



UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Game Based Learning a través de los videojuegos

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Fortea Martínez, Víctor

Tutor: González Molla, Jorge

Curso 2021-2022

Resumen

Este documento describe el proceso de desarrollar un proyecto de *Game Based Learning* o de aprendizaje basado en el juego creando un videojuego del género RPG (*Role Playing Game*).

El aprendizaje basado en juegos es un enfoque de la enseñanza en el que los juegos se utilizan como uno de los recursos del aprendizaje. Esto se aplica en este juego mediante la introducción de algunos conceptos que luego se utilizan en el propio grado en Ingeniería Informática.

El objetivo del proyecto es enseñar una introducción a los conceptos básicos de la programación a aquellas personas interesadas en la materia mientras se divierten jugando. El subgénero del videojuego es un *RPG Tactics* que cuenta con conocidos títulos como *Final Fantasy* o *Pokemon*.

Los *RPG Tactics* son juegos de combates por turnos donde se progresa en base a derrotar enemigos que te dan una recompensa al ser vencidos. Recompensas que se usan para derrotar a otros enemigos más poderosos. Al ser un juego por turnos, la lógica del juego se complica debido a la cantidad de habilidades, armas y personajes involucrados. Por otra parte, conceptos como los *FPS* (*Frames* por segundo) o la eficiencia de los algoritmos son menos importantes que en otros juegos.

Para este proyecto se utiliza Unity3D como motor para el desarrollo del videojuego, Microsoft Visual Studio como entorno de programación y C# como lenguaje de programación.

Palabras clave: Videojuego, Aprendizaje basado en juego, Unity3D, C#, RPG.

Abstract

This document describes the process of developing a game based learning project through a video game of the RPG genre (*Role Playing Game*).

Game based learning is a teaching approach where games are used as one of the learning resources. This is applied to the game by introducing some of the basics used on the degree.

The objective of the project is to teach an introduction to the basics of programming to those who are interested in this subject while they play a game. The video game sub-genre is an RPG Tactic that includes familiar games like *Final Fantasy*, *Pokémon*, or *Fire Emblem*.

RPG Tactic is a turn-based genre where you progress based on the defeat of enemies that will give a reward when they are defeated. Those rewards are used to defeat other enemies more powerful. On the one hand, being a turn-based game, the logic rules of the game are more complicated due to the big number of skills, weapons, and characters involved. On the other hand, concepts such as FPS (*Frames per second*) or the efficiency of algorithms are less important than in other genres.

For this project, Unity3D is used as the game development engine, Microsoft Visual Studio as the programming environment, and C # as the programming language.

Keywords: Video game, Game based learning, Unity3D, C#, RPG.



Agradecimientos

Me gustaría primero agradecer la realización de este proyecto a mis padres José Manuel y María Dolores. Ellos son los que han hecho que este proyecto pueda llevarse a cabo y quienes me han apoyado durante todo el camino hasta llegar aquí. Gracias por siempre estar.

Seguidamente, me gustaría agradecer la realización de este proyecto a Paulyn quien siempre me apoyó en los peores momentos del proyecto y me dio el empujón que hacía falta para poder seguir adelante.

También quiero hacer mención especial a mi tutor, Jorge, quien aceptó este proyecto y ha hecho posible que se haya podido finalizar y a Andrea, a quien le agradezco enormemente su talentosa participación artística en este proyecto.

Tampoco me quiero olvidar de todos aquellos amigos y familiares que han participado probando y sugiriendo mejoras para este proyecto en muchas de sus etapas. Personas como Javi, Lorena, Estefanía o Eva. Muchas gracias por participar desde los primeros recortes en cartón a los últimos despliegues en la web haciendo posible que este sea, hoy, mi Trabajo Final de Grado

Muchísimas gracias a todos,

Víctor Fortea Martínez.

Torrent, Valencia. 21 de noviembre de 2021.

Índice de contenidos

Game Based Learning a través de los videojuegos.....	¡Error! Marcador no definido.
Agradecimientos	5
Índice de tablas	10
Índice de diagramas	13
Índice de ilustraciones	14
1. Introducción	16
1.1 Motivación.....	16
1.2 Objetivos.....	16
1.3 Impacto esperado	16
1.4 Metodología.....	17
1.5 Estructura.....	17
1.6 Relación con asignaturas cursadas.....	18
1.7 Colaboraciones	18
2. Estado del arte	20
2.1 Game Based Learning.....	20
2.1.1 Game Based Learning en el contexto informático.....	21
2.2 Los videojuegos	21
2.2.1 El género RPG de combate por turnos.....	22
2.3 El motor del videojuego.....	22
2.4 Propuesta.....	22
3. Análisis del problema.....	25
3.1 Introducción al juego	25
3.2 Identificación de los retos no tecnológicos.....	25
3.3 Identificación de actores	25
3.4 Requisitos funcionales	25
3.5 Requisitos no funcionales	33
3.6 Casos de uso	34
3.6.1 Diagrama de casos de uso.....	34
3.6.2 Especificación de los casos de uso.....	35
3.7 Solución propuesta.....	38



3.8 Plan de trabajo	39
4. Reglas y diseño del juego	42
4.1 El jugador.....	42
4.1.1 Guerrero.....	42
4.1.2 Mago.....	43
4.1.3 Pícaro	44
4.2 Enemigos	45
4.3 Objetos.....	46
4.4 Interfaz de usuario	47
4.4.1 Diagrama de flujo	48
4.4.2 Tablas de las interfaces.....	49
4.5 Camino y mapa.....	63
4.6 Controles.....	64
4.7 Animaciones	64
4.8 Arquitectura	64
4.8.1 Arquitectura <i>hardware</i>	64
4.8.2 Arquitectura <i>software</i>	64
4.9 Diagrama de clases	64
4.9.1 Menú principal.....	65
4.9.2 Mapa, gestores de interfaz y oro.....	68
4.9.3 Datos	69
4.9.4 Gestor del personaje en combate	70
4.9.5 Gestores de la interfaz de combate	71
4.9.6 Enemigos	73
4.9.7 Base de datos para habilidades y objetos e inventario.....	75
4.9.8 Subida de habilidades y características.....	77
4.9.9 Tienda	78
4.9.10 Menú <i>in-game</i>	79
4.9.11 Victoria y derrota.....	80
5. Desarrollo de la solución.....	82
5.1 Organización del proyecto.....	82
5.1.1 Animations.....	82
5.1.2 Art	83
5.1.3 Scenes	84
5.1.4 TextMesh Pro.....	84
5.1.5 Scripts	84

5.2 Escenas en Unity y el proyecto.....	84
5.2.1 Escena del menú principal.....	85
5.2.2 Escena del juego.....	87
5.3 Implantación.....	90
6. Pruebas.....	92
6.1 Pruebas del diseño.....	92
6.2 Pruebas del software.....	93
6.2.1 Ciclo 1.....	93
6.2.2 Ciclo 2.....	96
6.2.3 Ciclo 3.....	99
6.2.4 Ciclo 4.....	105
6.2.5 Últimas correcciones.....	109
7. Conclusiones.....	111
8. Trabajo futuro.....	113
9. Referencias.....	115



Índice de tablas

Tabla 1- ACT-01 El jugador	25
Tabla 2- RQF-01 Movimiento	26
Tabla 3- RQF-02 Botones.....	26
Tabla 4- RQF-03 Elección de personaje.....	26
Tabla 5- RQF-04 Habilidades.....	26
Tabla 6- RQF-05 Interacción con personajes no jugables	26
Tabla 7- RQF-06 Cambio de interfaz tras interacción.....	26
Tabla 8- RQF-07Menú in-game	27
Tabla 9- RQF-08 Tutorial.....	27
Tabla 10- RQF-09 Mapa.....	27
Tabla 11- RQF-10 Crecimiento del personaje.....	27
Tabla 12- RQF-11 Cámara	27
Tabla 13- RQF-12 Interfaz del menú inicial.....	27
Tabla 14- RQF-13 Interfaz de combate	27
Tabla 15- RQF-14 Interfaz de la tienda	28
Tabla 16 - RQF-15 Interfaz del menú in-game.....	28
Tabla 17- RQF-16 Apertura de nuevas zonas.....	28
Tabla 18- RQF-17 Estadísticas siempre visibles	28
Tabla 19- RQF-18 Consulta de controles	28
Tabla 20- RQF-19 Consulta de habilidades y características	28
Tabla 21- RQF-20 Salir del juego.....	28
Tabla 22- RQF-21 Volver a elegir en el menú inicial	29
Tabla 23- RQF-22 Reelección de habilidades	29
Tabla 24- RQF-23 Reelección de características.....	29
Tabla 25- RQF-24 Subida de habilidades.....	29
Tabla 26- RQF-25 Subida de características	29
Tabla 27 - RQF-26 Arte.....	29
Tabla 28- RQF-27 Música.....	30
Tabla 29- RQF-28 Efectos de sonido	30
Tabla 30- RQF-29 Escalado de la dificultad	30
Tabla 31- RQF-30 Victoria en combate	30
Tabla 32- RQF-31 Derrota en combate	30
Tabla 33- RQF-32 Interfaz de Game Over	30
Tabla 34- RQF-34 Interfaz de combate	30
Tabla 35- RQF-34 Interfaz de los dados.....	31
Tabla 36- RQF-35 Ataque enemigo.....	31
Tabla 37- RQF-36 Espera al ataque enemigo.....	31
Tabla 38- RQF-37 Control de compra.....	31
Tabla 39- RQF-38 Control de invisibilidad.....	31
Tabla 40- RQF-39 Control de habilidades.....	31
Tabla 41 - RQF-40 Objetos en combate	31
Tabla 42- RQF-41 Escalado de dificultad durante el combate	32
Tabla 43- RQF-42 Control de movimiento.....	32
Tabla 44- RQF-43 Animación del movimiento.....	32

Tabla 45- RQF-44 Animación del combate.....	32
Tabla 46- RQF-45 Condicional de camino.....	32
Tabla 47 - RQF-46 Tiendas en pasillo.....	32
Tabla 48- RQF-47 Objetos en base de datos	32
Tabla 49- RQF-48 Habilidades en base de datos.....	32
Tabla 50- RQF-49 Opciones.....	33
Tabla 51- RQF-50 Victoria.....	33
Tabla 52- RQF-51 Interfaz de compra.....	33
Tabla 53- RQF-52 Interfaz de venta	33
Tabla 54- RNF-01 Game Based Learning	33
Tabla 55- RNF-02 Respuesta rápida.....	33
Tabla 56- RNF-03 Idioma	34
Tabla 57 – RNF-04 Dispositivos de Input y Output.....	34
Tabla 58- RNF-05 Plataforma	34
Tabla 59 - CU-01 Empezar partida.....	35
Tabla 60- CU-02 Elegir personaje	35
Tabla 61- CU-03 Elegir habilidades	35
Tabla 62- CU-04 Mapa.....	35
Tabla 63. CU-05 Mover personaje	35
Tabla 64- CU-06 Menú.....	36
Tabla 65 - CU-07 Consultar stats	36
Tabla 66- CU-08 Consultar habilidades	36
Tabla 67- CU-09 Consultar características.....	36
Tabla 68- CU-10 Consultar instrucciones	36
Tabla 69 - CU-11 Consultar inventario	36
Tabla 70 - CU-12 Combate.....	36
Tabla 71 - CU-13 Usar objeto.....	37
Tabla 72 - CU-14 Atacar	37
Tabla 73 – CU-15 Tienda	37
Tabla 74 - CU-16 Comprar.....	37
Tabla 75 - CU-17 Vender	37
Tabla 76 - CU-18 Créditos	37
Tabla 77 - CU-19 Salir	38
Tabla 78- Habilidades del guerrero	43
Tabla 79- Habilidades del mago	44
Tabla 80- Habilidades del pícaro	45
Tabla 81- Características de los enemigos.....	46
Tabla 82- Habilidades de los enemigos	46
Tabla 83- Objetos	47
Tabla 84 - UI-01 Menú principal.....	49
Tabla 85 - UI-02 Créditos.....	49
Tabla 86 - UI-03 Historia	50
Tabla 87- UI-04 Elegir personaje	50
Tabla 88 - UI-05 Tutorial	51
Tabla 89- UI-06 Elegir habilidades	51
Tabla 90- UI-07 Mapa	52
Tabla 91- UI-08 Tienda	52
Tabla 92- UI-09 Comprar	53



Tabla 93- UI-10 Vender.....	54
Tabla 94- UI-11 Menú in-game	55
Tabla 95- UI-12 Consulta de los Stats	55
Tabla 96- UI-13 Inventario	55
Tabla 97- UI-14 Consulta de las habilidades.....	55
Tabla 98 - UI-15 Consulta de las instrucciones.....	56
Tabla 99 - UI-16 Interfaz de combate.....	56
Tabla 100 - UI-17 Usar objeto.....	57
Tabla 101- UI-18 Dado de fuerza	58
Tabla 102- UI-19 Dado de magia	58
Tabla 103 - UI-20 Interfaz de invisibilidad	58
Tabla 104 - UI-21 Interfaz de venenos	59
Tabla 105 - UI-22 Dado de invisibilidad	59
Tabla 106 - UI-23 Oro ganado.....	60
Tabla 107 - UI-24 Reintentar.....	60
Tabla 108 - UI-25 Subida de características.....	61
Tabla 109 - UI-26 Subida de habilidades	62
Tabla 110 - UI-27 Game Over.....	62
Tabla 111 - UI-28 Victoria	63
Tabla 112 - Pruebas del ciclo 1.....	96
Tabla 113 - Pruebas del ciclo 2.....	99
Tabla 114 . Pruebas del ciclo 3	105
Tabla 115 – Pruebas del ciclo 4.....	108
Tabla 116 – Correcciones	109

Índice de diagramas

Diagrama 1- Casos de uso de Electroanima	34
Diagrama 2- Diagrama de flujo	48
Diagrama 3 - Menú inicial	67
Diagrama 4- Mapa y gestores	68
Diagrama 5 – Datos	69
Diagrama 6 - Gestor de personaje en combate	71
Diagrama 7 - Gestores de la interfaz de combate	73
Diagrama 8 – Enemigos.....	75
Diagrama 9 - Base de datos para habilidades y objetos e inventario.....	77
Diagrama 10 - Subida de habilidades y características.....	78
Diagrama 11 – Tienda	79
Diagrama 12 - Menú in-game.....	80
Diagrama 13 - Victoria y derrota.....	80



Índice de ilustraciones

Ilustración 1 - Prototipado de los personajes en cartón	40
Ilustración 2 - Ejemplo de inicialización de un script.....	65
Ilustración 3 - Animaciones jugador.....	82
Ilustración 4 - Animator de Unity para el movimiento del mago	83
Ilustración 5 - Animation Clip, mago move down	83
Ilustración 6 - Jerarquía de la primera escena.....	85
Ilustración 7 - Ejemplo de los componentes de la cámara.....	87
Ilustración 8 - Jerarquía de la segunda escena.....	87
Ilustración 9 - Jerarquía de combate y componentes del personaje	90

1. Introducción

1.1 Motivación

Los videojuegos han sido mi motivación desde que en mi infancia empecé a jugarlos por primera vez. Pronto se convirtieron en mi pasatiempo favorito. Ello motivó que mi elección para continuar mis estudios, con la finalidad de poder desarrollar posteriormente una carrera profesional en la industria del videojuego, fuera el Grado de Ingeniería Informático de la UPV.

Aunque durante el grado se introdujeron muchas materias, esto no cambió mi intención de desarrollar mi carrera como desarrollador de videojuegos. Por ello, cuando llegó la hora de elegir el Trabajo Final de Grado y tras revisar la oferta pública de proyectos disponibles decidí que quería realizar uno que se ajustara mejor a mis intenciones.

Antes de la realización de este trabajo, había formado parte de un proyecto de emprendimiento orientado a la creación de juegos de mesa para el aprendizaje de conceptos básicos de diferentes materias de la educación primaria y secundaria. Además, había desarrollado y diseñado varios videojuegos, tanto como pasatiempo, como académicamente.

La realización de este trabajo combina ambos trabajos previos en conjunto con mi anhelo profesional. Esto es, el trabajo en videojuegos previo y la enseñanza de conceptos básicos mediante el juego se combinan en este trabajo y sirve de iniciación profesional.

1.2 Objetivos

El siguiente proyecto consta de cuatro objetivos diferentes que se exponen a continuación.

El primer objetivo es crear un videojuego que sirva a personas de entre 16 y 18 años que estén pensando en empezar el estudio de la programación para familiarizarse con sus conceptos básicos.

El segundo objetivo es poner en práctica el aprendizaje obtenido durante los estudios realizados en este grado.

El tercer objetivo, comentado con anterioridad, es familiarizarse e introducirse al mundo del desarrollo de videojuegos, aprendiendo a realizarlos tanto en su vertiente técnica como de diseño. Aprendiendo también las herramientas profesionales usadas en la industria.

Finalmente, el cuarto y último objetivo es que, siendo un juego, sea divertido de jugar.

1.3 Impacto esperado

Aunque se ha expuesto en el apartado anterior y es un juego apto para casi todos los públicos, ya que no contiene apenas imágenes de miedo o escenas violentas, el juego está pensado para adolescentes que quieran iniciarse en la programación y conocer conceptos muy básicos de esta. Por tanto, se espera que mediante este juego se familiaricen con algunos conceptos básicos.

Se busca generar en el juego una emoción de reto, exploración y mejora en el jugador. Superar enemigos no es fácil y hay que elegir bien la estrategia.

Finalmente, se busca que el jugador aprenda a tomar decisiones basadas en las posibilidades que el juego ofrece. Por ejemplo, repetir una tirada para conseguir un tesoro a cambio de un coste, elegir que objetos comprar y vender cuando se disponga de poco oro o elegir si es conveniente derrotar al enemigo en ese momento.

1.4 Metodología

El desarrollo del juego sigue una metodología ágil por lo que el desarrollo es realizado en ciclos. Los ciclos previstos son:

1. Versión del juego en cartón y papel.
2. Desarrollo de un prototipo inicial en *software* con las partes necesarias para poder superar el juego.
3. Desarrollo de las partes que mejoran la jugabilidad del juego. Este ciclo puede separarse en diferentes ciclos.
4. Añadido del arte al juego y desarrollo de las características extras.

En todas estas fases se realiza un proceso de *testing*, tanto del nuevo desarrollo, como del anterior, detectando las partes que deben mejorarse para el siguiente ciclo.

Esta metodología otorga mayor flexibilidad a la hora de realizar cambios durante el desarrollo que ayuden a cumplir los objetivos.

1.5 Estructura

La memoria se divide en diferentes capítulos. Cada capítulo contiene a su vez diversos apartados. La estructura de capítulos es la siguiente:

- **Capítulo 1. Introducción:** Este capítulo contiene la motivación del proyecto, que se espera conseguir, la metodología usada y las colaboraciones en el proyecto.
- **Capítulo 2: Estado del arte:** En este capítulo se habla del estado actual de los videojuegos, del estado actual del *Game Based Learning* y se realiza una comparación entre las diferentes opciones disponibles para la elección del motor de juego.
- **Capítulo 3. Análisis del problema:** El capítulo tres se dedica al planteamiento de los retos que ofrece el proyecto y la propuesta para la resolución satisfactoria de estos.
- **Capítulo 4. Reglas y diseño del juego:** En este capítulo de la memoria se detallan las distintas partes de la solución, entrando en profundidad en cada parte del proyecto. Se explica detalladamente cada parte del proyecto poniendo el foco en el desarrollo de *software*.
- **Capítulo 5. Desarrollo de la solución:** En el capítulo cinco se muestran las distintas partes del desarrollo del proyecto y la implantación del juego.
- **Capítulo 6. Pruebas:** En este capítulo se explican todos los casos de prueba realizados en cada fase del desarrollo.
- **Capítulo 7. Conclusiones:** En este capítulo se realiza una reflexión sobre la consecución de los objetivos marcados.
- **Capítulo 8. Trabajo futuro:** En este capítulo se exponen las ampliaciones que se pueden realizar en un futuro que han quedado fuera del alcance de este proyecto.
- **Capítulo 9. Referencias:** Referencias bibliográficas presentes en la memoria.



1.6 Relación con asignaturas cursadas

Para la realización de este proyecto han sido muy útiles los conocimientos adquiridos durante la asignatura Introducción a la informática y a la programación (IIP), Programación (PRG) y Lenguajes, Tecnologías y Paradigmas de la Programación (LTP). Me enseñaron las bases para poder llevar a cabo este proyecto. Estas asignaturas usaban un lenguaje, en la mayoría de las ocasiones, orientado a objetos como Java. Esto ha sido útil ya que el lenguaje de programación usado en este proyecto es C#, otro lenguaje orientado a objetos.

También ha resultado útil la asignatura Deontología y Profesionalismo (DYP) para remarcar las prácticas profesionales en un proyecto como este y la asignatura Interfaces Persona Computador (IPC) que me ha ofrecido las herramientas para poder crear una interfaz de usuario más sencilla y entendible para el jugador.

La asignatura de Gestión de Proyectos (GPR) ha servido para poder gestionar un proyecto de esta envergadura y la asignatura de Ingeniería del Software (ISO) me ha permitido usar el lenguaje UML y organizar de manera más efectiva el proyecto.

Durante mi estancia en la *Amsterdam University of Applied Science (AUAS)*, encuadrado en el programa Erasmus, donde pude estudiar el programa semestral *Game Technology* y realizar mi primera práctica con un videojuego real en un equipo de trabajo internacional. En dicho programa se estudian asignaturas como Físicas aplicadas a videojuegos, Patrones de programación para videojuegos, Diseño automatizado de niveles o el propio proyecto académico del programa, consistente en un videojuego. Gran parte de estas asignaturas, aunque especialmente el proyecto del programa, han ayudado enormemente a la realización del proyecto de la memoria.

Finalmente, mis prácticas de empresa en convenio con la UPV en No Spoon Lab, una empresa de videojuegos de Valencia en la que algunos integrantes tienen más de 30 años de experiencia en el sector me ha servido para ampliar mi base de programación orientada a los videojuegos.

1.7 Colaboraciones

Andrea Espinosa Mateu. Autora de toda la parte gráfica del proyecto:

- *Sprites* de todos los personajes jugables.
- *Sprites* de todos los enemigos y vendedores.
- Fondos de escena y mapa.
- Elección de los colores.

2. Estado del arte

2.1 Game Based Learning

El *game based learning* es algo que ha existido desde que somos mamíferos. Se trata simplemente de aprender mediante el juego. Algo que realizan diferentes especies de manera intuitiva y natural cuando son infantes. Este concepto pretende extenderlo a otro tipo de actividades o edades en las que no suele usarse el juego para el aprendizaje. Los resultados de utilizar el juego como forma de aprendizaje han sido hasta ahora positivos, mejorando las competencias de la materia a la que se aplica, mejorando el interés del alumno en el estudio o actividad y, finalmente, mejorando también las competencias blandas y sociales ya que muchos de estos juegos son de carácter cooperativo o competitivo por grupos [1].

Se debe realizar en este punto una distinción entre el *game based learning* y la gamificación o ludificación, ya que son conceptos que suelen dar lugar a confusión. La ludificación es introducir elementos de los juegos como la experiencia, los niveles o la puntuación dadas por completar una tarea para motivar y mejorar el aprendizaje de una materia o herramienta. Al usar elementos de los juegos se hace más atractivo aprender estas materias o usar las herramientas ludificadas. Un ejemplo de uso de esta metodología está en la aplicación de aprendizaje de idiomas Duolingo, que emplea niveles, experiencia, retos y la competición entre los usuarios para fomentar el uso de la aplicación y el aprendizaje.

Por otro lado, el tema de este proyecto, el *game based learning*, se trata de utilizar los juegos como aprendizaje. Es decir, no es una materia o herramienta con apariencia de juego, sino que es un juego en el que se enseña mientras es jugado. Aunque todos los juegos enseñan alguna cosa, un juego que esté diseñado con el propósito de enseñar tiene más potencial para ello. Aunque no faltan ejemplos de juegos que no están diseñados para enseñar y, sin embargo, se puede aprender mucho durante su uso. Existen juegos como *Civilization* o la saga *Total War* donde manejas diferentes civilizaciones a lo largo de la historia enseñan durante su transcurso nombres de personajes históricos, la ubicación y ciudades de estas civilizaciones, en qué estaban especializadas, etc.

Por tanto, aunque la gamificación tiene más potencial de aprendizaje, tiene menos capacidad de generar diversión. En el caso del *game based learning*, al ser un juego, tiene un gran potencial para generar diversión, pero es complicado enseñar conceptos profundos. El *game based learning* tiene la capacidad de enseñar conceptos básicos y generar motivación e interés suficientes en el jugador para fomentar la posterior investigación sobre el tema que el juego pretende enseñar [2].

Aunque hemos hablado de videojuegos como *Civilization*, estos son juegos comerciales con un gran número de jugadores. Si buscamos el estado de desarrollo de proyectos de menor alcance nos encontramos con juegos muy interesantes para diferentes áreas. Un ejemplo de juegos de menor presupuesto diseñados para la educación es *Ancestors*, un juego diseñado por el profesor de arqueología Xavier Rubio Campillo de la universidad de Edimburgo que explica la historia de los asentamientos paleolíticos de Atapuerca haciendo dirigir al jugador a las diferentes tribus que habitan el lugar en dos tramos históricos distintos [3]. Otro ejemplo es *Innate Attack* [4], diseñada por la Federación de Científicos Americanos para enseñar a los estudiantes de educación secundaria biología, en el juego diriges un *nanobot* por las arterias del cuerpo humano que enseña

a cazar bacterias a las células. Como último ejemplo está *Kokoro Kids*, una aplicación desarrollada por la empresa del mismo nombre en Valencia y cofinanciada por fondos europeos donde se enseña a niños de entre 2 y 6 años, usando el método *kokoro*, a potenciar sus habilidades desde la infancia en áreas como la lógico-matemática, la científica, la comunicativa o la emocional, entre otras [5].

2.1.1 Game Based Learning en el contexto informático

Aunque no hay grandes títulos comerciales que usen el *game based learning* en el contexto informático, si se encuentran en el mercado algunos juegos, tanto de mesa como digitales.

En cuanto a juegos de mesa se destaca *Code Master* [6] y *Arqueras de Nand* [7]. El primero es un juego de mesa inspirado en la estética del famoso videojuego *Minecraft* en el que tienes que llevar el avatar a un portal siguiendo un camino por diferentes puntos y recogiendo todos los cristales del mapa en un número determinado de pasos, el juego llega a enseñar conceptos como los bucles. *Arqueras de Nand*, por otro lado, enseña el uso de selección de grupos en bases de datos mediante un mapa con ciudades de diferentes colores en el que hay que disparar flechas a los orcos que se asientan en estas ciudades, pero solo puedes hacerlo eligiendo colores e instrucciones como OR, AND, COUNT o NOT.

En el aspecto digital se destacan juegos como *Code Combat* [8] o *JSRobot* [9]. Ambos juegos tienen mecánicas y conceptos muy parecidos. En el primer juego, como en *Code Master*, el jugador encarna un avatar en un juego de rol con diferentes capacidades, donde el objetivo es superar un mapa con diferentes puntos de control. Antes de comenzar el nivel el jugador indica que instrucción se debe seguir en cada punto de control, por ejemplo, ataque, recoger tesoro, mover arriba, mover abajo, etc. Cada instrucción se presenta en formato de método de programación. El segundo juego consiste en controlar a un robot. En este juego se actúa directamente sobre código, indicando que debe hacer el robot. Ambos enseñan Javascript pero *Code Combat* también enseña otros lenguajes como Python. Son juegos diseñados no solo para enseñar a niños sino también para enseñar a adultos o profesionales de la programación.

2.2 Los videojuegos

Se considera que el primer videojuego nació en 1947, inspirado en los radares usados en la Segunda Guerra Mundial. Los videojuegos actuales son completamente diferentes en todos los sentidos, tanto desde el punto de vista técnico como desde el punto de vista artístico. En la actualidad la industria del videojuego ha crecido mucho y ha alcanzado campos impensables hace tan solo veinte o treinta años, campos como el de la rehabilitación motora o mental en casos como los ictus [10]. La industria se encuentra actualmente en auge en todo el mundo, en el año 2019 creció un 9.6% [11].

El mercado total en el año 2020 se calcula en 175000 millones de dólares, de los cuales un 49% fue gastado en móviles y tabletas, un 29% en consolas y un 20% en PC, dando a estas plataformas un total del 98% del total gastado en videojuegos. La región con más peso en el mercado es la región Asia-pacífico con más del 50% del peso total sobre la cantidad gastada en videojuegos en todo el globo. Esto es posible por el consumo en estados como China, Japón y la República de Corea, ocupando el primer, tercer y cuarto puesto respectivamente. Además, es una región con un crecimiento del 6.5% cada año. En contraste, Europa y América del norte crecen un 3.6% y un 2.6% respectivamente teniendo un peso en el mercado del 14% y 8% respectivamente. La región de mayor crecimiento en valor porcentual es África y Oriente Medio con un 8.8% anual [12].



Si se centra la atención en el estado español, es el noveno estado en el total de facturación de sus empresas [13]. Además, ocupa la décima posición del ranking global en tamaño total del mercado, con un total de 2656 millones de dólares gastado por consumidores de España. Esto contrasta con la posición de las empresas españolas en los *rankings* de videojuegos, en los que no se encuentra ninguna entre los veinticinco primeros puestos [12].

2.2.1 El género RPG de combate por turnos

En el género *RPG* por turnos, es decir, juego de rol por turnos, se encuentran juegos conocidos como *Pokémon* o *Final Fantasy*. En este tipo de juego es habitual la exploración de un mapa intercalado con combates por turnos.

Actualmente el género sigue siendo uno de los más importantes del mercado con juegos recientes como *Persona 5*, *Yakuza: Like a Dragon*, *XCOM 2* o la saga *Fire Emblem*. Las empresas y el mercado japonés es el dominante en este género al ser tres de los cuatro títulos anteriores de origen nipón.

2.3 El motor del videojuego

Existen en la actualidad diferentes motores que facilitan el desarrollo de videojuegos. En algunos de ellos incluso no es necesaria la programación para poder desarrollar un juego. Este es el caso de GameMaker [14].

Tras estudiar las diferentes opciones, valorando variables como la comunidad desarrolladora, la potencia del motor, el precio y los lenguajes de programación usados, quedaron únicamente dos alternativas: Unreal Engine [15] y Unity 3D [16].

Unreal Engine está desarrollado por Epic Games y usa C++ como lenguaje de programación mientras que Unity 3D está desarrollado por Unity Technologies y usa C#. Ambas son gratuitas para el alcance del proyecto. Las ventajas de un motor sobre el otro son las siguientes [17]:

- Ventajas de Unity 3D:
 - Una comunidad desarrolladora mayor.
 - Una curva de aprendizaje menor debido a la estructura de su editor.
 - Conocimiento previo de Unity 3D.
- Ventajas de Unreal Engine:
 - Mayor calidad en los acabados gráficos.
 - Posibilidad de no usar código en algunas partes del desarrollo usando *Blueprints*, herramienta nativa de esta Unreal Engine que sustituye la parte de *scripting* por conexiones de nodos y objetos.

2.4 Propuesta

La propuesta para este proyecto es un RPG por turnos, donde eliges a un personaje entre tres disponibles para conseguir rescatar al PC de un *ransomware*. Mientras se juega el jugador se puede familiarizar con conceptos como los condicionales *if-else*, la cláusula *switch*, el término *array* y sus costes de búsqueda o términos para algunos *malware* o procesos del PC, dándoseles cierto estilo humorístico.

Para ello, después de valorar sus ventajas frente a Unreal Engine, se hace uso de Unity, al necesitar menos tiempo de aprendizaje y tener disponible una comunidad mayor para avanzar en el desarrollo. Unreal Engine ofrece mejor acabado gráfico, pero al no ser un proyecto que lo

necesite no supone una ventaja real, además tampoco lo es el hecho de poder reducir algunas partes del desarrollo usando *Blueprints*.

Al usar Unity, se ha requerido el uso de C# como lenguaje de programación, para desarrollar en C# se ha utilizado Microsoft Visual Studio, por ser una herramienta con mucha compatibilidad con Unity. Además, C# es un lenguaje estandarizado por Microsoft, la misma empresa propietaria del entorno de desarrollo escogido y por tanto muy preparado para este lenguaje.



3. Análisis del problema

3.1 Introducción al juego

Electroanima es un juego en el que el PC es secuestrado por un *ransomware*. El PC te pide ayuda y el alma del jugador se ve arrastrada dentro del PC como un proceso de software más. El jugador debe ir derrotando a los diferentes enemigos que se le presentan y que le llevarán por caminos diferentes hasta derrotar al *ransomware* mientras cuida que su vida no se reduzca a 0.

Se trata de un RPG (*Role Playing Game*) por turnos en el que también se debe explorar un pequeño mapa definido. El mapa consiste en salas y pasillos. En cada pasillo hay una tienda donde se pueden vender los objetos que se poseen y comprar otros, siempre teniendo en cuenta que el máximo de armas permitidas es una y el máximo de objetos no consumibles (aquellos que no son de un solo uso) sin contar el arma es de tres. En las salas hay enemigos que se deben derrotar para avanzar por alguno de los pasillos que se abren al vencer el combate.

Durante el combate el turno inicial siempre corresponde al jugador. El jugador puede elegir una habilidad, ataque básico u objeto, acciones que implican el inicio del turno enemigo. El enemigo realiza su ataque y entonces es el turno nuevamente del jugador. Esto sigue así hasta que alguna de las dos partes alcance un valor de vida igual o inferior a cero.

3.2 Identificación de los retos no tecnológicos

El reto no tecnológico al que se enfrenta el proyecto es hacer un juego que deje alguna enseñanza al jugador mientras esto no lo vuelve un juego aburrido. Esta es la razón por la que se descartó la gamificación desde un primer momento y se optó por un juego.

Siendo un videojuego debe tener unas mecánicas, un escalado de la dificultad y una interfaz que hagan entretenido e intuitivo el juego. Se debe incluir, además, un arte atractivo.

3.3 Identificación de actores

En este apartado se detallan que actores hacen uso del juego. Al ser un juego jugado por un único jugador, se posee un solo actor.

ID	ACT-01 El jugador
Descripción	El usuario que juega e interactúa con el juego.

Tabla 1- ACT-01 El jugador

3.4 Requisitos funcionales

En este apartado se van a exponer los requisitos funcionales del juego en una serie de tablas.

Se establecen tres niveles de prioridad y tiempo necesitado.

Las categorías de prioridad son las siguientes: alto, que indica que es requisito fundamental para poder desarrollar el juego, medio, que indica que es requisito importante, pero el juego podría funcionar sin su desarrollo y bajo, que indica que con el desarrollo especificado en el requisito se añaden características extras sin las que el juego podría funcionar bien.

El tiempo necesitado tiene las mismas categorías y consiste en el tiempo estimado que se necesita para que el juego cumpla el requisito. Alto indica un tiempo estimado de entre cuatro y



ocho horas de trabajo, pudiendo ser superior en algunos casos, medio indica entre una y cuatro horas y bajo indica que el tiempo es inferior a una hora. En estos tiempos no se estima el coste temporal de las pruebas y posibles correcciones que se necesitara para cumplir cada requisito. Además, existe una cuarta categoría, externo, si es un requisito no es desarrollado realizado por el autor.

ID	RQF-01 Movimiento
Descripción	El jugador debe poder moverse y además hacerlo únicamente cuando está en el mapa, sea en un pasillo o en una sala.
Prioridad	Alta
Tiempo necesitado	Bajo

Tabla 2- RQF-01 Movimiento

ID	RQF-02 Botones
Descripción	Los botones del juego deben poderse clicar con el botón izquierdo del ratón.
Prioridad	Alta
Tiempo necesitado	Bajo

Tabla 3- RQF-02 Botones

ID	RQF-03 Elección de personaje
Descripción	El juego debe proveer de al menos dos opciones diferentes en la elección de personaje y debe mostrar información sobre ellos al ser seleccionados o si el cursor se sitúa sobre sus siluetas.
Prioridad	Media
Tiempo necesitado	Alto

Tabla 4- RQF-03 Elección de personaje

ID	RQF-04 Habilidades
Descripción	Por cada personaje diseñado, el juego debe tener al menos tres habilidades diferentes y debe mostrar la información de estas al mover el cursor por encima.
Prioridad	Alta
Tiempo necesitado	Medio

Tabla 5- RQF-04 Habilidades

ID	RQF-05 Interacción con personajes no jugables
Descripción	El juego debe asignar un único botón para iniciar todas las interacciones con jugadores no jugables que se encuentren en el mapa.
Prioridad	Alta
Tiempo necesitado	Bajo

Tabla 6- RQF-05 Interacción con personajes no jugables

ID	RQF-06 Cambio de interfaz tras interacción
Descripción	Tras la interacción del RQF-05 el jugador debe ser trasladado a la tienda si es un vendedor o a la sala de combate si es un enemigo.
Prioridad	Alta
Tiempo necesitado	Bajo

Tabla 7- RQF-06 Cambio de interfaz tras interacción

ID	RQF-07 Menú <i>in-game</i>
Descripción	Cuando se está en el mapa el juego debe proveer al jugador de un botón asignado para poder abrir un menú.
Prioridad	Media
Tiempo necesitado	Medio

Tabla 8- RQF-07Menú in-game

ID	RQF-08 Tutorial
Descripción	El juego debe poseer un tutorial antes de empezar la partida con una duración en su lectura inferior a cinco minutos.
Prioridad	Alta
Tiempo necesitado	Medio

Tabla 9- RQF-08 Tutorial

ID	RQF-09 Mapa
Descripción	El juego debe tener un mapa que permita recorrer diferentes caminos para llegar al mismo resultado final. Además, sólo debe permitir al jugador seguir un único camino en cada partida.
Prioridad	Alta
Tiempo necesitado	Medio

Tabla 10- RQF-09 Mapa

ID	RQF-10 Crecimiento del personaje
Descripción	El juego es un RPG. El personaje escogido debe tener parámetros para su crecimiento equilibrados y entendibles para el jugador promedio.
Prioridad	Alta
Tiempo necesitado	Alto

Tabla 11- RQF-10 Crecimiento del personaje

ID	RQF-11 Cámara
Descripción	El juego debe proveer de una cámara de vista aérea para el mapa y una cámara frontal para los combates e interfaces.
Prioridad	Alta
Tiempo necesitado	Bajo

Tabla 12- RQF-11 Cámara

ID	RQF-12 Interfaz del menú inicial
Descripción	El juego debe tener una interfaz para las elecciones de personaje y habilidades, así como otras opciones como los créditos. La interfaz no debe de ser confusa, es decir, el jugador promedio no debe perderse la primera vez que se enfrenta a ella.
Prioridad	Alta
Tiempo necesitado	Medio

Tabla 13- RQF-12 Interfaz del menú inicial

ID	RQF-13 Interfaz de combate
Descripción	El juego debe tener una interfaz para poder usar las diferentes habilidades y elementos del combate. La interfaz no debe de ser confusa, es decir, el jugador promedio no debe perderse la primera vez que se enfrenta a ella.
Prioridad	Alta
Tiempo necesitado	Alto

Tabla 14- RQF-13 Interfaz de combate

ID	RQF-14 Interfaz de la tienda
Descripción	El juego debe tener una interfaz para la compraventa de objetos. La interfaz no debe de ser confusa, es decir, el jugador promedio no debe perderse la primera vez que se enfrenta a ella.
Prioridad	Alta
Tiempo necesitado	Alto

Tabla 15- RQF-14 Interfaz de la tienda

ID	RQF-15 Interfaz del menú <i>in-game</i>
Descripción	El juego debe tener una interfaz para poder salir del juego desde el mapa, así como consultar sus <i>stats</i> , oro, características y habilidades. La interfaz no debe de ser confusa, es decir, el jugador promedio no debe perderse la primera vez que se enfrenta a ella.
Prioridad	Media
Tiempo necesitado	Medio

Tabla 16 - RQF-15 Interfaz del menú *in-game*

ID	RQF-16 Apertura de nuevas zonas
Descripción	El juego debe abrir una única puerta cuando finalice el combate con victoria del jugador.
Prioridad	Alta
Tiempo necesitado	Medio

Tabla 17- RQF-16 Apertura de nuevas zonas

ID	RQF-17 Estadísticas siempre visibles
Descripción	El juego debe mostrar, mientras se esté en el mapa o en la tienda, en todo momento, las estadísticas de vida, electricidad y oro. El oro no debe estar visible en combate, aunque si lo deben estar la vida y electricidad.
Prioridad	Baja
Tiempo necesitado	Bajo

Tabla 18- RQF-17 Estadísticas siempre visibles

ID	RQF-18 Consulta de controles
Descripción	El juego debe permitir consultar los controles desde el menú <i>in-game</i> .
Prioridad	Media
Tiempo necesitado	Bajo

Tabla 19- RQF-18 Consulta de controles

ID	RQF-19 Consulta de habilidades y características
Descripción	El juego debe permitir consultar las habilidades y características desde el menú <i>in-game</i> .
Prioridad	Baja
Tiempo necesitado	Bajo

Tabla 20- RQF-19 Consulta de habilidades y características

ID	RQF-20 Salir del juego
Descripción	El juego debe permitir salir del juego desde el menú <i>in-game</i> , desde el menú inicial, desde la interfaz de victoria y desde la interfaz de derrota.
Prioridad	Baja
Tiempo necesitado	Bajo

Tabla 21- RQF-20 Salir del juego

ID	RQF-21 Volver a elegir en el menú inicial
Descripción	El juego debe permitir al jugador, hasta que empiece la partida y se encuentre en el mapa, volver hacia atrás y cambiar sus elecciones.
Prioridad	Media
Tiempo necesitado	Bajo

Tabla 22- RQF-21 Volver a elegir en el menú inicial

ID	RQF-22 Reelección de habilidades
Descripción	El juego debe permitir reelegir las habilidades mientras el jugador esté en la interfaz de subida de habilidades que aparece después del combate, hasta que el jugador elija continuar. Debe mostrar la información de cada habilidad por la que el cursor sea movido.
Prioridad	Media
Tiempo necesitado	Medio

Tabla 23- RQF-22 Reelección de habilidades

ID	RQF-23 Reelección de características
Descripción	El juego debe permitir reelegir las características en la interfaz de subida de características después de combate hasta que elija continuar.
Prioridad	Media
Tiempo necesitado	Medio

Tabla 24- RQF-23 Reelección de características

ID	RQF-24 Subida de habilidades
Descripción	El juego debe permitir al jugador subir 0.6 puntos de habilidad cada dos niveles, redondeando el número hacia arriba, por cada dos opciones de habilidad que posea el personaje elegido. Es decir, un personaje con tres habilidades tendrá dos puntos de habilidad cada dos niveles y un personaje de cinco habilidades tendrá tres puntos. Al acabar debe volverse al mapa.
Prioridad	Alta
Tiempo necesitado	Alto

Tabla 25- RQF-24 Subida de habilidades

ID	RQF-25 Subida de características
Descripción	El juego debe permitir al jugador subir cinco puntos de característica con un máximo de dos puntos por característica por nivel. Al acabar, si es nivel impar debe volver al mapa y si es un nivel par, debe hacer una transición hacia la interfaz de subida de habilidades.
Prioridad	Alta
Tiempo necesitado	Alto

Tabla 26- RQF-25 Subida de características

ID	RQF-26 Arte
Descripción	El juego debe poseer un arte vistoso con un estilo unificado para todo el juego que cree la atmósfera necesaria para la inmersión del jugador.
Prioridad	Media
Tiempo necesitado	Externo

Tabla 27 - RQF-26 Arte

ID	RQF-27 Música
Descripción	El juego debe poseer una música que cree la atmósfera necesaria para la inmersión del jugador.
Prioridad	Baja
Tiempo necesitado	Medio

Tabla 28- RQF-27 Música

ID	RQF-28 Efectos de sonido
Descripción	El juego debe poseer unos efectos de sonido que creen la atmósfera necesaria para la inmersión del jugador.
Prioridad	Baja
Tiempo necesitado	Medio

Tabla 29- RQF-28 Efectos de sonido

ID	RQF-29 Escalado de la dificultad
Descripción	El juego debe tener un escalado de dificultad que permita que los jugadores se sigan enfrentando a retos conforme avanzan.
Prioridad	Alta
Tiempo necesitado	Medio

Tabla 30- RQF-29 Escalado de la dificultad

ID	RQF-30 Victoria en combate
Descripción	El juego debe cambiar de la interfaz de combate a la interfaz de subida de características cuando el jugador consiga que la vida del enemigo sea cero o menos, con la excepción del combate contra el enemigo final, donde debe cambiar a la interfaz de victoria.
Prioridad	Alta
Tiempo necesitado	Bajo

Tabla 31- RQF-30 Victoria en combate

ID	RQF-31 Derrota en combate
Descripción	El juego debe cambiar de la interfaz de combate a la interfaz de <i>Game Over</i> cuando el jugador se quede con cero o menos de vida.
Prioridad	Alta
Tiempo necesitado	Bajo

Tabla 32- RQF-31 Derrota en combate

ID	RQF-32 Interfaz de <i>Game Over</i>
Descripción	El juego debe dar la opción en esta interfaz de empezar desde el principio la partida o salir del juego.
Prioridad	Media
Tiempo necesitado	Bajo

Tabla 33- RQF-32 Interfaz de *Game Over*

ID	RQF-33 Interfaz de combate
Descripción	La interfaz de combate debe proveer de información al jugador sobre las posibles opciones del combate. Tanto la acción de atacar, como la acción de usar objeto y como de la habilidad que se quiere usar.
Prioridad	Alta
Tiempo necesitado	Alto

Tabla 34- RQF-34 Interfaz de combate

ID	RQF-34 Interfaz de los dados
Descripción	La interfaz de los dados debe tener un botón para lanzar los dados y debe mostrar el resultado de la tirada. Además, debe mostrar al jugador previamente como afecta el resultado del dado a la acción que se quiere realizar.
Prioridad	Alta
Tiempo necesitado	Alto

Tabla 35- RQF-34 Interfaz de los dados

ID	RQF-35 Ataque enemigo
Descripción	Se debe proveer de información visual sobre el ataque que está realizando el enemigo.
Prioridad	Media
Tiempo necesitado	Medio

Tabla 36- RQF-35 Ataque enemigo

ID	RQF-36 Espera al ataque enemigo
Descripción	El juego no debe permitir al jugador pulsar sobre la interfaz de combate hasta que el ataque enemigo no esté visualmente completado.
Prioridad	Alta
Tiempo necesitado	Medio

Tabla 37- RQF-36 Espera al ataque enemigo

ID	RQF-37 Control de compra
Descripción	El juego no debe permitir al jugador comprar un objeto para el que no disponga de oro o nivel suficiente. Tampoco debe poder comprar un objeto del que se incumpla la regla del juego de no poseer más de un arma (<i>WEAPON</i>) o más de tres ropas (<i>CLOTHES</i>).
Prioridad	Alta
Tiempo necesitado	Medio

Tabla 38- RQF-37 Control de compra

ID	RQF-38 Control de invisibilidad
Descripción	El juego debe controlar que siempre que un enemigo o jugador falle en su intento de invisibilidad pierda el turno.
Prioridad	Alta
Tiempo necesitado	Medio

Tabla 39- RQF-38 Control de invisibilidad

ID	RQF-39 Control de habilidades
Descripción	El juego no debe permitir pulsar sobre una habilidad que no pueda ser usada debido a su coste eléctrico ni mostrar una habilidad que aún no se haya adquirido.
Prioridad	Alta
Tiempo necesitado	Medio

Tabla 40- RQF-39 Control de habilidades

ID	RQF-40 Objetos en combate
Descripción	El juego solo debe mostrar al jugador, en la interfaz de combate, los objetos consumibles (<i>CONSUMABLES</i>).
Prioridad	Alta
Tiempo necesitado	Medio

Tabla 41 - RQF-40 Objetos en combate

ID	RQF-41 Escalado de dificultad durante el combate
Descripción	Se debe diseñar el juego haciendo que el combate sea entretenido y aumente la dificultad del combate cuando el enemigo pierda vida.
Prioridad	Media
Tiempo necesitado	Medio

Tabla 42- RQF-41 Escalado de dificultad durante el combate

ID	RQF-42 Control de movimiento
Descripción	El juego no debe permitir al jugador salirse de los límites del mapa.
Prioridad	Alta
Tiempo necesitado	Medio

Tabla 43- RQF-42 Control de movimiento

ID	RQF-43 Animación del movimiento
Descripción	El personaje debe estar animado para flotar en cada una de las cuatro direcciones.
Prioridad	Media
Tiempo necesitado	Medio

Tabla 44- RQF-43 Animación del movimiento

ID	RQF-44 Animación del combate
Descripción	El juego debe tener como mínimo animaciones básicas para los ataques del jugador y del enemigo.
Prioridad	Baja
Tiempo necesitado	Medio

Tabla 45- RQF-44 Animación del combate

ID	RQF-45 Condicional de camino
Descripción	Si hubiera más de una puerta en la sala, el juego debe mostrar la información de las condiciones de apertura de cada una de las puertas antes de comenzar el combate.
Prioridad	Media
Tiempo necesitado	Bajo

Tabla 46- RQF-45 Condicional de camino

ID	RQF-46 Tiendas en pasillo
Descripción	Todos los pasillos del juego deben tener al menos un vendedor.
Prioridad	Alta
Tiempo necesitado	Bajo

Tabla 47 - RQF-46 Tiendas en pasillo

ID	RQF-47 Objetos en base de datos
Descripción	La información de todos los objetos debe guardarse en una base de datos.
Prioridad	Media
Tiempo necesitado	Alto

Tabla 48- RQF-47 Objetos en base de datos

ID	RQF-48 Habilidades en base de datos
Descripción	La información de las habilidades, tanto del jugador como del enemigo, deben guardarse en una base de datos.
Prioridad	Media
Tiempo necesitado	Alto

Tabla 49- RQF-48 Habilidades en base de datos

ID	RQF-49 Opciones
Descripción	El juego debe tener un menú de opciones en el juego y en el menú principal para ajustar la resolución.
Prioridad	Baja
Tiempo necesitado	Alto

Tabla 50- RQF-49 Opciones

ID	RQF-50 Victoria
Descripción	El juego debe tener una interfaz de victoria cuando se derrote al último enemigo felicitando al jugador por su logro y permitiéndole salir del juego.
Prioridad	Alta
Tiempo necesitado	Bajo

Tabla 51- RQF-50 Victoria

ID	RQF-51 Interfaz de compra
Descripción	Se deben mostrar los objetos disponibles para comprar, como máximo nueve por página y se debe permitir avanzar a la siguiente página o volver a la página anterior si hay más de nueve objetos disponibles para la compra. Además, debe mostrar una descripción de los objetos cuando se seleccionen y permitir comprarlos si se tiene el oro suficiente. No se deben mostrar los objetos que no posean el nivel suficiente para ser mostrados y deben mostrarse, pero no ser interactivos, los objetos que no se puedan comprar debido a la restricción del número máximo de tipo de objeto o debido a no tener suficiente oro.
Prioridad	Alta
Tiempo necesitado	Alto

Tabla 52- RQF-51 Interfaz de compra

ID	RQF-52 Interfaz de venta
Descripción	Se deben mostrar los objetos para vender. Debe mostrar una descripción de los objetos cuando se seleccionen y permitir venderlos.
Prioridad	Alta
Tiempo necesitado	Alto

Tabla 53- RQF-52 Interfaz de venta

3.5 Requisitos no funcionales

En este apartado se van a exponer los requisitos que son externos al juego.

ID	RNF-01 <i>Game Based Learning</i>
Descripción	El juego debe enseñar al menos cuatro aspectos de la informática y estos aspectos deben estar completamente integrados en el juego.
Prioridad	Alta
Tiempo necesitado	Alto

Tabla 54- RNF-01 *Game Based Learning*

ID	RNF-02 Respuesta rápida
Descripción	El juego debe dar <i>feedback</i> al jugador cuando este introduzca un <i>input</i> en el sistema de manera ágil.
Prioridad	Alta
Tiempo necesitado	Bajo

Tabla 55- RNF-02 Respuesta rápida

ID	RNF-03 Idioma
Descripción	El juego es en español ya que es el idioma hablado por la mayoría de los jugadores.
Prioridad	Alta
Tiempo necesitado	Bajo

Tabla 56- RNF-03 Idioma

ID	RNF-04 Dispositivos de <i>Input</i> y <i>Output</i>
Descripción	El juego necesita de ordenador, pantalla, ratón y teclado para ser jugado.
Prioridad	Alta
Tiempo necesitado	Bajo

Tabla 57 – RNF-04 Dispositivos de *Input* y *Output*

ID	RNF-05 Plataforma
Descripción	El juego se jugará en la web y necesita un navegador como <i>Google Chrome</i> o <i>Mozilla Firefox</i> para poderse jugar y memoria RAM suficiente para poder abrir estos navegadores.
Prioridad	Alta
Tiempo necesitado	Bajo

Tabla 58- RNF-05 Plataforma

3.6 Casos de uso

Para poder ver la interacción entre las diferentes partes del sistema y el jugador se va a recurrir al lenguaje UML y a las tablas de casos de uso. Se va a omitir el actor principal en las tablas por haber un único actor.

3.6.1 Diagrama de casos de uso

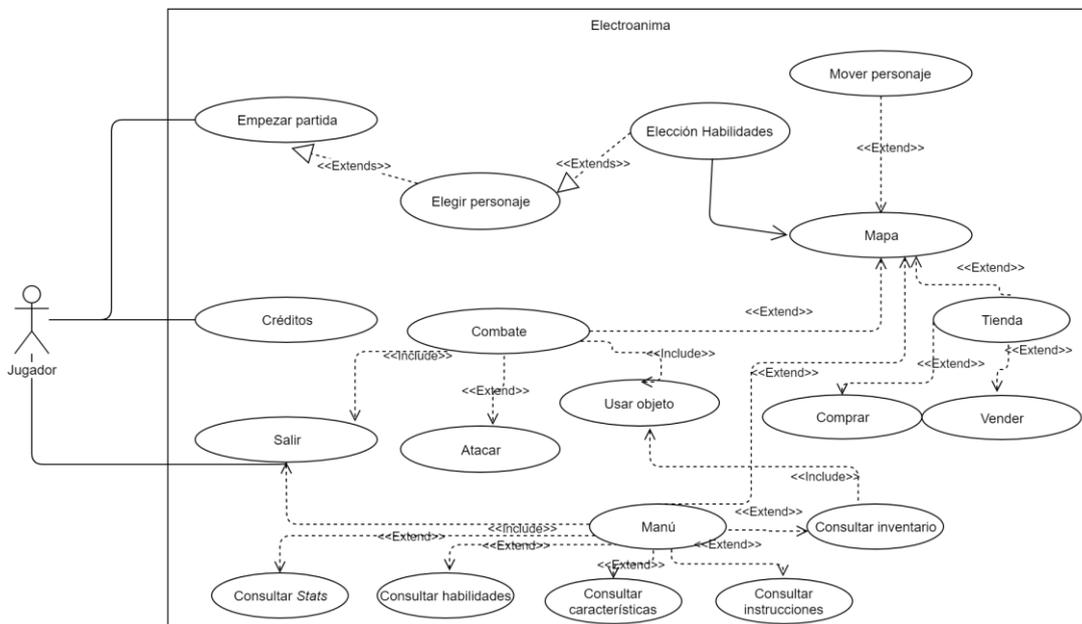


Diagrama 1- Casos de uso de Electroanima

3.6.2 Especificación de los casos de uso

ID	CU-01 Empezar partida
Descripción	Permite avanzar hacia donde se elige el personaje y empezar el juego.
Condición de activación	Apretar el botón Empezar en el menú inicial.
Precondición de uso	Encontrarse en el menú inicial.
Resultado esperado	La interfaz deja de ser el menú principal y avanza hacia la selección de personajes.

Tabla 59 - CU-01 Empezar partida

ID	CU-02 Elegir personaje
Descripción	Permite escoger personaje para empezar la partida y avanzar al tutorial y la elección de habilidades.
Condición de activación	Clicar sobre la silueta de alguno de los personajes que se muestran en la pantalla.
Precondición de uso	Encontrarse en la interfaz de elección de personaje.
Resultado esperado	La interfaz deja de ser la interfaz de selección de personaje y avanza hacia el tutorial y la selección de habilidades.

Tabla 60- CU-02 Elegir personaje

ID	CU-03 Elegir habilidades
Descripción	Permite escoger las habilidades del personaje para comenzar la partida.
Condición de activación	Elegir una habilidad en el caso del guerrero o dos habilidades en el caso del pícaro o mago.
Precondición de uso	Encontrarse en la interfaz de selección de habilidades.
Resultado esperado	El jugador se encuentra en el mapa para empezar el juego.

Tabla 61- CU-03 Elegir habilidades

ID	CU-04 Mapa
Descripción	Se muestra al avatar del jugador o personaje donde va a poder interactuar con todos los elementos y controles del juego.
Condición de activación	Acabar un combate, volver de la tienda, cerrar el menú o acabar de empezar el juego, es decir, avanzar desde la interfaz de elección de habilidades.
Precondición de uso	Encontrarse en el mapa.
Resultado esperado	Capacidad de activar cualquiera de los otros estados posibles desde el mapa: Combate, Tienda, Mover personaje y Menú.

Tabla 62- CU-04 Mapa

ID	CU-05 Mover personaje
Descripción	Permite al jugador mover al personaje en el mapa.
Condición de activación	Apretar la tecla W, A, S o D en el teclado.
Precondición de uso	Encontrarse en el mapa.
Resultado esperado	Si no hay una colisión que lo impida, el personaje se mueve en la dirección escogida por el jugador.

Tabla 63. CU-05 Mover personaje

ID	CU-06 Menú
Descripción	Permite al jugador consultar diferentes aspectos del juego como los stats, las habilidades, las características o los objetos. También permite salir del juego.
Condición de activación	Apretar el botón Q desde el mapa.
Precondición de uso	Encontrarse en el mapa.
Resultado esperado	Se abre una ventana con el menú sobre el mapa.

Tabla 64- CU-06 Menú

ID	CU-07 Consular stats
Descripción	Permite al jugador ver sus stats.
Condición de activación	Apretar el botón Stats en el Menú.
Precondición de uso	Encontrarse en el menú.
Resultado esperado	Se abre una interfaz sobre la que se sigue viendo el mapa que muestra los stats del jugador.

Tabla 65 - CU-07 Consultar stats

ID	CU-08 Consultar habilidades
Descripción	Permite al jugador ver sus habilidades.
Condición de activación	Apretar el botón Habilidades en el Menú.
Precondición de uso	Encontrarse en el menú.
Resultado esperado	Se abre una interfaz, sobre la que se sigue viendo el mapa, que muestra las habilidades del jugador.

Tabla 66- CU-08 Consultar habilidades

ID	CU-09 Consultar características
Descripción	Permite al jugador ver sus características.
Condición de activación	Apretar el botón Características en el Menú.
Precondición de uso	Encontrarse en el menú.
Resultado esperado	Se abre una interfaz, sobre la que se sigue viendo el mapa, que muestra las características del jugador.

Tabla 67- CU-09 Consultar características

ID	CU-10 Consultar instrucciones
Descripción	Permite al jugador las instrucciones básicas del mapa.
Condición de activación	Apretar el botón Instrucciones en el Menú.
Precondición de uso	Encontrarse en el menú.
Resultado esperado	Se abre una interfaz, sobre la que se sigue viendo el mapa, que muestra las instrucciones básicas usables en el mapa.

Tabla 68- CU-10 Consultar instrucciones

ID	CU-11 Consultar inventario
Descripción	Permite al jugador ver su inventario.
Condición de activación	Apretar el botón Inventario en el Menú.
Precondición de uso	Encontrarse en el menú.
Resultado esperado	Se abre una interfaz en la que se puede consultar el inventario actual y usar cualquiera de los objetos del tipo <i>CONSUMABLE</i> .

Tabla 69 - CU-11 Consultar inventario

ID	CU-12 Combate
Descripción	Interfaz de combate donde se puede elegir atacar o usar un objeto.
Condición de activación	Interactuar con un enemigo desde el mapa apretando la tecla E.
Precondición de uso	Encontrarse en el mapa y lo suficiente cerca de un enemigo.
Resultado esperado	Se muestra la interfaz de combate donde ocurren los eventos relacionados con este.

Tabla 70 - CU-12 Combate

ID	CU-13 Usar Objeto
Descripción	Permite al jugador usar un objeto para recuperar vida o electricidad.
Condición de activación	Apretar sobre un objeto y usarlo en la interfaz de combate o en el inventario del menú.
Precondición de uso	Estar en la interfaz de objetos en combate o en el inventario del menú.
Resultado esperado	El jugador recupera electricidad o vida.

Tabla 71 - CU-13 Usar objeto

ID	CU-14 Atacar
Descripción	Permite al jugador restar vida al enemigo para intentar ganar el combate.
Condición de activación	Apretar la opción de Atacar o alguna de las habilidades del personaje en la interfaz de combate.
Precondición de uso	Estar en la interfaz de combate.
Resultado esperado	Resta vida al enemigo. Si la reduce a cero o menos vuelve al mapa o gana la partida si es el último enemigo.

Tabla 72 - CU-14 Atacar

ID	CU-15 Tienda
Descripción	Permite al jugador acceder a la interfaz de la tienda donde comprar y vender objetos.
Condición de activación	Interactuar con Fifoxi, el vendedor, desde el mapa apretando la tecla E o volver de las interfaces de compra o venta.
Precondición de uso	Encontrarse en el mapa y lo suficientemente cerca de un vendedor.
Resultado esperado	Se muestra la interfaz de la tienda donde se compran y venden objetos.

Tabla 73 – CU-15 Tienda

ID	CU-16 Comprar
Descripción	Permite al jugador comprar objetos.
Condición de activación	Elegir un objeto y apretar en el botón Comprar.
Precondición de uso	Apretar el botón de comprar desde la interfaz de la tienda.
Resultado esperado	Añade el objeto al inventario, resta el oro de la compra al jugador y lo equipa si no es <i>CONSUMABLE</i> .

Tabla 74 - CU-16 Comprar

ID	CU-17 Vender
Descripción	Permite al jugador vender objetos.
Condición de activación	Elegir un objeto y apretar en el botón Vender.
Precondición de uso	Apretar el botón de vender desde la interfaz de la tienda.
Resultado esperado	Elimina el objeto del inventario y suma el oro de la venta al oro que tiene disponible el jugador.

Tabla 75 - CU-17 Vender

ID	CU-18 Créditos
Descripción	Permite ver los participantes en la creación del juego.
Condición de activación	Apretar el botón Créditos desde el menú inicial.
Precondición de uso	Estar en el menú inicial.
Resultado esperado	Enseña los créditos del juego.

Tabla 76 - CU-18 Créditos

ID	CU-19 Salir
Descripción	Permite al jugador salir del juego.
Condición de activación	Apretar el botón Salir desde el menú inicial, desde el menú en el mapa, al ser derrotado en el juego o al ganar la partida.
Precondición de uso	Estar en el menú inicial, en el menú del mapa, haber sido derrotado en el juego o haber vencido en el juego.
Resultado esperado	El jugador sale del juego.

Tabla 77 - CU-19 Salir

3.7 Solución propuesta

La solución propuesta para realizar este juego, la integración de algunos elementos de la informática para convertirlo en un *Game Based Learning*, así como el proceso para llegar a esta solución se van a exponer en este apartado. Para los patrones que se deben seguir cuando se diseña un juego se ha seguido las recomendaciones del libro *The art of game design* [18]. Para probar la solución propuesta se diseña un prototipo de cartón para convertirlo en un juego de mesa. Este prototipo sirve para realizar varias pruebas con las que ajustar los parámetros de los personajes y enemigos, eliminar y añadir nuevas habilidades y recortar algunas partes innecesarias o tediosas para el jugador.

Primero se procede a realizar un *brainstorming* para decidir como integrar elementos informáticos en el juego. Finalmente se opta por hacer un mapa con diferentes caminos donde el decisor del camino a seguir por el jugador es el número de turnos que se tarda en derrotar a un enemigo. Se usa el estamento *switch* y el estamento *if-else* para ello. Se piensa en usar el estamento *while* pero tras hacer varias pruebas se descartó por hacer el juego demasiado aburrido. Otro elemento informático incluido es el *array* y su coste de búsqueda, implementado a través de las ganancias de oro en cada combate y la opción de volver tirar el dado una vez obtenido el premio para poder encontrar un premio mayor gastando una pequeña parte del potencial premio. Además, se opta por usar nombres de procesos o *malware* que se pueden encontrar en un PC para nombrar a los enemigos.

Se repite esta ronda de *brainstorming* para las sucesivas elecciones, como, por ejemplo, elegir que personajes estarán presentes en el juego. Se ha tomado la decisión de tomar los arquetipos típicos de los RPG y dar al jugador la opción de elegir entre un guerrero, un pícaro o un mago. Aunque esto aumenta considerablemente la complejidad del proyecto se cree que dejar a los jugadores diferentes opciones va a mejorar la jugabilidad y diversión.

Para la elección de las habilidades se ha necesitado más tiempo. Se ha elegido que el guerrero tenga pocas habilidades y básicas. Se han asignado tres habilidades que se van modificando con las pruebas. El guerrero se piensa como un personaje ideal para principiantes. El mago usa magia y es el personaje más frágil y poderoso del juego, planteándolo para un jugador algo experimentado en el juego. Se le ha dotado de cuatro habilidades. El uso de las habilidades no es complejo, pero requiere un enfoque más estratégico que el del guerrero. Primero se pensó en incluir únicamente una habilidad que recuperara vida, pero durante las pruebas se ha visto el problema de quedarse sin electricidad en medio de un combate para un personaje como el mago, basado en su uso. Para solucionar el problema se ha incluido una habilidad de recarga que sustituye a la habilidad de recuperar vida y otra para recuperar vida mientras se ataca. Finalmente, se ha diseñado un personaje para jugadores expertos como el pícaro. El pícaro necesita de varias pruebas porque el concepto de invisibilidad y venenos dificulta la implementación y la comprensión del jugador. Finalmente se ha optado por otorgar la posibilidad de volverse invisible

sin la necesidad del uso de habilidad alguna y de poder potenciarla con dos habilidades, a la vez que se dispone de la posibilidad de elegir entre tres tipos de veneno que pueden ser usados en conjunto con la invisibilidad.

Se ha planteado poner enemigos en las salas y en los pasillos poner trampas y vendedores. Aunque tras las pruebas los enemigos se han quedado en las salas y los vendedores en los pasillos, se han eliminado las trampas para facilitar el juego. El problema de esta decisión es que deja la característica de detección muy poco útil. Se exploró la posibilidad de reintegrarla en la característica de sigilo, pero se ha decidido no hacerlo al pensar que los jugadores iban a ignorar esta característica si no escogían al personaje pícaro.

Una vez se ha decidido esta parte de diseño, se ha empezado a pensar en la implementación digital. Para ello se necesita uno o varios menús, un tutorial, debido a la pérdida del factor oral para explicar el juego de mesa e interfaces de combate y tienda. Además, se debe elegir el motor de videojuegos usado, que como se explicó anteriormente, es Unity.

Para los menús se ha determinado usar un menú inicial donde elegir al personaje y sus habilidades y un menú dentro del juego donde consultar datos relevantes para la partida. Se ha intentado integrar el tutorial dentro del mapa, pero la información transmisible en el mapa es insuficiente y se ha elegido integrarlo en el flujo de interfaces del menú inicial. Para la interfaz de combate y tienda se han creado flujos de acción que impliquen el uso de pocos pasos, a la vez, se intenta equilibrarlo con el objetivo de tener un número reducido de botones en cada interfaz para no confundir al jugador. Se ha decidido que los personajes tienen todas las opciones de ataque en la interfaz, exceptuando el pícaro, en el que se ha elegido mantener las funcionalidades separadas en diferentes interfaces dentro del combate debido al gran número de opciones. La tienda se ha separado en una interfaz de compra y otra de venta.

3.8 Plan de trabajo

Para el plan de trabajo se ha decidido usar una metodología ágil que divida en diferentes fases o ciclos el desarrollo. Además, se ha usado Jira para planificar las tareas de manera general, sin entrar en subtareas muy pequeñas. Se han incluido simplemente las categorías: pendiente, en curso y finalizada para dichas tareas.

Cada ciclo del desarrollo está pensado para un plazo de dos meses, con la excepción del prototipado de cartón y el ciclo final, que se alargará un poco para finalizar detalles del juego sin necesidad de iniciar un nuevo ciclo.

Las fases de un ciclo son las siguientes:

- **Planteamiento:** Se plantea como solucionar los problemas de la fase anterior y como implementar lo que va a ser necesario en el nuevo ciclo.
- **Diseño:** Con el planteamiento listo se crea en el diseño detallado y específico para el desarrollo de este ciclo.
- **Desarrollo:** Se codifica el diseño.
- **Despliegue:** El despliegue se realiza usando una web en la que se actualiza el juego en cada ciclo de *software*. Se usa este despliegue para realizar diferentes pruebas con jugadores.
- **Pruebas:** Se deja a varios jugadores probar el juego y se les pide *feedback* para mejorar en la siguiente fase.



Los ciclos de desarrollo ágil son los siguientes:

- Ciclo 1: Desarrollo de un prototipo de cartón: En este ciclo, por razones obvias, no hay fase de desarrollo, que es sustituido por una fase de dibujo y recorte de cartones. Tampoco contiene fase de despliegue. La fase de planteamiento se resuelve con los primeros *brainstorming*. Se crean los personajes y enemigos en cartón y se escribe en papel los datos de estos para poder ser usados a modo de fichas cuando sea probado.
- Ciclo 2: Se crea un *MVP* o Producto Mínimo Viable en PC en el que se pueda derrotar a enemigos sin ser aún atacado por estos, lo que implica crear una interfaz de combate, un menú de elección de personaje, un mapa en el que encontrarse enemigos y la capacidad de moverse por este mapa. Se implementa la victoria sobre el enemigo final para ganar el juego.
- Ciclo 3: Se implementa el ataque básico de los enemigos y la subida de nivel. Se corrigen errores de la fase anterior y se implementan las colisiones para no permitir salir al jugador del mapa. Se crea la pantalla de *Game Over* y se implementa la lógica de las puertas al finalizar el combate.
- Ciclo 4: Se corrigen errores de la fase anterior. Se implementa la tienda, la funcionalidad de conseguir oro. Se añade la funcionalidad de la invisibilidad y se añade habilidad a los enemigos y personajes. Se empieza a ajustar los parámetros de enemigos y personajes una vez se tiene tienda y habilidades.
- Ciclo 5: Se corrigen errores de la fase anterior. Se añaden animaciones y arte. Se añade el sonido y la música si está disponible. Se hacen los ajustes finales a los parámetros de los personajes y enemigos y se solucionan pequeños detalles que se hayan pasado por alto.



Ilustración 1 - Prototipado de los personajes en cartón

4. Reglas y diseño del juego

4.1 El jugador

El jugador tiene tres posibles personajes para seleccionar con sus diferentes habilidades y seis posibles características para mejorar cada vez que sube de nivel. Se sube de nivel cada vez que se finaliza un combate. A su vez, cada dos niveles, las habilidades se pueden mejorar o aprender nuevas. Los personajes son el guerrero, el pícaro y el mago. Cada personaje necesita mejorar características diferentes. Las características son Vida, Electricidad, Fuerza, Magia, Sigilo y Detección.

La vida es lo que determina si puedes seguir jugando o no. Si alcanza el valor 0 el juego avanza a la pantalla de *Game Over* se y debes empezar de nuevo desde el inicio para seguir jugando. La electricidad es la energía que obtienes del PC para conseguir canalizar tus hechizos, si el coste eléctrico de un hechizo es mayor a la electricidad que se posee no se puede realizar el hechizo. La fuerza mejora el daño en un golpe físico, la magia mejora el daño de los hechizos del mago, el sigilo es una característica útil para el pícaro y que se usa para volverse invisible y la detección se usa para detectar a los enemigos invisibles.

Cuando un combate finaliza el jugador sube de nivel. Si el jugador gana el combate, vuelve al mapa, avanzando primero por una serie de interfaces en las que subir sus características, subir sus habilidades y ganar oro.

En total se disputan cinco combates y hay cuatro transiciones entre salas o finalizaciones de combate que conllevan una subida de nivel. Se empieza en el nivel cero y cada combate ganado aumenta en uno el nivel del jugador. Por tanto, el jugador sube hasta el nivel cuatro.

4.1.1 Guerrero

El guerrero es el personaje más básico y simple de usar. Pensado para un jugador principiante. Un personaje basado en la fuerza física. Es excelente aguantando golpes enemigos, aunque tiene una capacidad menor de causar daño. Sus mejores características son la vida y la fuerza. El guerrero puede subir una habilidad en el menú inicial, dos habilidades cada dos niveles y cinco puntos de característica por nivel.

El guerrero empieza con las siguientes estadísticas:

- Vida 30
- Electricidad 10
- Fuerza 4
- Magia 0
- Sigilo 0
- Detección 1

El guerrero posee tres habilidades diferentes:

- *Berserker*: El siguiente golpe realiza un pequeño extra de daño.
- Defensa Imparable: El siguiente golpe recibido causa un daño menor.

- **Vigor Extremo:** La vida aumenta temporalmente durante tres turnos, el jugador puede perder la partida si se agota el efecto y su vida se vuelve menor o igual a cero.

Todas estas habilidades se usan mientras se realiza un ataque básico normal. Por tanto, todas las habilidades descritas en el guerrero realizan un ataque básico dependiente de la fuerza que se potencia con los efectos de las habilidades.

Esta es la tabla de habilidades:

Nombre habilidad	Efecto en	Coste nivel 1	Efecto nivel 1	Coste nivel 2	Efecto nivel 2	Coste nivel 3	Efecto nivel 3
<i>Berseker</i>	Daño	3	3	5	6	7	9
Defensa Imparable	Reducción de daño	4	1	6	3	7	5
Vigor Extremo	Vida temporal	3	5	5	10	7	17

Tabla 78- Habilidades del guerrero

En su turno el guerrero puede usar cualquiera de estas habilidades si tiene subido al menos un nivel la habilidad y tiene la electricidad necesaria. Además, puede realizar un ataque básico que solo depende de la fuerza sin coste eléctrico o consumir un objeto.

4.1.2 Mago

El mago tiene una dificultad media ya que se debe estar pendiente de no agotar la reserva eléctrica y es bastante frágil. El mago se basa en el poder mágico. Es excelente causando un gran daño, aunque ello lo hace, al contrario que el guerrero, no ser capaz de tener largos combates, causado porque ha derrotado al enemigo rápido gracias a su poder mágico o porque ha sido derrotado debido a su baja vida. Sus mejores características son la magia y la electricidad. El mago puede subir dos habilidades en el menú inicial, tres habilidades cada dos niveles y cinco puntos de característica por nivel.

El mago empieza con las siguientes estadísticas:

- Vida 15
- Electricidad 30
- Fuerza 0
- Magia 4
- Sigilo 0
- Detección 0

El mago posee cuatro habilidades diferentes:

- *Electroshock:* Un hechizo con poder mágico medio y coste bajo.
- *Electrovampirismo:* Este hechizo causa que el daño realizado al enemigo sea transferido en forma de curación a tu propia vida, sin poderse sobrepasar el valor máximo de vida. Tiene un poder mágico bajo y un coste bajo.
- *Recarga:* Recupera la electricidad el jugador para poder usar otros hechizos en los siguientes turnos.
- *Electrocarga:* El hechizo y habilidad más poderosos del juego. Un gran poder con un muy alto coste eléctrico.



Al contrario que el guerrero, las habilidades del mago no se suman a un ataque mágico básico, las habilidades usan la suma del poder mágico y su efecto para hacer daño.

Esta es la tabla de habilidades:

Nombre habilidad	Efecto en	Coste nivel 1	Efecto nivel 1	Coste nivel 2	Efecto nivel 2	Coste nivel 3	Efecto nivel 3
<i>Electroshock</i>	Daño	5	0	10	5	15	10
Electro-vampirismo	Daño y recuperación de vida	4	-6	8	-2	12	3
Recarga	Recuperación de electricidad	0	5	0	12	0	18
Electrocarga	Daño	12	5	20	12	28	20

Tabla 79- Habilidades del mago

En su turno el mago puede usar cualquiera de estas habilidades si tiene subido al menos un nivel la habilidad y tiene la electricidad necesaria. También puede consumir un objeto.

4.1.3 Pícaro

Debido a su mecánica el pícaro es el personaje más complicado de usar. Es frágil, aunque no tanto como el mago, y realiza una gran cantidad de daño, pudiendo superar al mago si se usan las combinaciones adecuadas. Mezcla una mecánica de debilitación paulatina del enemigo usando venenos combinada con otra mecánica de contundentes golpes realizados desde la invisibilidad. Puede usar un veneno y un ataque desde la invisibilidad en el mismo turno, pero debe vigilar la electricidad. La debilidad de la invisibilidad consiste en la probabilidad de ser detectado por el enemigo y perder el turno, para evitar esto debe subirse la característica de sigilo que combina con la de fuerza a la hora de causar daño. El pícaro no debe despreciar ninguna característica salvo la magia, aunque la más importante es el sigilo. El pícaro puede subir dos habilidades en el menú inicial, tres habilidades cada dos niveles y cinco puntos de característica por nivel.

El pícaro empieza con las siguientes estadísticas:

- Vida 20
- Electricidad 15
- Fuerza 1
- Magia 0
- Sigilo 3
- Detección 3

El pícaro posee cinco habilidades diferentes:

- Venenos:
 - Veneno de Vida: Resta un poco de vida al enemigo durante tres turnos.
 - Veneno de Vulnerabilidad: Se puede acumular hasta tres veces y hace al enemigo más frágil al daño recibido.
 - Veneno Paralizante: Si consigue golpear tras un duelo de sigilo con el enemigo, el turno del enemigo se salta.
- Invisibilidad:
 - Invisibilidad Letal: Aumenta el daño realizado desde un estado invisible.

- Invisibilidad Aumentada: Aumenta el sigilo total para llevar a cabo la acción de volverse invisible.

Las habilidades son muy diversas. El pícaro tiene un ataque básico que siempre realiza a lo que se le suma el daño hecho por veneno e invisibilidad. Puede elegir no usar ninguna o usar ambas. Volverse invisible no tiene ningún coste de electricidad y si se supera la tirada de sigilo contra la detección enemiga golpea sumando el sigilo y la fuerza. Sin embargo, se puede potenciar la invisibilidad usando alguna de las habilidades. Además, se puede sumar el veneno, aunque los venenos sí tienen un coste de electricidad.

Esta es la tabla de habilidades:

Nombre habilidad	Efecto en	Coste nivel 1	Efecto nivel 1	Coste nivel 2	Efecto nivel 2	Coste nivel 3	Efecto nivel 3
Veneno de vida	Daño paulatino	3	2	6	4	9	6
Veneno de vulnerabilidad	Baja defensa enemiga	2	1	5	2	7	3
Veneno paralizante	Sigilo	12	-3	18	0	24	4
Invisibilidad letal	Daño	3	3	6	6	9	9
Invisibilidad aumentada	Sigilo para invisibilidad	2	2	4	4	6	6

Tabla 80- Habilidades del pícaro

En su turno el pícaro puede usar cualquiera de estas habilidades si tiene subido al menos un nivel la habilidad y tiene la electricidad necesaria. Además, puede realizar un ataque básico sin aplicar veneno o invisibilidad o usar un ataque básico invisible. También se puede consumir un objeto.

4.2 Enemigos

No se ha elegido hacer los enemigos demasiado complicados. Algunos tienen tres fases de ataque y otros tienen una sola fase. El *spyware* tiene dos fases. Los enemigos poseen las mismas características que los jugadores. Algunos enemigos también poseen habilidades. Los enemigos con habilidades tienen un 50% de ejecutar su habilidad si tienen al menos un 50% de vida restante. Un 80% si tienen al menos un 20% de vida restante y si poseen una vida restante inferior ejecutan, siempre que su reserva eléctrica lo permita, la habilidad de la que dispongan. La excepción es el *Spyware* quien no tiene la regla de tener una vida restante inferior al 20% y con un nivel inferior al 50% de vida siempre realiza su ataque especial. Estos son los enemigos, sus características y habilidades:



Nombre enemigo	Vida	Electricidad	Fuerza	Magia	Sigilo	Detección	Nivel
<i>Firewall</i>	20	0	2	0	0	0	1
<i>Zombie</i>	35	0	2	0	1	2	2
<i>Eraser</i>	30	4	3	3	0	3	2
<i>Garbage Collector</i>	30	10	2	0	4	4	2
<i>Worms</i>	70	0	4	0	2	2	3
<i>Adware</i>	50	20	2	2	1	3	3
<i>Trojan</i>	60	0	5	0	4	5	3
<i>Spyware</i>	50	10	1	0	7	7	3
<i>Antivirus</i>	95	5	6	0	0	9	4
<i>Ransomware</i>	125	20	8	8	8	8	5

Tabla 81- Características de los enemigos

Nombre enemigo	Descripción habilidad
<i>Firewall</i>	No tiene.
<i>Zombie</i>	Si golpea inserta un veneno que resta vida paulatinamente.
<i>Eraser</i>	Su habilidad especial ataca a la energía eléctrica del jugador restando parte de ella.
<i>Garbage Collector</i>	Elimina de golpe un pequeño trozo de vida ignorando la defensa.
<i>Worms</i>	No tiene.
<i>Adware</i>	Ataque mágico bomba publicitario que resta vida extra.
<i>Trojan</i>	No tiene.
<i>Spyware</i>	Capacidad de hacerse invisible y realizar daño desde la invisibilidad.
<i>Antivirus</i>	Escudo que le hace protegerse de parte del próximo daño recibido.
<i>Ransomware</i>	Poderoso ataque mágico.

Tabla 82- Habilidades de los enemigos

4.3 Objetos

Existen tres tipos de objetos disponibles: *Weapon*, *Clothes* y *Consumable*. El número máximo del primer tipo es uno, el número máximo del segundo tipo es tres y para el tercer tipo el único límite es el máximo de inventario, que está fijado en nueve. El tipo *Consumable* es de un solo uso y se puede consumir en combate o desde el menú, en la sección de inventario. Los tipos *Weapon* y *Clothes* aumentan las características de los personajes continuamente hasta que sean vendidos. Los objetos se venden en las tiendas que Fifoxi, el vendedor, tiene en los pasillos.

Los objetos también tienen un nivel. Este nivel se asocia al nivel del personaje que se controla e implica que hasta no se alcance dicho nivel no se mostrará disponible el objeto en la tienda para ser comprado. Estos son los objetos del juego:

Nombre	Nivel	Efecto	Tipo	Nombre	Nivel	Efecto	Tipo
Espada de bronce	1	+1 Fuerza	Weapon	Capa invisible	1	+2 Sigilo	Clothes
Espada de plata	2	+4 Fuerza	Weapon	Afilador de espadas	1	+2 Fuerza	Clothes
Espada de oro	3	+8 Fuerza	Weapon	Lentillas de gato	2	+2 Detección	Clothes
Báculo de energía	1	+1 Magia	Weapon	Anillo de fuerza	3	+4 Fuerza	Clothes
Báculo de poder	2	+4 Magia	Weapon	Anillo mágico	3	+4 Magia	Clothes
Báculo de infinito	3	+8 Magia	Weapon	Anillo de sigilo	3	+4 Sigilo	Clothes
Daga de picaresca	1	+1 Sigilo	Weapon	Anillo de detección	3	+4 Detección	Clothes
Daga de espionaje	2	+4 Sigilo	Weapon	Pendiente de vida	4	+20 Vida Máxima	Clothes
Daga de fantasmas	3	+8 Sigilo	Weapon	Pendiente eléctrico	4	+20 Electricidad Máxima	Consumable
Poción eléctrica	1	+12 Electricidad	Consumable	Pergamino de vida	4	+15 Vida Máxima	Consumable
Poción eléctrica +	2	+25 Electricidad	Consumable	Pergamino eléctrico	4	+15 Electricidad Máxima	Consumable
Poción eléctrica ++	3	+42 Electricidad	Consumable	Pergamino de fuerza	4	+3 Fuerza	Consumable
Poción de vida	1	+14 Vida	Consumable	Pergamino de magia	4	+3 Magia	Consumable
Poción de vida +	2	+27 Vida	Consumable	Pergamino de sigilo	4	+3 Sigilo	Consumable
Poción de vida ++	3	+45 Vida	Consumable	Pergamino de detección	4	+3 Detección	
Colgante eléctrico	1	+2 Magia					

Tabla 83- Objetos

4.4 Interfaz de usuario

En este apartado del capítulo 4 se van a explicar las diferentes partes de la interfaz de usuario. Principalmente se divide entre menús, interfaces de combate, interfaces de subida de nivel, interfaces de objetos e interfaces de dados.



4.4.1 Diagrama de flujo

Este es el diagrama de flujo del juego. Se ha decidido usar Salir como una interfaz, aunque no lo sea, para esclarecer en qué casos se sale del juego. Salir no va a ser registrado como una Interfaz de Usuario en las tablas que se verán posteriormente. Su efecto, como el nombre indica, es salir del juego.

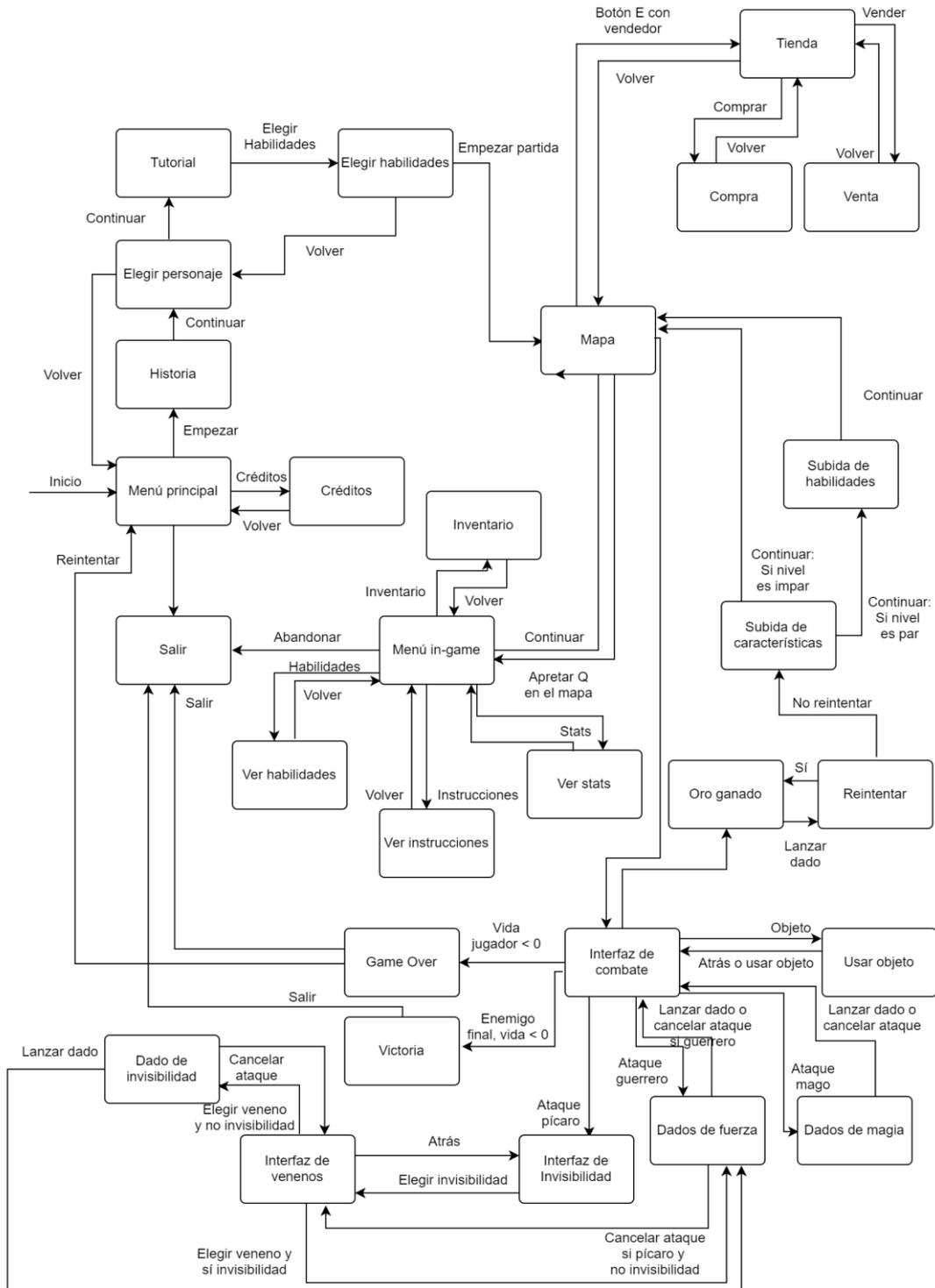


Diagrama 2- Diagrama de flujo

4.4.2 Tablas de las interfaces

En esta sección se va a especificar cada una de las posibles interfaces que se encuentran en el juego. En las interfaces que sean más complejas se va a añadir un ejemplo gráfico para ilustrar la explicación.

ID	UI-01 Menú principal
Descripción	Esta es la interfaz a la que se accede al empezar la partida. Se puede salir del juego, empezar la partida o ver los créditos del juego. Si se empieza la partida el jugador se encuentra una pequeña introducción que pone en contexto el juego y pulsando otra vez el botón Continuar accedes a la elección de personaje. El botón Opciones no realiza ninguna acción, es un botón preparado para futuras ampliaciones del juego.
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Empezar. Se accede a UI-03 Historia 2. Clicar sobre Créditos. Se accede a UI-02 Créditos 3. Clicar sobre Salir. Se sale del juego 4. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 5. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.
Ejemplo gráfico	

Tabla 84 - UI-01 Menú principal

ID	UI-02 Créditos
Descripción	Esta interfaz muestra a los participantes de este proyecto para el Trabajo Final de Grado en Ingeniería Informática para el curso académico 2021-2022. Dispone de un botón Volver para volver a UI-01 Menú principal.
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Volver. Se accede a UI-01 Menú principal. 2. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 3. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.

Tabla 85 - UI-02 Créditos

ID	UI-03 Historia
Descripción	Se cuenta un pequeño lore o historia para poner en contexto al jugador sobre el juego y su objetivo. Dispone de un botón Continuar para acceder a la interfaz de selección de personaje.
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Continuar. Se accede a UI-04 Elegir Personaje. 2. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 3. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.

Tabla 86 - UI-03 Historia

ID	UI-04 Elegir personaje
Descripción	En esta interfaz se va a elegir quién, de las tres opciones, va a ser el héroe escogido para jugar. El botón Continuar accede al tutorial, este botón empieza desactivado hasta que se seleccione alguna de las tres opciones de personaje. Se puede pasar por encima de cualquier personaje el cursor para ver sus descripciones. También se puede pulsar sobre el botón Volver para volver a la UI-01 Menú principal.
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Continuar. Se accede a UI-05 Tutorial 2. Clicar sobre Volver. Se accede a UI-01 Menú principal 3. El cursor pasa por encima de cualquiera de los botones. Se iluminan levemente de naranja. 4. El cursor se mueve por encima de cualquiera de los héroes. Se ilumina levemente el héroe y se muestra en los paneles inferiores los <i>stats</i> y la descripción del héroe hasta que el cursor salga de la zona del héroe. 5. Clicar sobre la silueta de un héroe. Se selecciona el héroe y se ilumina de un naranja más intenso que los anteriores y su descripción permanece hasta que pases el cursor por encima de otro héroe. Si el botón Continuar no estaba activado, se activa. 6. Se clica cualquiera de los botones o héroes. Se ilumina el botón clicado de un color naranja más intenso.

Ejemplo gráfico	
-----------------	--

Tabla 87- UI-04 Elegir personaje

ID	UI-05 Tutorial
Descripción	En esta interfaz se explica el funcionamiento del juego y del héroe escogido. Consiste en cinco páginas siendo la última la correspondiente al héroe elegido. Se explica primero el mapa, luego el combate, el funcionamiento del oro y la tienda en la tercera página y en la cuarta página el uso de los objetos. La quinta página contiene información específica de cada personaje y solo se muestra la página correspondiente al personaje elegido.

Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Continuar en las páginas uno, dos, tres y cuatro. Se accede a la siguiente página del tutorial. 2. Clicar sobre el botón Continuar en la página cinco del tutorial. Se accede a UI-06 Elegir Habilidades. 3. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 4. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.
-------------------------	--

Tabla 88 - UI-05 Tutorial

ID	UI-06 Elegir habilidades	
Descripción	<p>En esta interfaz están las habilidades que el jugador puede escoger dependiendo del héroe escogido previamente. El guerrero puede escoger una. El pícaro y el mago dos. Las habilidades son botones deshabilitados sobre los que si se pasa el cursor se muestra la descripción de la habilidad. Las habilidades se seleccionan pulsando el botón + que tienen a la derecha y que también muestran la descripción de la habilidad. Se deseleccionan pulsando el mismo botón.</p> <p>Una vez seleccionado el número de habilidades requeridas se activa el botón Comenzar Partida mediante el cual se accede a UI-07 Mapa. Para el ejemplo gráfico se ha escogido al mago.</p>	
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Empezar partida. Se accede a UI-07 Mapa. 2. Clicar sobre el botón Volver. Se accede a UI-04 Elegir personaje. 3. El cursor pasa por encima de cualquiera de los botones de habilidad. Se muestra una descripción de la habilidad. 4. El cursor se mueve por encima de cualquiera de los botones de +. Se muestra una descripción de la habilidad y se ilumina de tenue naranja el botón. 5. Clicar sobre el botón + sin seleccionar. Se selecciona esa habilidad y se muestra de un naranja más intenso. Si se cumple el requisito de número de habilidades seleccionadas se activa el botón Comenzar Partida. 6. Clicar sobre el botón + seleccionado. Se deselecciona esa habilidad y el botón vuelve a tener un color blanco. Si el botón Comenzar Partida estaba activado, se desactiva. 7. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 8. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso. 	
Ejemplo gráfico		

Tabla 89- UI-06 Elegir habilidades

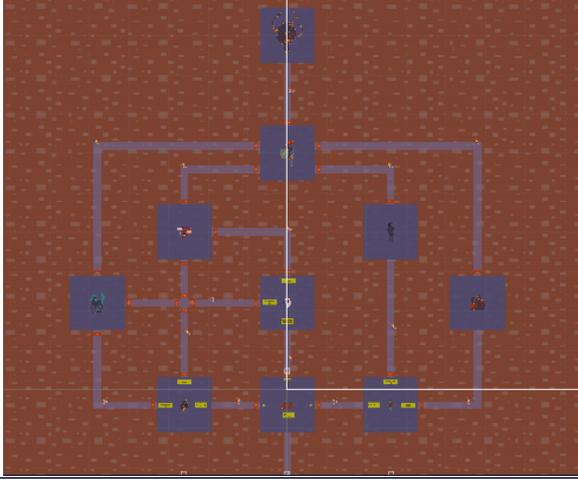
ID	UI-07 Mapa	
Descripción	En el mapa el jugador puede moverse, interactuar con enemigos y con el vendedor Fifoxi. La interacción con otros personajes se muestra iluminando la silueta del personaje con el que se va a interactuar.	
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar cualquiera de las cuatro teclas de movimiento. Se mueve el personaje si no hay una colisión que lo impide. Véase puertas cerradas o límites del mapa. 2. Acercarse a cualquiera de los personajes interactivos. Se ilumina una pequeña silueta negra a su alrededor. 3. Clicar el botón Q. Se accede a UI-11 Menú <i>in-game</i>. 4. Clicar el botón E sobre un enemigo con silueta activada. Se accede a UI-16 Interfaz de combate. 5. Clicar el botón E sobre un vendedor con silueta activada. Se accede a UI-08 Tienda. 	
Ejemplo gráfico		

Tabla 90- UI-07 Mapa

ID	UI-08 Tienda	
Descripción	En esta interfaz se ve un pequeño comentario del vendedor y tres opciones. Comprar objetos. Vender objetos. Volver al mapa.	
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón de Compra. Se accede a UI-09 Comprar. 2. Clicar sobre el botón de Vender. Se accede a UI-10 Vender. 3. Clicar sobre el botón de Volver. Se accede a UI-07 Mapa. 4. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 5. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso. 	

Tabla 91- UI-08 Tienda

ID	UI-09 Comprar
Descripción	<p>Esta es la interfaz en la que se pueden comprar objetos. Se muestran los objetos disponibles para el nivel del jugador. Los que se muestran desactivados son adecuados por nivel, pero no por otras reglas como el oro que se posee, el número máximo de <i>WEAPON</i> o el número máximo de <i>CLOTHES</i>.</p> <p>En la parte inferior izquierda se muestra el oro disponible y a la derecha se puede volver al menú de tienda.</p> <p>Hay un espacio para la descripción de los objetos en la parte inferior y en la parte central están los objetos expuestos para ser comprados. Hay nueve objetos por página, por tanto, si se ofrecen más de nueve objetos, los botones Siguiente y Anterior permiten acceder o volver del resto de páginas.</p> <p>Los objetos expuestos que se pueden clicar muestran, una vez clicados, la descripción del objeto y activan el botón de compra para la compra del objeto clicado.</p>
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Volver. Se accede a UI-08 Tienda. 2. Clicar sobre cualquier objeto que se pueda clicar sin estar clicado previamente. Se muestra la descripción del objeto y se activa el botón de Comprar. 3. Clicar sobre un objeto que ya estuviera clicado. Se deja de mostrar la descripción del objeto y el botón de Comprar se desactiva. 4. Clicar sobre Anterior. Se vuelve a la página anterior de objetos. 5. Clicar sobre Siguiente. Se avanza a la página siguiente de objetos. 6. Clicar sobre el botón Comprar. Se compra el objeto clicado y seleccionado y se añade al inventario. 7. El cursor se mueve por encima de cualquiera de los botones activos. Se ilumina levemente de naranja. 8. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso. 9. Se mantiene el último objeto clicado del mismo color naranja que en el caso de ser clicado.
Ejemplo gráfico	

Tabla 92- UI-09 Comprar

ID	UI-10 Vender
Descripción	<p>Esta es la interfaz en la que se pueden vender objetos. Se muestran los del inventario.</p> <p>En la parte inferior izquierda se muestra el oro disponible y a la derecha se puede volver al menú de tienda.</p> <p>Hay un pequeño espacio para la descripción de los objetos en la parte inferior y en la parte central están los objetos expuestos para ser vendidos. Los objetos expuestos muestran, una vez clicados, la descripción del objeto y activan el botón de venta para vender el objeto clicado.</p>
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Volver. Se accede a UI-08 Tienda. 2. Clicar sobre cualquier objeto que se pueda clicar sin estar clicado previamente. Se muestra la descripción del objeto y se activa el botón de Vender. 3. Clicar sobre un objeto que ya estuviera clicado. Se deja de mostrar la descripción del objeto y el botón de Vender se desactiva. 4. Clicar sobre el botón Vender. Se vende el objeto clicado y seleccionado. 5. El cursor se mueve por encima de cualquiera de los botones activos. Se ilumina levemente de naranja. 6. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso. 7. Se mantiene el último objeto clicado del mismo color naranja que en el caso de ser clicado.

Tabla 93- UI-10 Vender

ID	UI-11 Menú <i>in-game</i>
Descripción	<p>En este menú se puede salir de la partida o consultar diferentes estados de la partida. Como en el caso del menú principal, se tiene el botón opciones, aunque no tiene ninguna funcionalidad en este momento.</p>
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Continuar. Se accede a UI-07 Mapa. 2. Clicar sobre el botón Salir. Se sale del juego. 3. Clicar sobre el botón Stats. Se accede a UI-12 Consulta Stats. 4. Clicar sobre el botón Inventario. Se accede a UI-13 Inventario. 5. Clicar sobre el botón Habilidades. Se accede a UI-14 Consulta habilidades. 6. Clicar sobre el botón Instrucciones. Se accede a UI-15 Consulta instrucciones. 7. El cursor se mueve por encima de cualquiera de los botones activos. Se ilumina levemente de naranja. 8. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.
Ejemplo gráfico	

Tabla 94- UI-11 Menú in-game

ID	UI-12 Consulta de los Stats
Descripción	En esta interfaz se pueden consultar las características del personaje sumando los objetos que lleva en el Inventario que no son del tipo <i>CONSUMABLE</i> y las características.
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Volver. Se accede a UI-11 Menú <i>in-game</i>. 2. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 3. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.

Tabla 95- UI-12 Consulta de los Stats

ID	UI-13 Inventario
Descripción	En esta interfaz se muestran los objetos del inventario. Los objetos tipo <i>WEAPON</i> y <i>CLOTHES</i> están desactivados. Los objetos del tipo <i>CONSUMABLE</i> se pueden clicar y al hacerlo se activa el botón Usar para consumir el objeto. Además, al mover el cursor por encima de cualquiera de los objetos se muestra su descripción.
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Volver. Se accede a UI-11 Menú <i>in-game</i>. 2. Clicar sobre un objeto no seleccionado. Se ilumina el botón de naranja, se muestra su descripción y se activa el botón Usar. 3. Clicar sobre un objeto ya seleccionado. Se deja de mostrar su descripción y se desactiva el botón Usar. 4. Clicar sobre el botón Usar. Se elimina el objeto del inventario y se realiza el efecto de su descripción. 5. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja y si es un objeto muestra su descripción. 6. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso. 7. Se mantiene el último objeto clicado del mismo color naranja.

Tabla 96- UI-13 Inventario

ID	UI-14 Consulta de las habilidades
Descripción	En esta interfaz se pueden consultar las habilidades en su nivel actual y las posibles habilidades que se pueden escoger si aún no están escogidas.
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Volver. Se accede a UI-11 Menú <i>in-game</i>. 2. Mover el cursor por encima de alguna de las habilidades que ya se han escogido. Se muestra la información correspondiente a la habilidad. 3. Mover el cursor por encima de alguna de las habilidades que aún no se han escogido. Se muestra la información correspondiente a la habilidad si se subiera a nivel uno. 4. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 5. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.

Tabla 97- UI-14 Consulta de las habilidades



ID	UI-15 Consulta de las instrucciones
Descripción	En esta interfaz se pueden consultar los controles con los que se interactúa en el mapa.
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Volver. Se accede a UI-11 Menú <i>in-game</i>. 2. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 3. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.

Tabla 98 - UI-15 Consulta de las instrucciones

ID	UI-16 Interfaz de combate
Descripción	<p>Esta es la interfaz base para el combate. Desde esta interfaz se puede consumir un objeto, atacar usando una habilidad o ataque base y ganar o perder el combate. Cuando se ataque o se use un objeto, la interfaz queda paralizada hasta que el enemigo devuelva el ataque.</p> <p>Dependiendo del héroe escogido, atacar interactúa de una manera o de otra. Si se ha escogido al guerrero se abre una interfaz para tirar los dados de fuerza. Si se ha escogido al mago se abre una interfaz para tirar los dados de magia. Si se ha escogido al pícaro se abre una interfaz para elegir si hacer el ataque desde la invisibilidad.</p>
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el cualquier botón habilitado diferente al de objetos. Se accede a UI-18 Dado de fuerza si es el guerrero, a UI-19 Dado de magia si es el mago o a UI-20 Interfaz de invisibilidad si se trata del pícaro. 2. Clicar sobre el botón Objetos. Se accede a la UI-17 Usar objeto. 3. Ganar el combate final. Se accede a UI-28 Victoria. 4. Perder el combate. Se accede a UI-27 <i>Game Over</i>. 5. Ganar un combate diferente al combate final. Se accede a UI-23 Oro ganado. 6. El cursor se mueve por encima de cualquiera de los botones activos. Se ilumina levemente de naranja. 7. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.
Ejemplo gráfico	

Tabla 99 - UI-16 Interfaz de combate

ID	UI-17 Usar objeto
Descripción	<p>En esta interfaz se pueden ver los objetos del tipo <i>CONSUMABLE</i>, estos se pueden clicar, mostrando una descripción de los efectos del objeto y cambiando el botón Volver por el botón Usar.</p> <p>Se dispone también de los botones de la interfaz Página anterior y Página siguiente ya que la interfaz muestra seis objetos y se pueden poseer hasta nueve de ellos. Estos botones solo están habilitados si se tiene más de seis objetos del tipo nombrado. En la primera página solo está activado el botón Página siguiente y en la segunda página el botón Página anterior.</p> <p>Si se usa un objeto el jugador pierde el turno.</p>
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Volver. Se accede a UI-16 Interfaz de combate. 2. Clicar sobre un objeto no seleccionado. Se muestra la información sobre el objeto, se desactiva el botón Volver y se activa el botón Usar. 3. Clicar sobre un objeto ya seleccionado. Se deja mostrar la información sobre el objeto, se desactiva el botón Usar y se activa el botón Volver. 4. Clicar sobre el botón Usar. Se consume el objeto y finaliza el turno del jugador. 5. Clicar sobre Página siguiente. Muestra la segunda página de objetos. 6. Clicar sobre Página anterior. Muestra la primera página de objetos. 7. Clicar sobre el botón Volver. Se accede a UI-16 Interfaz de combate en estado paralizado para ser atacado por el enemigo. 8. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso. 9. Se mantiene el último objeto clicado del mismo color naranja que en el caso de ser clicado.

Tabla 100 - UI-17 Usar objeto

ID	UI-18 Dado de fuerza
Descripción	<p>En esta interfaz se lanza el dado para ejecutar el ataque. Cada número en un recuadro es el número del dado. Cada número debajo es el bono de ataque que entrega esa tirada en el dado.</p> <p>Si se clica el botón Lanzar dado se ejecuta el ataque. Si se clica sobre la habilidad elegida previamente, es posible modificar el ataque realizado si se trata del guerrero.</p> <p>Si se trata del pícaro es posible hacerlo si se ha decidido no aplicar invisibilidad.</p>
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Lanzar dado. Se ilumina de naranja el dado que se corresponde al resultado y desaparece la interfaz. Si no se gana el combate, se recibe el ataque del enemigo y si tampoco finaliza el combate, se vuelve a UI-16 Interfaz de combate. 2. Clicar sobre el botón de la habilidad escogida previamente para realizar el ataque. Cancela el ataque si es el guerrero o si es el pícaro y no se ha aplicado la invisibilidad, permitiendo elegir de nuevo el ataque. En el caso del guerrero se accede a UI-16 Interfaz de combate. En el caso del pícaro se accede a UI-21 Interfaz de venenos. 3. El cursor se mueve por encima del botón activo Lanzar dado. Se ilumina levemente de naranja. 4. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.

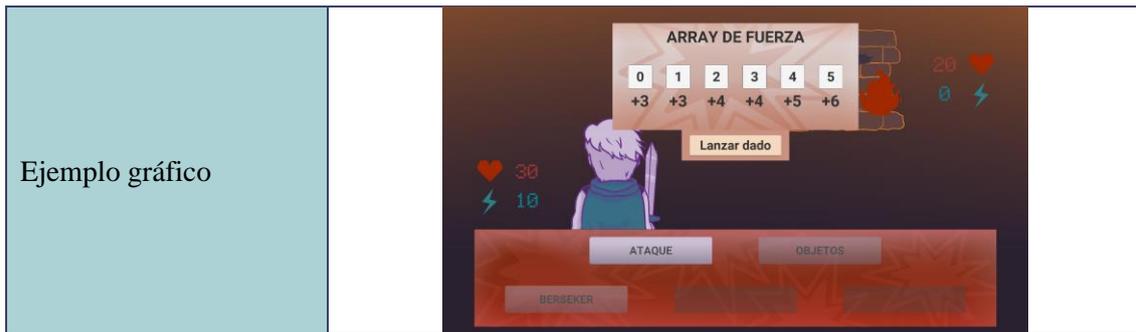


Tabla 101- UI-18 Dado de fuerza

ID	UI-19 Dado de magia
Descripción	<p>En esta interfaz se lanza el dado para ejecutar el ataque. Cada número en un recuadro es el número del dado. Cada número debajo es el bono de magia que entrega esa tirada en el dado.</p> <p>Si se clicla el botón Lanzar dado se ejecuta el ataque. Si se clicla sobre la habilidad elegida previamente, es posible modificar el hechizo realizado.</p>
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Cliclar sobre el botón Lanzar dado. Se ilumina de naranja el dado que se corresponde al resultado y desaparece la interfaz. Si no se gana el combate, se recibe el ataque del enemigo y si tampoco finaliza el combate, se vuelve a UI-16 Interfaz de combate. 2. Cliclar sobre el botón de la habilidad escogida previamente para realizar el hechizo. Cancela el ataque, permitiendo elegir de nuevo, y se vuelve a UI-16 Interfaz de combate. 3. El cursor se mueve por encima del botón activo Lanzar dado. Se ilumina levemente de naranja. 4. Se clicla cualquiera de los botones. Se ilumina el botón cliclado de un color naranja más intenso.

Tabla 102- UI-19 Dado de magia

ID	UI-20 Interfaz de invisibilidad
Descripción	<p>En esta interfaz se elige si se aplica la invisibilidad y, si se hace, de que tipo.</p>
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Cliclar sobre el botón Atrás. Se accede a UI-16 Interfaz de combate. 2. Cliclar sobre cualquiera de las opciones disponibles y activas. Se accede a UI-21 Interfaz de venenos 3. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 4. Se clicla cualquiera de los botones. Se ilumina el botón cliclado de un color naranja más intenso.

Tabla 103 - UI-20 Interfaz de invisibilidad

ID	UI-21 Interfaz de venenos
Descripción	En esta interfaz se elige si se aplica algún veneno y, si se hace, de que tipo.
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Atrás. Se accede a UI-20 Interfaz de invisibilidad. 2. Clicar sobre cualquiera de las opciones disponibles y activas. Si se ha elegido aplicar invisibilidad se accede a UI-22 Dado de invisibilidad. 3. Clicar sobre cualquiera de las opciones disponibles y activas. Si se ha elegido no aplicar invisibilidad se accede a UI-18 Dado de fuerza. 4. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 5. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.

Tabla 104 - UI-21 Interfaz de venenos

ID	UI-22 Dado de invisibilidad
Descripción	<p>En esta interfaz se lanza el dado para ejecutar el duelo de invisibilidades. Cada número en un recuadro es el número del dado. Cada número debajo es el bono de sigilo que entrega esa tirada en el dado.</p> <p>Si se clica el botón Lanzar dado se ejecuta el combate de invisibilidad para hacer o no al jugador invisible. Si se clica sobre la habilidad elegida previamente es posible modificar cualquiera de las decisiones en este turno del jugador.</p>
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Lanzar dado. Se ilumina de naranja el dado que se corresponde al resultado y se accede a UI-18 Dado de fuerza. 2. Clicar sobre el botón de la habilidad escogida previamente para realizar el ataque. Cancela el ataque, permitiendo elegir de nuevo, y se vuelve a UI-21 Interfaz de venenos. 3. El cursor se mueve por encima del botón activo Lanzar dado. Se ilumina levemente de naranja. 4. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.

Tabla 105 - UI-22 Dado de invisibilidad

ID	UI-23 Oro ganado
Descripción	En esta interfaz se dispone de un botón para lanzar el dado y obtener oro. Es la interfaz mostrada al ganar un combate diferente al combate final. En la parte superior se muestra el número de oro perdido del potencial premio.
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Lanzar dado. Se ilumina de naranja el dado que se corresponde al resultado y se accede a UI-24 Reintentar. 2. El cursor se mueve por encima del botón activo Lanzar dado. Se ilumina levemente de naranja. 3. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.
Ejemplo gráfico	

Tabla 106 - UI-23 Oro ganado

ID	UI-24 Reintentar
Descripción	Una ventana que da al jugador la opción de repetir la tirada del dado para ganar oro a cambio de un pequeño coste. Sí se acepta se vuelve a tirar los dados actualizando la información de la interfaz anterior y si se rechaza se accede a la interfaz de subida de características.
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Sí. Se accede a UI-21 Oro ganado y se actualiza la información en ella. 2. Clicar sobre el botón No. Se accede a UI-25 Subida de características. 3. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 4. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.

Tabla 107 - UI-24 Reintentar

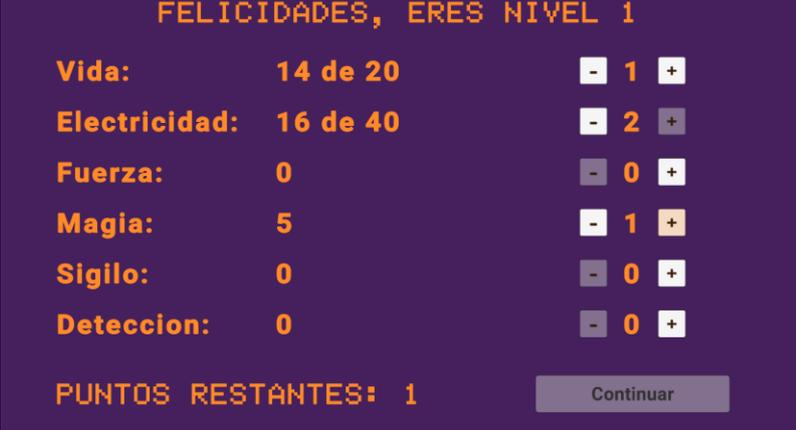
ID	UI-25 Subida de características
Descripción	<p>En esta interfaz se dispone de cinco puntos para repartir entre seis características. Sólo se pueden subir dos puntos de una misma característica. Para gastar uno de los puntos se debe pulsar el botón +. Si se quiere cancelar ese gasto de punto se puede pulsar el botón -.</p> <p>El botón - solo se activa en caso de haber gastado puntos en esa característica. Si ya se han gastado dos puntos en esa característica se desactiva el botón +.</p> <p>En la parte inferior se muestran los puntos restantes y en la parte superior el nivel adquirido.</p> <p>Cuando se han gastado todos los puntos se activa el botón Continuar, se desactiva si se recuperan los puntos guardados con el botón -.</p> <p>Al clicar sobre el botón Continuar, si el nivel es par se accede a la interfaz de subida de habilidades. Sino se accede al mapa.</p>
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón +. Añade un punto de característica. Si es el primero activa el botón - para esa característica. Si es el segundo se desactiva el botón + para esa característica. Resta un punto en los puntos restantes. Si los puntos restantes se reducen a cero se activa el botón Continuar. 2. Clicar sobre el botón -. Resta un punto de característica. Si el resultado de clicar el botón es que aún queda un punto gastado en la característica activa el botón + para esa característica. Sino se desactiva el botón - para esa característica. Suma un punto en los puntos restantes. Si se tenía cero puntos restantes y ahora se tiene uno desactiva el botón Continuar. 3. Clicar el botón Continuar en nivel par. Se accede a UI-26 Subida de habilidades. 4. Clicar el botón Continuar en nivel impar. Se accede a UI-07 Mapa. 5. El cursor se mueve por encima del botón activo Lanzar dado. Se ilumina levemente de naranja. 6. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.
Ejemplo gráfico	

Tabla 108 - UI-25 Subida de características

ID	UI-26 Subida de habilidades
Descripción	<p>En esta interfaz se dispone de dos o tres puntos, dependiendo del personaje, para repartir entre las habilidades disponibles. Sólo se pueden subir un punto de una misma habilidad. Para gastar uno de los puntos se debe pulsar el botón +. Si se quiere cancelar ese gasto de punto se puede pulsar el botón -.</p> <p>El botón - solo se activa en caso de haber gastado puntos en esa habilidad. Lo que desactiva el botón +.</p> <p>En la parte inferior se muestran los puntos restantes.</p> <p>Existe también un panel que muestra la información de la habilidad que se quiere modificar.</p> <p>Cuando se han gastado todos los puntos se activa el botón Continuar que se desactiva si se recuperan los puntos guardados usando el botón -.</p> <p>Al clicar sobre el botón Continuar se accede al mapa.</p>
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón +. Añade un punto de habilidad. Activa el botón - para esa habilidad. Desactiva el botón + para esa habilidad. Resta un punto en los puntos restantes. Si los puntos restantes se reducen a cero se activa el botón Continuar. 2. Clicar sobre el botón -. Resta un punto de habilidad. Activa el botón + para esa habilidad. Desactiva el botón - para esa habilidad. Suma un punto en los puntos restantes. Si se tenía cero puntos restantes y ahora se tiene uno desactiva el botón Continuar. 3. Clicar el botón Continuar. Se accede a UI-07 Mapa. 4. El cursor se mueve por encima del botón activo Lanzar dado. Se ilumina levemente de naranja. 5. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.
Ejemplo gráfico	

Tabla 109 - UI-26 Subida de habilidades

ID	UI-27 <i>Game Over</i>
Descripción	<p>Esta es la interfaz mostrada cuando se pierde en combate. Se puede reintentar el juego desde el principio o salir del juego.</p>
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Reintentar. Se accede a UI-01 Menú principal. 2. Clicar sobre el botón Salir. Se sale del juego. 3. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 4. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.

Tabla 110 - UI-27 *Game Over*

ID	UI-28 Victoria
Descripción	Esta interfaz muestra una felicitación al jugador por superar el juego al vencer en el último combate y un botón Salir para salir del juego.
Acciones en la interfaz	<ol style="list-style-type: none"> 1. Clicar sobre el botón Salir. Se sale del juego. 2. El cursor se mueve por encima de cualquiera de los botones. Se iluminan levemente de naranja. 3. Se clica cualquiera de los botones. Se ilumina el botón clicado de un color naranja más intenso.

Tabla 111 - UI-28 Victoria

4.5 Camino y mapa

Dependiendo de cómo derrote al enemigo el jugador, se toma un camino u otro para llegar finalmente al enemigo final, el *Ransomware*.

Este es el esquema del mapa del juego. El jugador empieza desde el enemigo llamado *Firewall*:

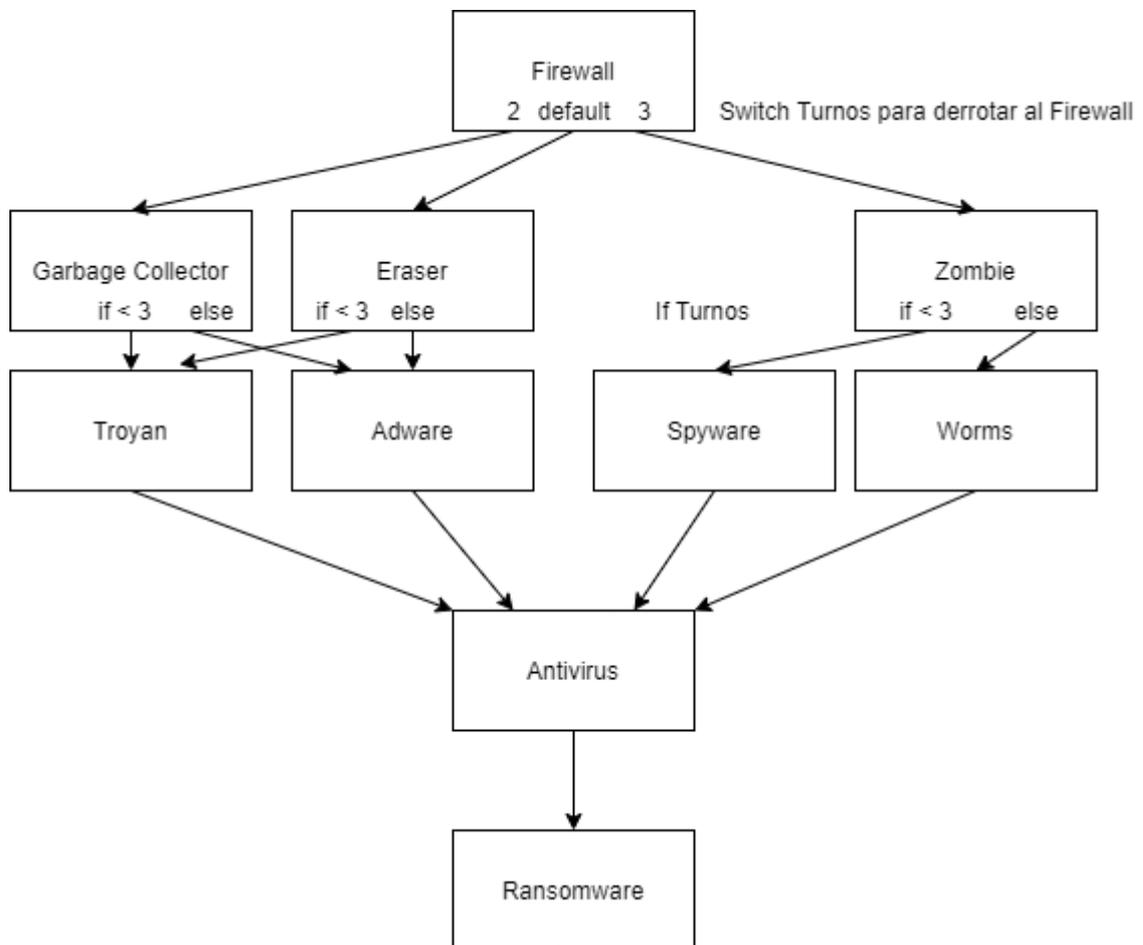


Diagrama 2 - Mapa del juego

Para dividir los caminos tenemos estamentos de programación en las salas del primer y segundo nivel. En la primera sala se trata de un *switch* de turnos. Si se tardan dos turnos el jugador se dirige hacia el *Garbage Collector*, si son tres turnos se dirige hacia el *Zombie* y en cualquier otro caso se dirige al *Eraser*. En las segundas salas se trata de un *if – else* también dependiente del número de turnos. Se pensó en cambiar los números de turno por el oro, aunque era muy confuso para el jugador. También se pensó en añadir al Antivirus un *while* para el oro acumulado,

pero se vio que hacía muy aburrido el juego. Finalmente se eligió los turnos como una forma de incentivar al jugador a elegir, si puede, la longitud del combate.

4.6 Controles

Existen siete controles en el juego. Seis controles en el mapa y uno fuera del mapa:

- Mapa:
 - Los controles de movimiento cuando se está en el mapa:
 - W mueve hacia delante el avatar.
 - S mueve hacia atrás el avatar.
 - A mueve hacia la izquierda el avatar.
 - D mueve hacia la derecha el avatar.
 - El botón E se utiliza para interactuar con los personajes no jugadores en el mapa.
 - El botón Q se utiliza para abrir el menú desde el mapa.
- Menús y combate:
 - El botón izquierdo del ratón se usa para seleccionar todos los botones.

4.7 Animaciones

Aunque el arte de la animación ha sido externo. Las transiciones y la animación de cada *sprite* en el motor de juego han sido realizadas por el autor.

Se poseen dos tipos de animaciones. Las animaciones de movimiento y las animaciones de combate.

Las animaciones de movimiento tienen, para cada uno de los tres personajes, animación de movimiento hacia arriba, hacia abajo, hacia la derecha y hacia la izquierda, priorizando, en caso de movimiento diagonal, la tecla pulsada primero.

Las animaciones de combate se tratan de un pequeño movimiento arriba y abajo del *sprite* tanto para el caso de ser el jugador quien ataque como en el caso de que sea el enemigo el atacante.

4.8 Arquitectura

4.8.1 Arquitectura *hardware*

Para jugar a este juego se necesita un ratón, una pantalla, un teclado, un ordenador y, si se juega en la *web*, conexión a Internet.

4.8.2 Arquitectura *software*

La arquitectura *software* se divide en dos capas, la capa de presentación y la capa de negocio. La primera capa contiene las clases encargadas de mostrar los elementos en la interfaz de usuario e interactuar con esta y la segunda capa se ocupa de la gestión de datos interna del juego. Se ha intentado mantener esta división durante el proyecto.

4.9 Diagrama de clases

En este apartado se va a explicar detalladamente como están diseñadas todas las clases y las diferentes secciones del código en formato UML.

Se ha dividido el juego en diferentes secciones. Para cada sección se va a detallar las diferentes clases que lo componen. Es importante destacar que por defecto en Unity todas las

clases heredan de la clase de Unity MonoBehaviour. Esta clase no se va a mostrar en los diagramas por cuestiones de espacio. Por tanto, si no se indica de otra manera, todas las clases heredan de esta.

Estas son las secciones:

- Menú principal
- Mapa, gestores de interfaz y oro
- Datos
- Gestor del personaje en combate
- Gestores de la interfaz de combate
- Enemigos
- Base de datos para habilidades y objetos e inventario
- Subida de habilidades y características
- Tienda
- Menú *in-game*
- Victoria y derrota

4.9.1 Menú principal

El primer diagrama es el del menú principal. En esta sección se encuentran diez clases que cubren desde que se entra al menú hasta que se aprieta el botón de Comenzar Partida.

Unity funciona mediante *scripts* que generalmente se asocian a un objeto en la escena de juego, este script es el responsable del comportamiento del objeto. Todas las clases aquí expuestas son *scripts* asociados a objetos con un par de excepciones que se nombrarán cuando aparezcan en el diagrama de clases UML.

La clase InitialMenu cubre la parte en la que se escoge héroe. Para esto se debe explicar una propiedad de Unity que es el *GameObject*. El *GameObject* es el objeto de la escena nombrado anteriormente y que puede contener diferentes propiedades como animaciones, posición, *Sprite*, botón o *scripts*.

Para inicializar estas variables en los *scripts* se pueden usar dos métodos: buscar al objeto en la escena al empezar la partida mediante código o tener la variable inicializada a través de la escena asociando un objeto de esa escena al *script*. Se puede asociar el objeto de la escena simplemente arrastrando el componente u objeto desde la escena o desde el menú de desarrollo hasta el *script* en la escena. En la *Ilustración 2* se puede ver un ejemplo usado en el *script* InitialMenu.

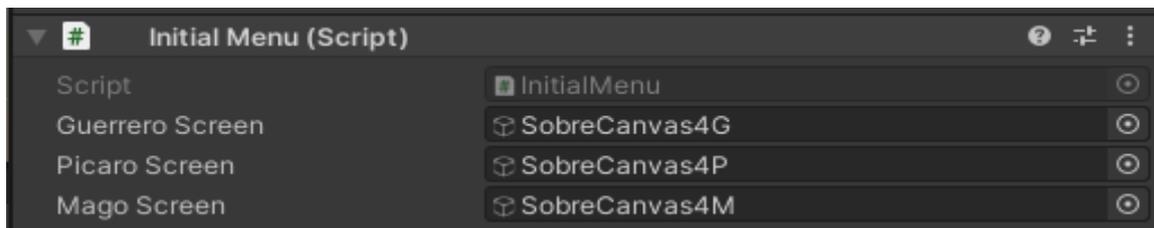


Ilustración 2 - Ejemplo de inicialización de un script

La clase InitialMenu también contiene la opción de salir del juego en el menú inicial y gestiona la pantalla a mostrar en el tutorial entre las tres posibles. Esta clase se complementa con

`OnHighlightedHeroes` que hereda de las clases de Unity `IPointerEnterHandler` y `IPointerExitHandler` que se encargan de gestionar cuando el cursor se mueve sobre el objeto que tiene asociado en la escena el *script*. `OnHighlightedHeroes` se encarga de gestionar la iluminación de los héroes cuando se mueve el cursor sobre ellos y de mostrar la información correspondiente a las descripciones y *stats* de los héroes. En este caso, para la elección de héroes, `InitialMenu` es la capa de negocio y `OnHighlightedHeroes` es la capa de presentación.

La gestión de las habilidades reside en siete clases. `Abilities` proporciona una estructura para que las clases específicas de cada personaje hereden esta estructura y puedan gestionar la elección de las habilidades. Esta sería la capa de negocio. La capa de presentación reside en cada una de las clases `OnHighLightened` respectivas a cada héroe que gestionan la información visual de cada uno de ellos y sus habilidades.

Finalmente, falta la explicación sobre `PlayGame`. `PlayGame` contiene un elemento nuevo que es una variable que contiene a la propia clase del *script*, es decir, que el *script* se contiene a sí mismo. Esta variable se usa como un *singleton*. Se necesita usar el patrón *singleton* porque Unity elimina por defecto los objetos de una escena cuando se avanza a otra. Al estar el menú inicial y el juego en escenas diferentes se perdería la información de este *script*. Como este *script* contiene todos los datos de las elecciones tomadas por el jugador queremos mantenerlo.

Para ello se le dice a Unity, mediante un procedimiento del propio Unity que no destruya el objeto que contiene el *script* al cargar la nueva escena. Pero para almacenar esta información se necesita crear en la segunda escena un objeto igual que contenga el *script*, pero que no contenga la información del *script* de la primera escena. Por tanto, se crea este *singleton* que asegura que cuando empiece la nueva escena la copia de la segunda escena se elimine y sea sustituido por el *script* creado en la primera escena [19].

Otro uso que se le puede dar a contener como *static* la propia clase en una variable es, que, si se hace pública, las demás clases en el juego pueden acceder a todos los datos del *script* directamente llamando a dicha variable. De esta manera no se corre el riesgo de acceder a datos incorrectos ya que solo existe un *casting* de este.

Los métodos `Start` y `Awake` son métodos propios de Unity que son ejecutados una única vez cuando se activan los objetos que contienen al *script*. Al iniciar una escena, `Awake` es ejecutado primero y seguidamente lo hace `Start` [20]. El método `Update` es un método que se ejecuta cada *frame* del videojuego y el método `OnEnable` se ejecuta cuando el objeto que contiene al *script* es activado en la escena sin límite de ocasiones como ocurría con `Start` y `Awake`.

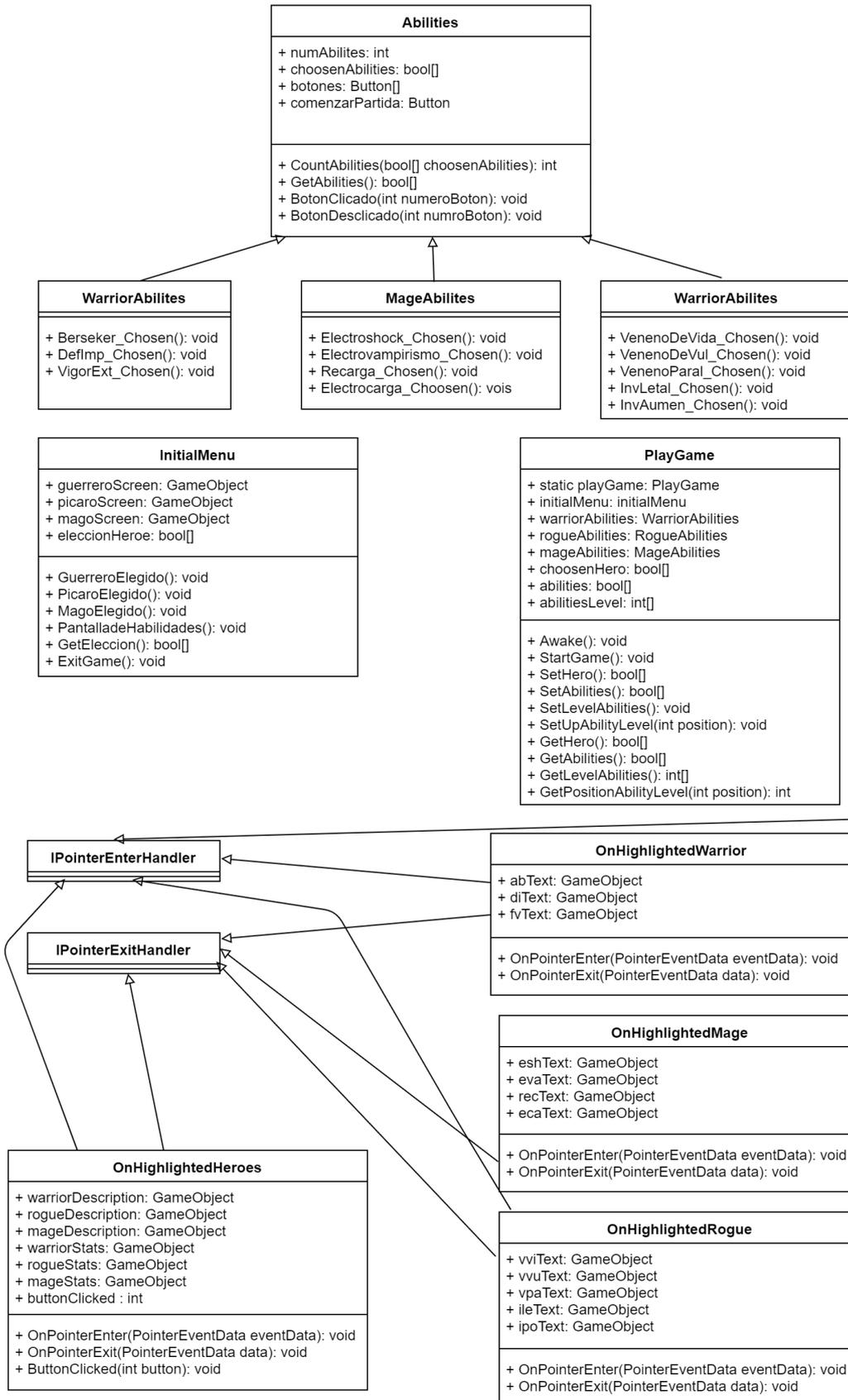


Diagrama 3 - Menú inicial



4.9.2 Mapa, gestores de interfaz y oro

Esta sección es la que se ocupa de gestionar el movimiento del personaje, la interacción con otros personajes, el oro que tiene el jugador y todas las interfaces mostradas desde cuando se entra en combate, hasta cuando se entra en la tienda y hasta cuando se muestra la interfaz de victoria.

Gold es una clase simple que gestiona el oro que tiene el jugador y lo hace fácilmente accesible al resto de clases.

DoorManager es también una clase muy simple que controla que puerta se abre al final de un combate. Obtiene la información del enemigo derrotado y los turnos que se han necesitado y en base a esa información decide que puerta se va a abrir.

MainCharacterController es la clase que se ocupa de gestionar el movimiento y las animaciones del jugador. La clase contiene un método llamado FixedUpdate. Este es un método que se usa para actualizar al personaje cuando se están usando físicas propias del motor Unity.

Finalmente, la clase StartToEndManager gestiona todas las transiciones entre interfaces del juego que tienen que ver con el flujo principal. Por ejemplo, no controla el flujo interno de la sala de combate o el flujo interno de la tienda, pero si es responsable de gestionar cuando la interfaz de la sala de combate o la tienda deben mostrarse.

No se ha incluido una clase RepetirOro que gestiona la repetición de la tirada cuando el combate se finaliza.

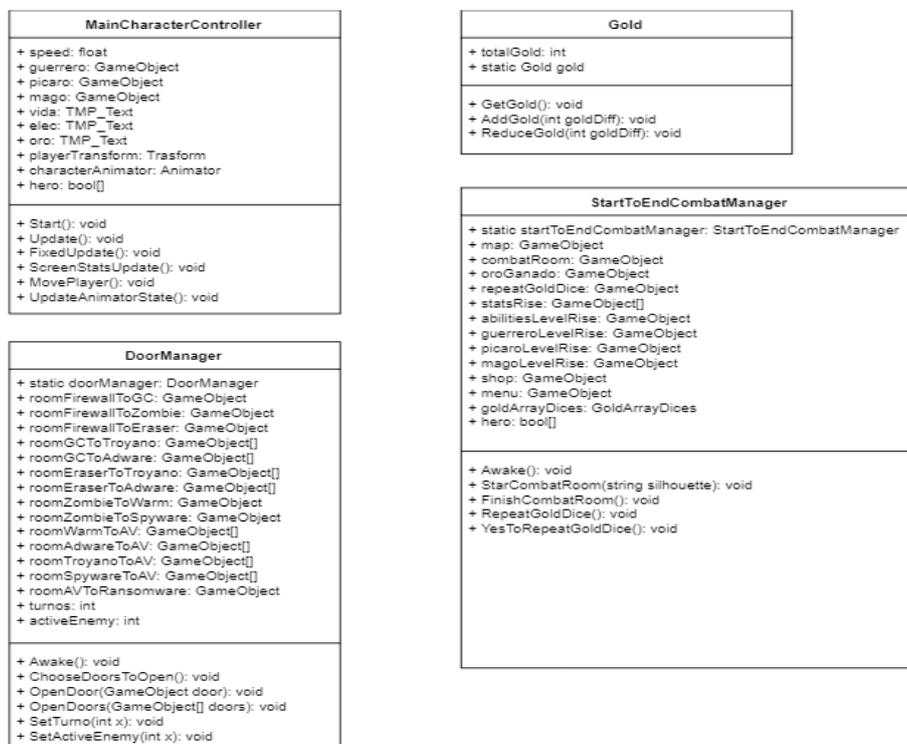


Diagrama 4- Mapa y gestores

4.9.3 Datos

En este apartado se muestran los diagramas de clase de las clases encargadas de los datos tanto en combate como fuera de combate.

DiceManager es la clase a la que otras clases que no son dados pueden acceder y mediante el acceso a esta clase pueden solicitar una tirada de cualquier tipo de dado. También se encarga de limpiar la interfaz cuando el dado ya ha sido lanzado. Contiene los cuatro tipos de dados y un quinto tipo que se utiliza si se quiere hacer referencia a todos los dados implicados en el combate al mismo tiempo.

Se puede observar que DiceManager contiene instancias de las otras clases de dados. Esto permite al *script* acceder a dichos *scripts* en la escena. De esta manera se puede activar el método GetResult de cualquiera de ellos desde el método UseDice una vez DiceManager sea llamado por otra clase indicando el tipo de dado del que desea una tirada.

El resto de las clases heredan de ArrayDices. Cada una de las clases que hereda tiene un atributo diceNumber que indica el último resultado en el dado. También contiene otros atributos y métodos, como, por ejemplo, el método para iluminar el dado resultante.

Cada clase que hereda implementa el método GetResult o GetGoldResult, que realiza la tirada dependiendo de los datos que se hayan introducido en la escena. La más complicada es la clase GoldArrayDices ya que almacena los dados para los cuatro niveles de combate.

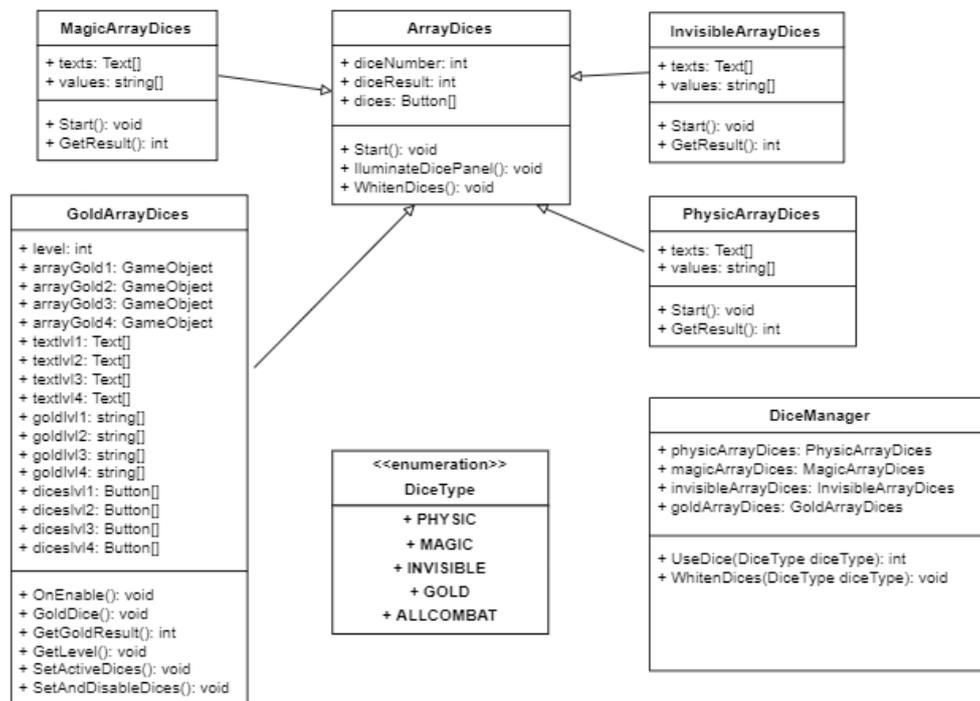


Diagrama 5 – Datos



4.9.4 Gestor del personaje en combate

En este apartado se expone la clase encargada de gestionar al personaje principal en combate, como se expuso anteriormente la clase encargada de la gestión del movimiento del personaje en el mapa. Esta clase es posiblemente la clase más compleja y extensa del juego. Debe controlar muchas variables y situaciones de juego. Desde la gestión de los *stats* del personaje, incluyendo el estar por debajo de cero de vida, hasta los aumentos de los objetos, el daño realizado por un ataque o el control de los efectos de las habilidades como el veneno, la invisibilidad o la defensa extra temporal para reducir el daño recibido.

Se contiene referencias a la mayoría de los gestores del juego para poder obtener sus datos. También se introduce el *struct Stats*, que aparecerá también en los enemigos. Es un *struct* que contiene los datos numéricos sobre las características actualizadas del personaje. Se tienen tres métodos relacionados con este *struct* en la clase. Uno calcula los bonificadores de los objetos, *UpdateObjectStats*, otro las características propias del personaje, *UpdatePlayerStats* y el último, *UpdateStats*, suma ambas. Además, contiene diferentes métodos *Get* para que se pueda obtener su información desde otras clases.

Un método importante es el método *AttackManager*. Este método recibe el estado de los botones pulsados para determinar qué acción de ataque se ha elegido y ejecutar su correspondiente método. Este método también invoca la animación del personaje al atacar y es el método que invoca el reinicio de las interfaces de combate y dados.

Otra clase presente es *PlayerStatsText* que simplemente hace uso del componente de *Unity Text* para mostrar y modificar la información de la vida y electricidad del jugador en la interfaz.

Finalmente, *CharacterImage* se encarga de gestionar cual es la imagen de personaje que se muestra dependiendo de la elección inicial del jugador y activa la animación de ataque.

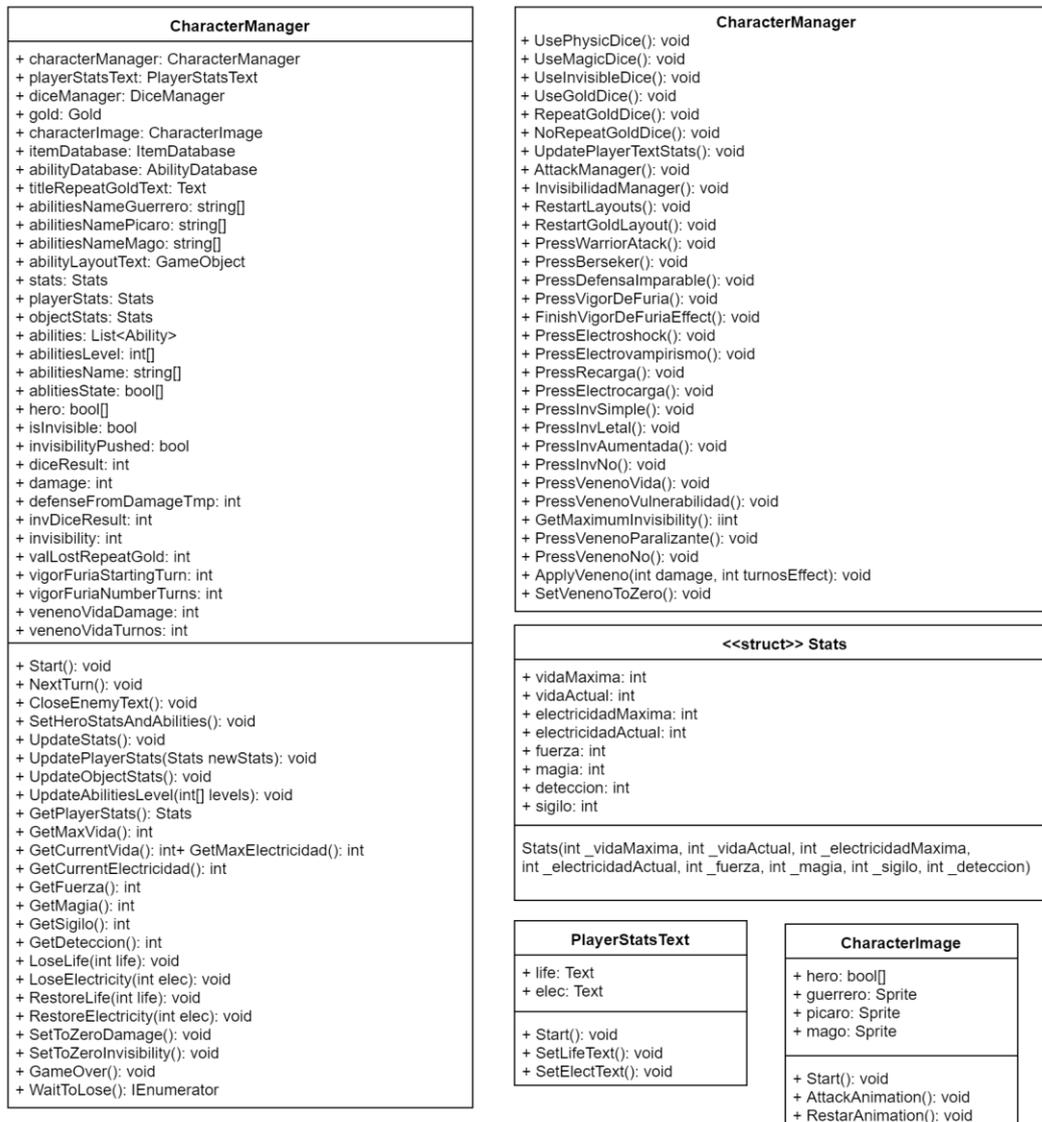


Diagrama 6 - Gestor de personaje en combate

4.9.5 Gestores de la interfaz de combate

Este apartado trata la gestión de las interfaces en el combate. Contiene la mayoría de las clases que gestionan la capa de presentación del juego. Aunque la clase Objects también contiene elementos de la capa de negocio.

Empezamos con las clases Objects y BackObjectButton. Estas dos clases gestionan la interfaz de los objetos en combate. BackObjectButton es la clase encargada de que cuando el jugador apriete el botón Volver se pueda volver a la interfaz correcta respecto al personaje elegido.

La clase Objects gestiona que objetos se muestran en la pantalla del jugador. Para ello obtiene la lista de ítems en el inventario, que, aunque se explicará más tarde, contiene las id de los objetos en la base de datos de los objetos y el *slot* de inventario que le corresponde al objeto. Además, gestiona los botones de ir a la siguiente página y volver de la anterior, gestiona el uso de un objeto y gestiona que se muestren sus descripciones cuando son seleccionados.



La clase `CombatRoomLayoutManager` gestiona la interfaz que debe mostrarse al iniciar el combate dependiendo del personaje escogido. Además, hace de intermediario entre el gestor de enemigos y el gestor del jugador en combate para indicar las finalizaciones de sus respectivos turnos y activa el panel de botones cuando el enemigo acaba su ataque.

`PressLayoutButton` gestiona la transición al pulsar el botón de habilidad y elegir el dado mostrado. Captura el botón que el jugador pulsa para que luego `CharacterManager` pueda gestionar los efectos. Además, reinicia las interfaces cuando recibe la activación desde `CharacterManager`.

Finalmente quedan las clases `GuerreroLayoutManager` y `OnHighlightWarriorButtons`. Para ambas clases existen sus correspondientes clases para pícaro y mago. Pero dado que ya se ha mostrado anteriormente que estas clases son muy parecidas y el hecho de que en este caso lo único que varía son los `GameObject` respectivos a los botones del panel inferior, se ha decidido no incluirlos en el diagrama.

La clase `OnHighlightWarriorButtons` se encarga de mostrar la información del daño que puede realizar la habilidad o el ataque cuando el jugador mueve el cursor sobre un botón de habilidad o ataque. Busca esta información en la base de datos de las habilidades y usa también la información de los *stats* de `CharacterManager`.

La clase `GuerreroLayoutManager` se encarga de mostrar los botones en el panel si la habilidad está al menos subida al primer nivel y una vez mostrada la habilidad, de desactivar la habilidad dependiendo del coste y la electricidad que el jugador tenga en el momento. También necesita hacer uso de la base de datos de las habilidades para obtener información sobre su coste.

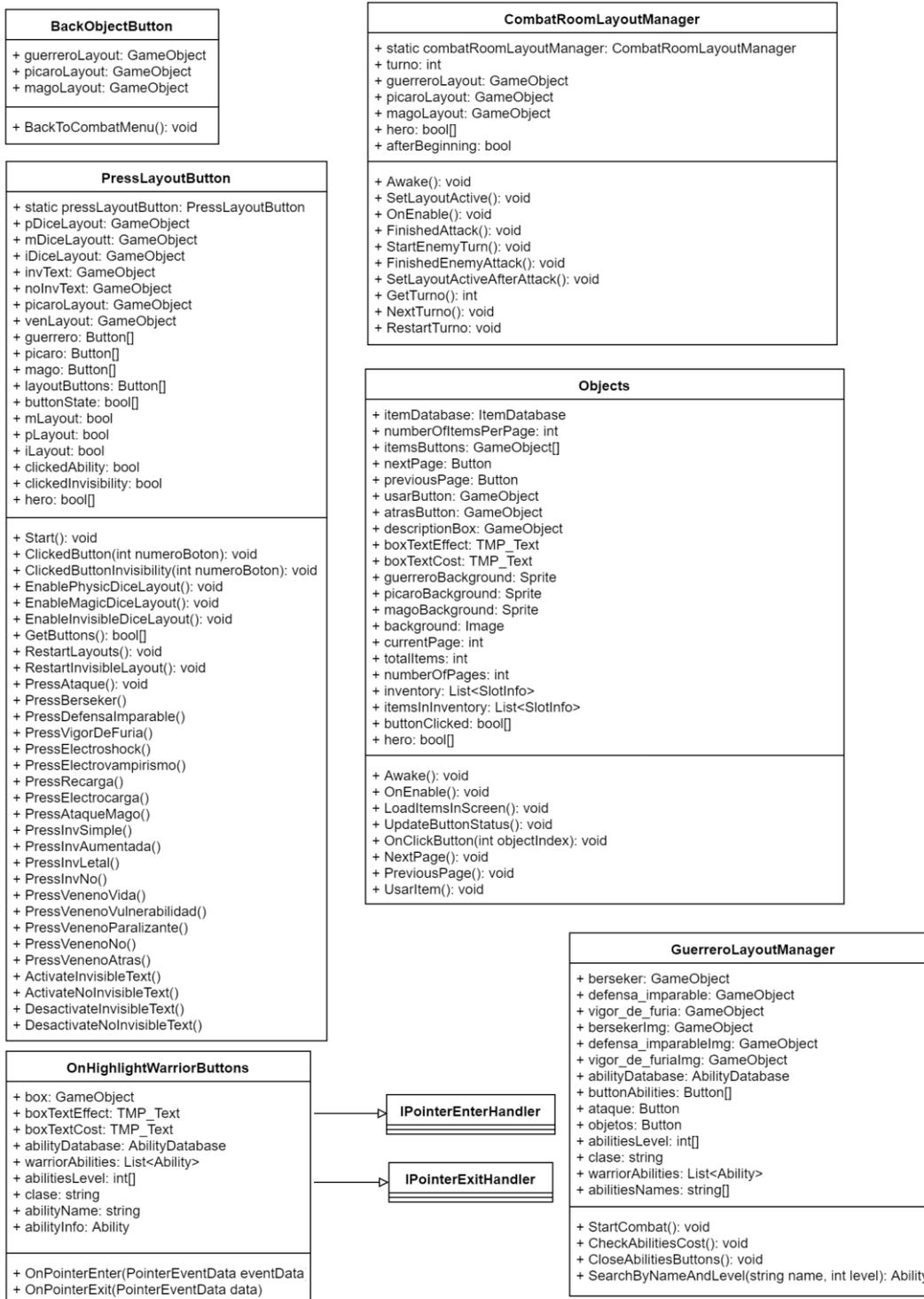


Diagrama 7 - Gestores de la interfaz de combate

4.9.6 Enemigos

En este apartado se explica la estructura de los enemigos. Se introduce una nueva función de Unity que es *virtual*. Esta palabra reservada permite a la clase implementar un método para luego ser sobrescrito por las clases que heredan de esta, aunque no obliga a hacerlo. Esto es usado por la clase genérica Enemy para que las clases específicas de los enemigos puedan usar la palabra reservada *override* y así conseguir que se ejecute el método específico para dicha clase.



EnemyManager es la clase encargada de comunicarse con el resto de las clases sobre el estado del enemigo activo. Es donde se calcula el resultado del duelo de invisibilidad del pícaro, es donde llega la información sobre los venenos que el jugador aplica al enemigo, es quien controla en todo momento que enemigo es el que hay en la escena y su nivel, es la clase que gestiona el inicio y final del turno del enemigo y es la clase que transmite el daño que el enemigo causa al jugador.

Enemy es la clase genérica de los enemigos que implementa la mayor parte su lógica, como la aplicación del daño recibido o el cálculo de la detección enemiga. La excepción son las particularidades de cada enemigo, como sus características y su ataque específico. Este método, Ataque, calcula para cada enemigo que ataque o habilidad debe usar y el daño que realiza.

Finalmente, EnemyStatsText, como en el caso del jugador, se encarga de mostrar en la interfaz los *stats* actuales del enemigo.

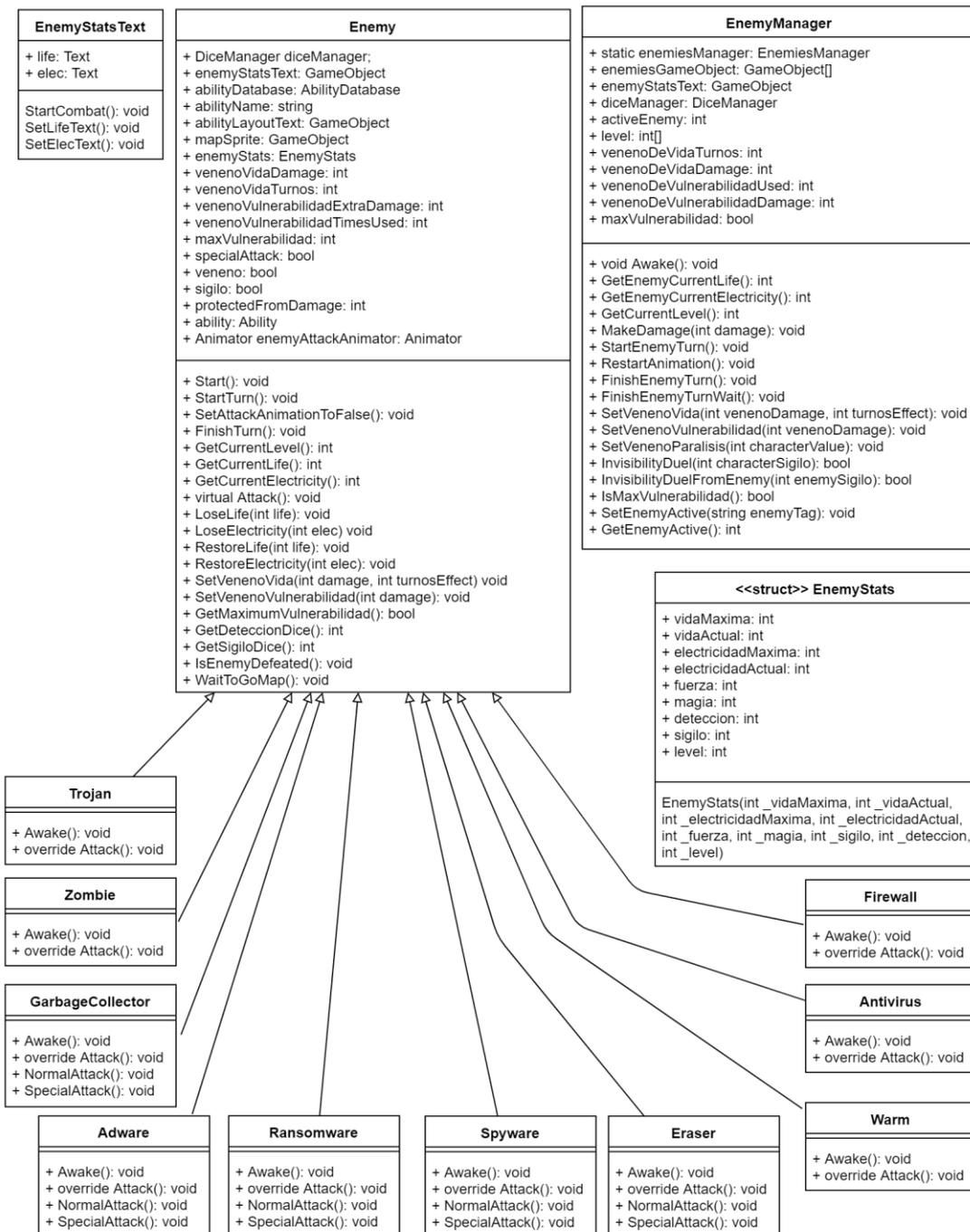


Diagrama 8 – Enemigos

4.9.7 Base de datos para habilidades y objetos e inventario

En este apartado se exponen las bases de datos que integran los objetos y las habilidades en el juego. Además, se expone el inventario que accede a la base de datos de los objetos y el gestor de las habilidades para el juego.

Este es un diagrama de clases peculiar. Nos encontramos con objetos que no heredan de MonoBehaviour por primera vez. En este caso se tienen seis clases que no heredan de MonoBehaviour: ItemDatabase y AbilityDatabase que heredan únicamente de ScriptableObject,



CSVToItemDB y CSVToDB que no tienen ninguna herencia y Ability e Item, que están contenidas en sus respectivas Database.

En Unity, se puede implementar una base de datos usando una clase que herede de ScriptableObject. Esta clase sirve para almacenar una gran cantidad de datos sin ver afectada la memoria cuando el juego se ejecuta como lo haría otra clase usando un gestor en la escena, ya que, una clase que herede de ScriptableObject, no necesita estar en la escena para poder ser accedida por otras clases.

Para acceder a los datos de las habilidades existe la clase CSVToDB, que es una clase que tiene una variable para definir la ruta por la que se accede al fichero .CSV que contiene toda la información de las habilidades. Usando esta clase se puede almacenar de forma permanente en la clase AbilityDatabase que hereda de ScriptableObject toda la información de este fichero .CSV. El guardado se realiza mediante una lista de la clase Ability, integrada en AbilityDatabase. Estas gestiones se pueden realizar de manera previa a ejecutar el juego.

La clase AbilitiesManager gestiona las habilidades y sus niveles actuales durante la ejecución del juego.

La sección de los ítems es muy parecida a la de las habilidades, razón por la que se ha decidido juntar ambas en un mismo apartado. En este caso la clase que accede al fichero .CSV que contiene todos los ítems y propiedades del juego es CSVToItemDB. Esta clase guarda la información en ItemDatabase que contiene una lista de elementos de la clase Item integrada en la propia ItemDatabase.

La clase SlotInfo contiene tres variables indicando el estado del *slot* en el inventario. Estas tres variables son *id*, que es el número de *slot* en el inventario, *isEmpty* que indica si el *slot* está libre e *itemId* que indica la id del ítem en la ItemDatabase.

Finalmente, está el Inventario. Es la clase que guarda información de los ítems que el jugador posee durante la partida. Para ello Inventory accede constantemente a la ItemDatabase. El inventario está implementado como una lista objetos de tipo SlotInfo. Cada vez que se necesita obtener información sobre un ítem específico se consulta la base de datos de los ítems. También permite eliminar un objeto del inventario, añadir un objeto, consultar los máximos objetos por tipo que permiten las reglas o devolver un ítem de la base de datos que otra clase necesite.

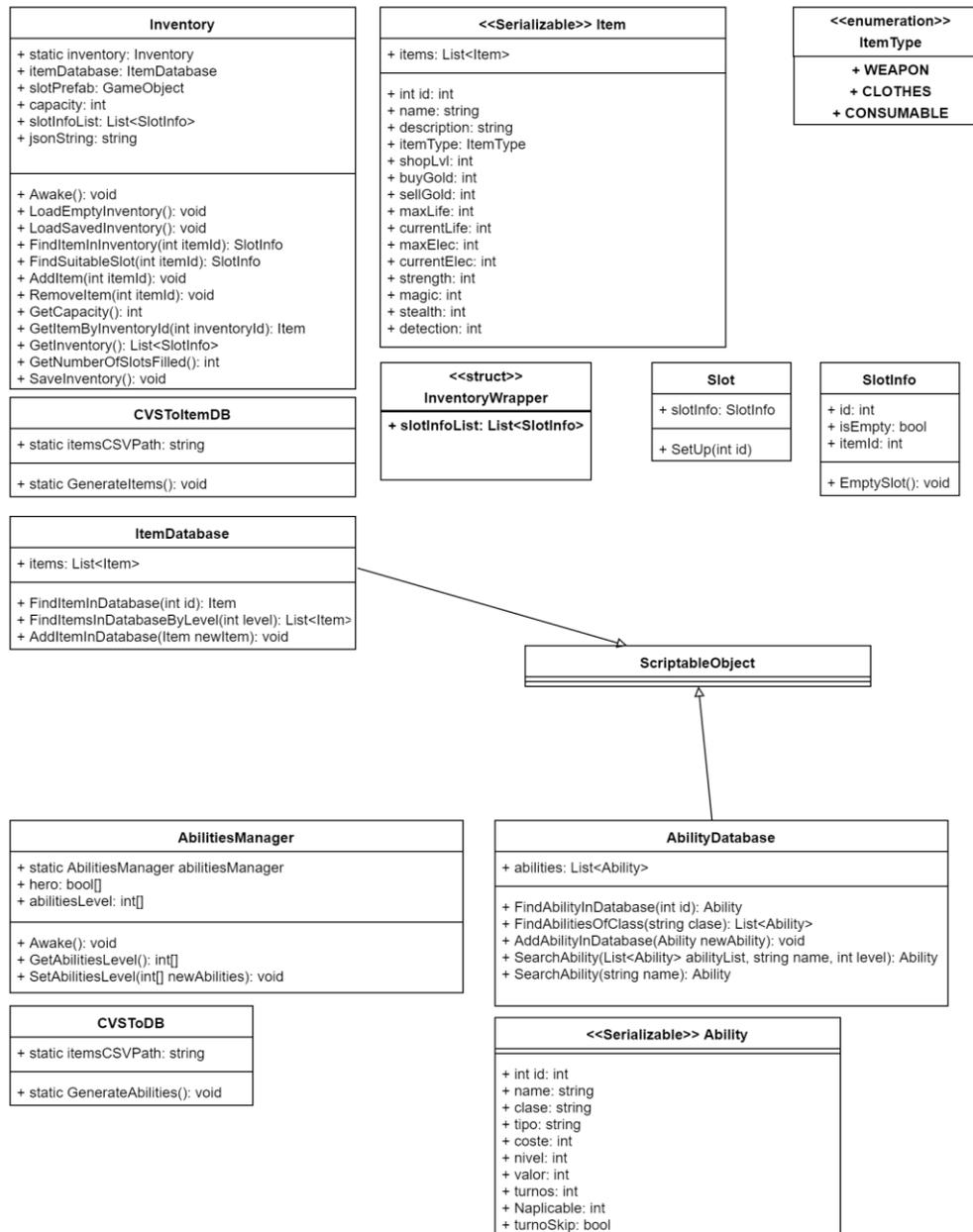


Diagrama 9 - Base de datos para habilidades y objetos e inventario

4.9.8 Subida de habilidades y características

En este apartado se expone como se gestionan las subidas de nivel. Tanto en el caso de las habilidades como en el caso de las características.

Las características las gestiona StatsLevelUpManager. Esta clase gestiona la desactivación y activación de botones de acuerdo con las reglas. También se encarga de detectar cuando el jugador pulsa continuar para avanzar a la siguiente pantalla. Aunque es StartToEndCombatManager la clase que gestiona si se va al mapa o a la interfaz de subida de características.

Las habilidades las gestiona AbilitiesLevelUpManager. La filosofía es la misma que en el caso anterior. La diferencia es que en este caso existen tres posibles interfaces y esta clase está



integrada dentro de tres objetos diferentes en la escena, cada uno con sus asignaciones propias de botones y número de habilidades permitidas.

OnHighlightMageAbilities controla la información que aparece cuando se mueve el cursor por encima de alguna de las habilidades. Muestra información de cómo evolucionaría la habilidad si se decide pulsar el botón sobre el que se está moviendo el cursor. También existen OnHighlightWarriorAbilities y OnHighlightRogueAbilities, pero, como se explicó con anterioridad, el cambio es mínimo entre las tres clases y se considera que una de ellas ya ilustra suficiente las tres.

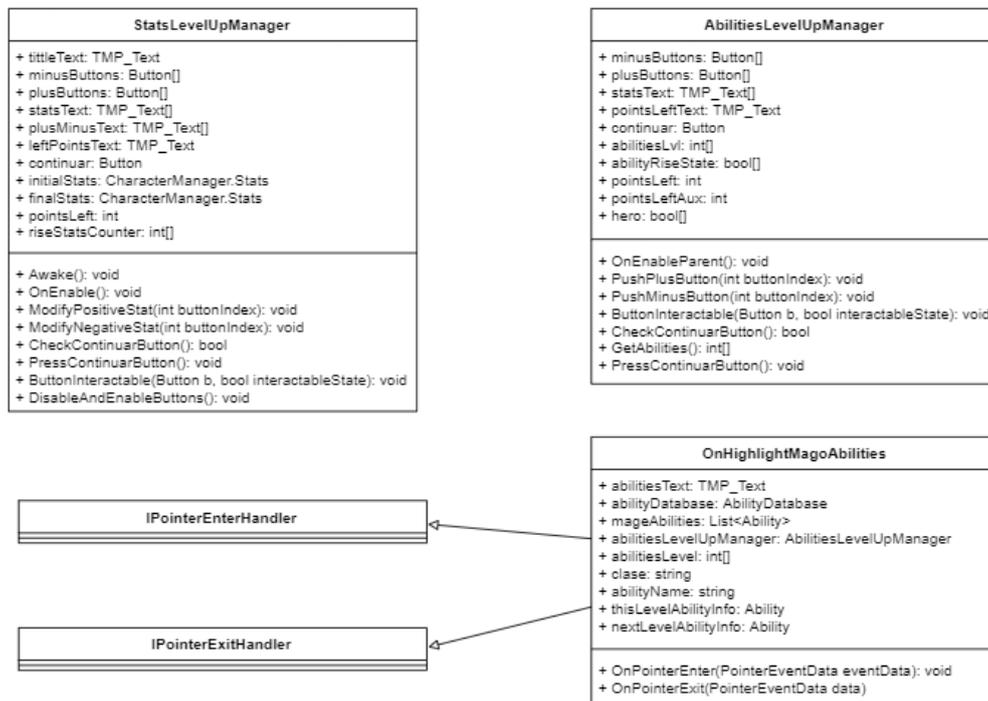


Diagrama 10 - Subida de habilidades y características

4.9.9 Tienda

La tienda está gestionada por dos clases, una para la compra y otra para la venta. Clases que, además, poseen muchas similitudes.

La clase ShopManager se encarga de gestionar la compra de objetos. Se encarga de filtrar que objetos se deben mostrar en cada página, retroceder una página, avanzar una página, comprar el objeto o mostrar la información de los objetos cuando sus botones son clicados.

La clase VentaManager se encarga de mostrar todos los ítems del inventario y enseña sus descripciones cuando son clicados. También gestiona con las clases Inventory y Gold la venta del objeto clicado.

No se ha incluido la clase ShopStats que gestiona mostrar las características de vida y electricidad del personaje durante el transcurso de las interfaces relacionadas con la tienda.

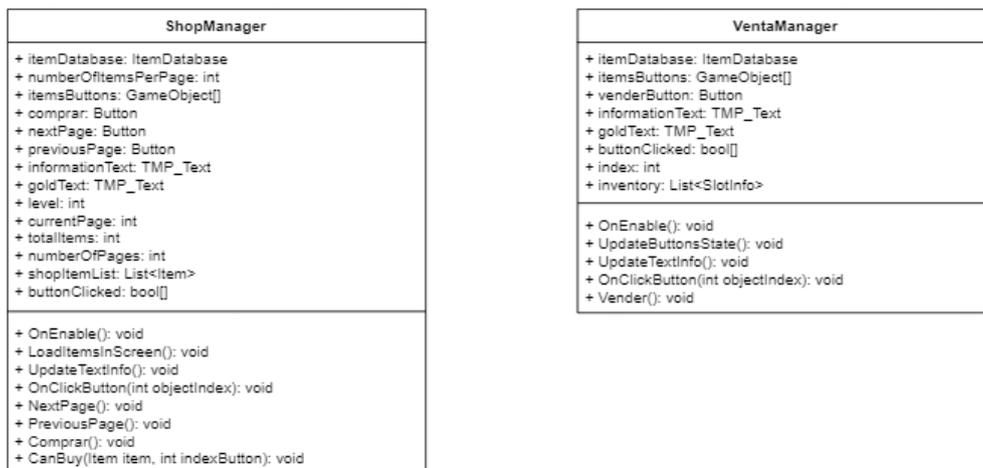


Diagrama 11 – Tienda

4.9.10 Menú *in-game*

En este apartado se expone la gestión del menú que está dentro del juego. Es decir, al que se accede desde el mapa.

MenuManager es la clase que gestiona que botón se pulsa. Es el *script* asociado al objeto del menú en la escena. Aunque cada botón de la escena gestiona el acceso a otra interfaz, esta clase gestiona la continuación del juego, la pausa o salir del juego.

StatsManager es una clase muy simple que muestra al jugador las características actuales del personaje.

La clase Inventario se encarga de mostrar todos los ítems del inventario, sus descripciones cuando se clican y de permitir usar aquellos ítems que sean del tipo *CONSUMABLE*. También gestiona la eliminación del objeto usado en la clase Inventory.

Finalmente, la sección de habilidades. Hay dos clases encargadas de la gestión. Una se encarga de mostrar las habilidades dependiendo del personaje que se haya escogido y de mostrarlas activadas o desactivadas dependiendo de si se ha subido al menos un nivel dicha habilidad. Esta clase es HabilidadesManager. La clase OnHighlightedAbilitiesMenu se encarga de mostrar la información del nivel y efecto de la habilidad si se mueve el cursor por encima de cualquier habilidad.

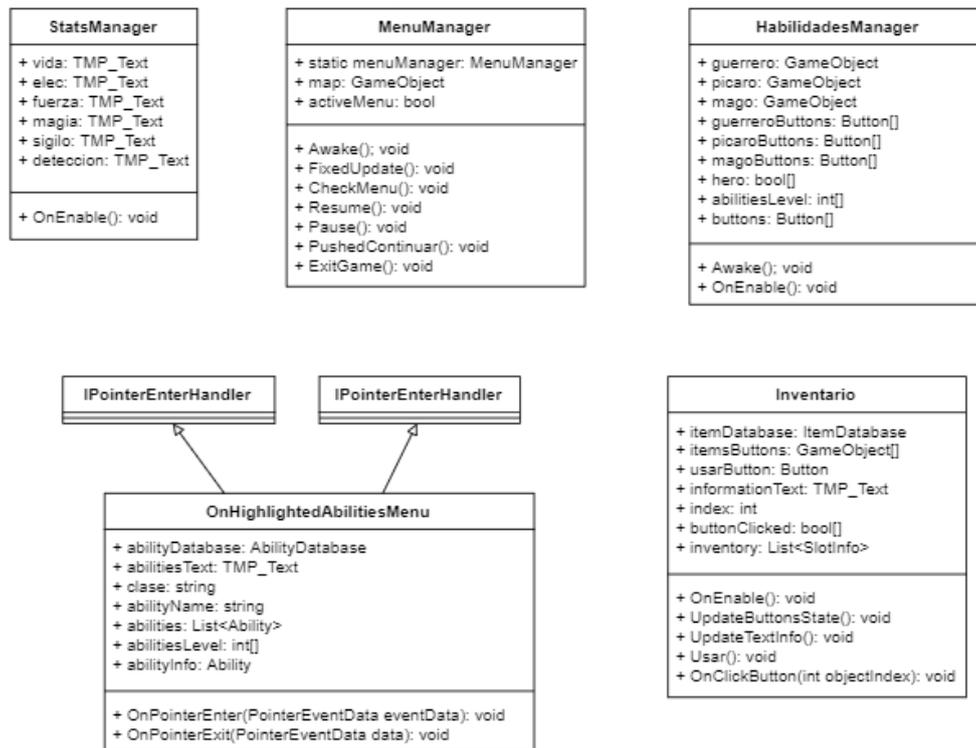


Diagrama 12 - Menú in-game

4.9.11 Victoria y derrota

El último apartado para los diagramas de clase contiene una única clase muy simple para ambas pantallas, la de victoria y la de derrota. Si se pierde la clase gestiona tanto la opción de reintentar, como la opción de salir. En el caso de victoria, la única opción es salir, que también está gestionada por esta clase.

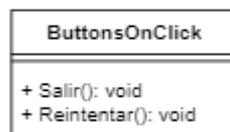


Diagrama 13 - Victoria y derrota

5. Desarrollo de la solución

5.1 Organización del proyecto

En este apartado del capítulo cinco se expone como se organiza en Unity el proyecto. El proyecto está orientado a gestionar los ficheros por áreas de desarrollo como por ejemplo el arte o la programación. Dentro del apartado de programación el proyecto se gestiona por funciones o zonas de actuación, véase: función enemigos, función sala de combate o función menú *in-game*.

5.1.1 Animations

La carpeta *Animations* del proyecto contiene todos los objetos Unity que se han usado para las animaciones. Se subdivide, a su vez, en cinco carpetas:

- EnemiesCombatRoom
- PlayerCombatRoom
- Mago
- Warrior
- Pícaro

Las dos primeras contienen las animaciones del combate. Las otras tres controlan el movimiento de cada personaje.

Así es como se ve en el proyecto las animaciones del personaje mago cuando se va a mover:



Ilustración 3 - Animaciones jugador

La *Ilustración 3* contiene dos objetos diferentes.

Animator Controller es un objeto que se introduce en el *GameObject* de la escena que va a ejecutar la animación. En la *Ilustración 3*, este objeto, corresponde al gráfico menos numeroso en la imagen. Es decir, el primer objeto.

El otro elemento en la carpeta es el objeto *Animation Clip* que es el objeto que contiene la propia animación. Estas animaciones se ejecutan de acuerdo con una máquina de estados que se define mediante la herramienta *Animator* de Unity. Aquí se muestra un ejemplo para la misma animación del mago:

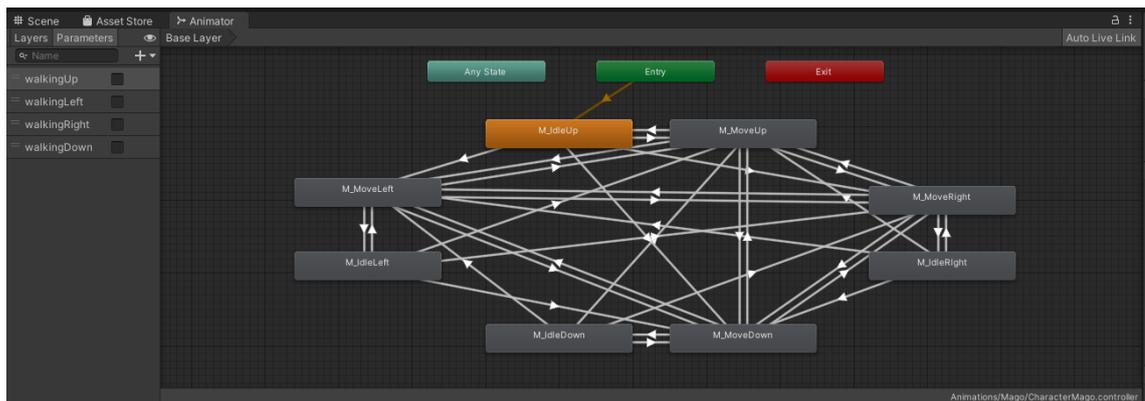


Ilustración 4 - Animator de Unity para el movimiento del mago

Cada recuadro en la *Ilustración 4* es un *Animation Clip*. Cada *Animation Clip* contiene una animación básica. Mediante el código, cambiando los valores del recuadro izquierdo, y la asociación visual de los *Animation Clip* en el *Animator* de Unity, podemos crear transiciones entre las animaciones.

Este es un ejemplo de cómo se usa la herramienta de animar para un *Animation Clip*:

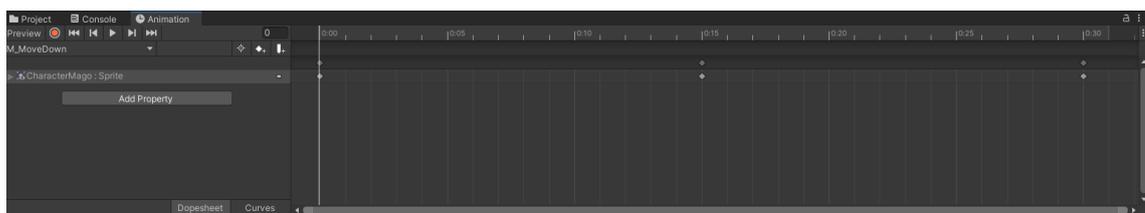


Ilustración 5 - Animation Clip, mago move down

Para animar el objeto se introducen los *Sprites* o imágenes necesarias, que veremos luego, y se comunica cuando se desea cambiar de *Spirite*. Cada punto blanco es un cambio de *Sprite* en la animación, un estado. En este ejemplo de la *Ilustración 5* se desea crear un efecto de movimiento del personaje al jugador. Por tanto, se cambia a los 15 *frames* el *Sprite*, volviendo en el tercer estado al *Sprite* inicial. Si se repite en bucle esta animación se crea el efecto de movimiento con tan sólo dos *Sprites*. Aunque, obviamente, los juegos de alto presupuesto usan muchísimos más *Sprites* y en caso de trabajar con objetos 3D, se usan otras herramientas.

5.1.2 Art

En este caso el arte es externo. Contiene ocho carpetas que contienen los *Sprites* para cada situación. Un *Sprite* es básicamente una imagen estática. En este caso el directorio *Art* se ha dividido en:

- *Backgrounds*: Fondos del juego.
- *Combat_Elements*: Elementos de combate como el corazón o el rayo.
- *Combat Menu*: La interfaz de combate para cada héroe.
- *Enemies*: Los enemigos del jugador.
- *MC_Back_Combat*: El personaje principal visto desde atrás para usar en combate.
- *MC_Movement*: Los *Sprites* del personaje principal necesarios para generar las animaciones de movimiento.
- *MenuHeroes*: *Sprites* usados en la selección de personaje.



- *Seller*: El vendedor, Fifoxi.

5.1.3 Scenes

Este directorio contiene las cuatro escenas del juego. Se explicarán en el apartado 5.2 Escenas en Unity y el proyecto, dedicado exclusivamente a las escenas.

5.1.4 TextMesh Pro

Librería de Unity usada para hacer los textos más estéticos. Contiene los directorios propios de la librería como las fuentes o los recursos.

5.1.5 Scripts

Este es el directorio que contiene todos los *scripts* del juego que se explicaron en el capítulo anterior. Se divide en nueve carpetas:

- *AbilitiesRise*: Dentro contiene la base de datos de las habilidades y la gestión de la subida de nivel de las habilidades.
- *CombatRoom*: Contiene todos los elementos del combate, incluyendo el *script* CharacterManager, la interfaz de los objetos en combate, la gestión de las interfaces de combate, la gestión de los dados, etc. No incluye a los enemigos ni su gestión.
- *Def&VicScreen*: Contiene el *script* que controla la interfaz de victoria y derrota.
- *Enemies*: Contiene a todos los enemigos y su gestión.
- *InGameMenu*: Contiene la gestión del menú activable desde el mapa.
- *IntentoryDatabase*: Contiene la base de datos de los objetos y el inventario del jugador.
- *Map*: Contiene diferentes máanager generales del juego como DiceManager. Contiene el controlador del movimiento y la interacción del personaje con personajes no jugables.
- *Menu & Abilities*: Contiene la gestión del menú principal donde se escoge personaje.
- *Shop*: Contiene los tres *scripts* encargados de gestionar la tienda.

5.2 Escenas en Unity y el proyecto

Una escena en Unity está compuesta de objetos, cada objeto tiene unos componentes. Puede ser un objeto que únicamente tenga coordenadas y no realice ninguna función o puede ser un objeto con muchos componentes como animaciones, *scripts*, *sprites*, físicas del motor o colisiones. La escena es el espacio en el que todos estos objetos ejecutan sus acciones.

Se suele realizar una escena por nivel. Aunque para este proyecto, dado que no existían niveles claros, se ha decidido dividirlo entre zona de menú inicial y zona de juego.

El proyecto está dividido en cuatro escenas. Aunque dos de ellas contienen simplemente la interfaz de victoria o derrota con sus respectivos botones.

Por tanto, el capítulo se va a centrar en las otras dos escenas nombradas, la escena que contiene el menú principal donde se elige al personaje y la escena donde se desarrolla el juego propiamente.

5.2.1 Escena del menú principal

Como se va a comprobar, los objetos de la escena coinciden, en muchas ocasiones, con el diagrama de flujo. Esta es la jerarquía o menú de la escena:

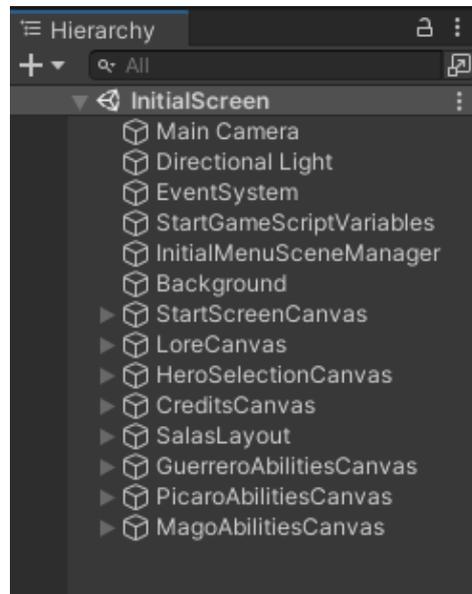


Ilustración 6 - Jerarquía de la primera escena

Cada uno de estos ítems de la jerarquía es un *GameObject* de la escena. Algunos contienen otros *GameObject* dentro de su jerarquía. Por ejemplo, *LoreCanvas* sería un *GameObject* padre de sus *GameObject* hijos que serían, la descripción de la historia, el título y el botón Continuar.

A continuación, se va a examinar uno por uno los elementos de la escena:

- *Main Camera*: Es la cámara de la escena, es el *GameObject* que hace posible la visión del juego en la pantalla. Viene por defecto en Unity al crear una nueva escena y contiene dos componentes.
 - *Transform*, que son las coordenadas del objeto en la escena. Es un componente que deben tener por obligación todos los objetos de la escena.
 - *Camera*, que es el componente que realiza propiamente la gestión de la cámara y contiene atributos modificables como el campo de visión.
- *EventSystem*: Es el objeto que se encarga de capturar los eventos de *input*, como, por ejemplo, clicar un botón con el ratón.
- *StartGameScriptVariables*: Este objeto es el ejemplo de cómo introducir en Unity un *script* sin asociarlo a ningún objeto gráfico en la escena. Este es un objeto que no tiene ninguna función más allá de ser el contenedor del *script* *PlayGame*, que es el *script* que transmite la información de esta escena a la siguiente. En la siguiente escena hay otro objeto igual que se destruye para mantener los datos del *script* que contiene este objeto. Todos los gestores de información en el proyecto tienen este aspecto y se han colocado en la parte superior de la jerarquía de objetos.
- *InitialMenuSceneManager*: Otro objeto-gestor. En este caso gestiona la pantalla inicial del menú principal y contiene el *script* *InitialMenu*.
- *Background*: Contiene el *Sprite* que se renderiza como fondo del menú.

- *StartScreenCanvas*: Todo elemento gráfico, generalmente, debe estar contenido en un *Canvas*. Entonces, hasta el final de la escena, todos los objetos en la cima de la jerarquía como padres, contienen un *Canvas* donde introducir los elementos gráficos propios de cada interfaz. En este caso, este objeto, contiene todos los botones de la escena inicial, así como el título y otras imágenes que componen la interfaz inicial al comenzar el juego.
- *LoreCanvas*: Como se ha comentado anteriormente, contiene la pequeña explicación de la historia o el contexto en el que se produce el juego.
- *HeroSelectionCanvas*: Este objeto contiene los elementos de la pantalla de selección de personaje. Así como los componentes de estos objetos. Por ejemplo, contiene el *script* que ilumina los personajes cuando se mueve el cursor sobre ellos.
- *CreditsCanvas*: Contiene la interfaz de los créditos.
- *SalasLayout*: Contiene las siete interfaces que forman el tutorial. Las cuatro páginas genéricas y la página específica de cada personaje.
- *GuerreroAbilitiesCanvas*: Este objeto contiene los objetos necesarios para gestionar la elección de habilidades del guerrero. Contiene los botones de elección, los textos descriptivos de las habilidades, el título y los botones con los nombres de las habilidades.
- *MagoAbilitiesCanvas*: Este objeto contiene los objetos necesarios para gestionar la elección de habilidades del mago. Contiene los botones de elección, los textos descriptivos de las habilidades, el título y los botones con los nombres de las habilidades.
- *PicaroAbilitiesCanvas*: Este objeto contiene los objetos necesarios para gestionar la elección de habilidades del pícaro. Contiene los botones de elección, los textos descriptivos de las habilidades, el título y los botones con los nombres de las habilidades.

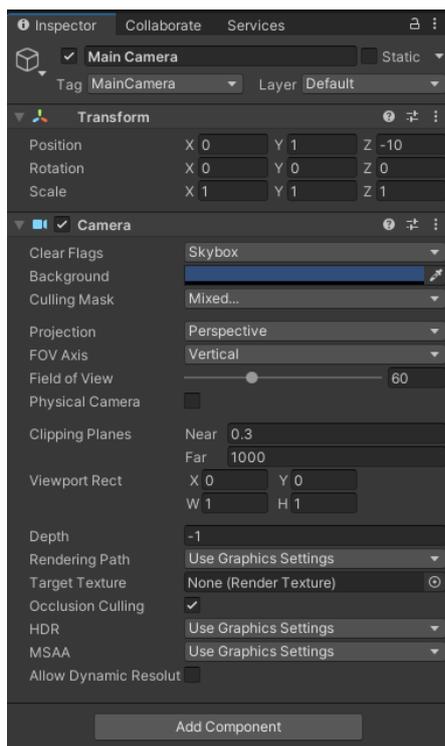


Ilustración 7 - Ejemplo de los componentes de la cámara

5.2.2 Escena del juego

En esta escena es donde ocurre el juego. Aquí está el mapa, los enemigos, el personaje o los vendedores entre otros. La escena se divide, básicamente, en diferentes *GameObject* de gestión, mapa, sala de combate, tienda y menú.

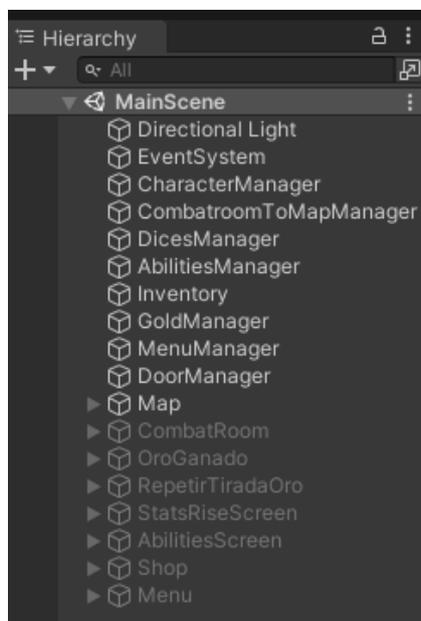


Ilustración 8 - Jerarquía de la segunda escena

- *CharacterManager*: Este *GameObject* u objeto es el gestor del *script* con el mismo nombre y que contiene todos los datos del personaje para el combate. Se ha decidido ubicarlo fuera del objeto de combate para, como en el caso de todos

los gestores, tener un acceso global al *script*. De esta manera se mantiene al *script* ajeno a todo flujo de activación y desactivación de objetos en la escena, siendo accesible en cualquier momento.

- *CombatroomToMapManager*: Este objeto contiene el *script* *StartToEndCombatManager* que maneja el flujo de activación y desactivación de objetos para mostrar las diferentes interfaces.
- *DicesManager*: Contiene el *script* del mismo nombre que controla el acceso por parte de otros *scripts* al lanzamiento de dados.
- *AbilitiesManager*: Objeto-gestor que controla el nivel actual de las habilidades.
- *Inventory*: Objeto que, como su nombre indica, contiene el *script* del mismo nombre que se encarga de gestionar el inventario del jugador durante el juego.
- *GoldManager*: Contiene el *script* *Gold* que controla la adición y resta del oro que se posee, así como la cantidad de oro actual.
- *MenuManager*: Objeto-gestor que contiene el *script* que controla la interfaz del menú *in-game*.
- *DoorManager*: El *script* contenido en el *GameObject* controla que camino se abre para el jugador una vez finalice el combate.
- *Map*: Es el objeto que contiene todos los elementos del mapa. Desde los personajes con sus *scripts* y animaciones hasta las baldosas del suelo. Se divide en varios objetos.
 - *Playable*: Contiene los elementos físicos del mapa como las puertas, los vendedores, los enemigos, etc. Todos ellos, a excepción del suelo y el fondo del juego, contienen un componente que no se ha introducido aún llamado *collider*. Este es el componente que detecta las colisiones con otros objetos de la escena. Puede configurarse para que el *collider* no deje pasar al objeto con el que colisiona, simulando a un objeto físico sólido, siempre que este objeto que entra en colisión contenga un componente llamado *RigidBody* (que se explica un poco más adelante). Otra configuración posible es no incluir la capacidad de simular un objeto físico sólido, pero dejar el *collider* como detector de cuando otro objeto con un *RigidBody* ha entrado al espacio definido por el *collider*. El espacio que define se puede modificar manualmente en la escena. Un ejemplo de este segundo uso es para activar la silueta negra que avisa al jugador de la posibilidad de interactuar con otro personaje del juego.
 - Los diferentes *Character*: Contienen la cámara del mapa, los elementos de vida, electricidad y oro que se muestran en pantalla y el propio personaje, con su *script* de control de movimiento y animaciones, su componente *Animator*, su *collider* y su componente *RigidBody*. *RigidBody* es un componente usado para simular la física en el objeto en el que se aplica. Por ejemplo, simula la gravedad del objeto en el juego o es el responsable de que sea posible que este objeto colisione con otros en la escena.
 - Los diferentes *Text*: Son objetos que se usan para el texto que aparece en las salas de combate y que sirven para informar al jugador de los posibles caminos disponibles antes de enfrentarse a un enemigo.
- *CombatRoom*: Este el objeto contiene todos los elementos necesarios para el combate. Contiene los fondos, la imagen del personaje que debe salir, el gestor

interno de las interfaces, las interfaces específicas de cada personaje, los diferentes dados y enemigos y un objeto Canvas, que contiene un componente del mismo nombre para renderizar todos los componentes del combate y la cámara de combate. A continuación, se nombran los objetos que contiene más destacados.

- *PlayerImage*: Contiene el *script* que decide que imagen se sitúa en el espacio del jugador y contiene el componente Animator para animar el ataque del jugador.
- *LayoutManager*: Este es un objeto-gestor propio de la sala de combate. Controla todas las transiciones entre las diferentes interfaces que no es posible manejar directamente en Unity. Por ejemplo, controla cuando se debe avanzar a la interfaz de alguno de los dados o almacena el valor pulsado por algún botón para luego poder ser usado por CharacterManager y conocer que ataque han decidido usar el jugador.
- *Enemies*: Contiene todos los enemigos con sus respectivas imágenes, componentes Animator y *scripts* específicos.
- Los diferentes objetos *Layout*: Contienen todos los botones de la interfaz de los diferentes personajes, sus *scripts* *LayoutManager* y los botones que se muestran en las interfaces. También, aparte de las interfaces básicas de los personajes, aunque con la misma función, existe la interfaz para los objetos, el veneno y la invisibilidad.
- Los diferentes objetos *Dices*: Contienen la información de la interfaz de cada dado y dentro de ellos contienen los *scripts* que implementan la lógica de las tiradas. En ellos se introduce manualmente, en el componente de la escena, los valores para cada dado.
- Los diferentes objetos *Stats*: Estos objetos contienen la información de la vida y electricidad, tanto del jugador como del enemigo, que se pueden ver en la pantalla.
- OroGanado: Este objeto contiene los dados que se lanzan para ganar oro en los cuatro niveles. Cada uno es un objeto diferente incluido en el mismo objeto Canvas. Contiene también el *script* que gestiona esta información, GoldArrayDices y una cámara.
- RepetirTiradaOro: Contiene la interfaz que aparece tras lanzar la tirada de ganar oro para preguntar al usuario si desea repetir la tirada a cambio de una pequeña parte del potencial premio. No contiene cámara al estar incluido dentro de la cámara de OroGanado.
- *StatsRiseScreen*: Este objeto contiene todos los objetos necesarios para subir las características cuando se vence en un combate. Tanto el objeto que contiene el componente Canvas, como la cámara, los botones y textos de la interfaz.
- *AbilitiesRiseScreen*: El caso de este objeto es el mismo que el anterior. La única diferencia es que el propio objeto contiene un *script* para decidir que interfaz de habilidades debe mostrarse dependiendo del personaje escogido y una interfaz para cada uno de los personajes.
- *Shop*: Contiene los objetos que gestionan la interfaz de la tienda, la interfaz de compra y la interfaz de venta. Contiene un objeto Canvas y una cámara. Dentro del objeto Canvas están contenidos los objetos que gestionan cada una de las tres interfaces.



- **Menu:** Este es el objeto que gestiona el menú interno del juego, el que es accesible desde el mapa. No contiene cámara porque se muestra sobre el mapa del juego que ya contiene su propia cámara. Contiene los elementos gráficos del menú y sus botones que están contenidos dentro del objeto Canvas. Dentro de este mismo Canvas existen también cuatro objetos que controlan las diferentes interfaces accesibles desde el menú.

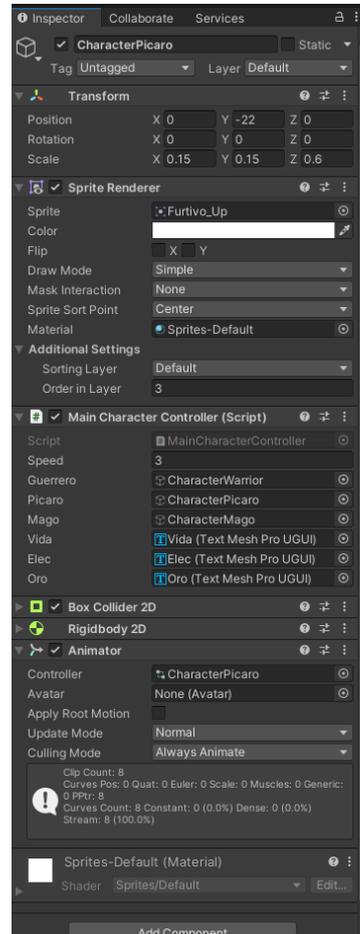
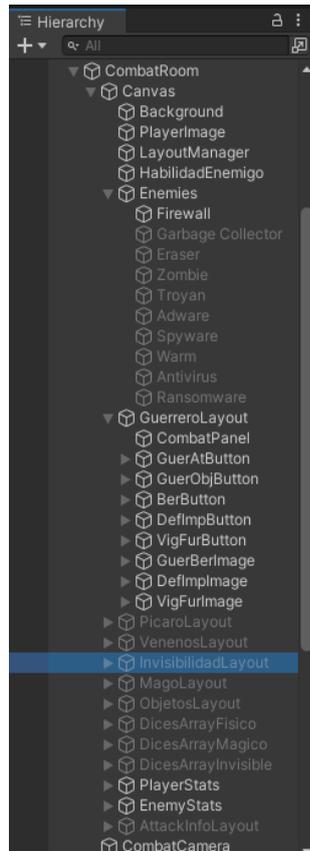


Ilustración 9 - Jerarquía de combate y componentes del personaje

5.3 Implantación

Aunque este apartado, en otras condiciones, podría ser un capítulo aparte. No se ha creído necesario crear un capítulo diferente.

La solución se ha desplegado en una página web gratuita, itch.io, donde se puede jugar al juego libremente si se tiene un navegador y una conexión a internet para acceder al juego.

La plataforma ofrece tres opciones para desplegar un juego. La primera opción permite que el jugador done libremente el dinero que considere al autor. La segunda opción establece un precio fijo para poder jugar al juego. La tercera opción, y la elegida para este proyecto, es la opción de que el juego sea gratuito y no acepte donaciones.

El juego está disponible en el siguiente enlace: <https://biku93.itch.io/electroanima>

Se recomienda usar el navegador Google Chrome.

6. Pruebas

En este capítulo se van a exponer las pruebas realizadas durante el desarrollo. Se han realizado dos tipos de pruebas. El primer tipo son las pruebas de diseño en las que han participado diferentes voluntarios probando el juego y dando su opinión de este. El segundo tipo son las pruebas de *software* que han consistido en diferentes pruebas de caja negra.

Se han producido siete etapas de pruebas. Las pruebas de diseño han consistido en seis fases y las pruebas de *software* en cinco. Esto ha sucedido porque durante el primer ciclo de *software* aún no se tenían suficientes elementos para ser probado desde el punto de vista del diseño. En el caso de las pruebas de *software*, por razones obvias, no se han realizado pruebas durante la etapa de cartón del juego. Aunque en la última prueba de *software* no se han añadido más funcionalidades y se ha limitado a probar que tras cambiar los parámetros del diseño el flujo de juego seguía siendo correcto.

6.1 Pruebas del diseño

Se han contemplado seis fases para estas pruebas. Dos en el estado de cartón y cuatro en el estado de *software*. Las pruebas más importantes para el diseño fueron la primera hecha sobre cartón y las dos últimas realizadas sobre *software*.

En la primera prueba realizada sobre cartón se decidió cambiar el concepto de los ataques tras verse que un escalado basado en multiplicadores y no en sumas perdía completamente la escalabilidad del juego. También se modificaron muchas características de los personajes.

En la segunda prueba realizada sobre cartón se decidió que el juego estaba suficiente depurado para implementarse en *software*. El único gran cambio que se decidió en esta etapa fue eliminar las trampas en los pasillos ya que se observó que los jugadores prestaban mucha atención a los combates, pero los pasillos eran interpretados como zonas de descanso entre los combates.

En la tercera prueba, la primera sobre *software*, fue más una prueba de funcionalidad, ya que solo se tenían los ataques básicos y los enemigos solo tenían una fase. Aunque en esta fase ya se observó que se disfrutaba del combate con los enemigos. El mayor problema que se observó es la confusión con las instrucciones del juego.

En la segunda prueba de *software* el proyecto estaba bastante avanzado, conteniendo ya instrucciones para el utilizar el personaje en el juego. Se introdujeron diferentes elementos facilitadores como la iluminación de los botones al mover el cursor sobre ellos. Aunque la información de las habilidades y el menú dentro del juego, por ejemplo, aún no estaban implementados. Se observó confusión en los jugadores sobre el uso de los personajes en combate, sobre la información de las características del personaje y los enemigos en la interfaz de combate y el efecto de usar cada una de las habilidades en el juego.

En la tercera prueba de *software* se introdujo el tutorial, información adicional para el enemigo y el jugador, se hizo permanente la información sobre la vida, electricidad y oro disponible, se creó el menú interno del juego y la información del efecto de cada habilidad al

mover el cursor sobre los botones. Se notó una agradable experiencia de juego en los participantes como voluntarios tanto por las mejoras en el diseño como por la introducción del arte.

La última prueba fue un pequeño ajuste en la escalabilidad de la dificultad en el juego ya que se detectó que los jugadores perdían muy a menudo.

6.2 Pruebas del software

Las pruebas de *software* consisten en cuatro ciclos de pruebas después del desarrollo de nuevas funcionalidades y un último ciclo en el que no se añaden nuevas funcionalidades. A continuación, se van a añadir las pruebas que se han superado en cada fase del desarrollo. Las primeras pruebas se han superado en cada uno de los ciclos de desarrollo, por lo que en las tablas se muestran los resultados de las pruebas para cada ciclo en los que se han superado. Las últimas pruebas solo se han superado en el último ciclo. Se ha querido volver a superar las primeras pruebas en ciclos posteriores para asegurar que el nuevo desarrollo no había dañado funciones que eran correctas anteriormente. A continuación, se va a exponer el resultado de las pruebas con un OK si la prueba ha funcionado bien y un KO si ha fallado y una explicación en la sección de comentario sobre el fallo.

6.2.1 Ciclo 1

Este es el ciclo de pruebas después de la creación del MVP. Se implementa el flujo del menú principal a excepción de la historia y el tutorial. Se implementa la interfaz de combate con los botones de las habilidades sin uso, aunque con el ataque básico implementado y se implementan los turnos para atacar al enemigo. Se crea el mapa en la escena y se habilita entrar y salir de la sala de combate cuando se den las condiciones.

ID Prueba	Descripción	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Comentarios
PU-01	El juego empieza sin errores.	OK	OK	OK	OK	
PU-02	El menú inicial contiene los cuatro botones: Empezar, Opciones, Créditos, Salir.	OK	OK	OK	OK	
PU-03	Al pulsar sobre el botón créditos se ve la interfaz de créditos y se puede volver al menú inicial de nuevo si se pulsa sobre el botón Volver.	OK	OK	OK	OK	
PU-04	Al pulsar el botón Empezar el juego avanza a la interfaz de elección de personajes. Cuando exista una pequeña historia se avanza a la interfaz de historia tras pulsar este botón.	OK	OK	OK	OK	
PU-05		OK	OK	OK	OK	



	Al pulsar el botón Volver desde la interfaz de selección de personaje se vuelve al menú principal.					
PU-06	El botón Continuar no se activa hasta que un personaje ha sido escogido.	OK	OK	OK	OK	
PU-07	Al mover el cursor por encima de alguno de los héroes se muestra una descripción del héroe y sus características.	OK	OK	OK	OK	
PU-08	Al clicar sobre un personaje queda elegido.	OK	OK	OK	OK	
PU-09	Si se pulsa el botón Continuar se avanza a la interfaz de selección de habilidades. Cuando exista el tutorial se avanza al tutorial tras pulsar este botón.	OK	OK	OK	OK	
PU-10	En la interfaz de elección de habilidades se muestran las habilidades correspondientes al personaje escogido.	OK	OK	OK	OK	
PU-11	Al mover el cursor por encima de cualquiera de las habilidades o sus respectivos botones de añadir habilidad se muestra la descripción y los efectos de la habilidad.	OK	OK	KO	OK	Al cambiar el nombre de la habilidad Recarga, que se llamaba Curación, se perdió el enlace en la escena y dejó de mostrar la información.
PU-12	Al clicar sobre un botón de Añadir habilidad se selecciona la habilidad. Desactiva el resto de las habilidades si con esta selección se tienen el máximo de habilidades para el personaje escogido.	OK	OK	OK	OK	

PU-13	Al clicar sobre un botón Añadir habilidad ya seleccionado se borra la selección la habilidad. Activando el resto de las habilidades si con este borrado de selección ya no se tienen el máximo de habilidades para el personaje escogido.	OK	OK	OK	OK	
PU-14	El juego avanza hasta el mapa permitiendo cualquier combinación de elección de personajes y habilidades.	OK	OK	OK	OK	
PU-15	Todos los botones activos de la interfaz y todos los personajes elegibles se iluminan de naranja al mover el cursor por encima.	OK	OK	OK	OK	
PU-16	El número de habilidades elegidas para cada personaje es el adecuado. Dos habilidades elegibles para el pícaro y el mago. Una habilidad para el guerrero.	OK	OK	OK	OK	
PU-17	Al pulsar el botón Comenzar partida empieza el juego y se muestra el mapa.	OK	OK	OK	OK	
PU-18	El mapa está completo y se puede mover al personaje por él.	OK	OK	OK	OK	
PU-19	Al pulsar el botón E sobre un enemigo la interfaz cambia a la sala de combate.	KO	OK	OK	OK	Algunos enemigos no tenían gráfico asignado y el juego se quedaba parado e inutilizable.
PU-20	Todos los elementos gráficos del combate están visibles.	OK	OK	OK	OK	

PU-21	Los botones son visibles para las habilidades escogidas.	KO	OK	OK	OK	Las dos invisibilidades del pícaro estaban intercambiadas y mostraba una al haberse elegido la otra en el menú.
PU-22	Se puede usar un ataque básico que resta vida efectiva al enemigo.	OK	OK	OK	OK	
PU-23	Cuando el enemigo llega a cero de vida y no es el enemigo final se puede volver al mapa y continuar moviéndose por este.	OK	OK	OK	OK	
PU-24	Si se derrota al último enemigo se termina la partida hasta que se implemente la interfaz de victoria.	OK	OK	OK	OK	

Tabla 112 - Pruebas del ciclo 1

6.2.2 Ciclo 2

Este es el ciclo de pruebas después de la finalización del segundo ciclo de creación de *software*. Se implementa el ataque básico para enemigos y se permite subir las características al finalizar el combate. Se añaden colisiones y puertas en el mapa para que pueda ser completado el flujo principal. Se corrigen los errores de la fase anterior y se crea la pantalla de *Game Over*.

ID Prueba	Descripción	Ciclo 2	Ciclo 3	Ciclo 4	Comentarios
PU-25	Cuando el pícaro, sólo hasta que se añada la invisibilidad, o el guerrero, pulsan el botón de atacar, se muestra el dado de fuerza y se desactivan el resto de los botones que no sean el pulsado previamente o el botón de Lanzar dado.	OK	OK	OK	
PU-26	Cuando el mago pulsa el botón de ataque del mago se muestra el dado de magia y se desactivan el resto de los botones que no sean el pulsado previamente o el de Lanzar dado.	OK	OK	OK	

PU-27	Cuando se mueve el cursor sobre el botón Lanzar dado se ilumina de naranja.	OK	OK	OK	
PU-28	Si se vuelve a pulsar el botón elegido para activar los dados, desaparece la interfaz de dado, se activan el resto de los botones y se puede elegir otra habilidad.	OK	OK	OK	
PU-29	Cuando se clica sobre el botón Lanzar dado se muestra de naranja el resultado del dado y se aplica correctamente su bono.	OK	OK	OK	
PU-30	Cuando se ha efectuado el ataque, desaparece la interfaz del dado, se impide realizar cualquier otra acción al jugador y el enemigo realiza su ataque restando vida al jugador de acuerdo con su daño y un dado interno que lanza el enemigo.	OK	OK	KO	Al añadir animaciones el tiempo de espera no es suficiente y la interfaz se activa antes de tiempo.
PU-31	Cuando el ataque del enemigo finaliza la interfaz de combate vuelve a estar habilitada.	OK	OK	OK	
PU-32	Si el jugador se queda con cero de vida, acaba el combate y se avanza a la interfaz de <i>Game Over</i> .	KO	OK	OK	El jugador seguía perdiendo vida sin cambiar la interfaz.
PU-33	En la interfaz de <i>Game Over</i> si se mueve el cursor por encima de los botones se iluminan de naranja.	OK	OK	OK	
PU-34	En la interfaz de <i>Game Over</i> si se pulsa el botón Reintentar se borran los datos de la partida anterior y se vuelve a la pantalla de menú inicial.	OK	OK	OK	

PU-35	En la interfaz de <i>Game Over</i> si se pulsa el botón Salir finaliza el juego.	OK	OK	OK	
PU-36	Las paredes del mapa tienen colisiones que impiden al jugador salirse del mapa.	KO	OK	OK	Impedían salir al jugador del mapa, pero las esquinas causaban en el jugador una rotación en el eje Z que dejaba inutilizable el juego.
PU-37	Al finalizar el combate, si no es el combate final, mientras la interfaz de ganar oro no está implementada, aparece la interfaz de subir características con todos los elementos gráficos necesarios en esta y con los puntos de subida de característica determinados, los botones - de cada característica deshabilitados, los botones + habilitados y el botón Continuar deshabilitado.	OK	OK	OK	
PU-38	Cuando se pulsa el botón + en la interfaz aumenta un punto el nivel de esa característica y si se pulsa el botón - lo resta. Esta acción debe ser solo posible si los botones están habilitados.	KO	OK	OK	Si se subía una característica y luego se subía otra. Al bajar la primera característica no deshabilitaba el botón - y dejaba seguir bajando puntos por debajo de cero.
PU-39	Si se sube un punto la característica debe habilitarse el botón -.	OK	OK	OK	
PU-40	Si se suben dos puntos una característica debe deshabilitarse el botón +.	OK	OK	OK	
PU-41	Si se resta un punto, teniendo dos, de una característica, debe habilitarse de nuevo el botón +.	OK	OK	OK	

PU-42	Si se resta un punto de una característica, teniendo uno, debe deshabilitarse el botón -.	KO	OK	OK	Si se subía una característica y luego se subía otra. Al volver a bajar no se deshabilitaba el botón - y dejaba seguir bajando puntos por debajo de cero.
PU-43	Si se han usado todos los puntos de característica debe habilitarse el botón Continuar.	KO	OK	OK	Realizando la mecánica nombrada antes, si se compensaba mediante otras características, el botón no se activaba.
PU-44	Si no se han usado todos los puntos de característica el botón Continuar debe permanecer deshabilitado.	OK	OK	OK	
PU-45	Si se pulsa el botón Continuar cuando está activado se vuelve al mapa en caso de nivel impar, en caso de nivel par, si está implementada la interfaz de subida de habilidades, se avanza a esta interfaz, sino al mapa.	OK	OK	OK	
PU-46	Cuando se vuelve al mapa el enemigo ha desaparecido y se ha abierto una de las puertas acorde a como está especificado en las reglas.	KO	OK	OK	El enemigo no desaparecía del mapa.
PU-47	Cuando en la interfaz de combate se pulsa el botón Objetos, avanza a la interfaz Objetos en combate. Al pulsar el botón Volver, se vuelve a la interfaz de combate. Muestra y permite usar los objetos del tipo <i>CONSUMABLE</i> cediendo el turno al enemigo al hacerlo.	OK	OK	KO	Al añadir el arte se pierde la interfaz del pícaro para los objetos.

Tabla 113 - Pruebas del ciclo 2

6.2.3 Ciclo 3

Este es el ciclo de pruebas después de la finalización del tercer ciclo de creación de *software*. Se habilitan las habilidades de los enemigos con sus diferentes fases y se activan las habilidades del jugador haciendo efectivas las subidas de nivel en las habilidades. Se crea la interfaz para ganar oro y repetir la tirada y se crea la tienda, permitiendo comprar y vender objetos. Se implementa posibilidad de usar objetos en combate. Se permite al pícaro volverse invisible con

los riesgos y efectos de hacerlo. Se corrigen los errores de la fase anterior y se crea la pantalla de victoria.

Se ha cometido un error durante este ciclo a la hora de medir la cantidad de desarrollo que se puede realizar en el mismo. En un intento de acelerar el proceso se ha añadido más carga de trabajo de la recomendada, lo que ha provocado una complicación importante para la depuración del ciclo. Se va a corregir este error en los siguientes ciclos.

ID Prueba	Descripción	Ciclo 3	Ciclo 4	Comentarios
PU-48	Cuando se avanza a la pantalla de subida de habilidades, se muestran todas las habilidades y puntos disponibles correspondientes al personaje elegido. Deben aparecer todos los elementos gráficos, los botones - de cada característica deshabilitados, los botones + habilitados y el botón Continuar deshabilitado.	OK	OK	
PU-49	Al mover el cursor sobre cualquiera de los botones - o + activados se muestra la información del efecto que causa subir la habilidad o devolverla al estado anterior en un panel de información que se encuentra en la parte inferior de la interfaz.	OK	OK	
PU-50	Si se sube un punto la habilidad debe habilitarse el botón - y debe deshabilitarse el botón +.	OK	OK	
PU-51	Si se resta un punto, de habilidad, debe habilitarse de nuevo el botón + y debe deshabilitarse el botón -.	OK	OK	
PU-52	Si se han usado todos los puntos de característica debe habilitarse el botón Continuar.	OK	OK	
PU-53	Si no se han usado todos los puntos de característica el botón Continuar debe permanecer deshabilitado.	OK	OK	

PU-54	Una vez pulsado el botón Continuar, se debe volver al mapa.	OK	OK	
PU-55	Cuando un enemigo es derrotado se muestra una interfaz en la que se permite hacer una tirada y conseguir oro.	OK	OK	
PU-56	Cuando el cursor se mueve por el botón Lanzar dado este se ilumina levemente de naranja.	OK	OK	
PU-57	Al pulsar el botón Lanzar dado se ilumina el dado resultante y se muestra una interfaz con la posibilidad de repetir la tirada con un cierto coste especificado en la interfaz.	OK	OK	
PU-58	Si se pulsa el botón Si se informa de la cantidad acumulada que se va a perder en la interfaz y se debe volver a la interfaz de tirar el dado para ganar oro.	KO	KO	No se muestra la información en pantalla de cuanto potencial oro se ha gastado.
PU-59	Si se pulsa el botón No se avanza a la interfaz de subida de características y se añade el oro obtenido en el dado, restándole el valor acumulado de pérdida de premio potencial.	OK	OK	
PU-60	En el mapa, en cada pasillo entre combates, debe aparecer un vendedor.	OK	OK	
PU-61	Si el jugador se aproxima a un enemigo o a un vendedor se muestra una sombra oscura que indica que ese personaje es interactivo.	KO	OK	No se implementó la sombra sobre los enemigos.

PU-62	Si se pulsa el botón E sobre un enemigo o vendedor cuando es interactivo se avanzan a sus respectivas salas, la de combate si es un enemigo y la tienda si es un vendedor.	OK	OK	
PU-63	En la interfaz de la tienda se muestran tres botones, uno para comprar objetos, otro para venderlos y otro para salir de la interfaz y volver al mapa.	OK	OK	
PU-64	Al mover el cursor sobre cualquiera de los botones se iluminan levemente de naranja.	OK	OK	
PU-65	Al presionar el botón Volver se vuelve al mapa.	OK	OK	
PU-66	Al presionar al botón Vender se accede a la interfaz de venta de objetos.	OK	OK	
PU-67	Al presionar el botón Comprar se accede a la interfaz de compra de objetos.	OK	OK	
PU-68	En la interfaz de venta de objetos aparecen los objetos que tiene el jugador disponible para vender y su precio de venta, el botón Vender desactivado y el botón Volver activado.	OK	OK	
PU-69	Si se pulsa el botón Volver se vuelve a la interfaz genérica de la tienda.	OK	OK	
PU-70	Si se selecciona un objeto de la interfaz se muestra una descripción de este en la parte inferior y se activa el botón Vender.	OK	OK	

PU-71	Si se pulsa el botón Vender el objeto desaparece, se obtiene el oro del precio de venta del objeto y se desactiva el botón Vender.	OK	OK	
PU-72	Si se pulsa un objeto ya seleccionado el botón Vender se desactiva y se deja de mostrar la descripción del objeto.	OK	OK	
PU-73	Al mover el cursor sobre cualquiera de los botones se iluminan levemente de naranja.	OK	OK	
PU-74	En la interfaz de compra de objetos, al entrar, deben aparecer los objetos disponibles para compra por nivel y aparecer desactivados los objetos que no se pueden comprar por razones de oro o de máximo de objetos, todos con su precio de compra. Debe aparecer un botón Comprar desactivado, un botón Siguiente activado si existen más de nueve objetos en venta, un botón Anterior desactivado y un botón Volver que vuelva a la interfaz genérica de la tienda.	OK	OK	
PU-75	Si se pulsa el botón Volver se vuelve a la interfaz genérica de la tienda.	OK	OK	
PU-76	Si se selecciona un objeto de la interfaz se muestra una descripción de este en la parte inferior y se activa el botón Comprar.	OK	OK	
PU-77	Si se pulsa el botón Comprar se pierde el oro del precio de compra del objeto y se añade al inventario del jugador.	OK	OK	

PU-78	Si se pulsa un objeto ya seleccionado el botón Comprar se desactiva y se deja de mostrar la descripción del objeto.	OK	OK	
PU-79	Al mover el cursor sobre cualquiera de los botones se iluminan levemente de naranja.	OK	OK	
PU-80	Si existen objetos en páginas posteriores el botón Siguiente debe estar activo y al ser pulsado avanzar a la página siguiente de objetos.	OK	OK	
PU-81	Si existen objetos en páginas anteriores el botón Anterior debe estar activo y al ser pulsado volver a la página anterior de objetos.	OK	OK	
PU-82	En combate, cuando el enemigo ataque, se muestra la información del ataque que está realizando.	OK	OK	
PU-83	Los enemigos usan habilidades según lo marcado en las reglas.	OK	OK	
PU-84	El pícaro dispone de una interfaz para usar la invisibilidad y otra para usar los venenos.	OK	OK	
PU-85	Al elegir un ataque el pícaro avanza a la interfaz de invisibilidad.	OK	OK	
PU-86	Al elegir el tipo de invisibilidad debe avanzar a la interfaz de venenos.	OK	OK	
PU-87	Al elegir un veneno se debe mostrar el dado de invisibilidad si se ha escogido usarla o el de fuerza si no se ha hecho.	OK	OK	

PU-88	Si se vuelve a pulsar sobre el botón de veneno seleccionado desde la interfaz de datos de invisibilidad se vuelve a la interfaz de venenos. También debe ocurrir esto en la interfaz de datos de fuerza si no se ha elegido usar invisibilidad. No debe ocurrir en la interfaz de datos de fuerza si se ha usado la invisibilidad.	KO	OK	Al pulsar el botón de habilidad desde la interfaz de los datos de fuerza habiéndose usado la invisibilidad el juego quedaba en un estado inconsistente.
PU-89	Al usar la invisibilidad se crea un duelo con el enemigo y si se gana se informa al jugador y se realiza el ataque, sino no se realiza ningún ataque. En ambos casos, empieza el turno del enemigo.	KO	OK	El duelo se realizaba, pero los parámetros elegidos para el enemigo estaban equivocados.
PU-90	Los venenos aplican correctamente su efecto durante el tiempo o para realizar el salto de turno.	KO	KO	Primero fallaba al calcular los parámetros para realizar el duelo de sigilo. Después fallaba en la acción de saltar el turno.
PU-91	Las habilidades de todos los personajes son funcionales y pueden subirse de nivel.	OK	OK	
PU-92	Las habilidades muestran su efecto cuando se mueve el cursor sobre estas.	OK	OK	
PU-93	Las habilidades realizan el efecto prometido en la descripción.	OK	OK	
PU-94	Las habilidades para las que no se disponga de electricidad suficiente aparecen desactivadas.	OK	OK	
PU-95	Las habilidades que no se posean no aparecen en la interfaz.	OK	OK	

Tabla 114 . Pruebas del ciclo 3

6.2.4 Ciclo 4



Este es el ciclo de pruebas después de la finalización del cuarto ciclo de creación de *software*. Se añaden las animaciones y el arte al juego. Se añade el tutorial y una pequeña introducción a modo de historia. Se corrigen los errores de la fase anterior y se empieza a crear el menú *in-game*.

ID Prueba	Descripción	Ciclo 4	Comentarios
PU-96	El personaje está animado al moverse en cada una de las direcciones.	KO	Una de las transiciones, al mover el personaje de abajo hacia el lado derecho mientras ya se tiene pulsado otro botón, paralizaba todas las animaciones de movimiento.
PU-97	El personaje está animado al atacar.	OK	
PU-98	El enemigo está animado al atacar.	OK	
PU-99	La vida, electricidad y oro del jugador están visibles en el mapa, en la tienda y en el menú del juego.	KO	No están visibles en la tienda.
PU-100	Al pulsar Empezar en el menú principal se muestra una pequeña historia.	OK	
PU-101	Al pulsar el botón Continuar desde la interfaz de historia avanza a la interfaz de selección de personajes.	OK	
PU-102	Al elegir personaje muestra la primera página del tutorial, si se pulsa el botón Continuar, se deben avanzar páginas del tutorial que son iguales para todos los personajes, hasta la página cinco del tutorial.	OK	
PU-103	La quinta página del tutorial depende del personaje escogido. Al pulsar el botón Continuar se muestra la interfaz de subida de habilidades del personaje escogido.	OK	
PU-104	Los botones del tutorial e historia se iluminan levemente de naranja al mover el cursor por encima de ellos.	OK	

PU-105	Al pulsar el botón Q desde el mapa sale un menú.	OK	
PU-106	Mientras se está en el menú las otras acciones sobre el mapa quedan deshabilitadas.	OK	
PU-107	El menú cuenta con los botones Continuar, <i>Stats</i> , Inventario, Habilidades, Instrucciones, Habilidades, Opciones y Abandonar.	OK	
PU-108	Al mover el cursor por encima de cualquiera de los botones de las distintas interfaces del menú se iluminan de naranja.	OK	
PU-109	Al pulsar el botón Continuar se regresa al mapa.	OK	
PU-110	Al pulsar el botón Abandonar se sale del juego.	OK	
PU-111	Al pulsar el botón <i>Stats</i> se avanza a la interfaz donde se muestran las características del jugador. Se puede volver al menú desde esta interfaz presionando el botón Volver con el clic izquierdo del ratón.	OK	
PU-112	Al pulsar el botón Instrucciones de muestran los controles sobre el mapa. Se puede volver al menú desde esta interfaz presionando el botón Volver con el clic izquierdo del ratón.	OK	
PU-113	Al pulsar el botón Habilidades se muestran las habilidades del jugador. Si se mueve el cursor por encima de cualquier habilidad se muestra su información. También se muestran las habilidades aún no elegidas, pero sin estar activadas. Se puede volver al menú desde esta interfaz presionando	OK	

	el botón Volver con el clic izquierdo del ratón.		
PU-114	Al pulsar el botón Inventario se abre el inventario con el botón Usar inactivo.	OK	
PU-115	En el inventario se muestran todos los objetos que el jugador posee. Los objetos del tipo <i>CONSUMABLE</i> se muestran activados y los objetos del resto de tipos desactivados.	OK	
PU-116	Al clicar sobre algún objeto activado se muestra su información y se activa el botón Usar.	OK	
PU-117	Si se clica el botón Usar, se consume el objeto seleccionado, desaparece del inventario y se deshabilita el botón Usar. Si se clica de nuevo, sin usar el botón Usar sobre el mismo objeto, se desactiva el botón Usar.	OK	
PU-118	Se puede volver al menú desde esta interfaz de inventario presionando el botón Volver con el clic izquierdo del ratón.	OK	
PU-119	Todos los botones se iluminan intensamente de naranja al ser clicados.	OK	
PU-120	Las interfaces que usan objetos como la interfaz de objetos de combate, el inventario o las interfaces de compra y venta mantienen el color naranja intenso al ser clicados si se trata de un objeto.	OK	

Tabla 115 – Pruebas del ciclo 4

6.2.5 Últimas correcciones

En este apartado se muestran las tablas de las pruebas de las últimas correcciones. Como no se añaden nuevas características ni pruebas no se considera un ciclo como los otros. Estas son pruebas para asegurar que las correcciones de los fallos en el ciclo 4 funcionan correctamente.

ID Prueba	Descripción	Ciclo 4	Comentarios
PU-30	Cuando se ha efectuado el ataque, desaparece la interfaz del dado, se impide realizar cualquier otra acción al jugador y el enemigo realiza su ataque restando vida al jugador de acuerdo con su daño y un dado interno que lanza el enemigo.	OK	
PU-47	Cuando en la interfaz de combate se pulsa el botón Objetos, avanza a la interfaz Objetos en combate. Al pulsar el botón Volver, se vuelve a la interfaz de combate. Muestra y permite usar los objetos del tipo <i>CONSUMABLE</i> cediendo el turno al enemigo al hacerlo.	OK	
PU-58	Si se pulsa el botón Si se informa de la cantidad acumulada que se va a perder en la interfaz y se debe volver a la interfaz de tirar el dado para ganar oro.	OK	
PU-90	Los venenos aplican correctamente su efecto durante el tiempo o para realizar el salto de turno.	OK	
PU-96	El personaje está animado al moverse en cada una de las direcciones.	OK	
PU-99	La vida, electricidad y oro del jugador están visibles en el mapa, en la tienda y en el menú del juego.	OK	

Tabla 116 – Correcciones



7. Conclusiones

En este apartado se va a reflexionar sobre el cumplimiento de los objetivos planteados inicialmente.

El primero de los objetivos planteados era conseguir familiarizar a personas de entre 16 y 18 años que estuvieran pensando en iniciarse en el estudio de la programación con conceptos básicos. Aunque se considera que se podría haber ampliado los aspectos enseñados y hubieron de recortarse algunos aspectos para mantener el juego divertido, se considera que se enseñan correctamente aspectos muy básicos.

El segundo de los objetivos era poner en práctica el aprendizaje obtenido durante los estudios del grado. Se considera este objetivo cumplido al poderse aplicar muchos de los conocimientos adquiridos tanto durante el programa de *Game Technology* realizado en Amsterdam como en las asignaturas estudiadas en la Universidad Politécnica de Valencia, especialmente en aquellas orientadas a la interacción con el usuario, a la gestión de proyectos de *software* y al aprendizaje de conceptos básicos de programación. La propia idea del juego se basa en los primeros pasos que se realizan en la asignatura de Introducción a la Informática y la Programación.

El tercer objetivo planteado consistía en familiarizarme e introducirme al mundo del desarrollo de videojuegos aprendiendo las últimas herramientas usadas en la industria. Este objetivo se ha cumplido. Unity es un motor de videojuegos muy usado en la industria actualmente y del cual, durante la realización de este proyecto, se ha amplificado el conocimiento que se tenía de la herramienta. Por ejemplo, el uso de las bases de datos en Unity, la asociación de los objetos en la escena con el código en los *scripts*, las físicas de Unity o el aprendizaje del propio lenguaje de programación que usa el motor, C#.

El último de los objetivos era mantener el juego divertido. Se considera cumplido este objetivo para los jugadores que les guste el género propuesto.

A pesar del volumen de trabajo, el proyecto ha sido llevado a cabo dentro de los plazos previstos con un considerable esfuerzo, pero gratificante al mismo tiempo. Plateado desde la pequeña idea concebida originalmente hasta el concepto que es ahora.

Por otro lado, no se puede realizar semejante trabajo sin un equipo de trabajo detrás, como Andrea, la artista del juego que apostó por incluir su arte y que ha hecho que este juego tenga muy buen aspecto desde el punto de vista estético. Además del soporte de la comunidad de Unity que siempre ha tenido la duda resuelta antes de ser planteada.

8. Trabajo futuro

El principal problema y trabajo futuro que necesita el juego es ser probado por varias personas de edades entre 16 y 18 años. No se ha podido hacer demasiadas pruebas sobre este grupo de edad, siendo el público objetivo del juego.

La principal parte del videojuego que no se ha podido realizar son los efectos de sonido y la música. Aunque es una parte esencial de un videojuego, no se ha conseguido realizar a tiempo.

Otro aspecto para trabajar en un futuro son las animaciones, se considera que los enemigos y los vendedores, por ejemplo, podrían tener algunas animaciones en el mapa para que parezcan más interactivos. También se hubiera querido introducir más animaciones en la sala de combate.

Tampoco se pudo introducir el menú de opciones para los jugadores por lo que se ha quedado un botón sin ninguna función. La idea era usar el botón para cambiar la resolución gráfica y el volumen del juego.

Diferentes personas que no hablan español han mostrado interés en el juego por lo que se valora la traducción al idioma inglés.

Finalmente, si se amplía la duración del juego y se mejoran las estrategias de combate podría llegar a ser un juego comercial.

9. Referencias

- [1] (Zabala-Vargas et al., 2020)
- [2] (*Vista de Gamificación y Aprendizaje Basado En Juegos.*, n.d.)
- [3] (*Ancestors: Historias de Atapuerca, Así Es El Juego de Divulgación Sobre Arqueología Disponible En IOS y Android*, n.d.)
- [4] (*Immune Attack / The Science Game Center*, n.d.)
- [5] (*Kokoro Kids - Crecer, Jugar, Aprender En Un Entorno Seguro.*, n.d.)
- [6] (*Code MasterTM*, n.d.)
- [7] (*Arqueras de Nand, Lo Hemos Jugado: Aprender Lenguaje SQL Con Arqueras y Orcos Es Posible*, n.d.)
- [8] (*CodeCombat - Coding Games to Learn Python and JavaScript*, n.d.)
- [9] (*JSRobot*, n.d.)
- [10] (*Rehabilitación Mediante Videojuegos - Wikipedia, La Enciclopedia Libre*, n.d.)
- [11] (AEVI. Asociación Española de Videojuegos, n.d.)
- [12] (*Libro Blanco Del Desarrollo Español de Videojuegos*, 2020)
- [13] (*Top Countries & Markets by Game Revenues / Biggest Games Markets*, n.d.)
- [14] (*GameMaker Studio 2 Manual*, n.d.)
- [15] (*Programming with C++ / Unreal Engine Documentation*, n.d.)
- [16] (*Unity - Manual: Unity User Manual 2020.3 (LTS)*, n.d.)
- [17] (*Unreal vs. Unity 3D: Elegir El Mejor Motor Para Tu Juego / Starloop*, n.d.)
- [18] (Schell J., 2008)
- [19] (*Singletons En Unity (o Cómo Conservar Objetos Entre Escenas) - YouTube*, n.d.)
- [20] (*Unity - Manual: Execution Order of Event Functions (Orden de Ejecución de Funciones de Evento)*, n.d.)

