Unity es más conocido por su uso en proyectos cuya finalidad es un videojuego en 3D, pero en las últimas versiones se ha visto un incremento de desarrolladores que están usando este motor para llevar a cabo sus creaciones indie, con lo que agranda su comunidad y promueve que sigan actualizando el entorno.

### Requisitos:

Requisitos	Windows	MacOS	Linux
<b>Sistema operativo</b>	Windows 7 (SP1+) y Windows 10, solo versiones de 64-bit.	Sierra 10.12.6+	Ubuntu 16.04, Ubuntu 18.04 y CentOS 7
<b>CPU</b>	X64 architecture with SSE2 instruction set support	Arquitectura x64 con soporte de instrucciones SSE2	Arquitectura x64 con soporte de instrucciones SSE2
<b>Tarjeta gráfica</b>	DX10, DX11, y con soporte para DX12	Metal-capable Intel y GPUs de AMD	OpenGL 3.2+ o Vulkan-capable, Nvidia y GPUs de AMD.
<b>Requisitos adicionales</b>	Los proveedores de hardware admitieron oficialmente los controladores	Los drivers oficialmente soportados por Apple	El entorno de escritorio Gnome se ejecuta sobre el sistema de ventanas X11, el controlador de gráficos oficial de Nvidia o el controlador de gráficos AMD Mesa. Otra configuración y entorno de usuario como stock provisto con la distribución compatible (Kernel, Compositor, etc.)

### Piskel

Cualquier videojuego que se precie actualmente no sería nada sin su apartado artístico, ya sea por los sprites utilizados para el entorno de los niveles o los personajes, sino también por la música que acompaña a dichos sprites.

Durante el inicio del proyecto, se valoró la opción de diseñar la gran mayoría de los sprites, si no su totalidad, de manera propia sin hacer uso de contenido gratuito externo.

Debido a los problemas acontecidos (coronavirus) durante el transcurso del período para realizar el TFM, se optó por adquirir los assets en la plataforma gratuita OpenGameArt [3], la cual aúna una gran cantidad de recursos, entre los que se encuentran efectos sonoros, dibujos de personajes, de armas, niveles, etc.

Para llevar a cabo las animaciones de los assets gratuitos, puesto que eran estáticos y no tenían, se utilizó la herramienta Piskel [2], que está disponible en el navegador de manera gratuita.

Al usar el estilo pixel-art, mencionado en puntos anteriores, se buscó una plataforma que fuera viable para realizar dichas animaciones, por lo que al no conocer en profundidad otro software como Adobe Photoshop o similares, se optó por usar esta herramienta, que cumple con lo requerido en el apartado artístico.



Imagen 7. Logo de Piskel

Los requisitos para utilizar esta plataforma no son específicos, como pueden ser los del motor Unity, puesto que se utiliza en cualquier navegador.

### Microsoft Visual Studio 2019

Como se explica en el punto anterior llamado “Plataformas de desarrollo”, otros entornos de desarrollo para videojuegos permiten su creación a través de una forma más sencilla y visual, que no sustituye al método tradicional de programación, pero si complementa ésta última.

Unity posee en su tienda personal algunos assets que nos permiten esta funcionalidad, puesto que por defecto no viene integrada. En este caso, no se ha utilizado éste método, ya que en muchas ocasiones algunos comportamientos no están controlados a través de ese tipo de programación más visual.

Para ello se ha utilizado C#, lenguaje necesario para crear un proyecto en Unity que, junto al entorno provisto por Microsoft Visual Studio Community nos permite

gestionar de manera sencilla todos los scripts que se utilicen en el videojuego gracias a su integración con Unity.

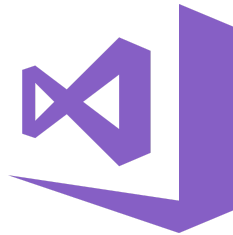


Imagen 8. Logo de Microsoft Visual Studio

## Sprites

El motor utilizado para desarrollar el proyecto dispone de una tienda que ofrece recursos tanto gratuitos como de pago, de manera que cualquier creador tenga a su disposición todo tipo de assets. En caso de no formar un gran equipo de desarrollo, esto facilita el trabajo a la hora de montar un proyecto.

Por lo tanto, para el proyecto se han usados los siguientes recursos:

- **Standard Assets**, un paquete que contiene assets, scripts y escenas de ejemplo que pueden usarse para iniciarse en un proyecto, sobre todo para los programadores que se estén iniciando en este mundo.  
<https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-2017-3-32351>
- **16x16 Dungeon Tileset**, que sirvió como base para todos sprites de los enemigos que se encuentran en el juego para posteriormente animarlos.  
<https://0x72.itch.io/16x16-dungeon-tileset>
- **Kyrise 16x16 Icon Pack**, sprites de los objetos de mejora durante el juego.  
<https://opengameart.org/content/kyrises-free-16x16-rpg-icon-pack>

Por otro lado, las animaciones creadas de manera personal se alojan en la plataforma de Piskel, donde se pueden realizar sin límite de espacio, lo que facilita ahorrarnos un lugar adicional donde guardarlas.

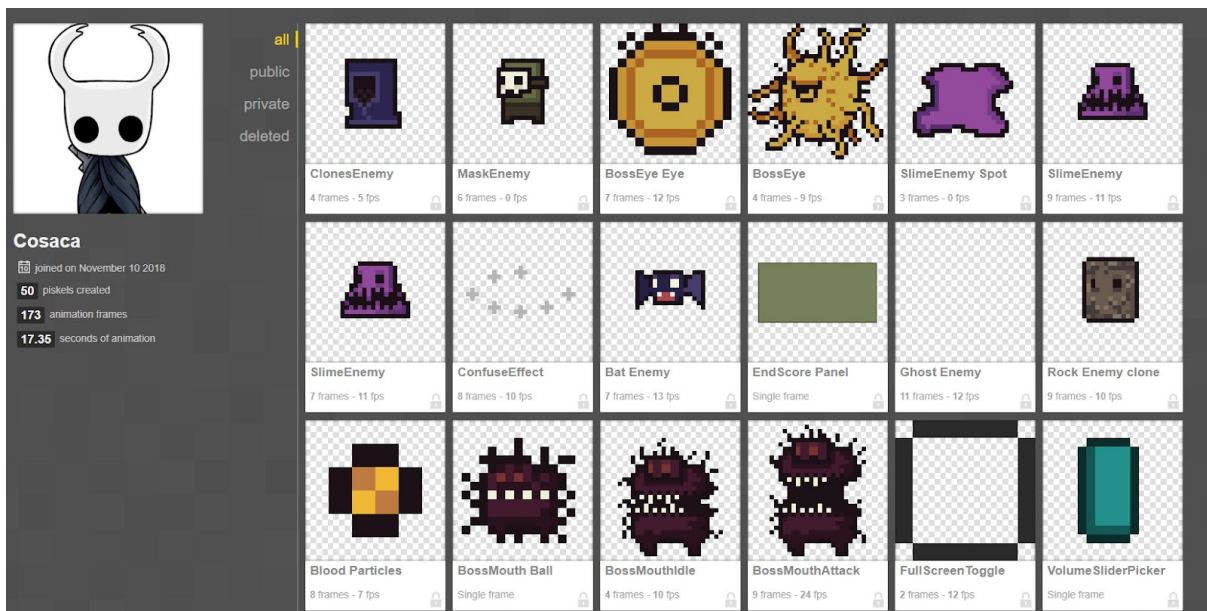


Imagen 9. Página principal de perfil en Piskel

Cada uno de ellos se tuvo que adaptar al tamaño que tiene la cuadrícula en Unity, puesto que las unidades son un tanto confusas a la hora de importar los sprites de los personajes, siendo su fuerte el estilo pixel art.

Las imágenes o frames de cada animación se encuentran entrando en cada uno de ellos, como se muestra en la imagen a continuación.

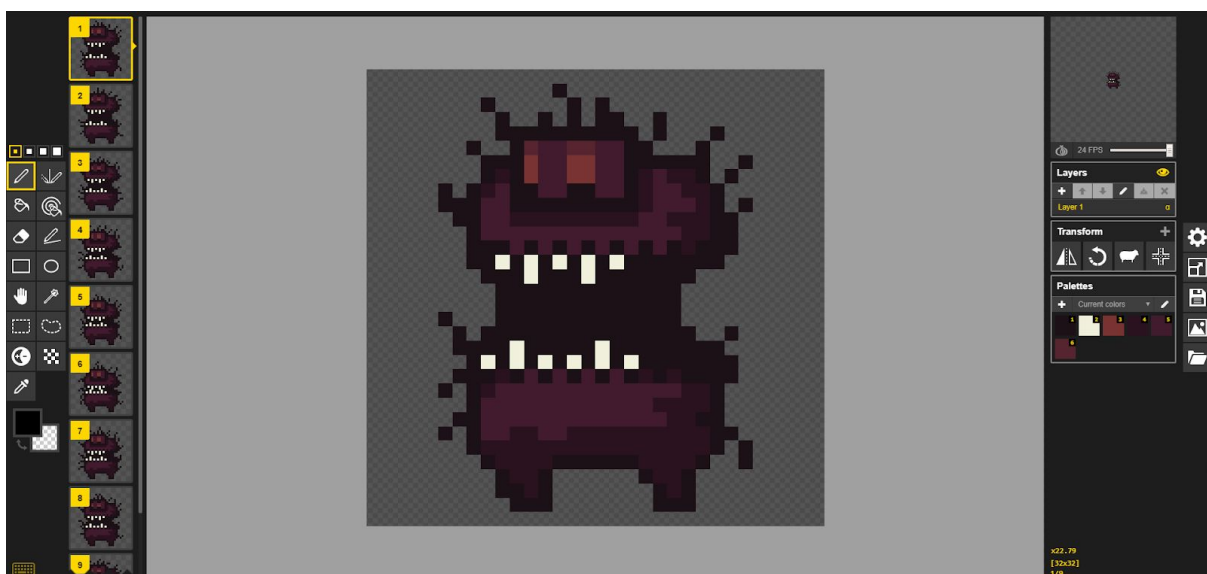


Imagen 10. Ejemplo de animación de enemigo



A la izquierda se muestran los diferentes frames, que después se agruparán en una única hoja, también llamado *atlas*, para posteriormente importarlos en Unity y separarlos para poder realizar la animación correctamente.

Existen otras herramientas de pago, como puede ser Aseprite (muy utilizada entre desarrolladores indie), pero al buscar una solución que fuera gratuita, se encontró Piskel, que ofrece una gran cantidad de posibilidades a la hora de crear sprites siguiendo este estilo.

## Sonidos

Por último, los efectos de sonido empleados en el videojuego, tanto de ambiente como los que corresponden a los disparos, enemigos, etc., se obtuvieron de la plataforma *Freesound* [4]. Este sitio web aloja multitud de música que puede ser utilizada tanto para proyectos que involucren videojuegos, como películas, cortos, etc.

Existe una gran comunidad detrás de esta página, y muchos de su contenido es libre de derechos de autor, por lo que es una fuente importante de recursos sonoros. De esta forma, no es necesario un desembolso de dinero para poder obtener sonidos de buena calidad para un proyecto.

## Arquitectura del juego

La creación de un videojuego en Unity conlleva una serie de componentes para que todo ensamblado formen el proyecto en su conjunto. Para ello, este motor distingue entre diferentes elementos que son necesarios:

### **Escena de inicio y controles**

Esta será la que contenga un menú principal que muestre el título del juego, con las opciones principales para acceder al juego, configuración del juego (resolución, volumen, etc.) y salir del juego.



Imagen 11. Jerarquía de la escena inicial y controles

Como se observa en la imagen superior, ambas escenas son bastante sencillas en cuanto a la cantidad de elementos que la componen.

Por un lado un canvas que engloba los botones y menús disponibles, el jugador y grid que se utiliza en uno de los niveles como fondo de pantalla y la música.

Por otro lado, en la escena de los controles, únicamente dos paneles que contienen la información necesaria con respecto a los controles y el objetivo de la partida.

### Escena de tutorial

Como se planteó al finalizar la entrega del primer documento de la idea inicial, podría ser necesario un nivel a modo de tutorial introductorio, que permitiera al jugador adecuarse a los controles y saber cuál es su objetivo.



Imagen 12. Jerarquía de la escena del tutorial

Muchos de los tutoriales de los videojuegos actuales, se basan en textos informativos durante el nivel. Para este caso, se ha diseñado únicamente un nivel que realice esta función, de manera que se irán mostrando cuadros de texto que guiarán al jugador en la interfaz que se mantendrá durante toda la partida, así como un enemigo de ejemplo para que observe como son los controles “in situ” al respecto del videojuego.

## Escenas de niveles

Puesto que este videojuego consiste en derrotar enemigos que se irán aproximando al personaje durante varias oleadas, los tres niveles diseñados para el proyecto final se componen de una jerarquía similar a la mostrada en la imagen inferior.

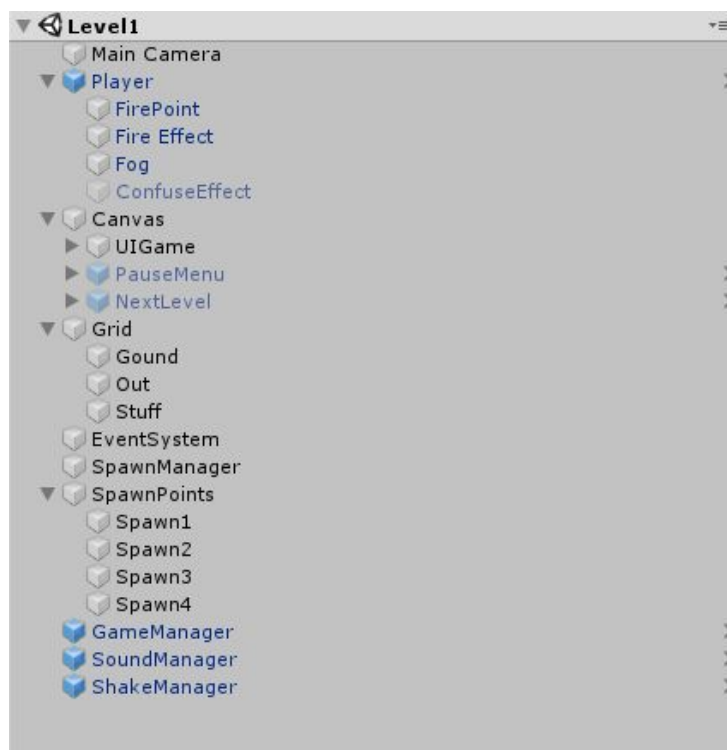


Imagen 13. Jerarquía de la escena de niveles

Entre los elementos más importantes en la escena, podemos encontrar el personaje principal, con su correspondiente comportamiento, el canvas con la información relevante de cara al jugador (puntuación, tiempo, etc.), la cuadrícula del nivel, el gestor de oleadas y de vibración de la cámara y, por último, los diferentes puntos en los que pueden aparecer los enemigos en el nivel.

## Escena de puntuación final

Una vez que se finaliza la partida, el jugador podrá observar la cantidad de enemigos que ha eliminado durante los niveles al igual que la puntuación final que ha conseguido, que estará calculada en base al tiempo que se ha consumido para completar todas las oleadas (menos tiempo, más puntuación).

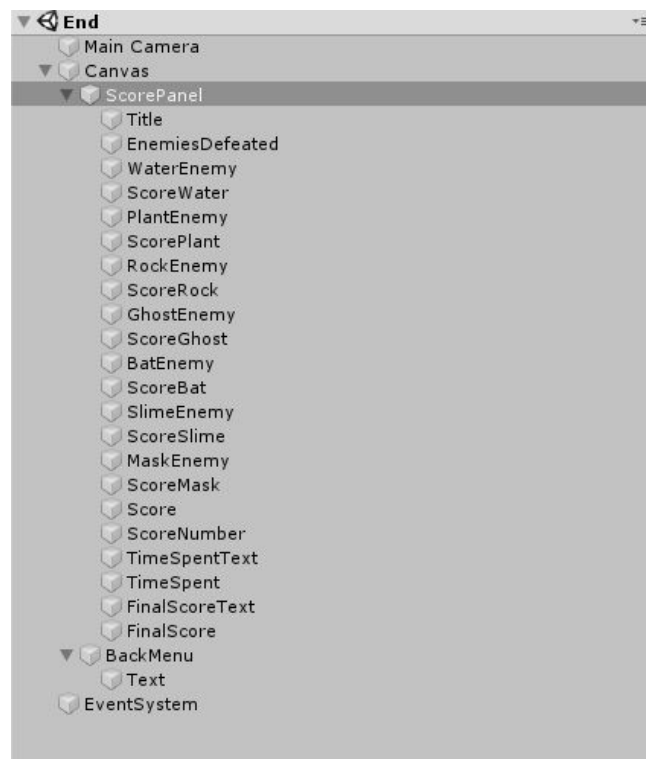


Imagen 14. Jerarquía de la escena final

## Inteligencia artificial de enemigos

Puesto que se han diseñado diferentes enemigos en las oleadas, para no crear una experiencia muy repetitiva, se han utilizado diferentes comportamientos entre ellos, aún siendo algunos similares entre sí.

Se intentó usar el componente NavMesh disponible en Unity, que utilizamos en asignaturas anteriores como “Programación en 3D”, de forma que los enemigos usaran algoritmos de búsqueda mínimo en el camino entre dos puntos, aún habiendo obstáculos. Al no poderse usar en 2D, se optó por una IA más sencilla.

Se podrían agrupar de la siguiente forma:

- Enemigos que siguen al jugador, donde únicamente persiguen al personaje hasta dar con él y hacerle daño.
- Otros que buscan puntos aleatorios en el nivel, de manera que su función sea molestar al jugador durante esa oleada.
- Y por último, los que disparan al jugador, pero no siguen un patrón específico en su camino.

En líneas futuras del trabajo, se intentará mejorar su comportamiento en cuanto a los obstáculos que puedan haber de por medio en el nivel, así como la variedad de enemigos, dado que un juego de este estilo suele destacar por su variedad, de forma que enganche al usuario.

## Diseño de niveles

Todo videojuego que ofrece un progreso en la historia de cara al jugador, suele comprender una serie de niveles, de forma directa o indirecta, con la intención de que sea atractivo y sea un aliciente para que al usuario le llame su atención.

Los juegos arcade, como éste que se describe para este proyecto, no suelen tener una estética llamativa, en cuanto a efectos visuales, pero destacan por su frenetismo y acción constante. Esto hace que su constante repetición, en busca de la mejor puntuación, no suponga un mayor esfuerzo sino todo lo contrario, ambición para ser el mejor jugador.

Por lo tanto, para este videojuego, se han diseñado tres niveles, que se diferencian en su disposición de obstáculos.

### Nivel 1



Imagen 15. Diseño del nivel 1

Este nivel tiene la misma disposición que el utilizado para el tutorial, por lo que resulta sencillo esquivar a los enemigos al no haber muros ni objetos que limiten el movimiento.

Durante el primer planteamiento, se pensó si utilizar un nivel en el que el jugador pudiera desplazarse libremente (teniendo en cuenta igualmente unos límites en el mismo), pero al final y para facilitar el manejo de los enemigos se modificó para tener siempre la cámara centrada en el nivel, que comprenderá el tamaño que se muestra en la imagen superior en todos los niveles.

## Nivel 2

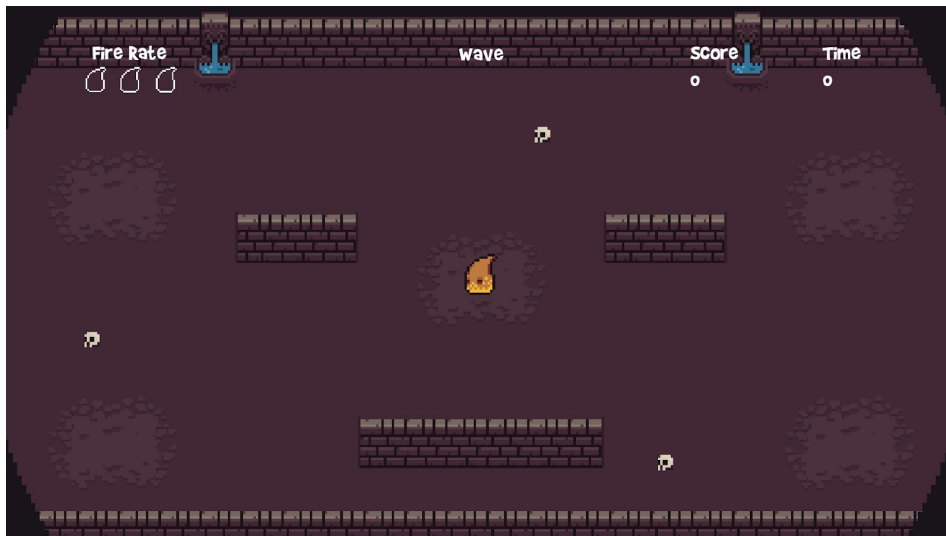


Imagen 16. Diseño del nivel 2

Al igual que el nivel 1, los spawns se encuentran en las cuatro esquinas que tienen un color diferente al resto del suelo del nivel, indicando que ahí será el lugar donde aparezcan los enemigos.

De la misma manera, el lugar central se usará como spawn de los objetos que puede obtener el personaje, con lo que no se solapará con el resto de puntos de aparición.

Este nivel tiene la peculiaridad que los muros colocados únicamente afectan al personaje, aumentando así el grado de dificultad para superar las oleadas, lo que supondrá un reto para el jugador.

## Nivel 3

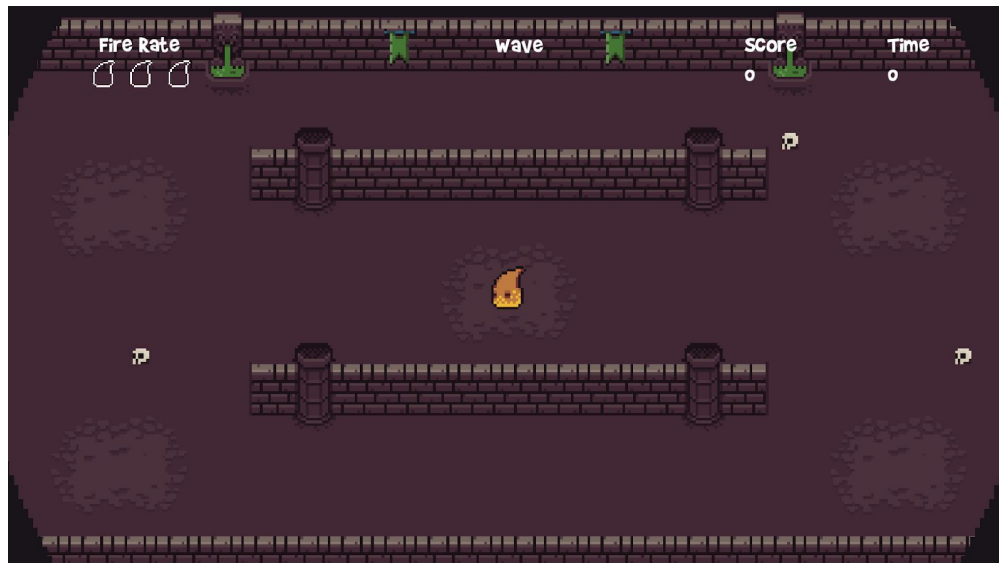


Imagen 17. Diseño del nivel 3

Por último, el nivel 3 que tendrá ocupado a su vez parte del nivel con obstáculos, pero esta vez diseñados para el jefe de éste, puesto que se usarán los 4 pedestales a los lados para su comportamiento.

Una vez completado este nivel, el jugador será conducido a la pantalla de estadísticas, donde se mostrará el número de enemigos eliminados durante la partida, a excepción de los jefes.

La puntuación estará basada en los puntos obtenidos de cada enemigo eliminado, así como el tiempo invertido para completar con éxito todos los niveles, que se restará a la puntuación global. Por lo tanto, a menos tiempo, más puntuación.



# Manual de usuario

## Requisitos

**Sistema operativo:** Windows 7 (SP1+) y Windows 10.

**CPU:** con soporte de instrucciones con arquitectura de x64 o x86.

**GPU:** tarjeta gráfica con capacidad para DX10, DX11 o DX12.

**Disco duro:** almacenamiento requerido de al menos 150Mb.

## Instrucciones de uso

Desde hace un tiempo hasta ahora, cada vez se valora en mayor medida la accesibilidad y usabilidad de las aplicaciones, debido a la cantidad de personas diferentes que pueden llegar a probar las mismas, ya sea gente con algún tipo de discapacidad motriz o de algún otro tipo.

Por ello, se ha de desarrollar un software que pueda ser accesible a todos los usuarios que quieran usarla.

Una de los detalles que se valoran en cualquier videojuego es que se sepa cuál es el objetivo final y una pantalla en la que se puedan observar los controles en todo momento.

En primer lugar, para ejecutar este proyecto no será necesario instalar ningún tipo de software adicional, como puede ser Unity, ya que sino no sería posible ejecutarlo a menos que se tuviera dicho software.

En cuanto a las instrucciones de uso, en la pantalla principal del juego, se ha habilitado una opción para poder ver los controles en todo momento, así como el objetivo principal de la partida.



Imagen 18. Pantalla inicial de The Power of Fire

La pantalla de controles se observa en la imagen inferior, con lo que resume de manera concisa lo esencial que debe saber cualquier usuario que quiera adentrarse a probar el videojuego.



Imagen 19. Controles de The Power of Fire

De la misma forma que se muestran los controles y objetivo al jugador, será posible realizar algunas configuraciones mínimas en cuanto a resolución del juego, volumen y si se desea poner en modo pantalla completa o no.

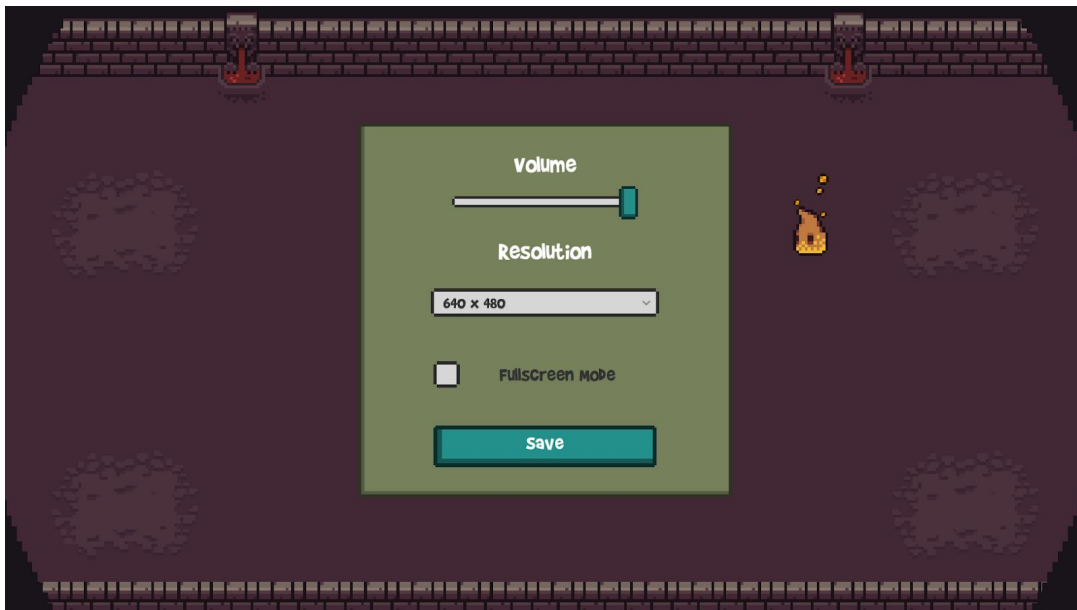


Imagen 20. Configuración de The Power of Fire

Una vez configurado y visto todo lo anterior, ya será posible acceder al tutorial del juego, pudiendo así comenzar una nueva partida.

## Tests realizados

Muchas veces, los desarrolladores, durante el proceso de creación y producción de un videojuego, no ven detalles o fallos simples, que al probar una persona ajena al proyecto puede detectar, así como consejos que se pueden dar como *feedback* de cara a mejorar el producto final.

Para este proyecto, no se ha hecho un estudio específico en cuanto al testing del videojuego en otras personas, pero han habido detalles que han servido para corregir la experiencia del jugador y, por ello, que el usuario final obtenga un proyecto refinado.

### **Velocidad del juego**

Durante las primeras etapas del desarrollo, se planteó que si se quiere hacer un juego de este estilo, se tiene que pensar en frenetismo, velocidad, que es lo que requiere el género arcade.

Al principio, al tener esa variación durante la partida de la velocidad de disparo, que puede verse aumentada al conseguir un objeto que la mejora, se decidió imponer una velocidad un tanto baja, de cara a ver un aumento significativo con dichos objetos.

Algunos de las personas que pudieron probar el juego en su fase temprana de desarrollo, sugirieron que esta velocidad fuera más alta desde un principio, por lo que se tuvo en cuenta de cara a subir la cadencia desde los inicios de la partida.

### **Traducción a inglés**

Es bien sabido que en el mercado de los videojuegos el idioma nativo que suele tener un proyecto por defecto es el inglés, debido a la diferencia de demanda en otros países en comparación con España, en cuanto a las compras de videojuegos.

De cara a tener una versión que pueda ser vista por cualquier persona, se decidió cerca del final del desarrollo, traducir el juego en su totalidad a inglés, ya que no supone una carga de trabajo adicional y los textos, en menor o mayor medida, pueden ser entendidos por cualquier persona que tenga un nivel mínimo de inglés.

### **Diseño de niveles**

Para que este juego resultara atractivo y permitiera su re jugabilidad, se intentó que hubiera variedad en las oleadas de enemigos. En edades tempranas del proyecto, los niveles carecían de contenido, puesto que no se habían añadido obstáculos, como los que se observan en puntos anteriores.

Esto hizo que algunas personas que probar el juego comentaran la posibilidad de añadir algún elemento diferenciador entre los niveles, por lo que se diseñó los muros y pedestales que, a su vez, ayudan a implementar nuevos comportamientos para los jefes de los niveles.

## Conclusiones

The Power of Fire ha sido un proyecto que se ha desarrollado con la intención de desarrollar un producto desde cero, con los conocimientos adquiridos durante los dos años de máster.

Ha supuesto una inversión de tiempo que no había tenido hasta ahora en una práctica de una de las asignaturas, puesto que se diferencia en que en este videojuego todos los ámbitos, artísticos y técnicos, tenían que correr a cargo de una sola persona, lo que supone un reto a superar.

La idea del proyecto inicialmente, podría parecer sencilla, pero con el paso del tiempo y de las diferentes tareas que se iban realizando, iban surgiendo problemas o ideas nuevas que se podían implementar en el videojuego, que no estaban marcadas en la planificación inicial.

Esta planificación no siempre se puede seguir como se ha pactado en un principio, ya que pueden surgir imprevistos entre medio que hagan que el tiempo de desarrollo se vea mermado.

En este sentido, en mi caso, como menciono en puntos anteriores, los sprites que se iban a diseñar desde un principio, no se pudieron continuar, debido a que no se disponía del iPad con el que se comenzaron, por lo que se optó por el estilo pixel art, más sencillo que este último.

Muchas de las ideas, que se han planteado durante el proceso, se han dejado a un lado, para priorizar lo principal que se tenía planteado inicialmente, que corresponden a las líneas futuras del proyecto:

- Cambiar el modo en el que se muestra la vida del personaje, utilizando las luces provistas por Unity en el modo 2D, y no la actual “máscara” alrededor, que se mejoran en versiones posteriores a la usada en el proyecto, así como aprender a usar *shaders* y añadir efectos que mejoren la interacción con el videojuego.
- Nuevos tipos de enemigos, que por tiempo no se han podido diseñar, ya que la animación de los mismos consume más tiempo del esperado, así como el aprendizaje que se necesita para animar estos enemigos.
- Más objetos que afecten a la experiencia de juego, ya sea que añadan ventaja al jugador, como desventajas para el mismo.

- Variedad en el diseño de niveles, de manera que pueda haber un mundo abierto, y estos sean accesibles desde diferentes partes del mapa, lo que lo haría más atractivo.
- Añadir un modo de juego que solo fuera de jefes finales, que supondría un reto para los jugadores más experimentados y que hayan completado el juego previamente.

## Glosario

**Sprites:** dentro de los recursos, los que se componen de imágenes correspondientes a personajes o entornos.

**Assets:** nombre que reciben los recursos utilizados un proyecto en Unity.

**IA:** inteligencia artificial, comportamiento que tienen los enemigos en este juego, es decir, cómo es su movimiento o ataque una vez que están en escena.

**Feedback:** opiniones recibidas acerca de un tema, de cara a usarlas para posteriormente mejorar, en este caso, sugerencias que pudieran mejorar la experiencia de usuario.

**Testing:** pruebas realizadas a la hora de jugar al videojuego, para encontrar posibles mejoras o fallos.

**Bug:** fallos encontrados durante la partida.

**C#:** lenguaje de programación orientado a objetos que está desarrollado por Microsoft.

**Shader:** son pequeños scripts que contienen los cálculos matemáticos y algoritmos para calcular el Color de cada píxel procesado, en función de la entrada de iluminación y la configuración del Material.

**Shooter:** engloba a un gran número de subgéneros que mayormente se centra en un personaje que dispone de un arma que puede ser utilizada durante la partida.

**Atlas:** en un proyecto en Unity 2D, usa los sprites y otras imágenes para crear los elementos visuales en la escena.

## Bibliografía

[1]. ¿Sabes lo que son los videojuegos AAA o triple A?  
<https://fxanimation.es/que-son-los-videojuegos-triple-aaa/>  
Consultado en mayo de 2020.

[2]. Plataforma Piskel  
<https://www.piskelapp.com/>  
Consultado en marzo de 2020.

[3]. Plataforma OpenGameArt  
<https://opengameart.org/>  
Consultado en abril de 2020

[4]. FreeSound  
<https://freesound.org/>  
Consultado durante todo el proyecto.

[5]. Unity Engine  
<https://unity.com/es>  
Consultado durante todo el proyecto.

**Enlace al pitch del videojuego final:**  
<https://www.youtube.com/watch?v=riARIAzqDaw>