



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Graduado en Ingeniería del Software

**Desarrollo de un videojuego serio como apoyo a la enseñanza de
conceptos y obras literarias de las épocas Edad Media,
Renacimiento y Barroco**

**Development of a serious video game as support for teaching
concepts and works of literature from the period of Middle Age,
Renaissance and Baroque**

Realizado por

Sergio Palacios Cabrera

Tutorizado por

Antonio José Fernández Leiva

Departamento

Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA

MÁLAGA, SEPTIEMBRE DE 2019

Fecha defensa: de de 2019

Fdo. El/la Secretario/a del Tribunal



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA

Resumen

Este Trabajo de Fin de Grado consiste en el desarrollo e implementación de un videojuego serio que sirva como método de enseñanza para estudiar y repasar conceptos de la literatura española en la Edad Media, el Renacimiento y el Barroco. Para ver si el objetivo es exitoso, se han realizado una serie de pruebas con diversos usuarios y se ha dado a probar a alumnos, de 3º y 4º de la ESO, que han estudiado estos conceptos recientemente.

Palabras clave: videojuego, juego serio, literatura, Unity, Scrum, puzzles, eventos.

Abstract

This Final Year Project consists of the development and the implementation of a serious videogame that can serve as a teaching method to study and review concepts of Spanish literature in the Middle Ages, Renaissance and the Baroque era. In order to know if the goal is completed, tests will be run with different users, and subsequently to students, of 3º and 4º of ESO, that have studied these concepts recently.

Keywords: videogame, serious game, literature, Unity, Scrum, puzzle, events.

Índice

Resumen	1
Abstract	1
Índice	3
1. Introducción	5
1.1 – Motivación.....	5
1.2 – Objetivos	5
1.3 – Estructura de la memoria	6
2. Antecedentes y los videojuegos	9
2.1 – Juegos de referencia	9
2.2 – Videojuegos serios.....	11
2.3 – Videojuegos serios aplicados a la educación.....	12
2.4 – Fases del desarrollo.....	12
2.5 – Metodologías ágiles y Scrum.....	14
2.6 – Entorno de desarrollo: Unity.....	17
3. Problema y diseño de la solución	19
3.1 – Diseño inicial	19
3.2 – Jugabilidad	21
3.3 – Flujo de juego.....	21
3.4 – Diseño de niveles	21
3.5 – Interfaz.....	22
4. Aplicación de Scrum a la solución	27
4.1 – Especificaciones iniciales y requisitos	27
4.2 – Evolución del proyecto.....	29
5. Implementación de la solución	37
5.1 – Diseño de clases	37
5.2 – Inteligencia artificial de los enemigos	44
5.3 – Interacción del usuario con el juego	47
5.4 – Flujo del juego.....	48
5.5 – Componentes del proyecto	49
6. Pruebas con usuarios	51
6.1 – Pruebas durante el desarrollo.....	51
6.2 – Pruebas con el juego completo.....	53

7. Conclusiones.....	57
7.1 – Objetivos cumplidos	57
7.2 – Aprendizaje personal	58
7.3 – Limitaciones y problemas	59
7.4 – Posibles mejoras	60
Referencias.....	63
A1 – Manual de Instalación.....	67
Forma de instalación y requerimientos:.....	67
A2 – Documento de Diseño de Juego (GDD).....	69
1 – Descripción	69
2 – Características técnicas	70
3 – Experiencia de juego	71
4 – Jugabilidad	71
5 – Flujo de juego	75
6 – Diseño de niveles.....	76
7 – Cámara, controles e interfaz.....	79
8 – Narrativa	92
9 – Ambientación.....	92
10 – Diseño de arte	92
A3 – Imágenes del juego.....	95

1. Introducción

Este apartado muestra los contenidos de todo el trabajo, la motivación que ha llevado a ser realizado, cuáles son los objetivos a alcanzar y que estructura va a tener.

1.1 – Motivación

Los videojuegos representan el área de entretenimiento que más ha crecido en los últimos años a nivel mundial y este campo ya genera más actividad comercial que el del cine y la música juntos. El videojuego está además reconocido por el Parlamento español como Industria Cultural desde el año 2009, lo que se traduce en una mayor cobertura a todos los niveles. Adicionalmente está demostrado que los videojuegos producen una mecánica positiva tanto a nivel personal como a nivel socio-económico en un sentido más general de sociedad, que va más allá del ocio, y pueden provocar efectos positivos en sectores transversales y en campos de aplicación que pueden afectar las vidas de miles de personas y contribuir al progreso y la mejora social.

Debido al auge de los videojuegos en el mundo actual, en los últimos años han ido surgiendo diversas categorías de videojuegos cuyo objetivo llega más allá de entretener al jugador, intentando enseñar conceptos nuevos o ayudar a reforzar datos que ya sabíamos. Este tipo de videojuego es a lo que se llama juego serio.

En este contexto, este proyecto busca desarrollar un videojuego serio, desde cero, donde el jugador deberá aprender sobre la literatura y ciertas obras literarias de las épocas de la Edad Media, el Renacimiento y el Barroco. El juego será una mezcla de acción con resolución de puzles, de tal forma que el jugador deberá enfrentarse a ciertos enemigos incultos, los cuales tendrá que vencer para convertirlos en cultos, además de resolver ciertos puzles que le harán preguntas sobre las obras literarias más importantes de cada época.

1.2 – Objetivos

El objetivo principal de este trabajo de fin de grado es el desarrollo de un videojuego serio para PC que busca enseñar al jugador sobre los conceptos

principales de la literatura de las épocas de la Edad Media, el Renacimiento y el Barroco, así como sus obras más importantes. Estos conceptos que se quieren enseñar están enfocados a los cursos de 3º y 4º de la ESO. El videojuego será diseñado y desarrollado completamente usando la metodología ágil *Scrum*, en la cual se pondrá especial énfasis en las etapas de desarrollo cubriendo las especificaciones iniciales y los requisitos, toda la fase de diseño, la implementación del código de cada componente, y la fase final de prueba del producto.

1.3 – Estructura de la memoria

La estructura que esta memoria sigue es la misma que la estructura que se hace a la hora de realizar un juego cualquiera:

1 – Introducción. En este punto se explica cuales son las motivaciones, los objetivos y como va a estar realizada la memoria. Sirve para poner en contexto al lector.

2 – Antecedentes y los videojuegos. En este punto se va a explicar en qué tipos de juegos nos basamos para hacer este proyecto y cuáles tomamos como referencia, al igual que se va a explicar lo que son los videojuegos serios. Se habla sobre los pasos iniciales que tenemos que seguir para desarrollar un juego, así como la metodología a seguir y la herramienta a usar.

3 – Problema y diseño de la solución. En este punto se va a explicar cómo ha sido diseñado el juego, cómo se ha obtenido la idea y se ha pulido hasta tener las mecánicas que queremos y cómo va a ser el juego.

4 – Aplicación de *Scrum* a la solución. En este punto se explica cómo hemos aplicado *Scrum* al proyecto, mostrando todos los pasos y reuniones hechas en el proceso de desarrollo, así como los sprints realizados.

5 – Implementación de la solución. En este punto se va a explicar el cómo se ha implementado el juego, así como el proceso de hacer el UML y los diferentes diagramas necesarios para desarrollar un software.

6 – Pruebas con usuarios. En este punto se va a mostrar los resultados de las diversas pruebas que se han realizado con los usuarios. Para tener

resultados más variados, se ha probado el juego con usuarios del público objetivo y usuarios fuera del público objetivo.

7 – Conclusiones. En este punto se va a explicar los resultados finales del proyecto y si se han cumplido los objetivos planteados o no.

2. Antecedentes y los videojuegos

En este apartado se va a mencionar y describir aquellos conceptos que deben ser comprendidos inicialmente por el lector para entender el resto del proyecto.

2.1 – Juegos de referencia

A la hora de hacer un videojuego lo primero es conocer otros videojuegos en los que basarte o en los que coger ideas. En este juego hemos hecho lo mismo. A continuación vienen algunos ejemplos de juegos que hemos tomado como referencia:

2.1.1 – The Binding of Isaac

The Bindgin of Isaac [[The Binding of Isaac Wiki, 2019](#)] es un juego *roguelike* de mazmorras, es decir, dos partidas no son iguales debido a que se generan proceduralmente las mazmorras cada vez que juegas, además de que los propios objetos son aleatorios. Uno de los puntos fuertes de este juego es el hecho de que podemos combinar dichos objetos entre sí para poder obtener objetos más poderosos y raros.

Sin embargo, este no es el punto en el que nos vamos a fijar para tomar referencias. *The Binding of Isaac* tiene un sistema de combate peculiar en el que vemos al personaje desde una posición cenital y nos movemos en los cuatro ejes y disparamos para uno de estos lados. Además, los enfrentamientos en dicho juego pueden llegar a ser hasta caóticos en algunos momentos debido a la de enemigos y proyectiles que hay. Tanto el sistema de combate como estos enfrentamientos más caóticos nos servirán como guía para las diferentes fases de nuestro juego.



Figura 2.1 - The Binding of Isaac

2.1.2 – Path of Exile

Path of Exile [Path of Exile Wiki, 2019] es lo que se conoce como un RPG de acción [Action RPG Wiki, 2019] en todo su esplendor. Es un juego que enfatiza en el combate a tiempo real donde puedes controlar a tu personaje, y todas las acciones o habilidades que hace, en todo momento. El combate de *Path of Exile* es muy frenético y con muchos enemigos y, conforme vamos avanzando subiendo de nivel, los enfrentamientos se vuelven bastante caóticos debido a la cantidad grandísima de enemigos que nos enfrentamos y al número tan grande de habilidades que tenemos a nuestra disposición.

Al igual que *The Binding of Isaac*, *Path of Exile* tiene un sistema de objetos muy amplio y, además, un sistema de habilidades y puntos de habilidades que le hacen famoso.



Figura 2.2 - Path of Exile

2.1.3 – The Legend of Zelda

The Legend of Zelda [[The Legend of Zelda Wiki, 2019](#)] es uno de los juegos más famosos en la historia de los videojuegos, con más de 10 juegos diferentes en la saga. *The Legend of Zelda* es un juego de aventura en el que eres Link y tienes que rescatar a la princesa Zelda del malvado Ganon. Es un juego de aventura clásico en el que conforme vas pasando niveles vas desbloqueando nuevos objetos y mejoras para tu personaje. Este es el aspecto que más vamos a usar como referencia en el juego.



Figura 2.3 - The Legend of Zelda

2.2 – Videojuegos serios

Un videojuego serio [[Serious game Wiki, 2019](#)] es un juego diseñado especialmente con un objetivo distinto de simplemente divertir, normalmente suele tener un objetivo educativo o del estilo. El género de los juegos serios tiene muchas similitudes con géneros como la simulación, dado que ambos quieren mostrar un punto más realista y quieren enseñar algo que ocurre en la realidad.

A lo largo de la historia los videojuegos serios han sido usado en diversos campos y para diferentes ocasiones. Desde los años 2000 estos han visto un aumento en la cantidad de ellos y en los campos en los que se aplicaban, pero no fue hasta los años alrededor de 2010 cuando realmente empezaron a surgir grandes juegos serios que han sido exitosos. Debido a esto, a día de hoy los juegos serios son un aspecto a considerar en todo momento a la hora de desarrollar un

videojuego. Algunos ejemplos de juegos serios a lo largo de la historia son: 10 juegos serios [[Growthengineering 10 serious games, 2019](#)] y juegos serios para aprender [[Game learn Serious games, 2019](#)].

2.3 – Videojuegos serios aplicados a la educación

Como ya se ha comentado anteriormente, los videojuegos son cultura y son un método muy sencillo para aprender. Pero no solo para aprender conceptos específicos, sino para trabajar la comunicación con otras personas, el trabajo en equipo, la coordinación entre lo que ven y lo que hacen, mejorar la comprensión lectora y su propia escritura, etc.

Todas estas características se podrían enseñar mediante otro tipo de actividades o estudios posiblemente, pero el punto fuerte de hacerlo mediante los videojuegos es que los alumnos se diviertan mientras lo hacen. No ven el jugar al Portal como aprender a trabajar en tres dimensiones y desarrollar la capacidad para resolver problemas, sino como un método divertido en el que pasar el tiempo. Por esto los videojuegos serios son cada vez más populares entre las aulas y dan buenos resultados. Algunos ejemplos de porque son buenos los videojuegos: artículo de EL PAÍS [[EL PAÍS Economía, 2019](#)] y este hilo de Twitter [[Twitter Clawster, 2019](#)].

2.4 – Fases del desarrollo

El desarrollo de un videojuego es muy parecido a cuando desarrollamos una aplicación software normal, tenemos que desarrollar la idea o necesidad y pulirla, luego plantear como se va a resolver dicha idea y que equipo necesitamos, después toca implementar el software que haga falta para realizarlo, y al final se hacen pruebas con los usuarios objetivos para ver si el juego funciona y donde puede ser cambiado para mejorarlo. Estas últimas dos partes al final se convierten en un proceso cíclico en el que implementamos partes del juego y lo probamos, y si no funciona como queríamos vamos para atrás e implementamos otras, y así hasta llegar al objetivo que queremos o hasta que nos quedemos sin tiempo/dinero.

Si nos fijamos, este proceso es muy parecido al proceso de hacer un Trabajo de Fin de Grado. Por ello, este trabajo de fin de grado se ha planteado de la

misma forma. A continuación vienen explicadas cada parte un poco más en detalle:

2.4.1 – Documento de diseño de juego

Lo primero a la hora de hacer un videojuego es diseñar la idea. Esto se hace en diversos pasos iterativos. Primero se empieza con un documento de concepto de juego (GCD) en el que se ponen por escrito las principales características de un juego, así como las mecánicas principales y los llamados USP, es decir, los puntos únicos de venta (Unique Selling Points).

Una vez que tenemos dicho documento, a base de darle vueltas a las ideas y de varias iteraciones, sacamos el documento de diseño de juego (GDD). En este documento viene explicado todo lo que hay en el juego, y como se va a plantear y desarrollar cada parte del juego. No obstante, muchas veces durante el proceso de desarrollo de un videojuego se vuelve hacia atrás en los conceptos definidos en el GDD y se cambian algunos.

2.4.2 – Planteamiento del proceso de desarrollo

Una vez que tenemos la idea más o menos definida tenemos que elegir como vamos a repartir el trabajo y que metodología vamos a seguir. En este punto se reparte el trabajo por departamentos y por prioridad de necesidad. Los artistas conceptuales son los primeros que empiezan a trabajar para poder tener una idea del arte que va a tener el juego, los personajes, los escenarios y que los animadores y artistas 3D puedan empezar a trabajar cuanto antes. Por el lado de la programación, los arquitectos de software son los que primero empiezan, detallando cuál va a ser la estructura que va a tener el código y como se va a hacer.

Respecto a la metodología, debido a la gran posibilidad a que un juego cambie de características en mitad del desarrollo, normalmente se suele elegir una metodología ágil a la hora de desarrollar un juego. Da mucha ventaja con el cliente que quiere ver resultados rápido, y sirve para darse cuenta de si una mecánica o característica principal del juego no termina de funcionar o no gusta lo suficiente y así poder cambiarla antes de que esté el desarrollo más avanzado.

2.4.3 – Implementación del software y el arte

Una vez que tenemos planteado como se va a hacer todo toca ponerse a hacerlo. En esta fase es cuando el equipo de programación realiza todas las funcionalidades anteriormente definidas, y donde crean la posible inteligencia artificial y las mecánicas que tiene el juego.

Al mismo tiempo, en este momento es cuando se empieza a realizar el arte y la música que saldrán luego en el juego.

2.4.4 – Pruebas del juego

Una vez que vamos teniendo ciertas mecánicas, ciertos escenarios o ciertas pruebas implementadas, es hora de probarlas con los usuarios objetivo que nos planteamos al principio del diseño. El objetivo de estas pruebas por los usuarios es ver si realmente el juego es divertido, si cumple lo que quería, si las mecánicas se entienden bien, si la historia se comprende, etc. Luego a los usuarios se les pueden hacer unas preguntas, o pasarle un cuestionario, para poder evaluar la experiencia y como ha ido. También, estas pruebas se usan para poder sacar todo tipo de errores o fallos que estén pero que no se hayan descubierto aún.

Adicionalmente, una vez que el juego esté listo y acabado, estas pruebas las volveremos a hacer, pero esta vez con más gente y en diferentes dispositivos, para comprobar que en todos los lugares funciona igual y que se consigue lo que quería con el juego.

2.5 – Metodologías ágiles y Scrum

A la hora de elegir que metodología vamos a usar tenemos que plantearnos cual nos va a servir mejor a la hora del desarrollo del software y cual nos pone las cosas más sencillas. Debido a que tenemos un cliente que satisfacer con resultados y la propia naturaleza cambiante de los videojuegos, elegimos una metodología ágil para desarrollar este trabajo, dadas las numerosas ventajas que nos proporciona. Vamos a ver que es una metodología ágil.

2.5.1 – Metodologías ágiles

¿Qué es una metodología ágil? Como indica el Manifiesto del desarrollo ágil del software [[Agilemanifiesto, 2019](#)], una metodología ágil es aquella que busca darle más importancia a las personas individuales y a las interacciones entre ellas por encima de los procesos y las herramientas. Además, valora más el trabajar mano a mano con el cliente que el simple hecho de hacer un contrato. Valora mucho más un software que funcione y cumpla sus objetivos, que una documentación extensa. Y, como punto muy importante, está a favor de responder al cambio cuando sea necesario, en vez de seguir un plan seguido sin capacidad para cambiar de rumbo cuando sea necesario.

Los métodos ágiles combinan el fomentar la satisfacción del cliente, el desarrollo incremental, tener equipos formados por pocas personas pero motivadas, usar métodos informales, y siempre tirar para hacer las cosas lo más simple posible. Con todo esto se quiere conseguir una comunicación continua y activa con el cliente mediante la entrega de pequeñas partes del producto en vez de analizar y diseñar todo.

Con todo lo mencionado, cuando estamos desarrollando un software y nos surge un problema que fuerza un cambio en el proyecto, gracias a la metodología ágil, los costes de estos cambios serán mucho menores. Donde las metodologías tradicionales son favorables al cambio en el principio del desarrollo, las metodologías ágiles son favorables al cambio durante todo el proceso. Es verdad que si un cambio grande ocurre en las fases últimas del desarrollo el coste va a ser grande, pero será mucho menor que si usamos una metodología tradicional.

Las metodologías ágiles defienden todo lo anterior ya que cuando empezamos un nuevo proyecto es difícil predecir que requisitos software se mantendrán y cuales cambiarán, además de que para muchos tipos de software, el diseño y la implementación son tareas que se entremezclan. Debido a toda esta incertidumbre y a esta impredecibilidad, el proceso de software tiene que ser adaptable en todo momento.

Por todas estas razones elegimos una metodología ágil para desarrollar este proyecto de fin de grado. Específicamente hemos elegido *Scrum*.

2.5.2 – Scrum

¿Qué es *Scrum*? *Scrum* [[Atlassian Scrum, 2019](#)] es una metodología ágil de gestión de proyectos que surgió sobre los años 90. Se basa en un proceso iterativo que define un conjunto de prácticas y roles o papeles. Los roles son los de *Scrum*:

- Scrum master: el que se encarga de dirigir el proceso de *Scrum* y es el líder en las reuniones.
- Equipo: el resto de personas que están en un grupo de *Scrum* para hacer una tarea.
- Product owner: el cliente que tiene la idea del producto.

Scrum lleva en uso ya muchos años y ha dado muy buenos resultados en muchos de los proyectos en los que ha sido aplicado, especialmente en proyectos de pequeño o mediano tamaño.

En *Scrum* se fomenta la comunicación entre los miembros del grupo, debe haber buen ambiente de trabajo entre los integrantes del grupo y tienen que haber interacciones continuas entre ellos. En *Scrum* se tiene muy en cuenta el optimizar los tiempos de trabajo, cuanto menores sean mejor. Se intenta siempre trabajar en ciclos cortos de tiempo, de forma que se disminuye los riesgos y, por ello, es más sencilla la adaptabilidad a cambios. En todo momento tiene que haber un consenso con el cliente y el cliente debe de estar al tanto de que se está produciendo y si le va gustando. Esto se hace para involucrar al cliente en todo el proceso y hacerlo uno más del equipo.

Durante todo el proceso de desarrollo suponemos que existirán cambios, más o menos grandes, por eso los ciclos de trabajo son cortos y siempre al final se hace una entrega de lo que se ha hecho hasta ese momento. Durante todo el proceso, el equipo participa en todas las fases y se plantean objetivos comunes para todos los integrantes del grupo. Después de cada sprint se hacen unas reuniones para hablar y reflexionar sobre lo que se ha hecho, lo que no se ha hecho y lo que se va a hacer. Todo esto se hace en cuanto a cada persona, sin ningún tipo de juicio y sin ninguna jerarquía en la que uno del grupo de trabajo esté por encima de otro.

Scrum trabaja en un sistema de sprints, que son un tiempo determinado de tiempo, normalmente una o dos semanas, durante los cuales se trabaja y al final de cada sprint se hace una reunión grupal para ver cómo va el proyecto.

Además, cada día se hace una reunión muy corta para ver cómo va el proyecto y como está el ambiente en el grupo. Estos sprints se van repitiendo hasta que el producto está listo para ser lanzado.

Como hemos visto, *Scrum* tiene muchísimas ventajas, sobre todo cuando se trata de un grupo pequeño. Por ello, es la que hemos elegido para seguir en este proyecto. Primero tendremos una reunión con el cliente para recoger los requisitos iniciales, que irán cambiando a lo largo del desarrollo, y, una vez empiece la implementación, tendremos reuniones semanales con el cliente para enseñarle el proceso que hemos realizado y para ver cómo avanza el producto y si los requisitos son los mismos.

2.6 – Entorno de desarrollo: Unity

Debido a que vamos a desarrollar un videojuego, la forma más sencilla de hacerlo es usando un entorno que nos ayude y esté preparado para ello. Por esto hemos elegido *Unity*.

Unity [Unity, 2019] es un entorno de desarrollo de videojuegos que trabaja con C#, totalmente gratuito y que ofrece muchas ventajas. Es una herramienta que es fácil de aprender para empezar, pero que hay que trabajar mucho con ella para sacarle todo el provecho posible. *Unity* te permite modificar su interfaz a gusto para que estés lo más cómodo posible trabajando con ella. Dicha interfaz es muy intuitiva y fácil de usar. Así se ve la interfaz:

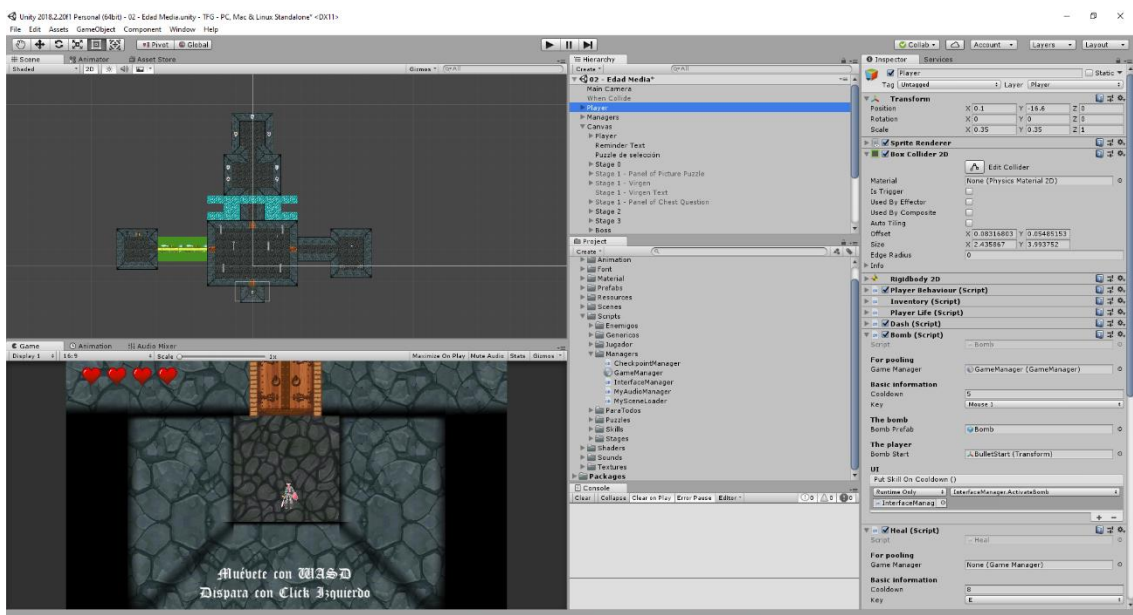


Figura 2.4 - Interfaz de Unity

Además, gracias a que es gratuito y fácil de acceder, hay multitud de guías y tutoriales en internet para hacer casi cualquier cosa, por lo que se facilita bastante el desarrollo.

3. Problema y diseño de la solución

Antes de empezar a hablar de la metodología, de la arquitectura o de la implementación del juego, tenemos que definir cómo va a ser el juego, las partes que tiene, los puntos clave, las mecánicas, los enemigos y todo lo que ello conlleva.

Depende del proyecto que sea, la idea del juego será nuestra o no, haciendo la tarea de diseño del juego más sencilla o más complicada. En nuestro caso, la idea principal viene de un cliente externo, por lo cual para diseñar el juego tenemos que hacer una serie de reuniones con dicho cliente para aclarar todos los aspectos del juego y poder definirlo de forma que sea viable y sea posible de hacer por una sola persona.

Como ya se comentará más adelante en la parte de Limitaciones en el punto Conclusión, en una de las primeras reuniones con dicho cliente se habló de hacer el juego sobre la Edad Media, el Renacimiento y el Barroco (como bien pone en el título del trabajo), sin embargo, debido al reducido tamaño del equipo (en programación, solo era yo) y al poco tiempo, en una de las primeras reuniones se desechó dicha idea y al final el juego será solo basado en la Edad Media.

Como ya se comentó en el punto anterior, a la hora de hacer el diseño del juego tenemos que realizar un Documento de Diseño de Juego. Debido a que la idea principal era de un cliente externo, el Documento de Diseño de Juego se ha realizado poco a poco, siendo ampliado cuando íbamos viendo las diferentes mecánicas y enfrentamientos que se iban implementando.

A continuación vienen algunos de los puntos clave de diseño realizados para este juego. Si se quiere leer en su totalidad, el Documento de Diseño de Juego se encuentra en el Apéndice 2. El nombre final del juego es *Incultura*.

3.1 – Diseño inicial

Incultura es un juego serio del género RPG de acción. Está basado en las diferentes obras de la literatura española de la Edad Media. El jugador tendrá que avanzar por las diferentes fases del juego, las cuales representarán una obra literaria cada una, aprendiendo los rasgos principales de la obra para completarla y liberando dichas obras de los enemigos incultos que la están arruinando.

El objetivo principal de *Incultura* es ayudar al aprendizaje y servir como recordatorio de diferentes obras importantes de la literatura española de la Edad Media. El contexto es que ha surgido un ente, la *Malvada Incultura*, la cual quiere destruir todas las obras de la literatura española y convertir a la gente en incultos. Para evitarlo, debes pasar por las diferentes obras derrotando a los incultos y culturizándolos, liberando así las obras del yugo de la *Malvada Incultura*.

Incultura está compuesto de tres obras o fases principales y un enfrentamiento contra el jefe final. Todas estas fases se desarrollan en el mismo escenario, solo que cada una de ellas estará en una de las salas disponibles. Dependiendo de la fase, los objetivos son diferentes, teniendo que resolver puzles y preguntas en algunos casos, hasta derrotar a enemigos en otros.

El combate es continuo, siendo capaz el jugador de entrar y salir del combate cuando quiera, pudiendo huir de los enemigos cuando esté en peligro. El jugador ataca principalmente a distancia, aunque puede usar otras habilidades, que conseguirá cuando avance en el juego, para poder derrotar a sus enemigos. Respecto a los enemigos, dependiendo del tipo de enemigo que sea, su ataque será a distancia o cuerpo a cuerpo, haciendo que el enfrentamiento varíe de unos a otros.

El jugador comienza la partida en una sala principal desde la cual puede acceder a las diferentes fases del juego, que tiene que recorrer en un orden específico. La primera fase del juego es la correspondiente a la obra *Los Milagros de Nuestra Señora*, la segunda fase corresponde a la obra *Coplas a la muerte de su padre*, y la tercera, y última, corresponde al *Cantar de mio Cid*. La última fase es una pelea contra el jefe final, la *Malvada Incultura*.

Conforme vayamos avanzando en el juego, iremos adquiriendo niveles, que aumentarán nuestra vida y nuestro daño, al mismo tiempo que iremos adquiriendo nuevas habilidades para ayudarnos a superar algunas tareas.

El juego está enfocado para los jugadores de edad entre 14 y 17 años, sobre el momento en el que en la enseñanza se da la literatura de la Edad Media (sobre 3º de la E.S.O.). El juego cuenta con un solo modo de juego en local, por lo que está pensando para una experiencia de un solo jugador. No obstante, aunque solo pueda jugar un solo jugador al mismo tiempo, puede tener a más personas cerca

para ayudarlo a resolver algunas de las preguntas y puzles que se encuentran a lo largo del juego.

3.2 – Jugabilidad

El jugador tiene disponibles las siguientes acciones, que puede realizar en cualquier momento: puede moverse por el escenario a su gusto, puede disparar cuando quiera y hacia donde quiera, puede poner en pausa el juego y volver al menú principal, y, una vez que desbloquee las habilidades, puede usar todas ellas siempre que estén disponibles y fuera de enfriamiento. El jugador puede interactuar con una serie de elementos, principalmente para acceder a diferentes puzles o pruebas, y con algún que otro NPC que le ayudará en la historia.

El juego presenta una serie de retos que el jugador deberá superar para pasarse el juego. Dichos retos van de menor a mayor dificultad, siendo los primeros enfrentamientos contra pocos enemigos y con facilidad de superar, luego están enfrentamientos cuanto más enemigos y de diferente tipo, hasta llegar al reto más difícil: la batalla contra el jefe final.

3.3 – Flujo de juego

El juego se desarrolla de manera que las fases se van complicando poco a poco, empezando por un nivel de dificultad bajo en la entrada y fase primera, y llegando a un nivel de dificultad alto en la batalla final. Esto se controla principalmente por el tipo de enemigo y el número de enemigos que hay en cada fase.

3.4 – Diseño de niveles

Los niveles estarán todos unidos en el mismo escenario, conectados desde una sala principal. Cada nivel consta de una o dos salas en las que tenemos que resolver una serie de pruebas. La vista de dichos niveles es desde arriba, siendo capaces de ver casi toda la sala al mismo tiempo. En dichas zonas puede haber obstáculos e impedimentos que nos dificulten la movilidad en algún momento.

3.5 – Interfaz

La interfaz es bastante minimalista. El menú principal consta del título y tres botones: uno para jugar, otro para ver los créditos del juego y otro para salir del juego. Una vez dentro del juego tenemos en la parte superior izquierda la vida y abajo a la derecha las habilidades que tengamos en ese momento. En alguna fase, como el jefe final, tenemos algo más, pero no mucho. En el menú de pausa, muerte y finalizar tenemos dos botones, uno que nos lleva al menú de vuelta y otro que, dependiendo de cuál sea, hace una cosa u otra: volver a la partida en el de pausa, cargar desde el último punto de guardado en el de muerte, y volver a jugar en el de finalizar.

Quitando menús y personaje, durante toda la partida tendemos un posible texto en la parte baja y centro de la pantalla donde nos saldrá texto informativo y que nos ayuda en algunos momentos.

Para cada puzle tenemos una interfaz específica, aunque todos ellos cuentan con un botón de salir y un botón de comprobar respuesta:

- Puzle de introducir texto: tenemos un texto en la parte superior que es una pregunta o indicativo y un campo de texto a rellenar respecto a la pregunta o preguntas.
- Puzle de mover imágenes: tenemos diferentes imágenes que tenemos que colocar en su lugar.
- Puzle de elegir opción correcta: tenemos una lista de marcadores asociados a textos entre los que tenemos que elegir.
- Prueba de pulsar rápidamente: tenemos dos barras que van subiendo, una asociada al oponente y otra a nosotros, además de un botón que nos sirve para aumentar un poco la barra.

A continuación se muestra el flujo de pantallas y algunas de las interfaces del juego. El diseño del resto de interfaces de usuario están disponibles en el Apéndice 2.

3.5.1 – Flujo de pantallas

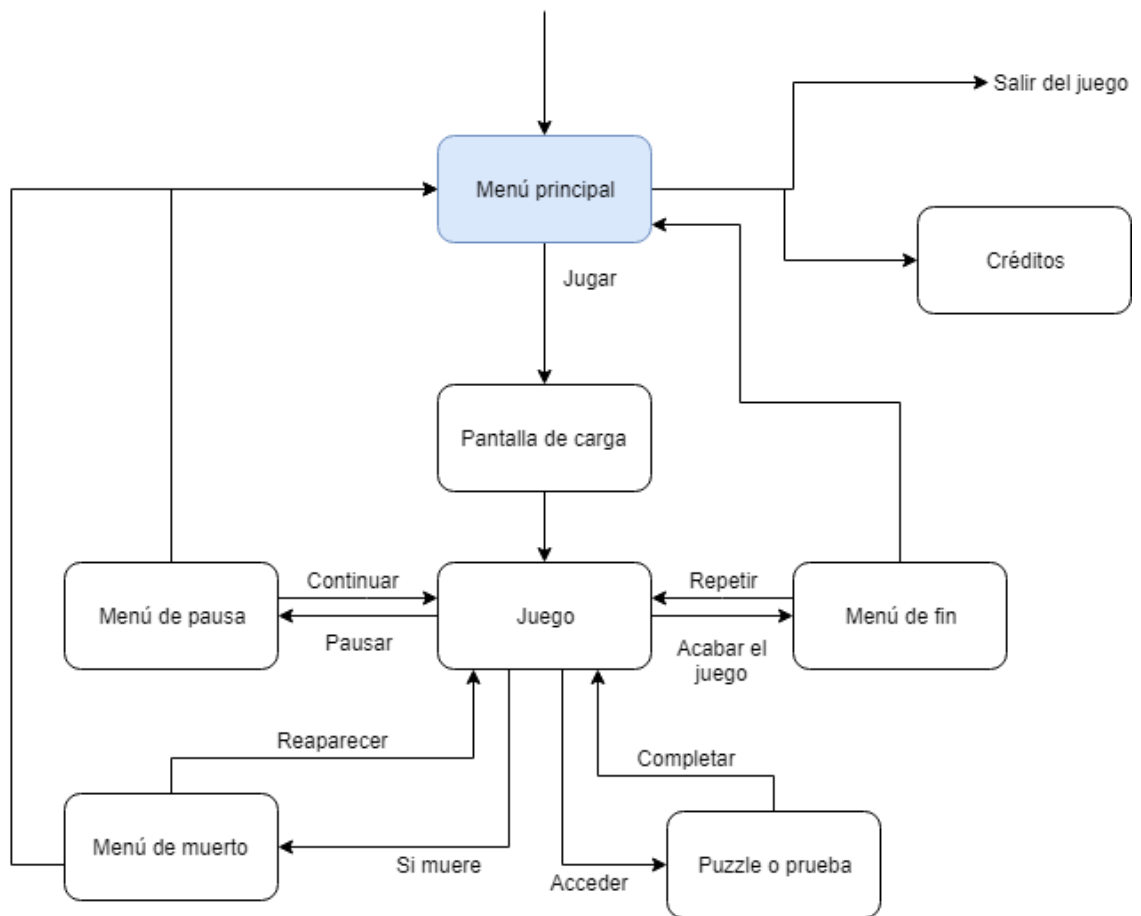


Figura 3.1 - Flujo de interfaz, realizado con draw.io [\[draw.io, 2019\]](https://draw.io)

Siguiendo el objetivo de tener la interfaz simple, el juego tiene pocas pantallas realmente. O estás en el menú principal o estás dentro del juego, en cuyo momento ya no cambias de pantalla en ningún momento. En todo caso aparecerá un mensaje por encima o la interfaz de un puzzle, pero en ningún momento cambiamos de interfaz de juego. Por esta misma razón se intentó que la interfaz propia del juego fuese lo más simple posible.

3.5.2 – Menú principal



Figura 3.2 - Esquema del menú principal

El objetivo del menú principal era mantenerlo simple y que se viese claramente las opciones que tienes, pudiendo jugar directamente sin pensarlo mucho.

3.5.3 – Interfaz durante la partida

Esta es la interfaz que más tiempo va a ver el jugador, así que no debe de ser muy liosa ni muy sobrecargada.

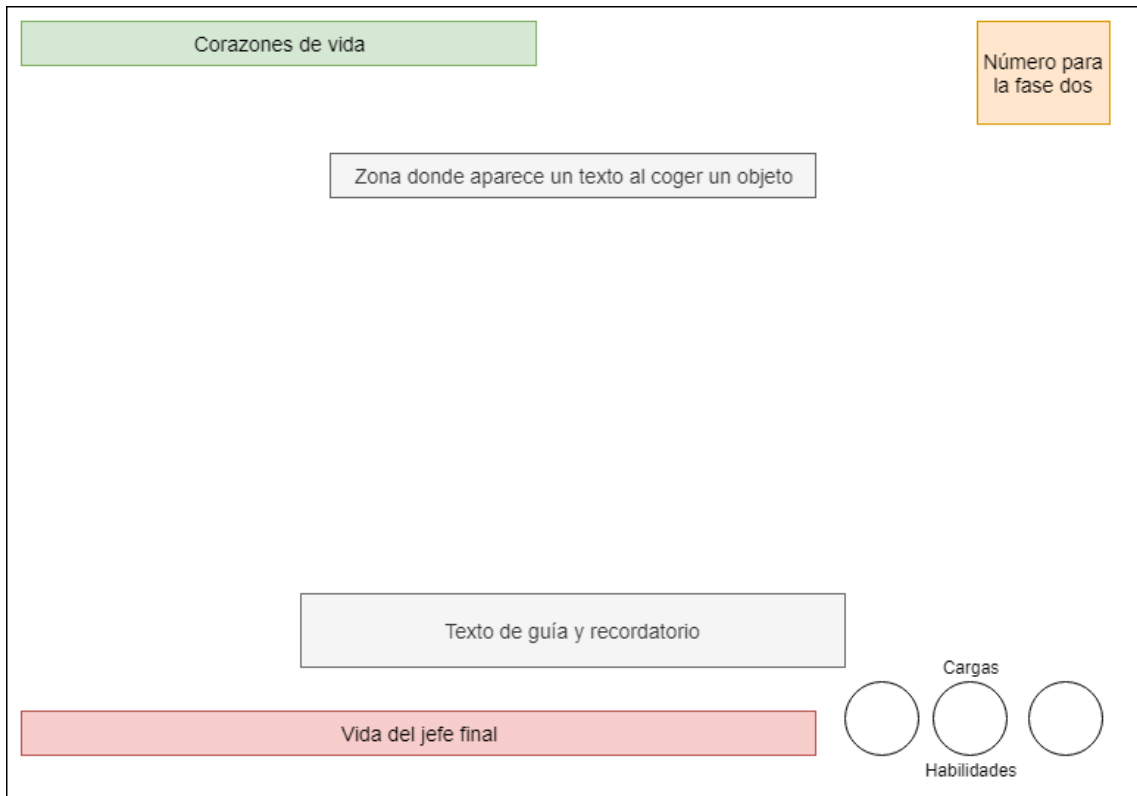


Figura 3.3 - Esquema de la interfaz de juego

4. Aplicación de Scrum a la solución

En este punto se va a mostrar el proceso de aplicación de *Scrum* a la solución diseñada.

4.1 – Especificaciones iniciales y requisitos

En las primeras reuniones con el cliente se recogieron las especificaciones técnicas iniciales y los requisitos que debería cumplir el juego. Aunque más adelante algunos cambiaron un poco, la mayoría se mantuvieron hasta el final:

4.1.1 – Especificaciones técnicas

Las especificaciones técnicas del juego eran las siguientes:

- Debe ser un juego para ordenador, pudiéndose jugar en cualquier ordenador actual.
- Debe ser un juego que se juega con teclado y ratón, haciendo uso de ambos.
- Debe estar disponible para ser jugado por cualquier persona.
- Tiene que ser un juego para un solo jugador local.
- Debe ser un juego con vista 2D.
- Tanto el arte como la música serán realizados por conocidos del cliente.

4.1.2 – Requisitos

Los requisitos iniciales fueron los siguientes:

- Debe ser un juego apto para todos los públicos.
- Debe servir como estudio o repaso de los conceptos que se definan.
- Debe tratar de las obras de la Edad Media, el Renacimiento y el Barroco.
- Cada obra deberá ser una fase separada.
- Las fases deberán estar ambientadas en la obra que representan.
- El escenario será una sala central con todas las puertas de las diferentes obras a las que podemos acceder.
- Deberemos introducir el nombre de la obra para que la puerta se habrá.

- Para poder abrir una puerta nos hacen falta las palabras de dicha obra, que se conseguirán derrotando enemigos o al pasarnos alguna fase.
- El jugador deberá ser capaz de elegir entre chico o chica.
- El personaje principal será capaz de moverse en todas las direcciones.
- El personaje podrá disparar en todas las direcciones.
- El personaje tendrá una vida que se mostrará en la interfaz.
- El personaje podrá subir niveles, aumentando su vida y su daño.
- El personaje podrá cambiar de atuendo.
- Existirán varios tipos de enemigos y un jefe final.
- Los enemigos básicos se diferenciarán en débil, mediano y fuerte.
- Los enemigos básicos tendrán diferente vida y daño dependiendo de su tipo.
- Con cada fase que vayamos avanzando los enemigos se harán más fuertes.
- El jefe final será un enfrentamiento más complicado que el resto, siendo una fase en sí mismo.
- El jefe final tendrá diferentes habilidades.
- Por cada era existirán tres fases.
- La primera fase representará la obra Los Milagros de Nuestra Señora.
- La primera fase deberá estar estructurada en cuatro partes, las mismas que la estructura de los fragmentos de la obra.
- La primera fase deberá acabar con una pregunta sobre la moraleja de la historia.
- La segunda fase representará a la obra de Coplas a la muerte de su padre.
- La segunda fase irá por oleadas de enemigos que tendremos que vencer.
- Entre cada oleada de la segunda fase deberá haber una pregunta sobre las coplas.
- Al final de la segunda fase deberá haber una pregunta sobre el número de coplas totales y la estructura de las coplas.
- La tercera fase deberá ser sobre la obra Cantar de mio Cid.
- La estructura de la tercera fase deberá representar los tres cantares de la obra.
- Al final de cada fase deberá haber un cofre que al abrirlo te aumente el nivel y te deje cambiar de aspecto.

Estos fueron los requisitos iniciales, pero, como se verá más adelante en la parte de Limitaciones en las Conclusiones, algunos de ellos se rebajaron o directamente se eliminaron de la lista, al igual que se añadieron nuevos conforme el desarrollo fue avanzando.

El principal requisito que fue desechado rápidamente fue el que el juego fuese de la Edad Media, el Renacimiento y el Barroco. Tanto el cliente como yo nos dimos cuenta de que no teníamos tanto tiempo ni equipo como pensábamos, y que preferíamos hacer una era bien a tres mal. Por tanto, el juego pasó a ser solo de la Edad Media.

Otro de los requisitos que cambiaron durante el desarrollo fue el personaje principal. Muchos de los requisitos se mantuvieron, pero algunos como el poder elegir sexo y el poder cambiar atuendos se eliminaron debido a la falta de artistas dispuestos a hacer el trabajo. Por otro lado, se añadió el requisito de que el personaje principal tuviera diferentes habilidades que fuese consiguiendo a lo largo del juego.

El último grupo grande de cambios de requisitos fueron los enemigos. En un principio teníamos tres tipos de enemigos diferentes, e irían apareciendo cada uno en una obra distinta. Una vez más, debido al tiempo y a la falta de artistas, estos se recortaron. Al final teníamos dos enemigos diferentes: uno que atacasen cuerpo a cuerpo y otros que atacasen a distancia, cada uno con sus estadísticas especiales. Además, como ya no teníamos tres eras, el jefe final tenía que ser más complicado y tener diferentes fases para suplir el hecho de que ya no eran tres sino uno.

4.2 – Evolución del proyecto

Durante todo el desarrollo del proyecto se ha ido guardando información de las diferentes reuniones al final de cada sprint, así como un recuento del trabajo diario. Aunque se empezó a trabajar la idea desde febrero de 2019, no fue hasta junio de 2019 que empezó a desarrollar bien el juego, debido a diferentes problemas y dificultad para coincidir con el cliente.

Debido a que no siempre se ha podido hacer reuniones con el cliente al final de cada sprint, vamos a separar este punto en dos sub puntos, las reuniones con el cliente y el equipo, y los sprints realizados.

4.2.1 – Reuniones con el cliente y el equipo

- 1ª Reunión – 13/02/2019

Esta primera reunión sirvió como toma de contacto entre el cliente, el productor y yo, para saber que íbamos a hacer y sobre que íbamos a trabajar. En esta reunión se habló de los tiempos que íbamos a tener, de la idea del juego muy por encima y de cómo lo íbamos a hacer. Principalmente sirvió de toma de contacto entre los integrantes del grupo.

- 2ª Reunión – 03/05/2019

Esta reunión fue donde se recogieron los principales requisitos y las características técnicas necesarias para el proyecto. Fue entre el cliente y yo. En esta reunión se aclararon y se pulieron muchas partes de la idea principal, se empezó a plantear como desarrollar el juego y se empezó a hablar del equipo necesario para poder realizar dicho juego. En esta reunión se diseñó principalmente la fase uno y la dos. Después de esta reunión ya pude empezar a trabajar en la estructura de la implementación.

- 3ª Reunión – 25/06/2019

Esta reunión fue la primera que estuvimos tanto cliente, como equipo de conocidos del cliente que se iban a encargar del arte y la música, como el programador, yo. En esta reunión fue cuando se recortaron los requisitos para como serían desde este punto hasta el final del desarrollo. Se pasó de tener tres épocas a solo una, quitamos un tipo de enemigo y el jugador ganó habilidades. Además, aquí se definieron los enemigos bien y se corrigieron algunas de las pruebas que se habían planteado para las diferentes fases. A partir de este punto empecé a implementar el juego y a trabajar por sprints de una semana.

- 4ª Reunión – 07/08/2019

Esta reunión fue la primera que el cliente vio ya una parte del juego desarrollado. Para este punto estaba hecha la fase inicial, las primeras pruebas y la primera fase, además de una de las habilidades. En este punto surgieron nuevos requisitos y se cambiaron algunos, pero todos ellos fueron cosas

menores. Principalmente, esta reunión sirvió para que el cliente viese como iba el juego, comentase que le gustaba y que no, que parte del equipo de arte mostrase que llevaba, y para desarrollar como sería la tercera fase del juego.

- 5ª Reunión – 22/08/2019

En esta reunión el juego estaba desarrollado ya hasta la tercera fase, además de casi todas las habilidades. Se le mostró al cliente y se hizo que una persona del público objetivo lo probase. Esta reunión sirvió para que el cliente volviese a decir que quería cambiar algunas cosas y para encontrar errores entre el juego. Además, el equipo de arte ya trajo escenarios y algunos personajes hechos. En esta reunión se terminó de definir la pelea con el jefe final y que habilidades iba a tener.

- 6ª Reunión – 29/08/2019

Esta fue la última reunión que se hizo antes de la entrega del proyecto. En esta estaba desarrollado hasta el jefe final, es decir, el juego estaba prácticamente acabado. No obstante, en esta se volvieron a cambiar un par de cosas, específicamente para modificar la dificultad de dicho jefe final, y para que no fuese aburrido. Además, el resto del arte fue mostrado, así como la música a usar para el juego. Después de todo esto, el cliente le dio el visto bueno al juego, y se terminó de implementar dichos cambios.

4.2.2 – Sprints

Debido a que el tiempo de desarrollo fue en verano, y que el cliente tenía que hacer vida aparte, no siempre se pudo hacer reunión con el cliente al final de cada sprint. Sin embargo, al final de cada sprint semanal hice un recuento sobre que había realizado, que necesitaba ser revisado aún, que no había hecho que debería, y que iba a hacer después.

El recuento de sprints empezó a partir del 25/06/2019, así que cada sprint se cuenta cada lunes después de dicha fecha.

- 1º Sprint – 01/07/2019

En este sprint el objetivo conseguido fue plantear la estructura del código y hacer el movimiento del personaje más el ataque del personaje. Además, se realizó algunas funciones del sistema como el pooling o el uso de los temporizadores. En el siguiente sprint se planeó hacer el sistema de vida para el jugador y los enemigos y empezar a hacer los puzles.

- 2º Sprint – 08/07/2019

El objetivo de este sprint era realizar la vida para todos los personajes del juego y plantear los puzles posibles. Sin embargo, en este sprint no se llegó a cumplir el objetivo de los puzles y solo se realizó la vida para todos los personajes. Para el siguiente sprint se planteó tener hecho la estructura base para todos los puzles, además de tener todos los tipos de puzles hechos.

- 3º Sprint – 15/07/2019

El objetivo de este sprint era realizar una estructura base para todos los puzles y realizar todos los puzles necesarios para las diferentes fases. Estos objetivos se cumplieron bien, además de haber realizado el comportamiento de las balas para los enemigos. Una vez que tenemos la lógica de los puzles, el siguiente objetivo es hacer la primera parte del escenario. Hacer la fase cero y la fase uno. Además, hacer el comportamiento de los enemigos cuerpo a cuerpo.

- 4º Sprint – 22/07/2019

En este sprint se cumplieron casi todos los objetivos planteados. Se realizó el comportamiento de los personajes cuerpo a cuerpo, además de la fase cero. Se ha hecho el principio de la fase uno, especialmente el puzle de poner en orden una imagen, que fue puesto como requisito más adelante. Sin embargo, se esperaba tener la fase uno terminada pero no se llegó a ello. El objetivo para la siguiente fase era terminar la fase uno y hacer funcionalidades necesarias para el juego, como el inventario de palabras, el sistema de las puertas y el sistema de puntos de guardado.

- 5º Sprint – 29/07/2019

En este sprint se cumplieron todos los objetivos planteados y algunos más. Se realizó enteramente la fase uno, se implementó el inventario para el jugador y los enemigos, se implementaron las palabras, se implementó el sistema de puertas que necesitan palabras para poder abrirse, y se implementó el sistema de puntos de guardado. Además, se creó el menú que aparece cuando mueres y se creó el sistema de cofres. Para el siguiente sprint se planteó mejorar la inteligencia artificial de los personajes cuerpo a cuerpo y gran parte de las funcionalidades de la interfaz, como que se muestre la vida del jugador y que se muestren mensajes al conseguir nuevas palabras. Además, se planteó hacer el sistema de niveles.

- 6º Sprint – 05/08/2019

En este sprint ocurrieron una serie de problemas y se cumplieron muy pocos objetivos. El único objetivo que se cumplió este sprint fue el que la interfaz mostrase la vida del jugador y se fuese actualizando conforme el jugador gana o pierde vida, además de que cuando consigas una nueva palabra salga un aviso. Por lo tanto, para el siguiente sprint se planteó mejorar la inteligencia artificial de los enemigos cuerpo a cuerpo, hacer el sistema de niveles, y empezar a hacer la habilidad del desplazamiento.

- 7º Sprint – 12/08/2019

En este sprint se cumplieron todos los objetivos que se plantearon en el sprint anterior y aún más cosas. Se hizo una máquina de estados simple para la inteligencia artificial de los enemigos, e implementó el sistema de niveles y se realizó la habilidad del desplazamiento. Además, en medio del sprint hubo una reunión con el cliente, lo cual hizo que se implementaran algunos cambios. Se implementaron una gran variedad de textos para recordar conceptos y guiar al jugador durante el juego. Se implementaron diversas pausas durante las fases para que el jugador tenga tiempo de ver que está ocurriendo entre cada parte de la fase. Encima, se empezó y casi se acabó de implementar la fase dos. Se realizaron todas las preguntas y los puntos de guardado durante la fase. Se implementó una forma de conseguir una habilidad

nueva y un nivel nuevo. Para el siguiente sprint se planteó terminar la fase dos y hacer la fase tres.

- 8º Sprint – 19/08/2019

En este sprint se cumplieron multitud de objetivos también. La fase dos se terminó de hacer, con la función de reinicio implementada. Se implementó la inteligencia artificial del enemigo a distancia, compuesta de un par de máquinas de estados. Además, se corrigió el sistema de perder vidas en la interfaz, que no funcionaba como debería. La fase tres se implementó en su totalidad, con todas sus partes e interacciones. Se implementó una prueba de pulsar rápidamente un botón para ganar que fue un requisito que surgió durante el sprint. Para el siguiente sprint se planteó hacer las habilidades restantes además de hacer el jefe final.

- 9º Sprint – 26/08/2019

En este sprint se cumplieron los objetivos planteados. Primeramente, se implementó el menú principal y un sistema para cambiar entre escenas. Se implementó tanto la bomba como la cura, la bomba con una animación y la cura con un efecto de partículas. En medio del sprint hubo una reunión con el cliente en la cual se descubrieron diferentes errores y cosas a cambiar, además de una necesidad de balancear las estadísticas de los enemigos. Una vez que estos errores fueron corregidos, se empezaron a hacer algunas de las habilidades para el jefe final, como el pilar de fuego y el rayo cargado. Sin embargo, no se acabó a tiempo. Para el siguiente sprint se planteó acabar el jefe final y sus habilidades, además de pequeñas cosas como textos varios y que pasa cuando vencemos al jefe.

- 10º Sprint – 02/09/2019

En este sprint se cumplieron los objetivos planteados y más, debido a la reunión que hubo en mitad del sprint. Se terminó de implementar las habilidades del jefe final y su inteligencia artificial, que apenas es nada. Las pequeñas cosas anteriores fueron implementadas junto con que pasa al vencer al jefe. Menú de pausa implementado. Después de esto vino la reunión, en la

cual se volvieron a añadir nuevos requisitos. El más importante de ellos es que la pelea contra el jefe final era muy sencilla y había que cambiarla, añadiéndole varias partes y diversos enemigos que van apareciendo durante la pelea. Se implementaron estos cambios, tanto las partes como los enemigos varios, además de balanceo alrededor de las habilidades del jefe y del jugador. En este punto el juego está prácticamente hecho y necesita ser probado para encontrar fallos y errores.

- 11º Sprint – 09/09/2019

En este último sprint lo que se realizó fueron diversas pruebas con usuarios para buscar errores y fallos, se añadieron los últimos artes necesarios y la música deseada. Lo principal ha sido arreglar diversos errores, varias veces en el caso de algunos, y comprobar que el juego es funcional y cumple su objetivo en todas las máquinas y para todas las personas. Además, en este sprint es cuando se ha realizado la memoria, con todos sus apartados y los diversos diagramas necesarios.

5. Implementación de la solución

A la hora de diseñar como vamos a implementar el juego, lo primero es ver cual va a ser la estructura de las clases y como se van a relacionar entre ellas. Otro punto importante a tener en cuenta es si los enemigos llevarán inteligencia artificial, que hará esta, y hasta donde queremos llegar con ella. Además, tenemos que plantearnos como va a interaccionar el usuario con el juego y que va a poder hacer, así como el desarrollo del flujo del juego.

Como partimos de que vamos a trabajar con Unity, podemos usar esto a nuestra ventaja. Podemos usar algunas de las características que nos da Unity ya hechas para simplificar el trabajo. Entre ellas hay dos que destacar que harán el desarrollo más sencillo en todos los aspectos: los eventos de Unity y el Collab.

Los eventos de Unity, bajo la clase `UnityEvent`, son una forma de llamar a varios métodos de diferentes clases y de diferentes objetos a la vez sin tener que escribir por código literalmente cuales son dichos métodos. Esto es muy útil cuando tenemos que actuar dependiendo de que este pasando en el juego, ya que no siempre vamos a hacer lo mismo, ni siempre queremos llamar a los métodos en un momento exacto, sino que queremos un poco más de libertad. Además, de esta forma conseguimos que haya menos dependencias entre el código, haciendo más limpia la estructura de clases (como se verá a continuación), y haciendo más sostenible el código.

La otra característica importante es el Unity Collab. Esta característica es una herramienta de control de versiones, es decir, hace las veces de un [\[GitHub\]](#) integrado en Unity. Esto ha servido para llevar un control claro del trabajo y ayudar a hacer el proceso de *Scrum*. Además, ha proporcionado la oportunidad de si algo ha dejado de funcionar en la versión actual, ir para atrás y cargar desde una versión anterior estable.

Ahora vamos a ver como esto ha afectado al diseño de la implementación del juego.

5.1 – Diseño de clases

Como ya se ha comentado, gracias a los eventos de Unity el diseño de clases se ha simplificado bastante y ha sido más sencillo la implementación de los

requisitos. Aun así, el diseño de clases y la implementación han sido laboriosas y de un tamaño considerable. A continuación, vamos a ver como ha quedado el diagrama de clases, así como todas las clases que tenemos en el software.

5.1.1 – Diagrama de clases

Como se ve en la Figura [5.1](#) más abajo, para este diagrama se ha intentado que solo aparezcan los nombres de las clases y las relaciones entre ellas. Esto es porque si hubiésemos puesto también los atributos y operaciones de cada clase, el diagrama hubiera sido demasiado grande y no se podría leer bien. Aun así, el diagrama que tenemos es bastante grande.

Como se puede apreciar, hay muchas clases que no están relacionadas con otras, o están puestas por bloque. Esto es debido a los eventos que antes hemos mencionado. Gracias a ellos, las dependencias son bajas. Principalmente, muchas clases usan diferentes tipos de temporizadores o sistemas para generar elementos sin gastar muchos recursos (pooling), por eso son dichas clases las más relacionadas con el resto.

Por otro lado, durante todo el desarrollo del software hemos usado bastante la herencia entre clases para poder generalizar los comportamientos para clases que se parecen bastante. De esta forma, ahorramos bastantes líneas de código que se repetirían y, cuando tengamos que ponernos a buscar componentes entre diferentes objetos, dicha tarea será más sencilla y genérica, siendo así más viable para un número mayor de objetos.

Vamos a ver ahora cuál es la funcionalidad de cada clase. Esto lo haremos por grupos.

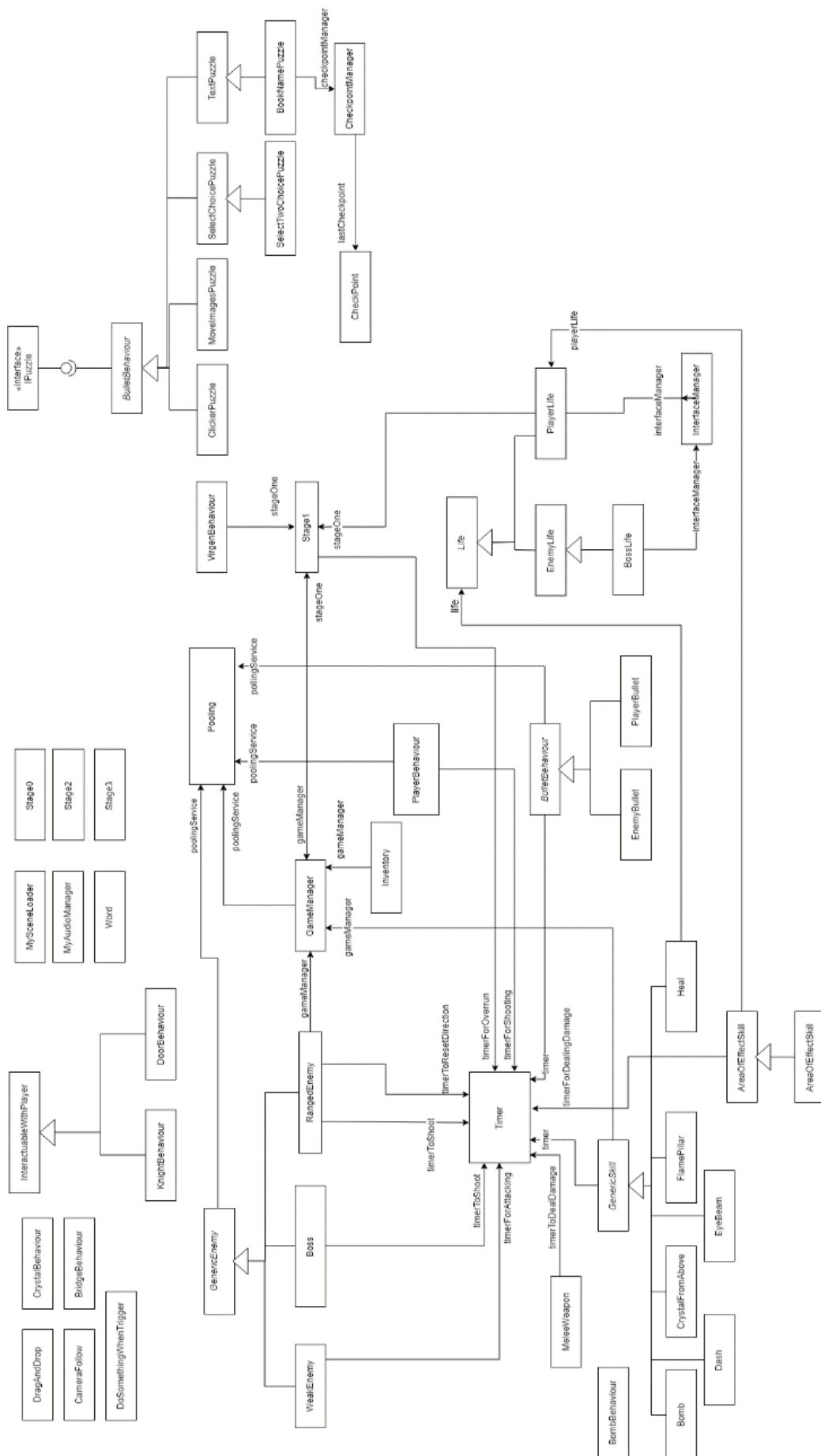


Figura 5.1 - Diagrama de clase

5.1.2 – Enemigos

En los enemigos tenemos cuatro clases principales: la clase abstracta `GenericEnemy`, de la cual todos heredan, y luego los tres enemigos del juego, `WeakEnemy`, `RangedEnemy` y `Boss`.

`GenericEnemy` es una clase abstracta de la cual extienden las otras tres. Esta clase sirve como base para todos los enemigos ya que contiene los elementos básicos de un enemigo, así como las definiciones para los métodos que los enemigos tendrán que completar.

`WeakEnemy` es la clase que controla el comportamiento de los enemigos cuerpo a cuerpo. Controla tanto los movimientos como los ataques.

`RangedEnemy` es la clase que controla el comportamiento del enemigo a distancia. Controla tanto los movimientos como los ataques.

`Boss` es la clase que controla tanto el comportamiento del jefe final como el funcionamiento de la pelea del jefe final, con todas sus fases y el recargar cuando pierdes.

5.1.3 – Genéricos

En la parte de genéricos diferenciamos entre dos tipos de clases, los comportamientos y funcionalidades.

`BridgeBehaviour` se encarga de controlar el comportamiento del puente en la fase tres.

`CrystalBehaviour` se encarga de controlar el comportamiento de los cristales en la primera parte de la pelea contra el jefe final.

`DoorBehaviour` se encarga controlar el comportamiento de las puertas. Extiende de `InteractableWithPlayer` debido a que las puertas pueden ser usadas por el usuario.

`KnightBehaviour` se encarga de controlar el comportamiento del Cid. Como las puertas, extiende de `InteractableWithPlayer` debido a que el jugador puede interaccionar con él.

VirgenBehaviour se encarga de controlar el comportamiento de la Virgen que aparece en la fase uno.

CameraFollow se encarga de que la cámara siga al personaje en todo momento, dejando al personaje siempre en el centro de la pantalla.

Checkpoint es una clase que representa a uno de los puntos de guardado. Será usada por el controlador adecuado.

DoSomethingWhenTrigger es una clase que se encarga de hacer lo que queramos cuando el jugador active un trigger de un objeto que digamos. El que hace lo podemos configurar desde la interfaz.

DragAndDrop controla que podamos coger un objeto con el ratón y soltarlo.

InteractableWithPlayer es una clase que hace que un objeto cualquiera pueda ser usado por el jugador, y que al ser usado haga lo que queramos.

Pooling es una clase que se encarga de mantener un sistema de ocultar y mostrar objetos cada vez que creamos uno nuevo, para que así mejoremos la eficiencia.

Timer es una clase que crea un temporizador del tiempo que deseemos.

5.1.4 – Jugador

PlayerBehaviour es la clase que se encarga de controlar todo el comportamiento del personaje principal.

5.1.5 – Controladores

En la parte de controladores tenemos todas aquellas clases que se encargan de manejar y controlar puntos importantes del juego.

CheckpointManager es la clase que se encarga de manejar los puntos de guardados: cuál fue el último por el que pasamos y cuál tenemos que usar al morir.

GameManager es la clase que se encarga de controlar algunos contadores genéricos del juego, sistemas base y las cosas genéricas que no tienen clase en específico.

InterfaceManager es la clase que se encarga de manejar todo lo que tiene que ver con la interfaz, desde la vida del jugador, a los mensajes que salen, los eventos que aparecen, etc.

MyAudioManager se encarga de manejar el audio del juego y cuando tiene que sonar una u otra cosa.

MySceneLoader se encarga de controlar el cambio entre escenas y el cómo se hace este.

5.1.6 – Comunes

En la parte de comunes se encuentran todas aquellas clases usadas tanto por los enemigos como por el jugador.

BulletBehaviour es una clase abstracta que hace el comportamiento de una bala común.

EnemyBullet es una clase que hace el comportamiento de las balas lanzadas por los enemigos. Extiende de **BulletBehaviour**.

PlayerBullet es una clase que hace el comportamiento de las balas lanzadas por el jugador. Extiende de **BulletBehaviour**.

MeleeWeapon se encarga de controlar el daño que hace las armas de los enemigos cuerpo a cuerpo.

Life es una clase abstracta que controla la vida de un personaje. En ella se encuentran las típicas opciones de ganar vida, perder vida y morir.

PlayerLife es una clase que extiende de **Life** y que se encarga de controlar la vida del personaje durante todo el juego.

EnemyLife es una clase que extiende de **Life** y se encarga de manejar las vidas de todo y cada uno de los enemigos del juego.

BossLife es una clase que extiende de **EnemyLife** y que se encarga de controlar la vida del jefe final.

Inventory es una clase que se encarga de manejar el inventario de palabras del personaje o de un enemigo.

Word es una clase que representa el objeto de una palabra.

5.1.7 – Puzles

En la parte de puzles se encuentran todas las clases que controlan el funcionamiento de un puzle o de una prueba.

IPuzzle es una interfaz que define las funcionalidades que van a tener todos los puzles.

GenericPuzzle es una clase abstracta que implemente **IPuzzle** y que define una serie de métodos y funcionalidades que serán comunes para todos los puzles.

ClickerPuzzle es una clase que extiende de **GenericPuzzle**. Se encarga de controlar la prueba de pulsar un botón rápidamente para superarla.

MoveImagesPuzzle es una clase que extiende de **GenericPuzzle**. Se encarga de controlar la prueba de resolver un puzle clásico a base de colocar las imágenes en su lugar.

SelectChoicePuzzle es una clase que extiende de **GenericPuzzle**. Se encarga de controlar la prueba de resolver una pregunta en la que tienes que elegir una opción correcta entre muchas.

SelectTwoChoicePuzzle es una clase que extiende de **SelectChoicePuzzle**. Se encarga de controlar la prueba de resolver una pregunta en la que tienes que elegir dos opciones correctas entre muchas.

TextPuzzle es una clase que extiende de **GenericPuzzle**. Se encarga de controlar la prueba de resolver una pregunta a base de introducir un texto en un cuadro de texto y comprobar que es el correcto.

BookNamePuzzle es una clase que extiende de **TextPuzzle**. Se encarga de controlar la prueba de introducir el nombre de un libro para entrar a una fase.

5.1.8 – Habilidades

En este grupo se encuentran todas las clases que se encargan de manejar las habilidades, tanto del personaje principal como del jefe.

GenericSkill es una clase abstracta que contiene métodos y funcionalidades que todas las habilidades tienen en común.

Dash es una clase que se encarga de manejar la habilidad del desplazamiento del jugador.

Heal se encarga de controlar la habilidad de la cura del jugador.

Bomb se encarga de controlar la habilidad de la bomba del jugador.

BombBehaviour controla el comportamiento de las bombas, así como el daño que hacen y a quién hacen daño.

FlamePillar es una habilidad del jefe y se encarga de generar pilares de fuego por la zona.

CrystalsFromAbove es una habilidad del jefe que hace que caigan cristales gigantes del techo.

EyeBeam es una habilidad del jefe que se encarga de controlar cuando lanzar un rayo.

AreaOfEffectSkill es una clase que está hecha para las habilidades que hacen daño en un área determinada. Se encarga que, en esa área, si entra un aliado o un enemigo, reciba una cantidad de daño por segundo según le indiquemos.

EyeBeamBehaviour extiende de **AreaOfEffectSkill** y sirve para controlar mejor como hace daño el rayo.

5.1.9 – Fases

Este grupo de clases se encarga cada una de manejar y controlar su fase asociada.

Stage0 se encarga de controlar la fase inicial.

Stage1 se encarga de controlar y manejar la fase uno del juego.

Stage2 se encarga de controlar y manejar la fase dos del juego.

Stage3 se encarga de controlar y manejar la fase tres del juego.

5.2 – Inteligencia artificial de los enemigos

En este juego tenemos solo tres tipos de enemigos, como ya hemos mencionado antes, y cada uno de estos tiene una inteligencia artificial diferente. En general, no queríamos complicarnos mucho con los enemigos ya que no son el punto fuerte del juego. Buscábamos unas inteligencias sencillas que nos sirviesen como enemigos y que nos presentasen un poco de reto al principio, pero que una vez que conocíamos que hacían fuese sencillo comprenderlos.

Los enemigos con inteligencia artificial son el enemigo cuerpo a cuerpo y el enemigo a distancia. El jefe final no tiene inteligencia artificial porque se le ha puesto más énfasis en las propias partes de la pelea y en esquivar bien los

diferentes obstáculos que están en el mapa, y las habilidades que nos lanza el jefe. El mecanismo usado para nuestras dos inteligencias artificiales ha sido el uso de sencillas máquinas de estado.

5.2.1 – Enemigo cuerpo a cuerpo

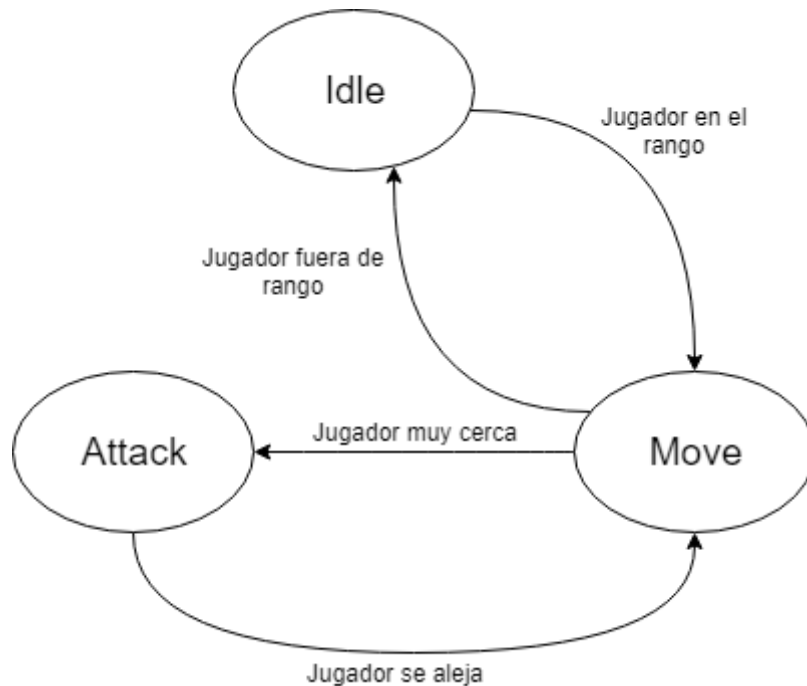


Figura 5.2 - Máquina de estados del enemigo cuerpo a cuerpo

El enemigo cuerpo a cuerpo tiene tres simples estados: quieto (Idle), moviéndose (Move) o atacando (Attack), y transitará de uno a otro dependiendo de donde esté el jugador.

5.2.2 – Enemigo a distancia

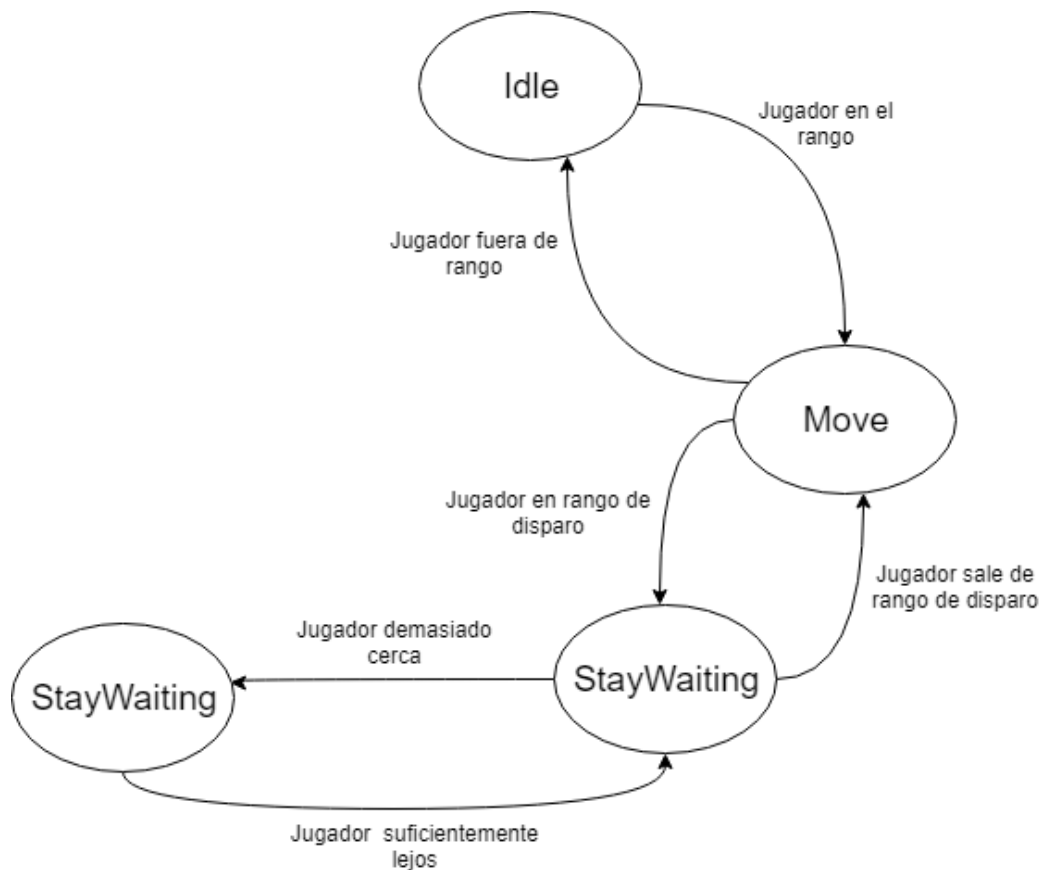


Figura 5.3 - Máquina de estados del movimiento de distancia

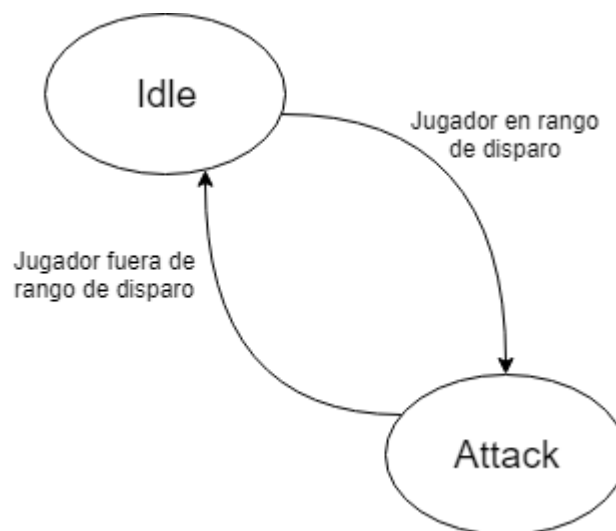


Figura 5.4 - Máquina de estados del ataque de distancia

El enemigo a distancia es un poco más complejo que el enemigo a cuerpo a cuerpo. Se compone de dos máquinas de estados: la de moverse y la de atacar.

En la de moverse está todo el rato mirando donde está el jugador: si está lejos pasa de él, si se acerca va hacia él y le dispara, pero si el jugador se le acerca

demasiado el enemigo intentará huir en una dirección aleatoria para que el jugador no le pueda golpear.

En la de atacar simplemente está esperando a que el jugador entre en el rango para poder dispararle.

La ventaja de tener las dos máquinas de estado es que el movimiento no depende del ataque ni al revés, por lo que ahora si que podemos tener al enemigo disparando y moviéndose, cosa que con el cuerpo a cuerpo no era posible.

5.3 – Interacción del usuario con el juego

Durante toda la duración del juego el usuario puede interaccionar con múltiples elementos:

- En el menú principal, el usuario puede pulsar cualquiera de los tres botones que existen, llevándole a una parte de la aplicación o cerrándola. En el caso en el que le de a los créditos, tendrá una ventana que puede cerrar en cualquier momento.
- Dentro del juego, el jugador puede moverse con total libertad cuando quiera pulsando WASD en el teclado, y el personaje se moverá en la dirección marcada.
- El jugador puede pulsar con el clic izquierdo en cualquier parte de la pantalla y el personaje disparará una bala hacia ese lugar.
- En cualquier momento, el usuario puede pulsar la tecla de Escape para acceder al menú de pausa. Ahí le puede dar a continuar por donde iba o volver al menú principal.
- Cuando desbloquee el desplazamiento, el jugador puede pulsar Espacio mientras que se mueve en una dirección para desplazarse en esa dirección unas unidades.
- Cuando desbloquee la cura, el jugador puede pulsar la tecla E para que se gaste una carga de cura, si tiene, y curarse un corazón de vida.
- En algunos puntos del juego, el jugador puede recoger corazones del suelo que le otorgarán una nueva carga para la cura.
- Cuando desbloquee la bomba, el jugador puede pulsar clic derecho para poner una bomba en sus pies que explotará al poco tiempo de ser puesta.

- El jugador puede interaccionar pulsando la F con diversos elementos del escenario como puertas, libros, carteles, escudos, cofres y personajes.
- En el puzle de mover imágenes, el jugador puede arrastrar cualquier imagen de las disponibles a cualquier parte de la pantalla simplemente pulsando clic izquierdo con el ratón y soltándolo cuando quiera.
- En el puzle de introducir texto, el jugador puede pulsar con el clic izquierdo sobre la caja o cajas de texto que haya para poder introducir texto. Una vez dentro de la caja, el jugador puede escribir lo que sea usando un teclado y pulsando cualquiera de sus teclas.
- En el puzle de elegir opción correcta, el jugador puede pulsar en cualquiera de las opciones cuando quiera, solo siendo posible que una esté marcada al mismo tiempo.
- En la prueba de pulsar un botón rápidamente, el jugador puede pulsar dicho botón tan rápido como quiera en cualquier momento.
- En todos los puzles, el jugador puede darle al botón de comprobar respuesta para saber si lo ha resuelto bien o si lo ha resuelto mal.
- En todos los puzles, el jugador puede darle al botón de cerrar puzle en cualquier momento y volver a él en otro momento.

5.4 – Flujo del juego

El flujo del juego es bastante sencillo debido a que es un juego lineal el cual cuenta una historia. Esto significa, el jugado irá pasando de fase en fase, de una en una, hasta llegar al final, sin tener que realmente volver a una fase anterior para nada. Aunque esto tiene una excepción que es la fase cero ya que, al ser la que une todas las fases, el jugador pasa por ella repetidas veces. El flujo es el siguiente:

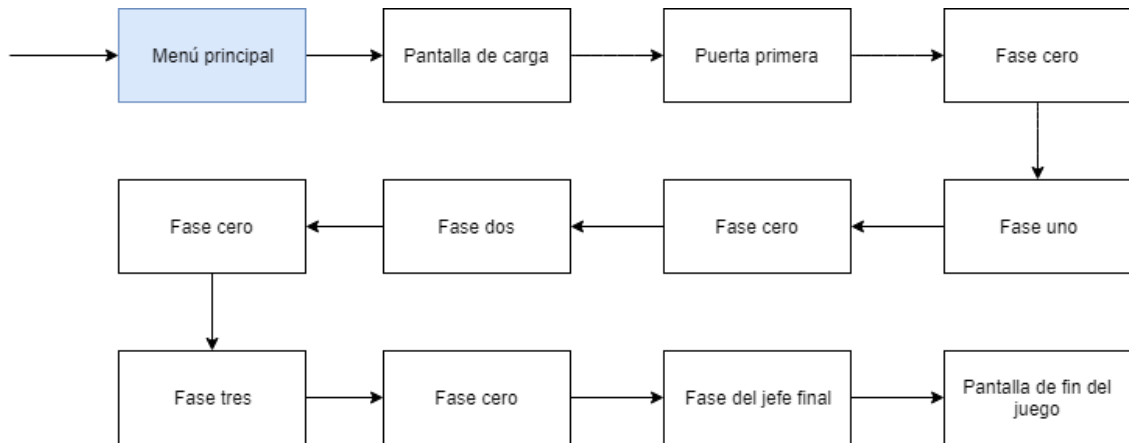


Figura 5.5 - Flujo del juego

Como se ve, el flujo es efectivamente lineal. Aun así, el jugador en cualquier momento puede volver y visitar las fases anteriores en las cuales se puede encontrar un regalo sorpresa.

5.5 – Componentes del proyecto

En este punto se va a ver como está compuesto el proyecto de Unity de este juego, ya que hay mucha variación entre un proyecto software normal y un proyecto software en Unity usando diferentes tipos de recursos.

Un proyecto normal se suele dividir en parte de programación, parte de diseño y parte de arte. En la parte de arte suelen entrar todos los recursos de arte visual, como las imágenes y texturas que se van a usar durante el juego o los modelos 3D, todos los materiales que se encuentren en el juego, los recursos de audio, como sonidos y música, toda la parte de animaciones y la parte del arte técnico, que suelen ser shaders y efectos especiales como los efectos de partículas.

En la parte de programación entran todas las clases y scripts que hagamos para el funcionamiento del juego.

En la parte de diseño entran todas las escenas y su composición, así como los prefabs (objetos definidos de antemano, y guardados, para poder ser usados en repetidas veces y que siempre se mantengan con valores parecidos) que nos encontraremos en la escena, y que iremos generando a lo largo del juego, y toda la configuración de objetos que tengamos en el juego.

El juego se monta en diferentes escenas, cada una representando una parte que decidamos. Dichas escenas se componen de diferentes objetos, que son los que luego tendrán comportamiento cuando les asignemos los diferentes scripts.

A continuación, tenemos un diagrama que nos muestra como está todo compuesto y como se relacionan las cosas unas con otras:

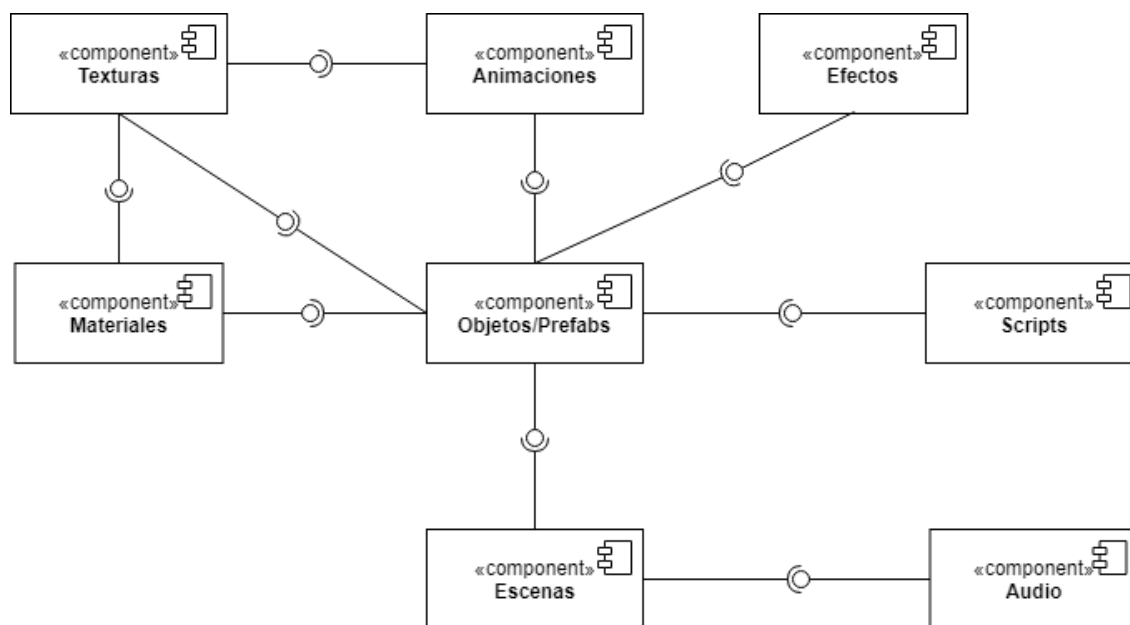


Figura 5.6 - Diagrama de componentes de Unity

Como se puede ver, al final la escena está compuesta de muchos objetos que, a su vez, estos necesitan el resto de recursos que existen tanto de arte como de programación. Como los prefabs son objetos ya configurados y guardados para su múltiple uso, se pueden considerar como si fuesen el mismo componente.

6. Pruebas con usuarios

Este punto se va a dividir en dos apartados: pruebas con usuarios mientras el juego estaba desarrollándose y pruebas con usuarios una vez que el juego estaba hecho.

6.1 – Pruebas durante el desarrollo

Durante todo el desarrollo se han ido haciendo pruebas con distintos usuarios, más cercanos al público objetivo y menos, y, sobre todo, con el cliente cerca para que pudiese ver como estaba yendo el proceso y si le parecía bien. El principal objetivo de estas pruebas era encontrar fallos y errores en la implementación, buscar si faltaba algo que debería estar hecho en ese punto, y ver si el mensaje que se quiere enviar es correcto. Los usuarios con los que se probó el juego durante el desarrollo fueron los mismos todo el rato, alumnos que acaban de estudiar 4º de la ESO, por lo que se consideran público objetivo y el juego debería funcionar para los estudios que tienen ellos.

- La primera prueba se realizó una vez que la fase cero estaba hecha, es decir, las preguntas iniciales y los primeros cinco enemigos. El objetivo de estas pruebas era ver si el combate y el movimiento se entendían y le convencía al cliente, así como si las primeras preguntas eran sencillas de responder. Con estas primeras pruebas descubrimos que el cliente prefería un tipo distinto de cámara que siguiera al personaje y que el movimiento le parecía bien. Además, las preguntas iniciales se cambiaron un poco para que fuese más fácil que la gente pudiese entenderlas y responder correctamente.
- La segunda prueba se realizó una vez que la fase uno estaba realizada y la primera habilidad, el desplazamiento, estaba hecha. Gracias a estas, se encontraron diversos fallos en la implementación y en como funcionaban los enemigos. Respecto al contenido, nos dimos cuenta que hacía falta una pausa entre las diferentes partes de una fase para que le jugador tuviese tiempo a leer que estaba ocurriendo y pudiese prepararse para la siguiente. Respecto a los conceptos, el cliente le pareció bien como estaban las

preguntas y puzles de la fase, y los usuarios pudieron resolverlas de forma sencilla. Respecto a la habilidad, tanto a los usuarios como al cliente les gustó cómo funcionaba.

- La tercera prueba se realizó una vez que la fase dos estaba hecha. Al igual que las otras veces, se encontraron diversos fallos que ocurrían en algunas partes del juego, así como en los enemigos. Respecto a la fase dos, esta estaba bien realizada con los objetivos que queríamos. Las oleadas funcionaban como queríamos y el nivel de dificultad era el justo para que un usuario se lo pasase sólo muriendo una vez en medio de ellas. Luego, a la hora de responder las preguntas sobre las coplas nos dimos cuenta que se ayudaron entre los usuarios para poder sacar la palabra que hacía falta para poder avanzar, lo cual nos sirve para ver que el juego fomenta el comunicarse entre varios para solucionarlo. Lo que también nos dijo este comportamiento es que quizás hacían falta unas pistas para cuando el jugador se quedase atascado en una pregunta. Al cliente le gustó el formato en el que estaban hechas, así como el funcionamiento de las oleadas y los mensajes subliminales de ayuda. Respecto a la pregunta del final, teniendo los conceptos que tenían los usuarios, esta fue sencilla de responder.
- La cuarta prueba se realizó una vez que la fase tres estaba hecha. Al igual que las otras veces, se encontraron diversos fallos que ocurrían en algunas partes del juego, así como en los enemigos. Una de las primeras cosas que se cambiaron al hacer estas pruebas fueron los textos que había. En esta fase es cuando aparece un NPC por primera vez y los textos puestos para este no eran del agrado del cliente, así que se cambiaron varios de estos. Respecto a los conceptos, el cliente le gustó como quedó el planteamiento final de la fase y solo tuvimos que cambiar un par de textos para que estuviese contento con los contenidos. Principalmente, estas pruebas sirvieron para ajustar valores de las estadísticas de los enemigos, porque en momentos en los que había demasiados era más complicado de lo que debería.
- La quinta y última prueba durante el desarrollo fue cuando el jefe final estaba terminado y el resto de habilidades. Al igual que las otras veces, se encontraron diversos fallos que ocurrían en algunas partes de la pelea, sobre

todo una vez que moríamos y teníamos que reintentar el jefe. Tanto la habilidad de la cura como la habilidad de la bomba le parecieron bien al cliente y a los usuarios les pareció intuitivo usarlas, sobre todo conforme la pelea iba avanzando. En esta pelea final no había ningún concepto que aprender, por lo que todo lo que importaba es que la pelea fuese buena. Una de las principales carencias de la pelea era la dificultad, era demasiado sencilla. Después de una sesión entera probando con diferentes valores de daño para el jefe, de enemigos que apareciesen, de cambiar habilidades, llegamos a la conclusión de que el jefe iba a estar separado en varias fases, una en la que tienes que cumplir un objetivo secundario antes de enfrentarte al jefe, y una en la que te pegas cara a cara contra el jefe. Además, cambiamos valores en la cura y le pusimos cargas para que no pudiese ser abusada. También, hicimos que el jefe generase enemigos cada vez que bajaba de un porcentaje de vida. Gracias a todo esto, la pelea del jefe duraba un poco más y era más difícil.

6.2 – Pruebas con el juego completo

Una vez que el juego estuvo completo (con algunos pequeños errores que fueron corregidos rápidamente luego), se le dio a probar a varios usuarios, tanto del público objetivo como usuarios fuera de ese rango, para ver hasta qué punto los conceptos iban a servir y se iban a comprender.

6.2.1 – Público no objetivo

1º - El primer conjunto de personas a las que se les mostró el juego y con el que se probó fue con usuarios en el rango de 25 – 30 años, ya habiendo terminado los estudios y trabajando. Estos usuarios hacía ya muchos años que estudiaron los conceptos que ese presentan en este juego, aunque están acostumbrados a jugar a juegos parecidos, como el anteriormente mencionado *The Binding of Isaac* [5].

Para estos usuarios las preguntas teóricas, como las que están al principio del juego o en la fase dos, les costaron más, haciendo que fuesen a buscarlas en internet o estuviesen debatiendo entre ellos cuál era la correcta. Sin embargo, es verdad que se dieron cuenta antes de los consejos subliminales y de las pequeñas

guías del juego. A la hora de los enfrentamientos, fueron los que más fácil los tuvieron, solo muriendo una vez contra el jefe final y una vez en la fase tres.

Partiendo de la base de que no conocían las obras excepto una, al final de cada fase y al final del juego se les preguntó si se habían enterado de que iba cada obra y cuáles eran los puntos importantes. La mayoría entendió la estructura y la moraleja de Los Milagros de Nuestra Señora, todos entendieron los temas de Coplas a la muerte de su padre y solo algunos reconocieron los tres cantares del Cantar de mio Cid.

2º - El segundo conjunto de personas a las que se les mostró el juego y con el que se probó fue con usuarios en el rango de 40 – 50 años. Al igual que los anteriores, estos usuarios hacía muchos años que estudiaron estos conceptos, si es que los llegaron a estudiar en algún momento. A diferencia del anterior conjunto de usuarios, estos usuarios no están acostumbrados a jugar a juegos de este estilo.

Respecto a los usuarios anteriores, para estos usuarios las preguntas teóricas fueron bastante más sencillas, siendo capaces de responder casi todas en todos los casos, solo fallando alguna pregunta pequeña al principio. En casi ningún caso tuvieron que buscar nada en internet o preguntarse entre ellos. Sin embargo, la parte de combate y enfrentamiento fue bastante complicada, hasta llegar el punto de que algunos no fueron capaces de pasar de la fase dos. Para ello, se cogió un usuario del rango 25 – 30 que hiciese las partes de combate, mientras que dichos usuarios del rango 40 – 50 respondían a las preguntas y a los eventos.

Al igual que con los otros usuarios, al final de cada fase y al final del juego se les preguntó si habían entendido de que iba cada obra. La mayoría comprendió tanto Los Milagros de Nuestra Señora como el Cantar de mio Cid, con algunos fallando en las Coplas a la muerte de su padre.

6.2.2 – Público objetivo

1º - El primer conjunto de personas que se probó fueron aquellos usuarios que acaban de estudiar 3º y 4º de la ESO, que es a los que está enfocado el juego. Estos usuarios tenían los conceptos recientes y se notaba a la hora de resolver los puzzles y reconocer las obras, así como los temas principales de estas.

En la parte de jugabilidad, dependía mucho del usuario. Había usuarios que estaban más acostumbrados a juegos de este estilo para los que el juego fue una experiencia sencilla. Para otros menos acostumbrados a este estilo de juego, fue algo más complicado, llegando a morir varias veces en la fase dos y fase tres. Aun así, una vez que pasaban la fase dos, en la fase tres morían como mucho una vez y en el jefe casi nunca morían, demostrando que se acostumbraban bien al juego y que se adaptaban.

En la parte de los conceptos, todos los usuarios conocían las obras y conocían su estructura. La mayoría recordaba gran parte de las preguntas que venían dentro de cada fase, solo teniendo que pararse un poco más para pensar algunas preguntas del inicio y de la fase dos. En los casos en los que alguno se quedaba un poco más atascado, rápidamente preguntaba a sus compañeros y entre ellos debatían cuál podría ser la respuesta correcta.

Después de cada fase se les preguntó por qué habían aprendido o recordado de cada obra y la mayoría dijo que lo principal ha sido el recordar algunos conceptos que tenían más fríos. Ninguno tuvo que buscar nada en sitios externos para poder responder las preguntas y todos entendieron las estructuras de las obras.

2º - Adicionalmente, se le dio a probar el juego a alumnos que acababan de terminar 2º de Bachillerato, los cuales no entran directamente en el público objetivo, pero están recién salidos de este, así que algo deberían comprender.

En cuanto a jugabilidad pasaba igual que con los de 3º y 4º de la ESO: aquellos que habían jugado a juegos del estilo anteriormente lo pasaron fácil, y aquellos que no empezaron lentamente, pero una vez acabada la fase dos se adaptaron y mejoraron rápidamente.

Respecto a los conceptos, aquí sí que había mayor diferencia. Dependiendo del alumno o alumna, conocían las obras o no, siendo Los Milagros de Nuestra Señora la obra que menos recordaban. Con las preguntas pasaba un poco como con los usuarios del rango de 25 – 30 años, tuvieron que buscar en sitios externos las respuestas a algunas preguntas, pero en menos ocasiones que aquellos de 25 – 30. Sin embargo, en las obras del Cantar de mio Cid y Los Milagros de Nuestra Señora entendieron rápidamente los conceptos, la estructura de la obra y lo que

se quería transmitir. No tan rápido como en el caso del público objetivo, pero casi igual.

Después de cada fase se les preguntó lo mismo y la mayoría dijeron que cuando empezaron la fase no recordaban muy bien la obra o de que iba, pero una vez que la pasaron sabían de qué iba, su estructura y los temas principales. Como excepción, la obra del Cantar de mio Cid todos los alumnos la reconocieron rápido y recordaron su estructura casi al instante.

7. Conclusiones

En este punto vamos a ver si los objetivos planteados al principio del proyecto se han llegado a cumplir, además de ver que se ha aprendido en el proceso de desarrollo. También, se van a ver cuáles han sido las limitaciones del proyecto y las posibles mejoras que se podrían realizar sobre este mismo.

7.1 – Objetivos cumplidos

Este proyecto de fin de grado tiene dos objetivos claros: crear un videojuego serio para PC que busca enseñar al jugador sobre los conceptos principales de la literatura de las épocas de la Edad Media, el Renacimiento y el Barroco, así como sus obras más importantes, y aplicar una metodología ágil, específicamente *Scrum*, al proceso de desarrollo. Vamos a ver si se han cumplido ambos.

7.1.1 – Desarrollo de videojuego serio

En este proyecto se buscaba adquirir la idea de un cliente sobre hacer un juego serio para enseñar conceptos de la literatura española de la Edad Media, el Renacimiento y el Barroco y desarrollarlo, pero ya vemos que de primeras no se ha cumplido todo el objetivo. Aunque el objetivo inicial era que representase las tres eras anteriormente mencionadas, cuando llevábamos poco del desarrollo del juego nos dimos cuenta de que esto iba a ser imposible debido al tamaño del equipo y el tiempo disponible, como se verá más tarde en el apartado de Limitaciones.

Sin embargo, aunque al final el juego solo representa la Edad Media, el juego cumple totalmente su objetivo de enseñar tres de las obras más importantes de dicha era. Después de todas las pruebas hechas con los usuarios de diferente rango con el cliente delante, el cliente le dio el visto bueno y estaba contento con el resultado final. Además, después de haberlo a prueba con gente que no tenía los conceptos de las obras recientes, se ha comprobado que al pasar las fases los usuarios aprendían que iba la obra y sus características principales.

Además, el juego no es restrictivo para ningún usuario. Como dieron resultados las pruebas, incluso sin estar acostumbrado a este estilo de juegos,

conforme vamos jugando y pasando las fases, nos acostumbramos al sistema y al final podemos pasarnos el juego sin importar nuestro historial de jugador, obteniendo así la experiencia completa.

Por lo tanto, aunque al final no tiene todo el contenido que al principio queríamos, el objetivo principal del proyecto ha sido cumplido con éxito.

7.1.2 – Aplicar Scrum al desarrollo

El otro objetivo importante de este proyecto era el aplicar una metodología ágil, específicamente *Scrum*, a un proyecto software grande, para aprender a cómo usarla en un entorno profesional.

Como se ha visto en el punto Metodología usada: *Scrum*, durante todo el desarrollo del proyecto se ha ido aplicando dicha metodología. Desde el principio, se ha ido recogiendo toda la información de todas las reuniones y todos los encuentros con el cliente, apuntando siempre que se hacía en cada reunión y los requisitos del software final, al igual que cuando cambiaban dichos requisitos y a que cambiaban.

Aparte, cuando empezó realmente el desarrollo del trabajo, se empezó a llevar una cuenta de sprints realizados y qué se hizo en cada sprint, qué no se hizo y qué se hará en el siguiente. Se llevó una cuenta diaria sobre qué se había hecho y en qué punto del desarrollo nos encontrábamos.

Debido a la organización llevada y a la documentación realizada, este objetivo también se da como cumplido. *Scrum* ha sido aplicado al 100% a un proyecto de software, en el cual teníamos un cliente y un equipo que controlar.

7.2 – Aprendizaje personal

Una de las cosas principales de este proyecto es el hecho que está realizado con Unity [[Unity, 2019](#)], una tecnología que conocía, pero nunca había trabajado en serio hasta que empecé a trabajar en este proyecto. Por tanto, una de las grandes cosas que he aprendido en este proyecto es a usar la herramienta Unity y poder sacarle el mayor provecho posible. Una fuente grande de aprendizaje vino de la mano del *Máster en Creación de Videojuegos de la UMA* [[Máster, 2019](#)], que cursé mientras realizaba este trabajo.

Respecto al lenguaje de programación, C# ya lo conocía de antes, pero lo había trabajado poco. Durante el grado la mayoría del tiempo he estado usando Java, así que usar C# ha sido una experiencia nueva. Esta ha sido el aprendizaje más sencillo debido a las similitudes entre Java y C#.

Otra de las cosas importantes que he aprendido en este proyecto es justo lo que se ha mencionado en el punto anterior: el uso de *Scrum*. Hasta este proyecto conocía la teoría de *Scrum* y sabía cómo había que usarlo, pero nunca lo había llegado a poner en práctica realmente, y menos aún con un cliente. Que es el último punto.

El último punto valioso de aprendizaje de este proyecto es el trabajar con un cliente, siendo este el que tiene la idea principal del proyecto y el que elige que requisitos serán los que tendrá el software al final. Lo normal durante el grado es hacer un software aplicado según lo que los profesores eligen o sobre lo que el alumno elige, pero en ningún caso es lo mismo que tener un cliente externo que no entiende como se hace el software y que simplemente sabe cuál es su idea. Este es uno de los puntos de aprendizaje más valiosos de todo este proyecto.

7.3 – Limitaciones y problemas

Como ya se ha comentado varias veces durante este documento, la principal limitación de este proyecto ha sido el tamaño del equipo y el tiempo. Originalmente, el juego debería haber encapsulado las tres eras de la Edad Media, el Renacimiento y el Barroco, pero cuando el desarrollo llevaba poco tiempo, tanto el cliente como yo nos dimos cuenta de que era un objetivo imposible.

Otra de las limitaciones importantes del juego, aunque no tan relacionado con la parte de la programación, es el arte. Se suponía que un grupo de personas conocidas del cliente iban a realizar el arte y la música del juego, pero al final la mitad de estas personas no llegaron a realizar nada o realizaron poca cosa. Debido a eso, el juego tiene pocas o ninguna animación y la ambientación de las fases está peor conseguida. Además, la música es la misma para casi todo el juego y el menú principal no tiene música. Todas estas cosas están fuera de mi rango de trabajo, pero igualmente limitan al producto final.

Aunque las limitaciones anteriores son importantes, la realidad es que muchos de los problemas que han surgido debido a ellas se hubieran podido solucionar con un diseño de juego profesional. La gran limitación de este proyecto es que el cliente no conoce conceptos profesionales del diseño de videojuegos y eso se ha notado en el diseño final. Con recientes conocimientos sobre el diseño de videojuegos aprendido en el *Máster en Creación de Videojuegos* [[Máster, 2019](#)], he intentado corregir el diseño y que las especificaciones se pudiesen adaptar lo mejor posible, pero, en general, he tenido que seguir las especificaciones del cliente y no había mucho margen de cambio. Como se comentará en el siguiente punto, si esta tarea hubiese sido realizada por un profesional en el diseño de videojuegos, el resultado final mejoraría bastante.

Respecto a los problemas, uno de los principales problemas que surgieron en el desarrollo del juego, y que tardó varias semanas en ser resuelto, es el hecho de que cuando un enemigo moría, si el jugador moría al mismo tiempo, dicho enemigo se quedaba en un estado de muerte continuo por lo que nunca más podía ser derrotado y el juego no podía avanzar. Esto ocurría debido a que la animación de muerte del enemigo se cortaba en mitad y no llegaba a llamar al método que estaba al final de la animación. Esto fue algo que aprendí en los últimos días de desarrollo, y, si hubiese conocido mejor como funcionaban las animaciones en Unity, posiblemente este error se podría haber corregido mucho antes.

Durante el desarrollo se encontraron otra serie de problemas, pero la mayoría de estos fueron cosas menores que, conforme se fueron descubriendo, se fueron corrigiendo. El único error grande que encima limitaba la jugabilidad era el anteriormente nombrado.

7.4 – Posibles mejoras

Las posibles mejoras claras para este proyecto es cumplir aquellos objetivos que se eliminaron al principio del desarrollo. El juego se puede ampliar a múltiples eras, tanto anteriores como posteriores, ya que la estructura de las otras eras serían la misma y solo habría que cambiar el contenido de estas.

Hablando sobre lo mismo, otra de las posibles mejoras es añadir nuevas fases al juego ya existente. El sistema de fases, subida de niveles, de puzzles, de pruebas, de enemigos, etc, está ya realizado. Teniendo toda la estructura ya montada,

añadir nuevas fases es añadir nuevas salas y cambiar los textos y los objetivos. Al mismo tiempo que se habla de añadir nuevas fases, se puede hablar de añadir nuevas habilidades, nuevos enemigos y nuevos jefes finales.

Otra de las posibles mejoras para el juego es el hacer que sea jugable en cualquier navegador web. Esto no costaría demasiado y sería fácilmente aplicable. En el mismo rango, hacer que el juego sea jugable en Android. Esto llevaría bastante más trabajo ya que el sistema de movimiento y combate tendría que cambiar para estar adaptado a Android.

Otra mejora posible sería añadir el poder jugar con un mando en vez de ratón y teclado solamente, ya que hay jugadores que se encuentran más cómodos jugando con un mando que con teclado y ratón.

Como última mejora notable, si se hubiera contado con un profesional en el diseño de videojuegos, todo el diseño general de juego cambiaría a mejor y el producto final sería de mejor calidad.

Referencias

- [Action RPG Wiki, 2019] "Action role-playing game",
https://en.wikipedia.org/wiki/Action_role-playing_game, (Fecha de acceso 05/09/2019)
- [Agilemanifesto, 2019] "Manifiesto for Agile Software Development",
<http://agilemanifesto.org/>, (Fecha de acceso 06/09/2019).
- [Atlassian Scrum, 2019] "Scrum - what it is, how it works, and why it's awesome", <https://www.atlassian.com/agile/scrum>, (Fecha de acceso 06/09/2019).
- [Dörner y otros, 2016] Dörner, R., Göbel, S., Effelsberg, W., Wiemeyer, J. (Eds.). "Serious Games: Foundations, Concepts and Practice". *Springer International Publishing*. 2016.
- [draw.io, 2019] "draw.io", <https://www.draw.io/>, (Fecha de acceso 04/09/2019 y 07/09/2019).
- [EL PAÍS Economía, 2019] "Por qué los videojuegos pueden ayudar a educar | Economía | EL PAÍS",
https://elpais.com/economia/2017/12/11/actualidad/1513005770_763013.html, (Fecha de acceso 10/09/2019).
- [Game learn Serious games, 2019] "Serious games examples that explain all you need to know", <https://www.game-learn.com/all-you-need-to-know-serious-games-game-based-learning-examples/>, (Fecha de acceso 06/09/2019).

- [GitHub, 2019] "The world's leading software development platform. GitHub",
<https://github.com/> (Fecha de acceso 06/09/2019)
- [Growthengineering 10 serious games, 2019] "10 Serious Games that Changed the World", <https://www.growthengineering.co.uk/10-serious-games-that-changed-the-world/>, (Fecha de acceso 06/09/2019)
- [Lasa et al., 2017] Lasa, Carmen et al. "Métodos Ágiles. Scrum, Kanban, Lean". ANAYA, (2017).
- [Máster, 2019], "Máster en Creación de Videojuegos",
<http://www.mastervideojuegos.uma.es/>, (Fecha de acceso 11/09/2019)
- [Path of Exile Wiki, 2019] "Path of Exile",
https://en.wikipedia.org/wiki/Path_of_Exile, (Fecha de acceso 05/09/2019).
- [Portal Wiki, 2019] "Portal (video game)",
[https://en.wikipedia.org/wiki/Portal_\(video_game\)](https://en.wikipedia.org/wiki/Portal_(video_game)), (Fecha de acceso 11/09/2019)
- [Serious game Wiki, 2019] "Serious game",
https://en.wikipedia.org/wiki/Serious_game, (Fecha de acceso 06/09/2019).
- [Skarin, 2015] Mattias Skarin. "Real-World Kanban: Do Less, Accomplish More with Lean Thinking (Inglés)", *Pragmatic Bookshelf*; Edición: 1 (3 de Julio de 2015)
- [The Binding of Isaac Wiki, 2019] "The Binding of Isaac (videogame)",
[https://en.wikipedia.org/wiki/The_Binding_of_Isaac_\(video_game\)](https://en.wikipedia.org/wiki/The_Binding_of_Isaac_(video_game)),
(Fecha de acceso 05/09/2019)

[The Legend of Zelda Wiki, 2019] "The Legend of Zelda",
https://en.wikipedia.org/wiki/The_Legend_of_Zelda, (Fecha de acceso
05/09/2019).

[Twitter Clawster] "Claudia on Twitter, Hilo sobre los beneficios de jugar a
videojuegos en niños",
https://twitter.com/Clawser_/status/1171020953368825857, (Fecha de
acceso 11/09/2019)

[Unity, 2019] Unity 3d. Desarrollo de videojuegos. Game Engine.
<https://unity.com/es> (Fecha acceso 30/05/2019)

Apéndice 1

A1 – Manual de Instalación

Forma de instalación y requerimientos:

Una vez que tenemos el videojuego desarrollado en Unity no hace falta ningún método de instalación debido a que Unity directamente nos da un ejecutable. Lo único que necesitamos es un ordenador con un teclado y ratón y conexión a internet para descargarlo.

El juego se puede descargar desde:

<https://github.com/RedScoresby/InculturaTFG/releases>

Necesitarás estar en la parte de releases, bajarte el .zip de la última versión y ejecutar el archivo que se llama Incultura.

Apéndice 2

A2 – Documento de Diseño de Juego (GDD)

A continuación, viene el Documento de Diseño de Juego, con todos sus puntos y todas las imágenes usadas durante el proceso de diseño. Algunos de los puntos ya han sido explicados anteriormente en el apartado de Problema y Diseño de la solución.

1 – Descripción

Incultura es un juego serio del género RPG de acción. Está basado en las diferentes obras de la literatura española de la Edad Media. El jugador tendrá que avanzar por las diferentes fases del juego, las cuales representarán una obra literaria cada una, aprendiendo los rasgos principales de la obra para completarla y liberando dichas obras de los enemigos incultos que la están arruinando.

El objetivo principal de *Incultura* es ayudar al aprendizaje y servir como recordatorio de las diferentes obras importantes de la literatura española de la Edad Media. El contexto es que ha surgido un ente, la *Malvada Incultura*, la cual quiere destruir todas las obras de la literatura española y convertir a la gente en incultos. Para evitarlo, debes pasar por las diferentes obras derrotando a los incultos y culturizándolos, liberando así las obras del yugo de la *Malvada Incultura*.

Incultura está compuesto de tres obras o fases principales y una pelea contra el jefe final. Todas estas fases se desarrollan en el mismo escenario, solo que cada una de ellas estará en una de las salas disponibles. Dependiendo de la fase, los

objetivos son diferentes, teniendo que resolver puzles y preguntas en algunos casos, hasta derrotar a enemigos en otros.

El combate es continuo, siendo capaz el jugador de entrar y salir del combate cuando quiera, pudiendo huir de los enemigos cuando esté en peligro. El jugador ataca principalmente a distancia, aunque puede usar otras habilidades, que conseguirá cuando avance en el juego, para poder derrotar a sus enemigos. Respecto a los enemigos, dependiendo del tipo de enemigo que sea, su ataque será a distancia o cuerpo a cuerpo, haciendo que el enfrentamiento varíe de unos a otros.

El jugador comienza la partida en una sala principal desde la cual puede acceder a las diferentes fases del juego, que tiene que recorrer en un orden específico. La primera fase del juego es la correspondiente a la obra *Los Milagros de Nuestra Señora*, la segunda fase corresponde a la obra *Coplas a la muerte de su padre*, y la tercera, y última, corresponde al *Cantar de mio Cid*. La última fase es una pelea contra el jefe final, la *Malvada Incultura*.

Conforme vayamos avanzando en el juego, iremos adquiriendo niveles, que aumentarán nuestra vida y nuestro daño, al mismo tiempo que iremos adquiriendo nuevas habilidades para ayudarnos a superar algunas tareas.

2 – Características técnicas

Este juego está pensado para lanzarse en PC, tanto en formato descargable como en Web.

El juego se juega con ratón y teclado. El ratón nos permite movernos por menús, resolver puzles y atacar, principalmente. Con el teclado nos podremos mover, usar diferentes habilidades y responder preguntas, principalmente. Nos moveremos por el escenario con WASD o con las flechas de las direcciones, atacaremos pulsando el clic izquierdo del ratón, y lanzaremos habilidades con las teclas del teclado y pulsando el clic derecho del ratón.

Durante todo el juego, la cámara nos seguirá y nos permitirá ver que hay a nuestro alrededor, incluso, en algunos casos, darnos una pista de que viene a continuación.

3 – Experiencia de juego

El juego está enfocado para los jugadores de edad entre 14 y 17 años, sobre el momento en el que en la enseñanza se da la literatura de la Edad Media (sobre 3º de la E.S.O.). El juego cuenta con un solo modo de juego en local, por lo que está pensando para una experiencia de un solo jugador. No obstante, aunque solo pueda jugar un solo jugador al mismo tiempo, puede tener a más personas cerca para ayudarle a resolver algunas de las preguntas y puzles que se encuentran a lo largo del juego.

Los tipos de jugadores objetivo de este juego son todos aquellos que quieran resolver retos, puzles, y que quieran mejorar su personaje y tener enfrentamientos difíciles. También tiene algún que otro componente para todos aquellos jugadores que les gusta más explorar.

Como juego que es, *Incultura* intentará transmitir una experiencia divertida para todos los jugadores que lo jueguen, aunque no es su objetivo principal, aunque es uno que intentará conseguir. Debido a su naturaleza de juego serio, *Incultura* intentará enseñar algo nuevo, o recordar algo ya sabido, a todas aquellas personas que jueguen, ya sea una fase o el juego entero.

4 – Jugabilidad

El jugador tiene disponibles las siguientes acciones, que puede realizar en cualquier momento: puede moverse por el escenario a su gusto, puede disparar cuando quiera y hacia donde quiera, puede poner en pausa el juego y volver al menú principal, y, una vez que desbloquee las habilidades, puede usar todas ellas siempre que estén disponibles y fuera de enfriamiento. El jugador puede interactuar con una serie de elementos, principalmente para acceder a diferentes puzles o pruebas, y con algún que otro NPC que le ayudará en la historia.

El juego presenta una serie de retos que el jugador deberá superar para pasarse el juego. Dichos retos van de menor a mayor dificultad, siendo los primeros enfrentamientos contra pocos enemigos y con facilidad de superar, luego están enfrentamientos con más enemigos y de diferente tipo, hasta llegar al reto más difícil: la batalla contra el jefe final.

4.1 – Habilidades

En el juego existen tres habilidades diferentes que se van desbloqueando conforme nos vayamos pasando las fases:

- Desplazamiento. El desplazamiento es la primera habilidad que conseguimos. Este nos permite desplazarnos rápidamente en la dirección que nos estemos moviendo, para así poder esquivar algunos ataques o llegar antes a nuestro destino. El desplazamiento tiene un segundo de enfriamiento, durante el cual no podemos usar la habilidad.
- Cura. La cura es la segunda habilidad que desbloqueamos. La cura nos permite recuperar un punto de vida que hayamos perdido anteriormente contra los enemigos. Esta habilidad funciona con un sistema de cargas y enfriamiento: empezamos con dos cargas de cura y cada vez que usemos la habilidad iremos gastando dichas cargas. Una vez que llegemos a cero, no podremos usar de nuevo la habilidad. Sin embargo, por el escenario hay unos objetos que nos dan una carga de cura, y, durante la pelea contra el jefe final, aparecen más objetos de cura para ayudarnos. Además de las cargas, la cura tiene un enfriamiento de ocho segundos, siendo así la habilidad más restrictiva del juego, debido a que haría los enfrentamientos demasiado sencillos si no fuese por esto.
- Bomba. La bomba es la tercera y última habilidad. La bomba nos permite dejar una bomba en el sitio en el que estemos y que un segundo después explote, haciendo daño a todos los enemigos que estén alrededor. Es muy útil cuando somos perseguidos por muchos enemigos, y es necesaria para poder hacer la primera fase del jefe final. La bomba tiene un enfriamiento de cinco segundos.

4.2 – Niveles del jugador

El jugador empieza el juego al nivel cero, teniendo cuatro corazones de vida y haciendo un solo punto de daño cada vez que ataca. Conforme vaya avanzando por el juego, conseguirá más niveles hasta llegar a un máximo de nivel 3. Cada vez que suba de nivel será curado completamente, aumentará su vida máxima un corazón, y su daño de ataque aumentará un punto. De esta forma, los enemigos

que al principio le costaban más serán siendo más sencillos conforme avance la partida.

4.3 – Tipos de enemigo

Durante el progreso del juego nos encontraremos dos enemigos principales: el enemigo que golpea cuerpo a cuerpo y más tonto, y el enemigo que pega a distancia, pero algo más listo.

4.3.1 – Enemigo cuerpo a cuerpo

El enemigo cuerpo a cuerpo tiene tres puntos de vida y cada ataque suyo hace medio punto de daño. Este enemigo intentará perseguir al jugador siempre que esté en su rango de alcance, y cuando esté lo suficientemente cerca le golpeará.

4.3.2 – Enemigo a distancia

El enemigo a rango tiene dos puntos de vida y cada ataque suyo hace un punto de daño. Este enemigo perseguirá al jugador una vez que entre en su rango de ataque, pero siempre manteniendo una distancia para atacarle. Una vez que el jugador se vaya a acercar, este enemigo intentará escapar en direcciones aleatorias para dificultar golpearle, manteniendo una distancia y atacando al mismo tiempo.

4.3.3 – El jefe final: la *Malvada Incultura*

La *Malvada Incultura* es el jefe final del juego y solo aparece en la última batalla, aunque ya sabemos de ella nada más empezar la partida. En la batalla, la *Malvada Incultura* tiene 300 puntos de vida y sus ataques hacen un punto de daño. Sus ataques son a distancia, como los de los enemigos a distancia, solo que, a diferencia de estos, sus ataques son aleatorios y mucho más rápidos, para que el jugador tenga que esquivar más en vez de solo correr. Además de esto, el jefe contará con tres habilidades diferentes, y la pelea se diferenciará en dos fases, con pequeños eventos en una de ellas. Sin importar en qué fase estemos, durante toda la batalla se irán generando corazones que nos darán cargas para la Cura.

4.3.3.1 – Habilidades

Además de atacar, la *Malvada Incultura* cuenta con tres diferentes habilidades, que irá desbloqueando conforme el enfrentamiento vaya avanzando.

- Pilar de fuego. El jefe hace aparecer un pilar de fuego en una zona del escenario, y se queda ardiendo hasta que la pelea acaba. El jefe genera dos nuevos pilares cada cinco segundos hasta llegar a un máximo de 25 pilares. Si el jugador pasa por encima del fuego pierde medio corazón por cada segundo que esté encima.
- Rayo cargado. El jefe lanza un rayo grande en dirección al jugador. Dura cinco segundos y si el jugador pasa por el rayo pierde dos corazones. Tanto cuando sale como cuando desaparece, el rayo tarda un poco en cargarse, así que el jugador tendrá tiempo de esquivarlo si está atento. Puede haber más de un rayo al mismo tiempo.
- Lanzar cristal grande. El jefe hace caer un cristal grande desde el techo en la posición del jugador. Si este es golpeado pierde dos corazones. Al poco de caer, el cristal desaparece. El jefe irá lanzando cristales cada cuatro segundos.

4.3.3.2 – Fases

1^a - Cristales de la inmortalidad

Esta es la primera fase del jefe final. Una vez empiece el encuentro el jefe empezará a atacarnos mientras que va creando Pilares de fuego por toda la zona. En esta fase el jefe es inmune a daño debido a cuatro cristales que están repartidos por el escenario. Para poder hacerle daño al jefe, el jugador tiene que romper estos cuatro cristales usando la Bomba. Sin embargo, esta tarea no será tan sencilla: cada cristal está guardado por tres enemigos a distancia que nos lo pondrán difícil. Una vez que hayamos destruido los cuatro cristales, la siguiente fase empieza.

2^a - Enfrentamiento

Esta es la segunda y última fase del jefe en la que te enfrentas directamente contra ella. Una vez que la fase empiece, el jefe empezará a usar también la habilidad de Rayo cargado. Durante esta fase, el jefe no para de atacar y lanzar rayos todo el rato. Además, dependiendo de en qué porcentaje de vida se encuentre ocurrirán más cosas.

Cuando el jefe baje del 75% de vida se activará la habilidad de Lanzar cristal grande. Además, se generarán cuatro enemigos en el escenario: dos de cuerpo a cuerpo y dos de distancia.

Cuando el jefe baje del 50% de vida se generarán seis enemigos: dos de cuerpo a cuerpo y cuatro de distancia. Además, el enfriamiento del Rayo cargado y de Lanzar cristal grande bajará considerablemente.

Cuando el jefe baje del 25% de vida se generarán los últimos enemigos: tres de cuerpo a cuerpo y seis de distancia.

4.4 – Puzles y pruebas

En este juego tenemos cuatro tipos diferentes de puzles, o pruebas de no combate, que el jugador deberá resolver y superar para poder avanzar en la trama. Son los siguientes:

- Puzle de introducir texto: en este puzle el jugador deberá deducir que palabra o palabras son las que encajan en el hueco en blanco dependiendo del texto que haya o la pregunta mostrada.
- Puzle de mover imágenes: un puzle clásico. El jugador deberá mover las piezas y colocarlas en su lugar para poder avanzar.
- Puzle de elegir la opción correcta: en este puzle el jugador deberá elegir la elección correcta entre varias que se le mostrarán dependiendo de la pregunta que se le haga.
- Prueba de pulsar rápido: en esta prueba el jugador deberá pulsar una tecla rápidamente para vencer la velocidad a la que sube la barra del oponente.

5 – Flujo de juego

El juego se desarrolla de manera que las fases se van complicando poco a poco, empezando por un nivel de dificultad bajo en la entrada y fase primera, y llegando a un nivel de dificultad alto en la batalla final. Esto se controla principalmente

por el tipo de enemigo y el número de enemigos que hay en cada fase. Al principio solo existen enemigos de cuerpo a cuerpo, que son más sencillos de derrotar, y no habrá muchos de ellos. Conforme vayamos pasando de fase, los enemigos de cuerpo a cuerpo irán desapareciendo y empezarán a salir más enemigos de distancia, además de que su número aumentará.

Para que esta subida de dificultad no se haga imposible de escalar, al mismo tiempo que el jugador va superando fases irá subiendo de nivel, adquiriendo más vida y haciendo más daño, además de consiguiendo nuevas habilidades. De esta forma, aunque los enfrentamientos son más complicados, siguen siendo posibles de superar.

Para acompañar la subida de dificultad de cada fase, los puntos de guardado serán más generosos conforme vayamos pasando las diferentes fases. Con la primera el punto de guardado está al principio, en la segunda existen dos puntos de guardado, uno al principio y otro en medio, en la tercera fase existe un punto de guardado continuo cada vez que avanza un poco. Ya con la pelea final sí que existe un solo punto de guardado, en el cual si mueres en la pelea tendrás que repetirla entera.

6 – Diseño de niveles

Los niveles estarán todos unidos en el mismo escenario, conectados desde una sala principal. Cada nivel consta de una o dos salas en las que tenemos que resolver una serie de pruebas. La vista de dichos niveles es desde arriba, siendo capaces de ver casi toda la sala al mismo tiempo. En dichas zonas puede haber obstáculos e impedimentos que nos dificulten la movilidad en algún momento.

Específicamente, los tres niveles o fases son los siguientes:

6.1 – Primera fase. Los Milagros de Nuestra Señora

- Tipo de espacio. Narrativo - Confrontación. El nivel cuenta sobre que va la obra y sus puntos importantes, mientras que te enfrentas a algunos enemigos.
- Tema. Acción y resolución de pruebas.

- Ambientación. Ambientado como en la obra que representa, con toques religiosos.
- Narrativa. La obra ha sido corrompida por la *Malvada Incultura* y tienes que salvarla. Tienes que ayudar al devoto a superar sus problemas y darse cuenta de que siempre debe confiar en la Virgen.
- Objetivo. Ayuda al devoto, aprende la estructura de los fragmentos de la obra y la moraleja de cada fragmento.
- Jugabilidad. Resuelve puzzles y derrota los enemigos para poder avanzar.
- Mecánicas. Moverse, atacar, mover piezas para formar una imagen, y elegir una opción entre varias en una pregunta.
- Número de enemigos. Inicialmente cinco enemigos de cuerpo a cuerpo, aunque luego saldrán indefinidos hasta vencerte.

En esta primera fase el jugador entra por un pasillo idílico, haciendo referencia a la primera parte de la obra, que da lugar a una sala con toques religiosos. Aquí debe resolver un puzzle para encontrarse con el devoto, al cual tiene que ayudar a derrotar a unos enemigos. Una vez derrotados, el devoto tiene más problemas, pero tú solo no puedes con todos ellos. Cuando te derrotan, la Virgen viene a salvarte y luego aparece un cofre con la pregunta sobre cuál es la moraleja de la historia.

Todo esto hace referencia a la estructura que tienen los diferentes fragmentos que hay dentro de la obra: introducción alegórica, presentación del devoto, problemas del devoto, intervención de la Virgen, y moraleja.

6.2 – Segunda fase. Coplas a la muerte de su padre

- Tipo de espacio. Narrativo - Confrontación. El nivel cuenta sobre que va la obra y sus puntos importantes, mientras que te enfrentas a algunos enemigos.
- Tema. Acción y resolución de pruebas.
- Ambientación. Ambientado como en la obra que representa, con toques de muerte.

- Narrativa. La obra ha sido corrompida por la *Malvada Incultura* y tienes que salvarla. Tienes que aprender los temas principales de la obra mientras lees algunas de las famosas coplas.
- Objetivo. Aprende los cuatro temas principales de la obra, así como la estructura de las coplas y el número de coplas que hay.
- Jugabilidad. Resuelve puzzles y derrota los enemigos para poder avanzar.
- Mecánicas. Moverse, atacar, usar habilidades, explorar el escenario, rellenar con una palabra un hueco y elegir una opción entre varias en una pregunta.
- Número de enemigos. Hay cuatro oleadas de enemigos, cada una compuesta por 10 enemigos. Dependiendo de la oleada habrá más de cuerpo a cuerpo o de distancia.

En esta segunda fase el jugador entra en una sala en la que hay un atril con un libro y unas tumbas. Al interactuar con el atril vemos una de las coplas de la obra, la cual tenemos que completar con uno de los temas principales de la obra. Después de esto, tenemos que derrotar a diez enemigos para poder acceder a la siguiente copla. Tenemos que repetir esto cuatro veces, que son los cuatro temas de la obra: fama, tiempo, muerte y vida. Una vez acabemos con las oleadas, se nos hace una pregunta sobre la estructura de las coplas y el número de coplas que hay en dicha obra.

6.3 – Tercera fase. Cantar de mio Cid

- Tipo de espacio. Narrativo - Confrontación. El nivel cuenta sobre que va la obra y sus puntos importantes, mientras que te enfrentas a algunos enemigos.
- Tema. Acción y resolución de pruebas.
- Ambientación. Ambientado como en la obra que representa, con toques de caballeros y castillos.
- Narrativa. La obra ha sido corrompida por la *Malvada Incultura* y tienes que salvarla. Tienes que ayudar al Cid a recuperar su honra y que el Rey vuelva a confiar en él.

- Objetivo. Aprende la historia del Cid, así como los diferentes cantares que existen dentro de la obra y lo que pasa en estos.
- Jugabilidad. Resuelve puzzles y derrota los enemigos para poder avanzar.
- Mecánicas. Moverse, atacar, usar habilidades, explorar el escenario, coger objetos, elegir una opción entre varias en una pregunta, y pulsar rápidamente una tecla para vencer una prueba.
- Número de enemigos. Cada vez que falles con el escudo saldrán cuatro enemigos (2 cuerpo a cuerpo, 2 a distancia). En la siguiente mini fase hay 12 de distancia. En la mini fase final hay nueve enemigos (3 cuerpo a cuerpo, 6 de distancia).

En la tercera fase el jugador nada más entrar se encuentra con el Cid, el cual le sirve como guía durante la fase y es el que le dice que tiene que hacer durante las mini fases que hay. Primero debemos encontrar el escudo del Cid para devolverle su honra y después tenemos que conseguir una serie de regalos para que el Rey vuelva a aceptar al Cid. Más adelante, nos toca enfrentarnos a la guardia de los Infantes de Carrión que maltratan a las hijas del Cid, todo para que luego los venzamos en un juicio.

Cada parte de la fase hace referencia a uno de los cantares del Cid: Cantar del destierro (recuperar la honra), Cantar de las bodas (regalos al Rey), y Cantar afrenta de Corpes (vencer a los Infantes).

7 – Cámara, controles e interfaz

La cámara del juego es una cámara casi enteramente cenital, con un poco de perspectiva. Todo el rato sigue al jugador y con suficiente rango como para poder ver todo lo que ocurre a nuestro alrededor. Menos en la sala inicial, en el resto abarcamos casi toda la sala con la cámara siempre.

El jugador tiene un control total del personaje en todo momento que no sea una pregunta o puzzle, durante los cuales el personaje estará quieto y no puede hacer nada hasta acabar dicha pregunta o puzzle. El jugador se puede mover con WASD o con las flechas direccionales y atacar pulsando el clic izquierdo del ratón. Para las habilidades, el jugador puede usar el Desplazamiento pulsando Espacio

mientras que se mueve en cualquier dirección. Para usar la Cura, el jugador puede pulsar la E y gastar una carga de cura. Para la Bomba, el jugador puede pulsar el clic derecho del ratón para poner una bomba a sus pies que explotará en un segundo y medio.

En todos los menús, el jugador puede moverse por ellos con el ratón y pulsando en la opción que quiera. En los puzles y pruebas igual, el jugador puede usar el ratón para resolverlos y el teclado para introducir las respuestas a aquellos puzles o pruebas que requieran una entrada de texto.

La interfaz es bastante minimalista. El menú principal consta del título y tres botones: uno para jugar, otro para ver los créditos del juego y otro para salir del juego. Una vez dentro del juego tenemos en la parte superior izquierda la vida y abajo a la derecha las habilidades que tengamos en ese momento. En alguna fase, como el jefe final, tenemos algo más, pero no mucho. En el menú de pausa, muerte y finalizar tenemos dos botones, uno que nos lleva al menú de vuelta y otro que, dependiendo de cuál sea, hace una cosa u otra: volver a la partida en el de pausa, cargar desde el último punto de guardado en el de muerte, y volver a jugar en el de finalizar.

Quitando menús y personaje, durante toda la partida tendemos un posible texto en la parte baja y centro de la pantalla donde nos saldrá texto informativo y que nos ayuda en algunos momentos.

Para cada puzle tenemos una interfaz específica, aunque todos ellos cuentan con un botón de salir y un botón de comprobar respuesta:

- Puzle de introducir texto: tenemos un texto en la parte superior que es una pregunta o indicativo y un campo de texto a rellenar respecto a la pregunta o preguntas.
- Puzle de mover imágenes: tenemos diferentes imágenes que tenemos que colocar en su lugar.
- Puzle de elegir opción correcta: tenemos una lista de marcadores asociados a textos entre los que tenemos que elegir.
- Prueba de pulsar rápidamente: tenemos dos barras que van subiendo, una asociada al oponente y otra a nosotros, además de un botón que nos sirve para aumentar un poco la barra.
-

7.1 – Flujo de pantallas

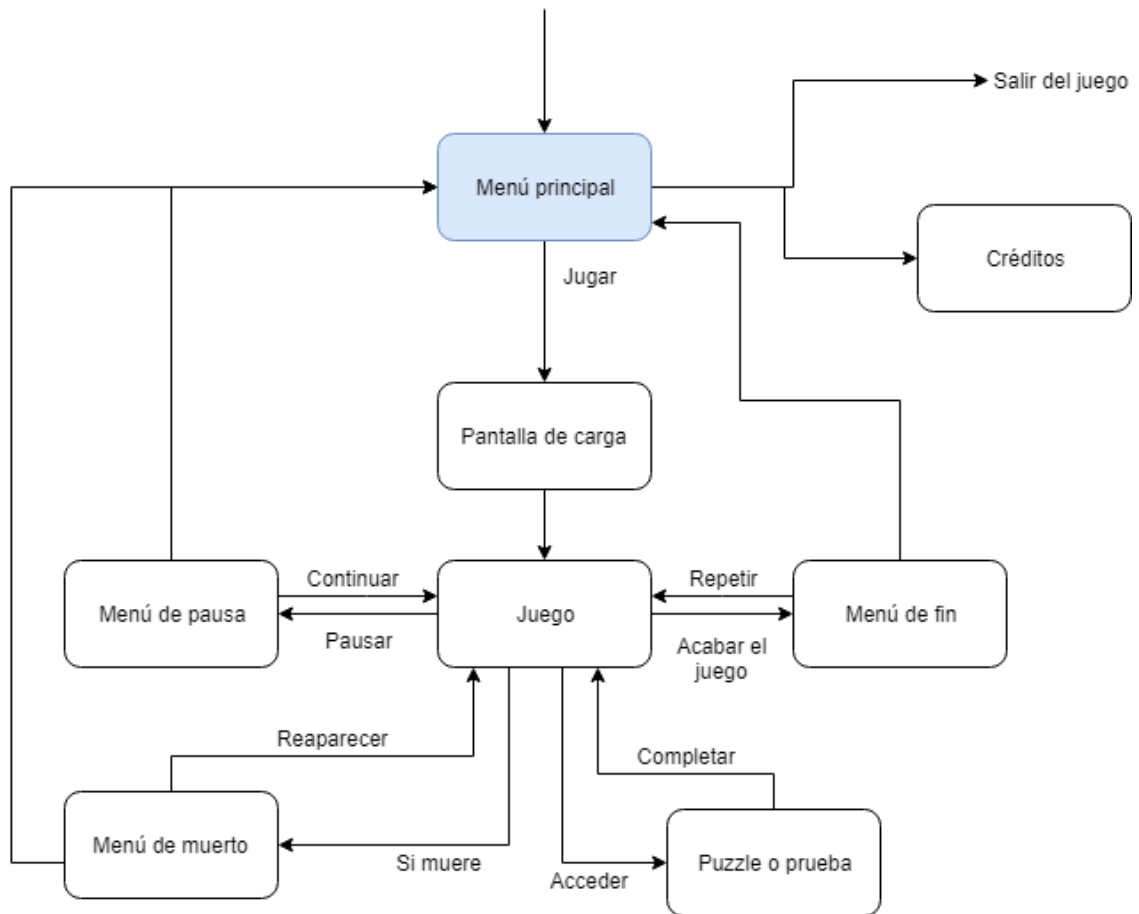


Figura 7.1 - Flujo de interfaz

Siguiendo el objetivo de tener la interfaz simple, el juego tiene pocas pantallas realmente. O estás en el menú principal o estás dentro del juego, en cuyo momento ya no cambias de pantalla en ningún momento. En todo caso aparecerá un mensaje por encima o la interfaz de un puzzle, pero en ningún momento cambiamos de interfaz de juego. Por esta misma razón se intentó que la interfaz propia del juego fuese lo más simple posible.

7.2 – Menú principal



Figura 7.2 - Esquema del menú principal

El objetivo del menú principal era mantenerlo simple y que se viese claramente las opciones que tienes, pudiendo jugar directamente sin pensarlo mucho.

7.3 – Créditos

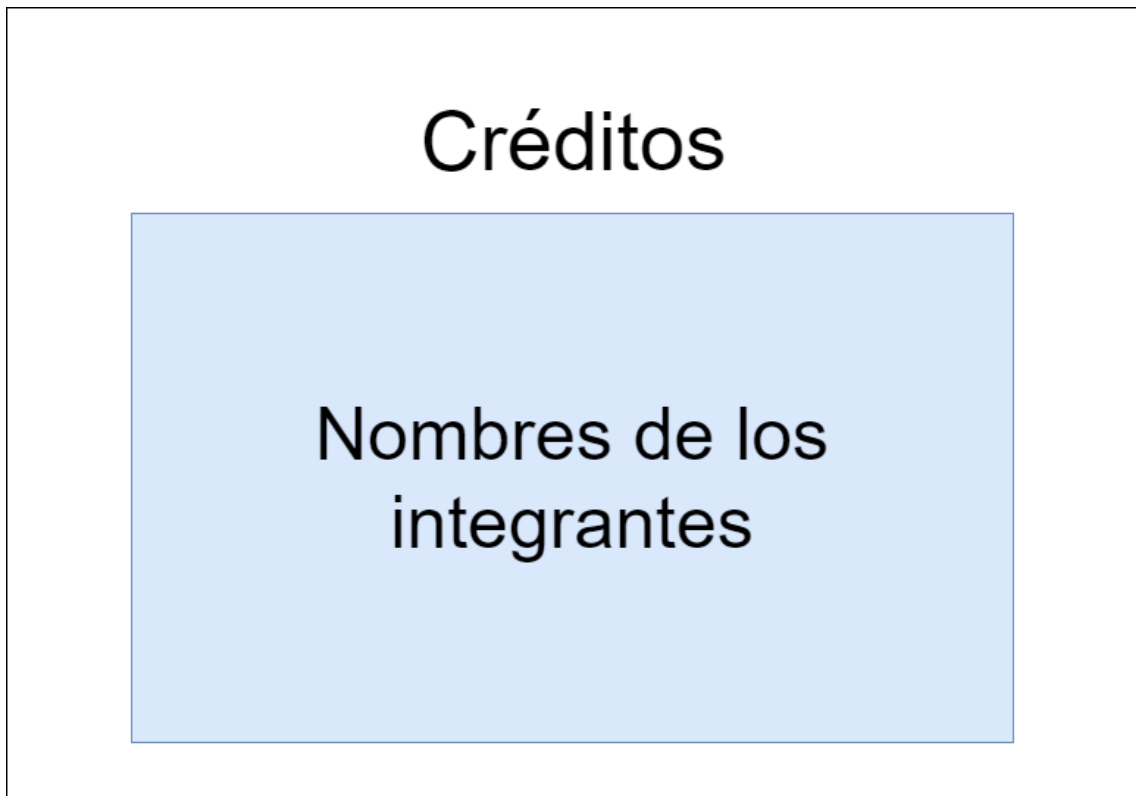


Figura 7.3 - Esquema de los créditos

Para los créditos se buscaba también algo sencillo que incluyese el nombre de todos los que hemos trabajado en el juego.

7.4 – Menú de pausa, menú de muerte y menú de fin del juego



Figura 7.4 - Esquema base de los menús

Tanto el menú de pausa, como el menú de muerte, como el menú de fin del juego, tienen la misma estructura. Buscamos un botón que nos devuelva a la acción y un botón que nos lleve de vuelta al menú principal por si queremos salir.

En el caso del menú de pausa, el botón de acción nos devolverá a la partida donde estábamos. En el caso del menú de muerte, el botón de acción nos devolverá al último punto de guardado por el que hayamos pasado. En el caso del menú de fin de juego, el botón de acción empezará una nueva partida.

7.5 – Interfaz durante la partida

Esta es la interfaz que más tiempo va a ver el jugador, así que no debe de ser muy liosa ni muy sobrecargada. La interfaz está compuesta de varios elementos:

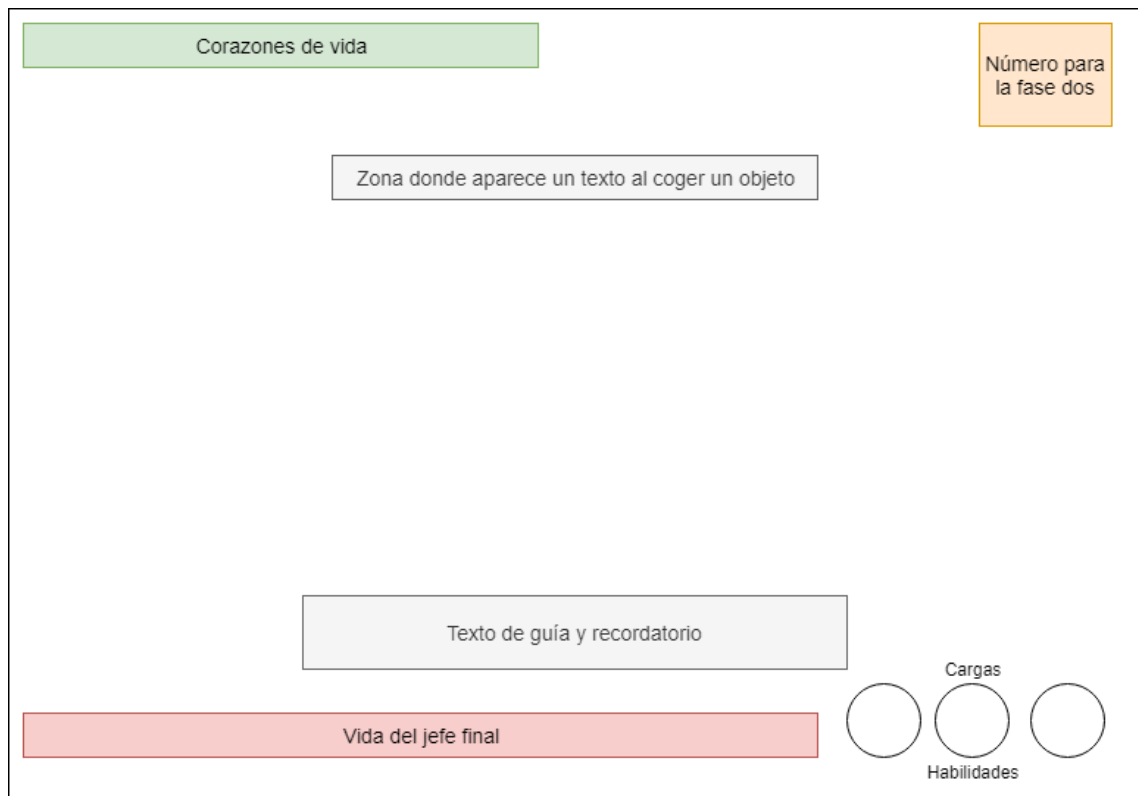


Figura 7.5 - Esquema de la interfaz de juego

- Arriba a la izquierda está la vida del jugador, representada con corazones. Estos corazones se irán perdiendo (enteros o una mitad) o ganando, dependiendo de cuál sea la vida del jugador y si se cura o es dañado.
- Abajo a la derecha estarán las habilidades. Conforme el jugador desbloquee nuevas habilidades un icono representativo aparecerá. Cuando dicha habilidad se use, encima del icono aparecerá un número que irá bajando poco a poco haciendo referencia al tiempo de enfriamiento de la habilidad. En el caso de la cura, encima del icono tendrá un número haciendo referencia a las cargas de uso que tiene dicha habilidad.
- En el centro y abajo irán apareciendo textos para explicar cosas que pasan y para servir de guía al jugador.
- En la parte central y arriba saldrán textos cuando el jugador obtenga un nuevo objeto o un nuevo nivel.
- En la segunda fase, arriba a la derecha aparecerá un número representando el número de enemigos derrotados.
- Cuando estemos en la pelea contra el jefe final, en la parte inferior aparecerá una barra de vida que representa la vida del jefe.

De todos los puntos anteriores, solo la vida y las habilidades son permanentes. El resto de elementos irán apareciendo y desapareciendo dependiendo de donde estemos y como vayamos interactuando con el juego.

7.6 – Interfaz de interacción con un personaje



Figura 7.6 - Esquema del cuadro de texto de los personajes

Durante el juego nos encontramos con un par de personajes, ya sean aliados o enemigos. En ambos casos, cuando interactuamos con ellos sale un cuadro de texto en el cual vemos su imagen y el texto que nos dicen. Este cuadro se pondrá por encima de la actual interfaz del juego.

7.7 – Interfaz de puzle de mover piezas

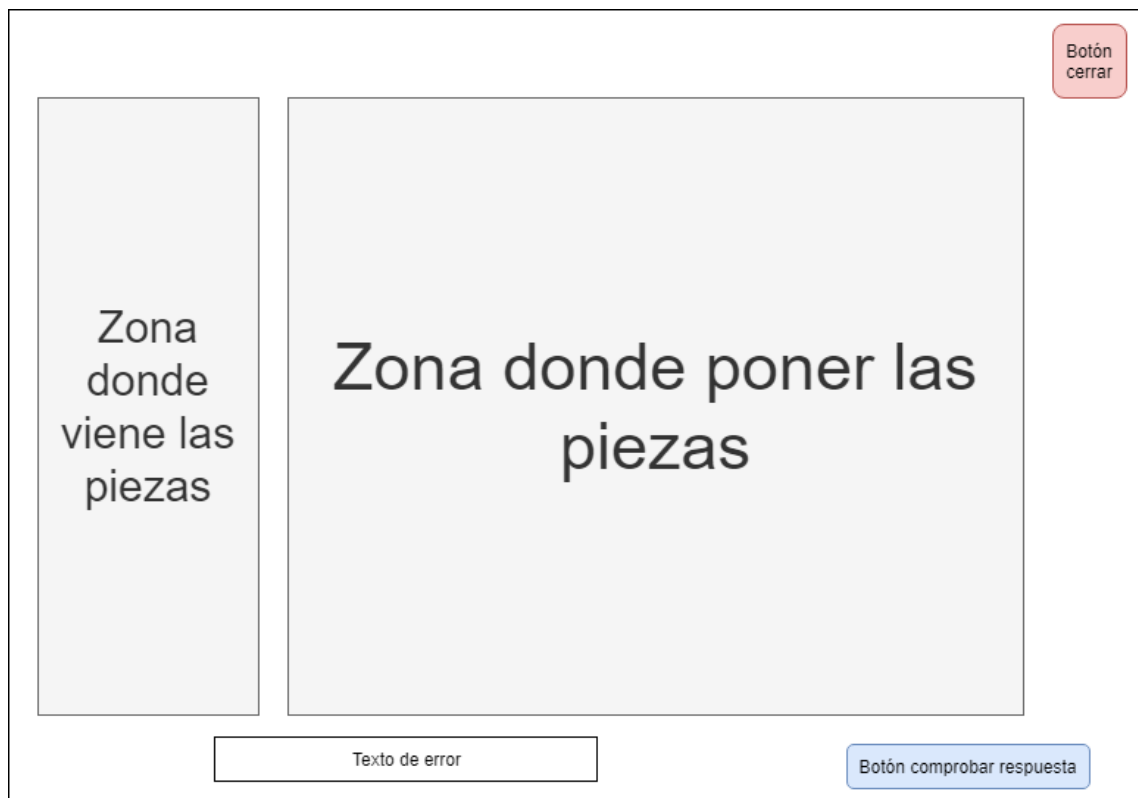


Figura 7.7 - Esquema del puzle de mover piezas

En este puzle tenemos un lugar donde vienen las piezas y otro lugar donde debemos colocarlas todas. Arriba a la derecha hay un botón para poder cerrar el puzle en cualquier momento. Abajo a la derecha está el botón para ver si es correcta nuestra respuesta. En el centro y abajo saldrá un texto si nos hemos equivocado con nuestra respuesta.

7.8 – Puzle de elegir opción correcta

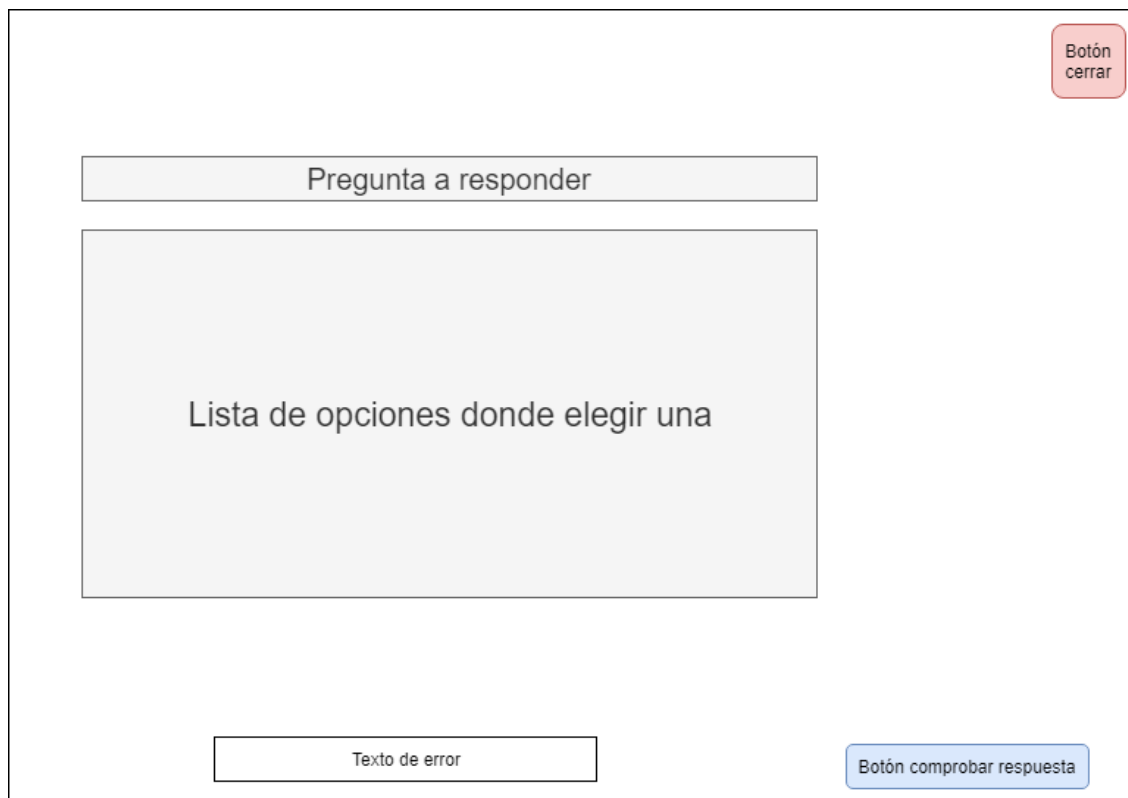


Figura 7.8 - Esquema del puzle de elegir una opción

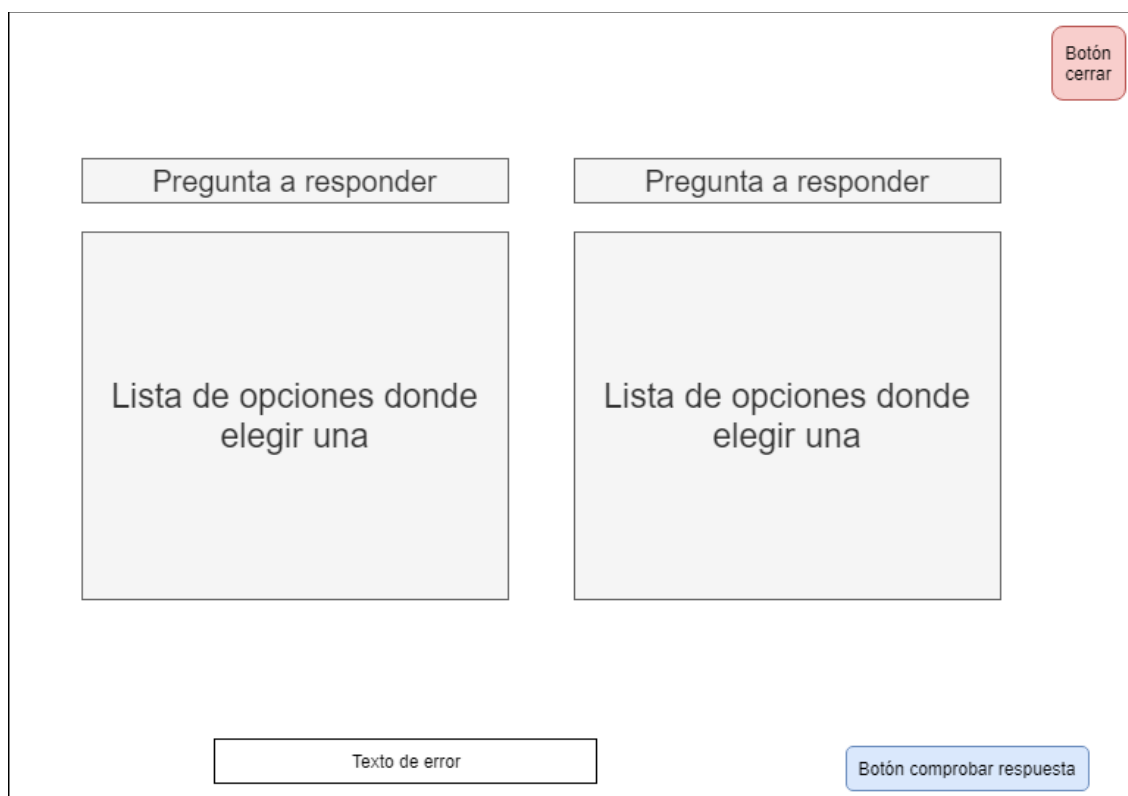


Figura 7.9 - Esquema del puzle de elegir dos opciones

Como podemos ver, este puzle lo tenemos en dos variantes diferentes: una con una sola pregunta con la que elegir la opción correcta y otra en el que tenemos dos preguntas distintas, cada una con su elección única. En ambos casos, arriba de cada lista de opciones hay un texto que es la pregunta. Las opciones son posibles respuestas a la pregunta. Arriba a la derecha está un botón para poder cerrar el puzle cuando queramos. Abajo a la derecha está el botón que comprueba si la respuesta es correcta. En el centro y abajo aparece un texto cuando la respuesta no es la correcta.

7.9 – Puzle de introducir texto

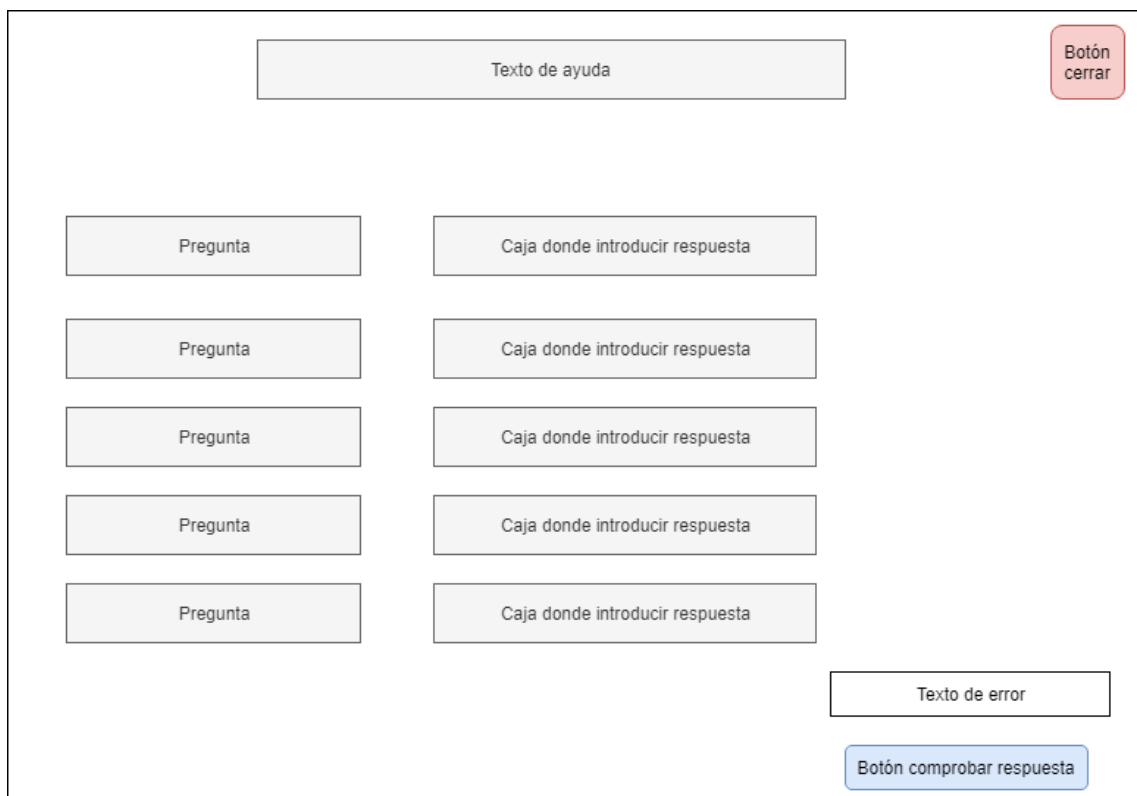


Figura 7.10 - Esquema del puzle de preguntas iniciales



Figura 7.11 - Esquema del puzle de una puerta

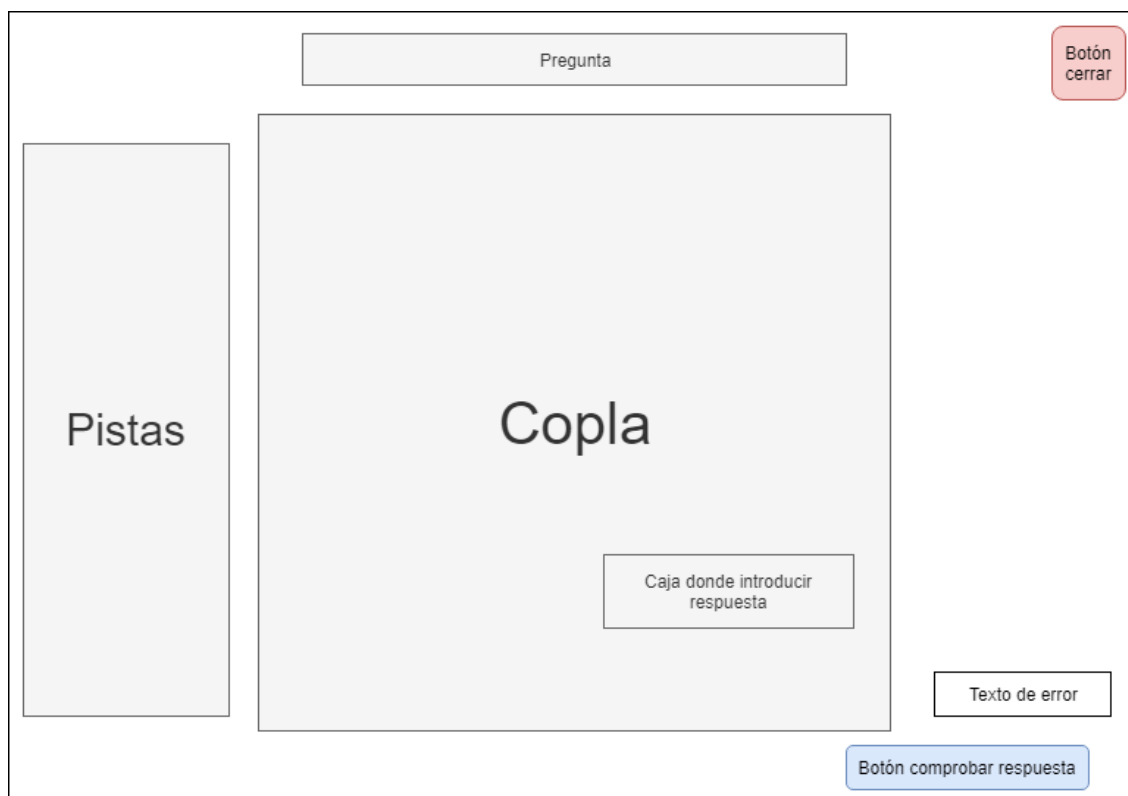


Figura 7.12 - Esquema del puzle de una copla

En este tipo de puzle también tenemos varias variantes: las preguntas iniciales, las preguntas para pasar una puerta y las coplas. Aun así, todos tienen los mismos

componentes, más alguno específico. Los tres tienen un texto que representa la pregunta y un cuadro para introducir texto. Luego, todos tienen un botón arriba a la derecha para cerrarlo y un botón abajo a la derecha para comprobar si la respuesta es correcta.

En el caso del puzzle de las puertas, tenemos un texto debajo de la caja de introducir texto donde nos dice que palabras tenemos. En el caso del puzzle de las preguntas iniciales, tenemos, en vez de una pregunta y una caja, tenemos varias preguntas, cada una con su caja de respuesta. En el caso de las coplas, tenemos un texto en el centro de la pantalla, que es la propia copla, con una caja para introducir texto tapando donde iría la palabra que tenemos que rellenar.

7.10 – Prueba de pulsar rápidamente un botón

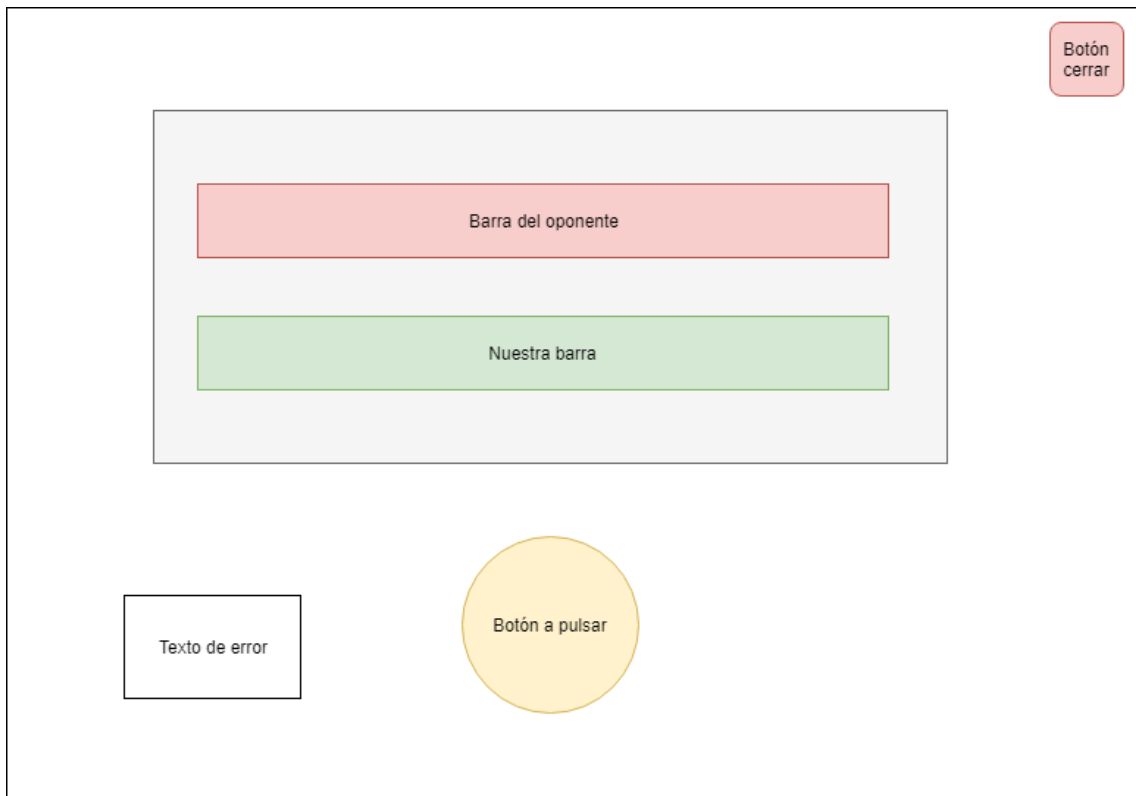


Figura 7.13 - Esquema de la prueba de pulsar un botón

Esta prueba se compone de un cuadro con dos barras, una correspondiente a nuestros enemigos y otra correspondiente a nosotros. Hay un botón en el centro que al ser pulsado aumenta nuestra barra de progreso. Además, arriba a la derecha hay un botón para salir y abajo a la izquierda sale un texto cuando hemos fallado en nuestra tarea.

8 – Narrativa

Incultura cuenta la historia de cómo la *Malvada Incultura* ha ido corrompiendo todas las obras de la literatura española de la Edad Media. Ha ido por todas ellas y ha convertido a todas las personas que están allí en incultos, en personas que todo lo que quieren es destruir la literatura. Para poder salvar la literatura española, tú, un alumno elegido, has sido enviado a dichas obras para liberarlas del yugo de la *Malvada Incultura*.

La historia transcurre por tres de las obras más importantes de aquella época: *Los Milagros de Nuestra Señora*, *Coplas a la muerte de su padre* y *Cantar de mio Cid*. Con la tu ayuda y tu aprendizaje, vas liberando cada obra, una a una, y aprendiendo lo que ellas cuentan. Conforme vas aprendiendo cosas vas ganando fuerza y nuevas habilidades que te servirán para derrotar a la *Malvada Incultura*.

9 – Ambientación

La ambientación general del juego es la Edad Media. La sala principal, y la más grande, está ambientada en una mazmorra de la Edad Media, con diversas columnas y cajas. Luego, cada sala de cada fase está ambientada con toques de la obra a la que representan. Aun así, el juego tiene toques de fantasía y magia en algunos momentos.

10 – Diseño de arte

El diseño del arte está mayormente realizado por gente externamente. Se busca un toque más alegre y desenfadado, sin llegar a algo muy infantil y tonto, aunque todo está a decisión de los artistas.

Respecto al sonido y el audio del juego, el juego tiene una música tranquila de fondo para no molestar al jugador durante la partida. En algunos momentos hay sonidos que nos ayudan a saber qué pasa en la historia y que nos guían. Para el jefe final se usa una música más animada para agitar al jugador por dentro y que se emocione en la pelea.

Algunos de los conceptos de personajes para el juego:

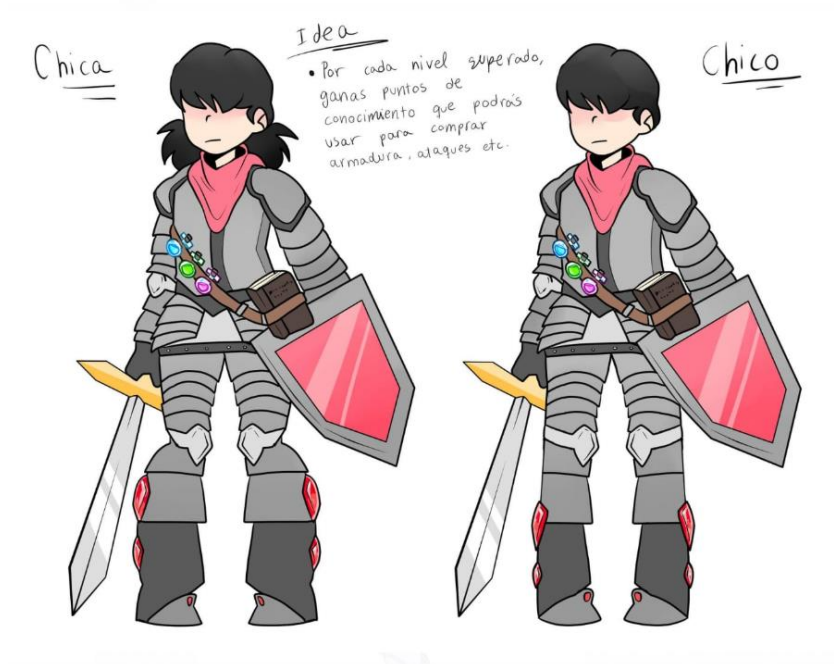


Figura 10.1 - Concepto de personajes



Figura 10.2 - Concepto de Malvada Incultura



Figura 10.3 - Concepto de enemigo cuerpo a cuerpo

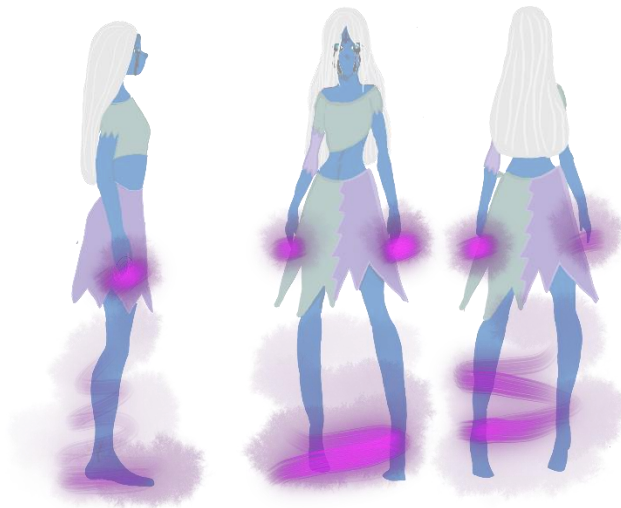


Figura 10.4 - Concepto de enemigo a distancia

Apéndice 3

A3 – Imágenes del juego

A continuación, vienen unas imágenes sacadas directamente del juego final. En ellas se ven todo lo que se ha ido comentando en la memoria.



Figura A3.1 - Menú principal 1

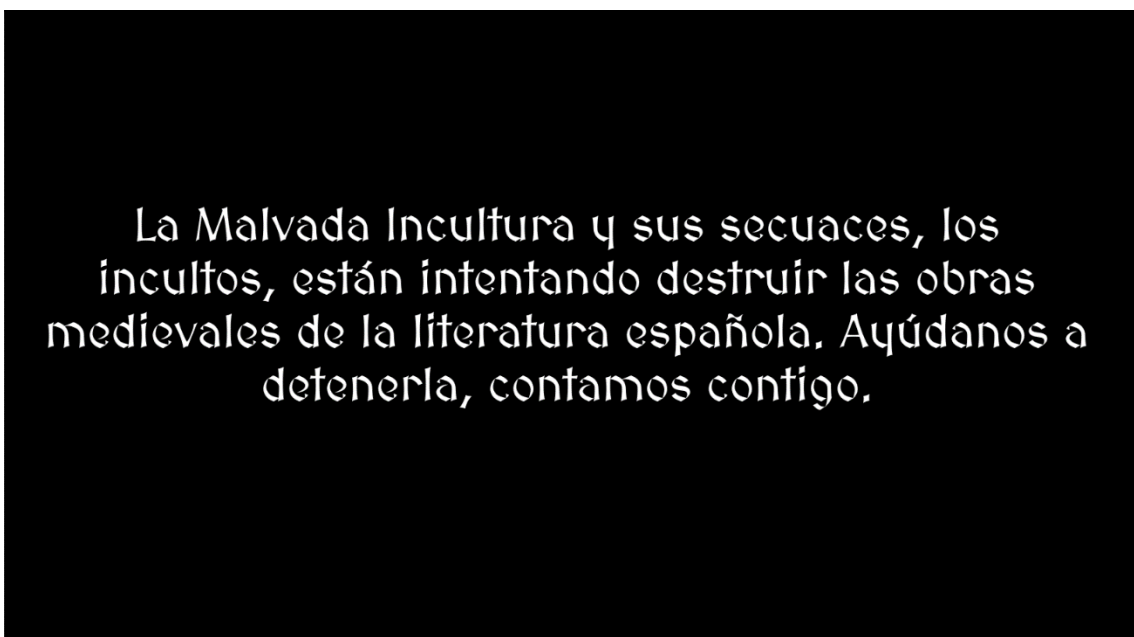


Figura A3.2 - Pantalla de carga



Figura A3.3 - Nada más empezar el juego

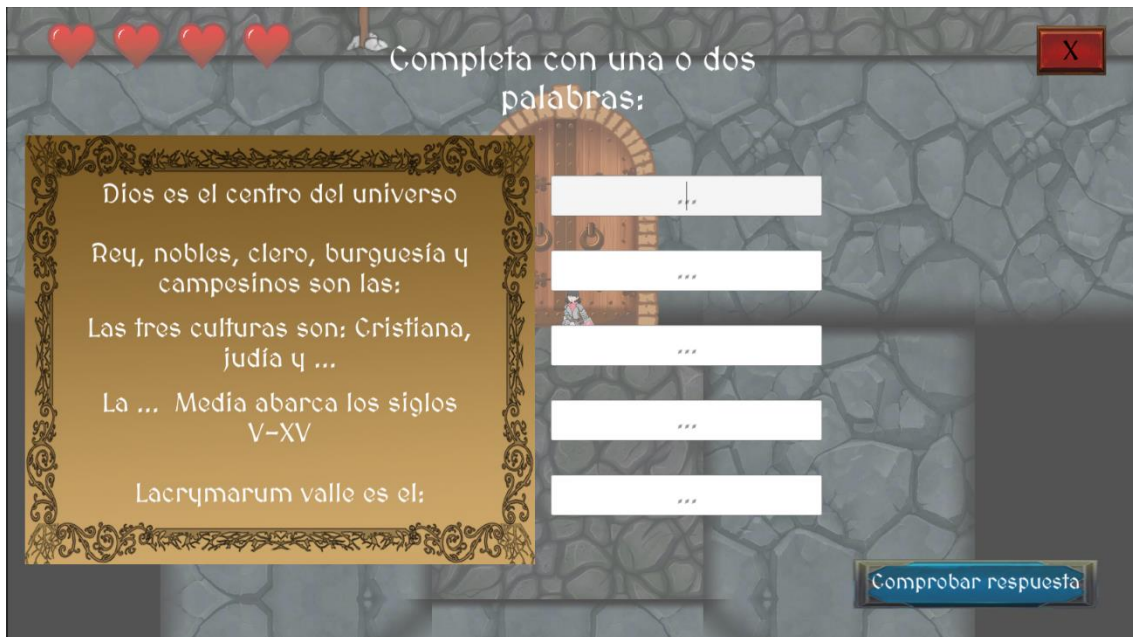


Figura A3.4 - Pregunta inicial



Figura A3.5 - Fase cero



Figura A3.6 - Puzzle en fase uno

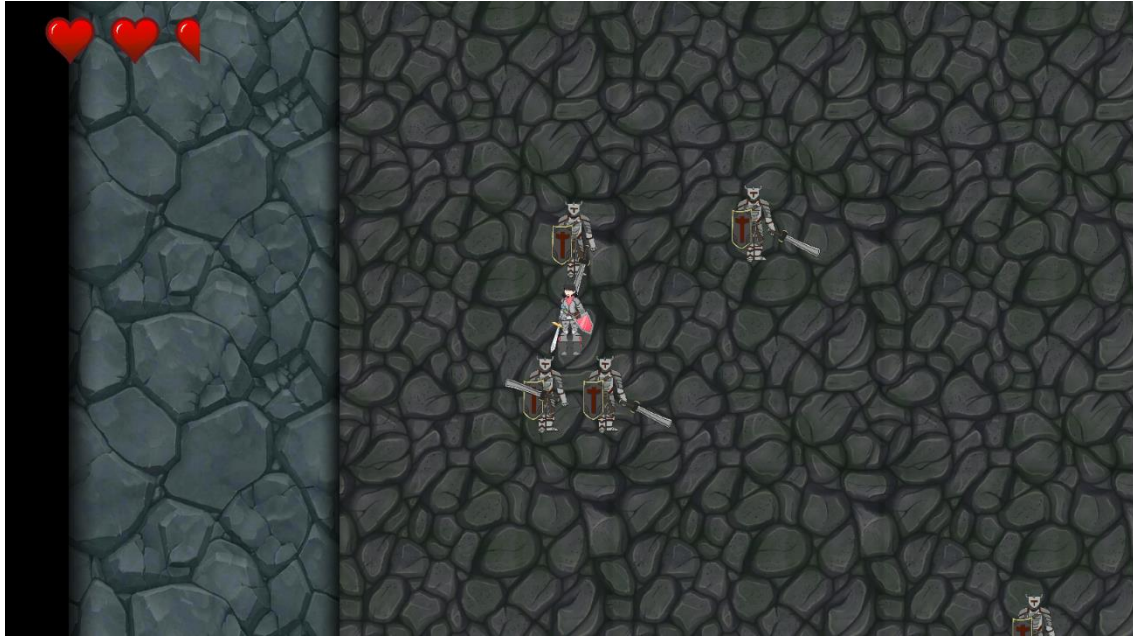


Figura A3.7 - Pelea en fase uno

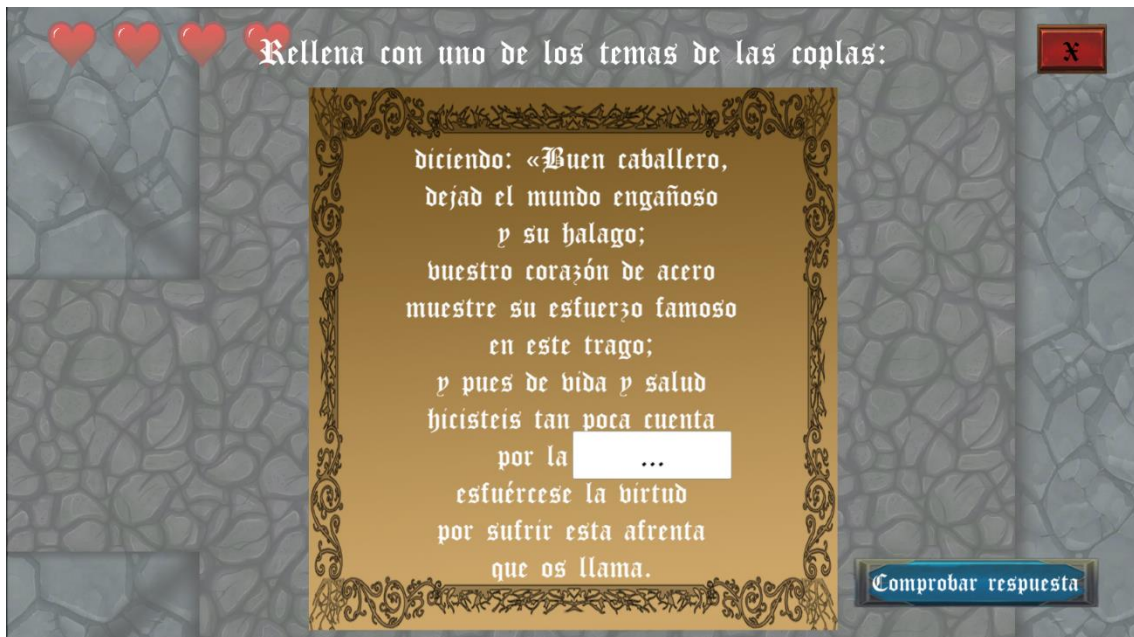


Figura A3.8 - Pregunta sobre una copla en fase dos



Figura A3.9 - Pelea en fase dos



Figura A3.10 - Primera interacción con el Cid en fase tres



Figura A3.11 - Enfrentamiento en fase tres

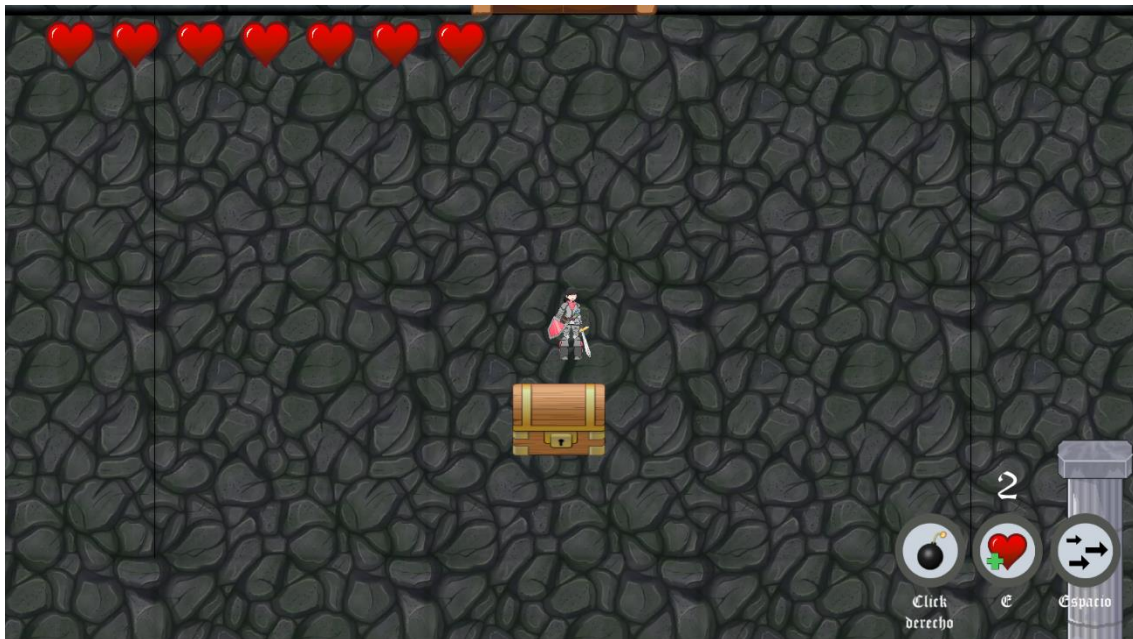


Figura A3.12 - Personaje con todas las habilidades y al máximo nivel



Figura A3.13 - Primera parte del enfrentamiento contra el jefe final



Figura A3.14 - Segunda parte del enfrentamiento contra el jefe final