



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

El camino del alquimista, un juego roguelike 2D
desarrollado en Unity

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Cerverón Carot, Andrea

Tutor/a: Abad Cerdá, Francisco José

CURSO ACADÉMICO: 2021/2022

Agradecimientos

Gracias a mis padres por su apoyo.

Gracias a mi hermana Sandra por su colaboración e inspiración.

Gracias a mi tutor Paco Abad por su ayuda y apoyo durante todos estos meses de trabajo, además de guiarme en este proyecto.

Gracias a mi compañero Tomás con el que he podido desarrollar este videojuego, sin el que no habría sido posible.

Resum

L'objectiu d'aquest projecte és desenvolupar per complet un videojoc 2D, des de la idea inicial fins a aconseguir un producte funcional, passant per totes les fases de desenvolupament. El joc en qüestió 'El camí de l'alquimista' profundeix en la mecànica de l'exploració de masmorres, que es generaran proceduralment, per a aconseguir que no hi haja dues patides iguals i dar varietat als entorns. L'atre pilar d'aquest joc és el seu sistema de creació d'objectes mitjançant receptes, per les quals el jugador podrà crear nous elements a partir de les matèries primeres que va trobant en la seua aventura. L'art d'aquest projecte és original, i ha sido desenvolupat per un estudiant de Belles Arts per al seu Treball Final de Grau. S'ha utilitzat metodologies àgils per al desenvolupament d'aquest TFG, i el motor de videojocs utilitzat ha sigut Unity.

Paraules clau: videojoc, unity, roguelike, generació procedural, diseny de jocs, desenvolupament de software, joc d'artesanía, RPG, C#

Resumen

El objetivo de este proyecto es desarrollar por completo un videojuego 2D, desde la idea inicial hasta conseguir un producto funcional, pasando por todas las fases de desarrollo. El juego en cuestión 'El Camino del Alquimista' profundiza en la mecánica de la exploración de mazmorras, que se generarán procedualmente, para conseguir que no haya dos partidas iguales y para dar variedad a los entornos. El otro pilar de este juego es su sistema de creación de objetos mediante recetas, por las que el jugador podrá crear nuevos elementos a partir de las materias primas que va encontrando en su aventura. El arte del proyecto es original, y ha sido desarrollado por un estudiante de Bellas Artes para su Trabajo Final de Grado. Se ha utilizado metodologías ágiles para el desarrollo de este TFG, y el motor de videojuegos utilizado ha sido Unity.

Palabras clave: videojuego, unity, roguelike, generación procedural, diseño de juegos, desarrollo de software, juego de artesanía, RPG, C#

Abstract

The objective of this project is to develop a complete 2D videogame, from the first scratch until getting a functional product, going through all the development phases. The game 'The path of the alchemist' take a closer look at the exploration of dungeons that are procedurally generated in order to get more variety in the environment as well as not to have two equal games. The other backbone of this game is its crafting system using recipes, in which the player will make new elements using the prime materials found on his/her adventure. This project's art is original, it has been done by a student from BeauX-Arts for his thesis. For this game's development, agile methodologies have been used as well as the game engine Unity.

Key words: videogame, unity, roguelike, procedural generation, game design, software development, craft game, RPG, C#

Índice general

| | |
|---|-------------|
| Índice general | V |
| Índice de figuras | VII |
| Índice de tablas | VIII |
| <hr/> | |
| 1 Introducción | 1 |
| 1.1 Motivación | 2 |
| 1.2 Objetivos | 4 |
| 1.3 Metodología | 4 |
| 1.4 Estructura de la memoria | 5 |
| 1.5 Colaboraciones | 5 |
| 2 Estado del arte | 7 |
| 2.1 Juegos de mazmorras | 7 |
| 2.1.1 Interpretación de Berlín | 7 |
| 2.1.2 Rogue | 8 |
| 2.1.3 Mystery Dungeon | 9 |
| 2.1.4 Hades | 10 |
| 2.2 Juegos con artesanía | 11 |
| 2.2.1 Minecraft | 12 |
| 2.2.2 Animal Crossing: New Horizons | 12 |
| 2.3 Comparación con la propuesta | 13 |
| 3 Análisis del problema | 15 |
| 3.1 Especificación de requisitos | 15 |
| 3.1.1 Requisitos funcionales | 15 |
| 3.1.2 Requisitos no funcionales | 20 |
| 3.2 Modelado conceptual | 20 |
| 3.3 Análisis del marco legal | 21 |
| 3.4 Solución propuesta | 21 |
| 3.5 Plan de trabajo | 22 |
| 4 Diseño | 25 |
| 4.1 Arquitectura del Sistema | 25 |
| 4.2 Diseño detallado | 26 |
| 4.2.1 Diseño del juego | 26 |
| 4.2.2 Diseño del personaje | 27 |
| 4.2.3 Diseño de la mazmorra | 28 |
| 4.2.4 Diseño de la IA de los enemigos | 30 |
| 4.2.5 Diseño de clases | 31 |
| 4.3 Tecnologías utilizadas | 31 |
| 4.3.1 Unity | 31 |
| 4.3.2 GitHub | 37 |
| 5 Desarrollo de la solución | 39 |
| 5.1 Comienzo del proyecto | 39 |
| 5.2 Desarrollo enfocado al usuario | 40 |

| | | |
|----------|--|-----------|
| 5.3 | Desarrollo del personaje jugador | 40 |
| 5.4 | Implementación de la alquimia e inventario | 42 |
| 5.4.1 | Lógica | 42 |
| 5.4.2 | Inventario de ítems | 43 |
| 5.4.3 | Libro de recetas y sistema de alquimia | 44 |
| 5.5 | Desarrollo del sistema compra-venta | 46 |
| 5.6 | Desarrollo de los enemigos | 46 |
| 5.7 | Desarrollo de niveles | 48 |
| 5.7.1 | Generación de mazmorras | 49 |
| 5.7.2 | Elementos de las mazmorras | 51 |
| 6 | Implantación y pruebas | 53 |
| 6.1 | Implantación | 53 |
| 7 | Conclusiones y trabajos futuros | 59 |
| 7.1 | Conclusiones | 59 |
| 7.2 | Trabajos futuros | 60 |
| | Bibliografía | 61 |
| <hr/> | | |
| | Apéndices | |
| A | Objetivos desarrollo sostenible | 63 |
| B | Alquimia - manual de juego | 67 |
| C | El Camino del Alquimista - Documento de diseño de juego | 77 |
| D | Recetario | 89 |
| E | Glosario | 97 |

Índice de figuras

| | | |
|------|--|----|
| 1.1 | Representación de los cuatro elementos de Empédocles | 2 |
| 1.2 | Alquimia, juego de cartas | 3 |
| 1.3 | Concept art, jugador y su hermana | 5 |
| 2.1 | Rogue, versión original | 9 |
| 2.2 | Pokémon Mundo Misterioso: Equipo de rescate azul, Nintendo DS | 10 |
| 2.3 | Hades, Supergiant Games | 11 |
| 2.4 | Receta de berenjenas rellenas del videojuego Don't Starve (Klei Entertainment, 2016) | 12 |
| 2.5 | Sistema de artesanía de Minecraft | 12 |
| 2.6 | Sistema de bricolaje de Animal Crossing: New Horizons | 13 |
| 3.1 | Diagrama de contexto | 21 |
| 3.2 | Diagrama casos de uso | 22 |
| 3.3 | Lista de tareas del proyecto | 23 |
| 3.4 | Sprint semana 11 | 23 |
| 4.1 | Diagrama de componentes UML | 25 |
| 4.2 | Diagrama de navegación de pantallas | 26 |
| 4.3 | Golden Sun, Camelot Software Planning, 2001 | 27 |
| 4.4 | Inazuma Eleven, Level-5, 2008 | 27 |
| 4.5 | The Legend of Zelda: The Minish Cap, 2004 | 28 |
| 4.6 | Objetos equipados, imagen del Camino del Alquimista | 28 |
| 4.7 | Mazmorra sin pasillos | 29 |
| 4.8 | Mazmorra con pasillos | 29 |
| 4.9 | Pasos disponibles en una cuadrícula | 30 |
| 4.10 | Mazmorra generada con el algoritmo de la caminata aleatoria, imagen obtenida de la página de la Wikipedia en inglés <i>Random Walk</i> | 30 |
| 4.11 | Máquina de estados, IA enemigos cuerpo a cuerpo | 31 |
| 4.12 | Máquina de estados, IA enemigos a distancia | 31 |
| 4.13 | Diagrama de clases UML del jugador | 32 |
| 4.14 | Diagrama de clases UML de las mazmorras | 33 |
| 4.15 | Editor de Unity | 34 |
| 4.16 | Ventana de jerarquía de Unity | 35 |
| 4.17 | Vista de escena de Unity | 35 |
| 4.18 | Ventana de proyecto de Unity | 36 |
| 4.19 | Ventana de inspector de Unity | 36 |
| 4.20 | Ventana de juego de Unity | 37 |
| 5.1 | Bocetos de Nico, realizados por mi compañero Tomás | 39 |
| 5.2 | Mando de Super Nintendo Entertainment System | 40 |
| 5.3 | Objetos equipados, imagen del Camino del Alquimista | 41 |
| 5.4 | Interfaz de la bolsa en el videojuego | 43 |
| 5.5 | Interfaz del baúl en el videojuego | 44 |

| | | |
|------|---|----|
| 5.6 | Primer boceto para el libro de recetas | 44 |
| 5.7 | Segundo boceto para el libro de recetas | 45 |
| 5.8 | Libro de recetas implementado en el juego | 45 |
| 5.9 | Sistema de alquimia del videojuego | 46 |
| 5.10 | Interfaz de la tienda de la vendedora ambulante | 47 |
| 5.11 | Boceto de los enemigos | 48 |
| 5.12 | Imagen de distintos enemigos | 48 |
| 5.13 | <i>NavMesh</i> de una mazmorra | 49 |
| 5.14 | Imagen de una mazmorra del videojuego | 50 |
| 5.15 | Matriz de caracteres que representa una mazmorra | 51 |
| 6.1 | Página del Camino del Alquimista en itch.io | 53 |
| 6.2 | Resultados del formulario, dificultad del juego | 54 |
| 6.3 | Resultados del formulario, se entiende qué hacer | 54 |
| 6.4 | Resultados del formulario, si es frustrante el juego | 55 |
| 6.5 | Resultados del formulario, controles intuitivos | 55 |
| 6.6 | Resultados del formulario, mecánicas preferidas del juego | 56 |
| 6.7 | Resultados del formulario, sistema de alquimia del juego | 56 |
| 6.8 | Resultados del formulario, obtención de recetas en el juego | 56 |
| 6.9 | Resultados del formulario, idioma preferido del juego | 57 |
| 6.10 | Resultados del formulario, controles preferidos en el juego | 57 |
| 6.11 | Resultados del formulario, acabar la mazmorra | 57 |
| 7.1 | Boceto del jefe de la mazmorra | 60 |

Índice de tablas

| | | |
|------|---|----|
| 2.1 | Características de <i>rogue-like</i> | 14 |
| 2.2 | Características de la artesanía | 14 |
| 5.1 | Requisitos cumplidos en el desarrollo enfocado al usuario | 40 |
| 5.2 | Requisitos cumplidos en el desarrollo del jugador | 41 |
| 5.3 | Requisitos cumplidos en el desarrollo de la lógica de ítems, recetas e inventarios | 42 |
| 5.4 | Requisitos cumplidos en el desarrollo de las UI de los inventarios de ítems | 43 |
| 5.5 | Requisitos cumplidos en el desarrollo del libro de recetas y el sistema de alquimia | 46 |
| 5.6 | Requisitos cumplidos en el desarrollo del sistema compra-venta | 47 |
| 5.7 | Requisitos cumplidos en el desarrollo de los monstruos | 49 |
| 5.8 | Significados de los caracteres de la representación ASCII de una mazmorra | 52 |
| 5.9 | Requisitos cumplidos en el desarrollo de la generación de la mazmorra | 52 |
| 5.10 | Requisitos cumplidos en el desarrollo de los elementos de la mazmorra | 52 |

CAPÍTULO 1

Introducción

El proyecto software a desarrollar para este trabajo fin de grado es un videojuego. Los videojuegos son un medio de entretenimiento muy joven pues, aunque se puede decir que nacieron en la década de los 1950s, estos eran todavía muy primitivos, y estaban muy lejos del gran público. Los videojuegos tal y como los conocemos hoy en día comenzaron en la década de los 70s, cuando nació una de las primeras máquinas arcade conocida como Pong (Atari, 1972), que es considerada como el producto que dio comienzo a la industria de los videojuegos. En esta década también comenzaron a comercializarse los primeros ordenadores personales. Gracias a esto, nacieron los primeros videojuegos para ordenador como Adventure (William Crowther, 1976) que, si no fue el primero, sí que fue uno de los más influyentes de la época. En esta década también comenzó el tercer pilar de los videojuegos: las consolas caseras como la Atari 2600 (Atari, 1977). La industria del videojuego había nacido y fue expandiéndose cada vez más hasta el día de hoy. [1]

Debido a la gran diversidad existente, los videojuegos se clasifican en distintos géneros. El Camino del Alquimista pertenece al género de rol (RPG) y más concretamente al subgénero conocido como *rogue-like*. Este subgénero, también conocido como exploración de mazmorras, debe su nombre a Rogue (Epyx, 1980) el primer juego del género que inspiró a infinidad de títulos que vinieron después. Estos se caracterizan en que su mecánica principal es la exploración de grandes calabozos laberínticos donde hay que enfrentarse a toda clase de criaturas y monstruos, para encontrar los tesoros que estos custodian. Lo particular de estas mazmorras es que son generadas al azar, lo que hace que cada partida sea diferente, pues nunca hay dos calabozos iguales. Otra característica de este género es que la muerte es un punto de no retorno, lo que significa que, al morir el jugador, debe volver a comenzar la mazmorra de nuevo, la cual habrá cambiado. Esta aleatoriedad es el principal atractivo de este género. [2]

El Camino del Alquimista se enfoca además en una mecánica recurrente en muchos RPG: la artesanía (*craft* en inglés), que consiste en la fabricación de objetos a partir de otros objetos más simples, y cuyos resultados se pueden utilizar para crear elaboraciones más complejas. Esta mecánica se basa en la obtención de recetas, la recolección de ingredientes y la utilización de un puesto de fabricación para crear los nuevos objetos. La artesanía se ha utilizado en infinidad de videojuegos de los géneros de rol y aventuras, siendo Minecraft (Mojang Studios, 2011) uno de los títulos más populares que ha explorado esta mecánica en profundidad. Esta mecánica es la principal en torno a que gira El Camino del Alquimista.

1.1 Motivación

Dentro de los productos de software, los videojuegos son sin duda uno de los más atractivos para el consumidor. Los videojuegos son, en general, la primera toma de contacto de muchos usuarios con el mundo de la informática. Pues son principalmente diversión y entretenimiento.

En los últimos años cada vez más títulos han ido introduciendo la mecánica de la artesanía, volviéndose cada vez más popular. Sin embargo, esta no suele ser el eje central del título, sino un añadido del juego. Es por eso por lo que, en *El Camino del Alquimista*, la artesanía, la alquimia, es el centro del juego, como bien se puede ver en su título.

La artesanía que se puede encontrar en este videojuego está basada en la alquimia. Esta fue una protociencia y disciplina filosófica, fuertemente relacionada con la química, la metalurgia, la astrología e incluso la medicina. La alquimia se practicó durante cientos de años por todo el mundo, desde el antiguo Egipto, la antigua Grecia y el mundo islámico hasta en la Europa medieval. Un concepto muy importante en la alquimia, concebido por Empédocles (495a.C. - 435a.C.) y desarrollado por Aristóteles (384 a.C. – 322 a.C) es que todas las cosas del universo están compuestas por los cuatro elementos: aire, agua, tierra y fuego. Una representación de los elementos se puede ver en la Figura 1.1. Este concepto se traslada al videojuego *El Camino del Alquimista*, ya que estos cuatro elementos son la base para fabricar el reto de ingredientes del juego. En la Europa medieval la alquimia se fue extendiendo por los filósofos, religiosos y burgueses del continente. Durante la baja Edad Media, los alquimistas se centraron en la búsqueda de la piedra filosofal que se cree que Nicolás Flamel (1330 - 1417) halló; piedra clave para transmutar el plomo en oro. El otro objetivo principal fue la búsqueda del elixir de vida, una poción que se creía conceder la vida eterna o bien, ser una panacea universal capaz de curar todas las enfermedades. Este elixir es el objetivo a conseguir en *El Camino del Alquimista*. [3, 4]

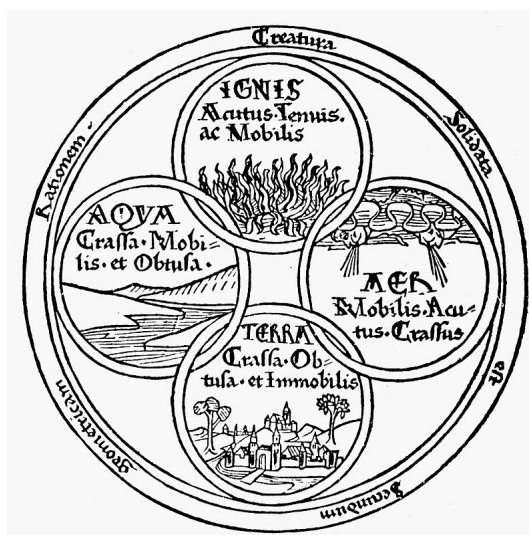


Figura 1.1: Representación de los cuatro elementos de Empédocles

La idea original para explorar esta mecánica no fue un videojuego, si no un juego de cartas. En la Pascua de 2021, teniendo en mente explotar más la fabricación de objetos en forma de juego, creé junto a Sandra Cerverón Carot un juego de mesa al que bautizamos como *Alquimia*. En este juego el objetivo era fabricar un objeto: el elixir de vida. El juego

se basa principalmente en hacerse con los ingredientes necesarios, ya sea mediante las cartas o haciendo trueques con los otros jugadores, para fabricar dicho elixir y así ganar la partida. El juego está compuesto por tres mazos diferentes, uno con cartas de lugares, uno con las cartas de los ingredientes y objetos, y otro con cartas de los monstruos, además de un recetario donde ver todas las recetas del juego, y monedas, dados y fichas. Alquimia es un juego de cartas de 2 a 7 jugadores. Para comenzar el juego es necesario colocar los mazos de lugares y monstruos en el centro de la mesa. A continuación, se reparten dos monedas a cada jugador, que podrán ser utilizadas para comprar ingredientes.

Alquimia es un juego por turnos. En su turno, cada jugador debe levantar primero una carta de lugar, y leer lo que indique. Si en dicha carta no hay escrita ninguna instrucción, esto significa que es una zona de monstruo, por lo que a continuación, se ha de levantar una carta de monstruo y enfrentarse a él con ayuda de los dados. Si el número obtenido al lanzar los dados supera la cifra indicada en la carta, dicho monstruo es derrotado, y como recompensa el jugador recibe una cantidad de fichas de un tipo concreto; el tipo y cantidad están indicados en la carta del monstruo. Estas fichas son los ingredientes más básicos del juego a partir de los cuáles se puede fabricar cualquiera de las recetas del juego. En su turno, un jugador también puede proponer un trueque a otro, fabricar algún objeto o utilizar alguno de los objetos con función contra otro jugador, por ejemplo, para robar un ingrediente. El juego termina cuando uno de los jugadores consigue fabricar el elixir de vida y se proclama ganador. Una imagen de dicho juego se puede ver en la Figura 1.2.



Figura 1.2: Alquimia, juego de cartas

Este juego de mesa ha inspirado el proyecto de mi TFG. El Camino del Alquimista es una forma de adaptar dicho juego de mesa para varios jugadores a un videojuego para un solo jugador que mantiene varios elementos comunes con el juego original, como puede ser en algunas de las maneras de obtener ingredientes. Estas son bien enfrentándose a monstruos y obteniendo dichos ingredientes como recompensa, aunque de una

forma mucho más activa al tratarse de un videojuego, o bien, comprándolos en la tienda a cambio de monedas como también se podía hacer en el juego de cartas. Una gran parte de los ingredientes y recetas del juego de cartas puede encontrarse también en el videojuego. Por último, el objetivo de ambos juegos es conseguir fabricar el mítico elixir de vida. Viendo estas similitudes entre ambos, se puede decir que El Camino del Alquimista conserva la esencia del juego de mesa que lo inspiró.

1.2 Objetivos

A continuación se enumeran los distintos objetivos de este trabajo final de grado.

- Desarrollar un proyecto software desde cero, participando en todas las fases del desarrollo, comenzando por el análisis de requisitos, siguiendo con el diseño y planificación del proyecto, la implementación del mismo, el control y finalizando con el mantenimiento o en este caso el cierre del proyecto.
- Aprender a utilizar metodologías ágiles para desarrollar un proyecto, en concreto el marco de trabajo conocido como Scrum.
- Trabajar en un proyecto con un equipo multidisciplinar como se hace en los proyectos reales de las empresas.
- Aprender a utilizar el motor de desarrollo de videojuegos Unity, en concreto la vertiente 2D y enfocado a la plataforma PC.

1.3 Metodología

En este proyecto se han utilizado metodologías ágiles para llevarlo a cabo y seguir un desarrollo constante. Al principio del proyecto, se recopilaron los requisitos tanto funcionales como no funcionales que debería cumplir el videojuego. Una vez definidos dichos requisitos, se dictaron las distintas tareas que, en su conjunto, englobaban el producto final que se deseaba conseguir. Dichas tareas se pueden modificar a lo largo del desarrollo del proyecto ya que la metodología ágil permite un desarrollo más flexible donde las distintas fases del proyecto se solapan entre sí. Teniendo ya las tareas definidas, se dividió el tiempo que se disponía hasta la fecha cuando el proyecto software debía estar terminado, en sprints de dos semanas. Un sprint, en este tipo de metodologías, es un periodo de tiempo al que se asignan unas tareas para desarrollar. En cada sprint, se pueden ver las tareas asignadas a cada miembro del grupo, qué tareas están en desarrollo actualmente, cuáles se han finalizado y cuáles están todavía por comenzar. De esta manera, la organización de un proyecto a largo plazo como este fue mucho más sencilla y eficiente. En todo momento sabíamos qué habíamos hecho y qué quedaba por hacer, además que evitamos así una sobrecarga de trabajo, pues desde el primer momento repartimos de manera equilibrada las tareas entre los diferentes sprints.

1.4 Estructura de la memoria

La memoria está compuesta por una introducción, a la que pertenece esta sección. A continuación, nos encontraremos con el estado del arte, donde se explorarán distintos productos similares al desarrollado en este proyecto, y se realizará una comparativa entre sus características y las de nuestro videojuego. También se analizarán las distintas alternativas que había para desarrollar diferentes mecánicas del juego y por qué se eligieron las implementadas finalmente. Seguidamente, se realizará un análisis del problema a resolver y el proyecto a desarrollar, donde se recogerán los requisitos tanto funcionales como no funcionales. A continuación, se realizará el diseño del videojuego y sus componentes así como se explicarán las tecnologías utilizadas en este proyecto. Se continuará con el desarrollo del proyecto, cómo se ha llevado a cabo, así como su implementación. Y se finalizará analizando los resultados de las pruebas hechas con usuarios, las conclusiones y los trabajos futuros del proyecto. Al final del documento existe una bibliografía con las referencias que se han utilizado para la redacción de esta memoria además de varios documentos anexos, estos incluyen el GDD del videojuego, un manual del juego de mesa que inspiró este proyecto, un recetario en excel de todas las recetas que aparecen en el juego y el documento de los ODS además de un glosario con todas las abreviaciones utilizadas en este documento.

1.5 Colaboraciones

Este videojuego se ha realizado junto con mi compañero José Tomás Puertas Caballero de la Facultad de Bellas Artes de la UPV, quien se ha encargado de la parte artística del juego, realizando el diseño de personajes, *concept arts*, sprites, *tilemaps* y animaciones. Este trabajo ha sido también parte de su trabajo fin de grado 'El Camino del Alquimista: Desarrollo conceptual y gráfico de un videojuego'. Para más información, consultar su trabajo que se puede encontrar en la web Riunet de la UPV en la sección de la Facultad de Bellas Artes.

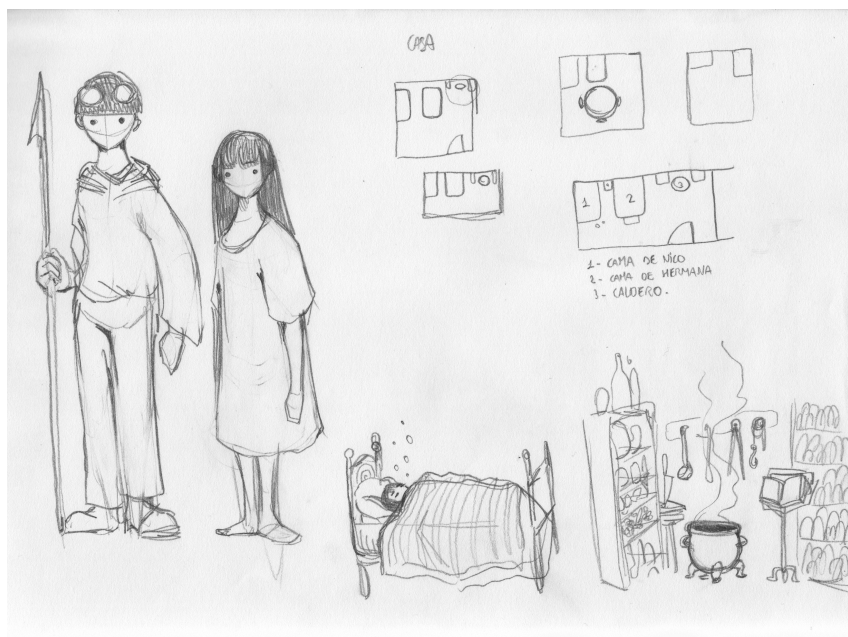


Figura 1.3: Concept art, jugador y su hermana

CAPÍTULO 2

Estado del arte

En este capítulo estudiaremos el contexto tecnológico de nuestro videojuego. Lo compararemos con juegos del mercado con los que comparta características y/o género para así ver qué lo hace destacar del resto. También se estudiará más en profundidad el origen de las mecánicas principales de El Camino del Alquimista: la generación de mazmorras aleatorias y el sistema de artesanía. Además, se evaluarán distintas alternativas para las principales mecánicas del juego, así como se justificará la elección final de las mismas entre las alternativas para el proyecto final.

2.1 Juegos de mazmorras

Los juegos de mazmorras consiguen que cada partida sea diferente gracias a su algoritmo generador de calabozos. Este tipo de mecánica permite ahorrar recursos, tanto en diseñadores como en tiempo, pues no es necesario diseñarlas manualmente. A continuación analizaremos las principales características de estos juegos, además de algunos de los títulos más representativos.

2.1.1. Interpretación de Berlín

En 2008 se realizó la primera Conferencia Internacional de Desarrollo de Roguelike (IRDC) en Berlín, donde se definió el término. Además, se definieron unos puntos que cumplir, para poder determinar si un juego pertenece o no a este género. Estos están divididos en factores de valor alto, los más importantes, y factores de valor bajo. No es necesario cumplir todos estos puntos para ser considerado videojuego de mazmorras.

Primero explicaremos los factores de valor alto, los que se consideran más esenciales para ser considerado dentro del género:

- Generación aleatoria del escenario: el mapa, tesoros y monstruos son aleatorios cada partida.
- Muerte permanente: si muere, el nivel se comienza de nuevo, perdiendo todo el progreso.
- Juego basado en turnos: cada movimiento es un turno.
- Juego basado en celdas: el mapa esta dividido en celdas, tanto el jugador como los enemigos y elementos del mapa ocupan una celda.

- No modal: El movimiento, combate y otras acciones son accesibles desde el mismo módulo y en cualquier momento del juego.
- Complejidad: El juego tiene una complejidad que permite seguir diferentes soluciones para llegar a un mismo objetivo.
- Gestión de recursos: El jugador tiene una cantidad limitada de recursos (como comida o pociones) que tendrá que gestionar con cuidado.
- *Hack and slash* (También conocido como mata mata en español): El combate es una parte importante de este género. Las mazmorras suelen estar repletas de enemigos que el jugador deberá matar si quiere llegar al final.
- Exploración y descubrimiento: El juego debe requerir de exploración por parte del jugador para encontrar todos los secretos y objetos especiales que ocultan.

A continuación, expondremos los factores de valor bajo, que no son necesarios para que un juego entre dentro del género:

- Un solo jugador, un solo personaje: El juego es para un solo jugador, donde este controla a un único personaje. Cuando este muere acaba la partida.
- Monstruos similares al jugador: Los enemigos son similares al jugador, les afecta los mismos elementos, como las trampas. Además, también disponen de las mismas armas, armaduras...
- Reto táctico: El jugador debe aprender las tácticas y trucos de las mazmorras antes de conseguir hacer grandes progresos. Esto lo aprenderá por medio de prueba y error muriendo durante sus primeras partidas.
- Gráficos ASCII: En los *rogue-like* tradicionales todos los elementos del juego son representados mediante letras, número y símbolos.
- Mazmorras: Los juegos de este género contienen mazmorras compuestas por diversos niveles de habitaciones y pasillos.
- Números: En todo momento se muestran las características del jugador mediante números en la pantalla, como puede ser los puntos de vida.

Estos son todos los factores que se decidieron en dicha convención como los necesarios para poder clasificar a un videojuego en este género. [5]

2.1.2. Rogue

Rogue (Epix, 1980) fue el videojuego que precedió al género conocido hoy en día como juegos de mazmorras o *rogue-like* (parecidos a *rogue* en inglés). Por aquel entonces los 'ordenadores' de los que disponían la mayoría de estudiantes de entonces eran terminales tontas, las cuales consistían en una pantalla de salida y un teclado de entrada con el que se podían transmitir datos a una computadora remota, que sí tenía capacidad de almacenamiento y procesamiento. Estas terminales tampoco disponían de ninguna capacidad gráfica. Los programas que ejecutaban simplemente mostraban texto en pantalla.

Por esta época, nació la librería *curses* que permitía simular imágenes tan solo con letras, símbolos y números. Dicha librería se utilizó para crear *Rogue*, siendo así una de

las primeras aventuras gráficas, como se puede ver en la Figura 2.1. El objetivo de Rogue era ser un videojuego que pudieran disfrutar los creadores del mismo, lo cuál era difícil si conocían todos los secretos y puzzles de este. Para conseguir este objetivo, decidieron que lo mejor era que el mismo programa generara la mazmorra cada partida de manera aleatoria. Además, así conseguirían dar al jugador una nueva aventura cada vez que jugaba. Así fue cómo nació Rogue, el juego que inspiraría un género entero.[6]

Rogue consiste en explorar una mazmorra laberíntica de varios pisos hasta llegar al piso más profundo donde se encuentra un medallón, el Amuleto de Yendor, y volver a subir al primer piso para salir al exterior con este en su poder. Antes de conseguir el amuleto, el jugador no puede volver a los pisos anteriores. Durante la exploración, el jugador se irá encontrando con monstruos que serán más fuertes conforme se adentre en la mazmorra. Trampas y tesoros, como pociones o armaduras, le ayudarán en su aventura. Rogue es un juego basado en turnos. Cada movimiento del jugador es considerado un turno como puede ser dar un paso, hacer un ataque o tomar una poción, entre otros. Los enemigos sólo pueden realizar un movimiento por turno.[7]



Figura 2.1: Rogue, versión original

2.1.3. Mystery Dungeon

Mystery Dungeon, conocido en España como Mundo Misterioso, es una serie de videojuegos que introdujo el género *rogue-like* a las consolas. Aunque hubo algún intento previo por parte de Sega, no alcanzaron el éxito. Esta serie comenzó con la idea de desarrollar un título de la popular saga *Dragon Quest* (Enix, 1986), basado en Rogue, un juego que fuera más sencillo y comprensible, enfocado esencialmente los jugadores de consola. Este juego fue *Taloon's Great Adventure: Mystery Dungeon* (Chunsoft, 1993).

La serie de Mundo Misterioso está entrelazada no solo con la saga *Dragon Quest*, sino además con otras sagas como *Final Fantasy* con títulos como *Chocobo's Dungeon 2* (Square, 1998) o con *Pokémon*, con varios títulos como *Pokémon Mundo Misterioso: equipo de rescate rojo y azul* (Chunsoft, 2006) que se puede ver en la Figura 2.2 esta última subsaga de *Pokémon Mundo Misterioso* es la que más éxito ha conseguido dentro de los juegos *Mystery Dungeon*. Los juegos pertenecientes a esta serie se caracterizan en tener un protagonista, o a veces un equipo, que se dedica a explorar distintas mazmorras generadas al azar. Una diferencia entre esta serie y Rogue, es que no tiene muerte permanente. Cuando el jugador muere o cae derrotado, se restablece su progreso al principio de la mazmorra pero normalmente sin perder todos los objetos y dinero encontrados durante el recorrido. Estos juegos se basan en turnos, una de las características del género. Una característica diferenciadora con la mayoría de juegos del género, es que los personajes

tienen un nivel, que al combatir y ganar puntos de experiencia van mejorando, haciendo cada vez más sencillo avanzar en las mazmorras. En general, se puede decir que esta serie es una adaptación del género de juegos de mazmorras a un público más general. [8]



Figura 2.2: Pokémon Mundo Misterioso: Equipo de rescate azul, Nintendo DS

2.1.4. Hades

En las últimas décadas, desde finales de los 2000, este género ha sido cada vez más popular entre los videojuegos independientes. Los desarrolladores comenzaron a combinar este género con géneros totalmente distintos, y aquí nacieron títulos como *Spelunky* (Derek Yu, 2008) que combina elementos *rogue-like* con plataformas resultando en algo nunca antes visto. A este le siguió *The Binding of Isaac* (Edmund McMillen, 2011) que combinó elementos de acción RPG con características de los juegos de mazmorras como la exploración de mazmorras siendo aleatorias y la muerte permanente, pero con un combate sin turnos. Estos títulos dieron un aire fresco al género, mostrando que habían infinitas posibilidades.

Siguiendo la estela de los ya mencionados, finalmente hablaremos de uno de los títulos más recientes de este género, *Hades* (Supergiant Games, 2020) un *rogue-like* indie que fue nominado a Juego del año en los Game Awards de 2020. Al igual que *The Binding of Isaac*, *Hades* combina la acción con la exploración de mazmorras. Además, este juego tiene un buen equilibrio entre la aleatoriedad de los calabozos y el nivel de decisión del jugador, pues la mazmorra de *Hades* funciona por estancias, sin pasillos. En cada cámara hay una recompensa, que podrá conseguir el jugador una vez acabe con los enemigos que la custodian. Una vez derrotados, y obtenida la recompensa, aparecen varias puertas que llegan a las siguientes cámaras. Las puertas indican la recompensa que aguarda en su interior, dando la posibilidad de elegir al jugador qué recompensa será mejor en su aventura, ej. más vida o una nueva habilidad. *Hades* fue un gran éxito también por su forma de utilizar el ciclo vida-muerte de los *rogue-like* en una forma que produce satisfacción para el jugador, en lugar de frustración. Después de cada muerte, el jugador vuelve a la casa, donde puede comprar mejoras permanentes y nuevas armas con los puntos que ha conseguido en su último recorrido de la mazmorra. También se desbloquean nuevos diálogos con los personajes no jugables (pnj o *npc* en inglés) que le permite descubrir nuevas partes de la historia. Estas pequeñas recompensas consiguen animar al jugador a volver a intentarlo y mantener su ilusión, algo esencial en este tipo de videojuegos, donde la

muerte no es el final sino un nuevo comienzo, una nueva oportunidad de llegar un poco más lejos. [9]



Figura 2.3: Hades, Supergiant Games

2.2 Juegos con artesanía

La artesanía es una mecánica que ha sido introducida en muchos juegos y géneros distintos. Es muy fácil encontrarla en gran parte de los títulos más recientes, pues esta es cada vez más popular. Incluso se pueden encontrar en sagas bien asentadas donde nunca se había visto algo parecido, como puede ser el caso de Pokémon Leyendas: Arceus (Game Freak, 2022), primer juego de la franquicia en introducir dicha mecánica. La artesanía o *crafting* es una parte importante de cada vez más juegos, pero ¿en qué consiste esta mecánica?

El *crafting* consiste en crear, construir, fabricar objetos a partir de otros más simples o materias primas. Esto produce satisfacción en el jugador, como una pequeña recompensa al esfuerzo de recolectar los ingredientes necesarios o descubrir dicha receta. Esta mecánica puede tener finalidades muy distintas dependiendo del género donde se implemente. En un RPG clásico como puede ser Dragon Quest IX: Centinelas del firmamento (Level-5, 2010) la artesanía se emplea para conseguir mejores armas y armaduras para combatir. En un juego de simulación de vida como es Animal Crossing: New Horizons (Nintendo, 2020) la artesanía sirve para crear herramientas, muebles y decoración. En un juego tipo *sandbox* como Minecraft (Mojang Games, 2011), el *crafting* es uno de los pilares esenciales del *gameplay* y sirve para obtener la gran mayoría de objetos del juego.

La artesanía se ejecuta de distintas maneras en cada título, siendo o no una parte importante del juego en sí. Normalmente, para fabricar un objeto es necesario seguir una receta, que consiste en los ingredientes con sus cantidades necesarias para construir un determinado objeto. Las recetas pueden estar perfectamente definidas, indicando los ingredientes y cantidades necesarias para fabricar un objeto o pueden ser paramétricas, lo que significa que distintos ingredientes pueden dar el mismo resultado. La Figura 2.4 muestra la receta para cocinar berenjenas rellenas (resultado en la casilla derecha), utilizando una berenjena (primera casilla), una verdura a elección del jugador (segunda casilla) y dos ingredientes más sin restricciones (casillas con la interrogación). Por último, hay recetas no definidas, que se descubren experimentando con distintos materiales. Este sistema de construcción libre permite crear objetos que sorprenden hasta a los mismos desarrolladores del juego. Esta creatividad y cantidad de posibilidades es uno de los

principales atractivos de juegos como Minecraft o Terraria (Re-logic, 2011), entre otros. Otro factor a diferenciar en la artesanía es dónde se pueden fabricar los objetos. Hay dos opciones: una es que el jugador puede fabricar donde y cuando quiera, o por el contrario, el jugador tiene que ir a una estación o mesa de trabajo para fabricar lo que desee. [10]



Figura 2.4: Receta de berenjenas rellenas del videojuego Don't Starve (Klei Entertainment, 2016)

2.2.1. Minecraft

Minecraft es un videojuego de mundo abierto (o *sandbox* en inglés). Estos juegos hacen una analogía con una caja de arena, donde los niños pueden hacer volar su imaginación dando a la arena infinidad de formas. En este tipo de juegos los desarrolladores dan libertad al jugador sin objetivos que cumplir. Minecraft consiste esencialmente en dos aspectos, como indica su nombre: *mine* o sea minería, recolectar materiales usualmente mediante un pico de minero; y *craft* lo que nos concierne en este análisis. Este juego consiste básicamente en un ciclo de exploración del mundo y recolección de ingredientes y materiales. Con ellos se fabrican nuevos picos, armas... que te permiten recolectar nuevos materiales, sobrevivir y explorar un poco más lejos, consiguiendo así nuevos ingredientes con los que fabricar nuevos objetos. En la Figura 2.5 se puede ver cómo es el sistema de *crafting* de Minecraft. Hay una estación de trabajo compuesta por una matriz de 3x3 donde se colocan los materiales. Dependiendo de la distribución elegida y los materiales, se puede fabricar objetos distintos. Las recetas se pueden consultar en el libro de recetas, que se van desbloqueando a medida que se obtienen ingredientes nuevos. Minecraft es uno de los juegos que más han explorado esta mecánica, y como hemos visto la ha hecho una de las bases de su *gameplay*.

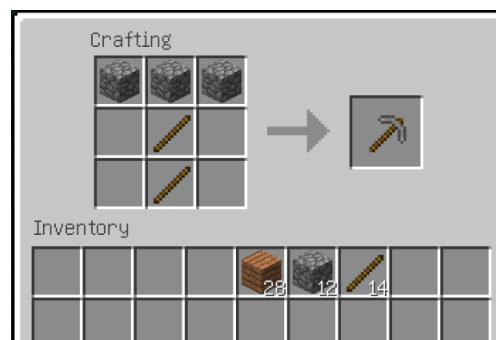


Figura 2.5: Sistema de artesanía de Minecraft

2.2.2. Animal Crossing: New Horizons

Animal Crossing: New Horizons es el título más reciente de la saga Animal Crossing. Este juego es un simulador de vida, donde el jugador comparte un pequeño pueblo con unos simpáticos animales antropomórficos que son sus vecinos. Este último título añadió por primera vez el *crafting* o bricolaje (como se denomina en el juego) a la saga haciéndolo uno de los ejes principales del *gameplay*. En este Animal Crossing se pueden fabricar herramientas como cañas de pescar, hachas, palas... También hay cientos de muebles que no se pueden comprar y que hay que fabricar. Estos son muy útiles en un juego donde

el principal objetivo es transformar una isla desierta y salvaje en una bonita isla llena de vida que anime a más gente a vivir allí. Las recetas en este juego pueden obtenerse tanto gracias a los vecinos que las regalan al jugador como en botellas que llegan a la orilla de las playas, además de compartirlas con amigos gracias a las funciones multijugador. No es posible fabricar algo sin su receta. El sistema de bricolaje de este juego se basa en seleccionar la receta del objeto que se quiere fabricar y, si se dispone de los ingredientes necesarios, apretar el botón indicado. Los ingredientes necesarios para fabricarlas se obtendrán normalmente de los recursos naturales de la isla, como madera, piedra, oro, bambú, conchas... o bien a partir de otros objetos más simples. Al ser un juego con cambio de estaciones, hay materiales y recetas exclusivas de determinadas épocas del año como puede ser copos de nieve en invierno, bellotas y setas en otoño o huevos de Pascua en Semana Santa, entre otros. Esto le da un aire fresco al sistema clásico de artesanía.



Figura 2.6: Sistema de bricolaje de Animal Crossing: New Horizons

2.3 Comparación con la propuesta

A continuación explicaremos cómo la propuesta, *El Camino del Alquimista*, ha implementado las dos mecánicas vistas en este apartado. Haremos una comparación de ellas con los títulos mencionados. Comenzamos viendo qué características propias de los *rogue-like* cumple y de qué manera.

El Camino del Alquimista cumple varios de los factores de Berlín, pudiendo considerarse un *rogue-like*. Primero comenzaremos a explicar qué factores de valor alto cumple. El primero sería que los mapas que se recorren en dicho juego son generados automáticamente y aleatoriamente mediante un algoritmo, siendo este es uno de los factores más importantes. Pese a no tener muerte permanente, cuando muere, el jugador pierde una parte de los tesoros recogidos en su partida, además de perder el progreso y tener que comenzar el recorrido desde el principio en un mapa nuevo, al igual que sucedía con los juegos pertenecientes a *Mundo Misterioso*. También cumple el punto de *Hack and slash*, pues el jugador deberá matar a los monstruos que se encuentra si no quiere morir en sus garras. La gestión de recursos será también algo a tener en cuenta por parte del jugador, pues antes de entrar en una mazmorra y comenzar el recorrido, podrá equiparse con objetos limitados que le ayudarán en su aventura como pociones curativas o bombas. La exploración es uno de los elementos más importantes del juego, pues la mazmorra tendrá tesoros escondidos que son esenciales para avanzar en el juego. Continuamos con los factores de valor bajo que se pueden encontrar en *El Camino del Alquimista*, como son las mazmorras que se exploran en el juego, además que sólo hay un personaje a controlar por el jugador. La propuesta coge el enfoque al gran público que tiene la saga *Mundo Misterioso* que se puede ver en la muerte, dónde se pierde sólo parte de lo conseguido en

el recorrido, evitando una muerte permanente que puede ser frustrante para jugadores novatos; un enfoque a los combates dinámicos de acción que se pueden encontrar en Hades o The binding of Isaac, en lugar de los clásicos combates por turnos. Así, da como resultado un juego dinámico y apto para personas que nunca se habían acercado a este género. Finalmente, compararemos El Camino del Alquimista con varios juegos de mazmorras y así analizaremos sus características en la Tabla 2.1.

| Comparación títulos de mazmorras | | | | |
|----------------------------------|-------------------|-------|-------|-----------------------|
| | Camino Alquimista | Rogue | Hades | Pkmm Mundo Misterioso |
| Gen. procedural | sí | sí | sí | sí |
| Muerte permanente | no | sí | sí | no |
| Basado en turnos | no | sí | no | sí |
| Gestión recursos | sí | sí | no | sí |
| <i>Hack & slash</i> | sí | sí | sí | sí |
| Exploración | sí | sí | no | sí |

Tabla 2.1: Características de *rogue-like*

Ahora hablaremos de cómo es el sistema de artesanía de El Camino del Alquimista. Para fabricar un determinado objeto, será necesario no sólo disponer de los ingredientes y cantidades pertinentes, sino además tener su receta en el libro de recetas. Estas recetas se encontrarán como tesoros dentro de las mazmorras que explora el jugador. Los ingredientes también se obtienen dentro de estos calabozos, ya sea derrotando monstruos, encontrándolos en los cofres o bien comprándolos a la vendedora ambulante que se puede encontrar dentro de este laberíntico lugar. Una vez obtenidas las recetas e ingredientes necesarios, se deberá ir a la casa del protagonista, donde se encuentra el caldero de alquimia (la estación de trabajo), con el que se puede fabricar todas las recetas disponibles en el libro. Esto se hace simplemente seleccionando la receta deseada y apretando el botón pertinente. Este sistema de alquimia es similar al sistema de bricolaje visto en Animal Crossing: New Horizons, tanto en la forma de crear los objetos como en la obtención de nuevas recetas. En la Tabla 2.2 veremos cómo distintos juegos afrontan esta mecánica:

| Comparación títulos con artesanía | | | | |
|-----------------------------------|-------------------|-----------|-----------------|----------------|
| | Camino Alquimista | Minecraft | AC:New Horizons | Don't Starve |
| Recetas | fuertes | fuertes | fuertes | fuertes/param. |
| Mesa de trabajo | sí | no | sí | sí |
| Recetas bloqueadas | sí | no | sí | sí |

Tabla 2.2: Características de la artesanía

CAPÍTULO 3

Análisis del problema

El problema a resolver es la creación de un videojuego que pertenezca al género *rogue-like* y tenga la mecánica de artesanía. Como hay muy pocos títulos de este género que incluyan dicha funcionalidad, este proyecto es una oportunidad de innovar en el género.

3.1 Especificación de requisitos

La primera fase de cualquier proyecto software es la especificación de requisitos los cuales se dividen en requisitos funcionales, que dictan “qué debería hacer un sistema” definen funcionalidades del software. Por otra parte, están los requisitos no funcionales, que se refieren a aspectos más generales del producto, definen “qué debería ser un sistema” como por ejemplo, un requisito no funcional con respecto a la interfaz de usuario del juego sería que debe ser intuitiva para el jugador y no mostrar datos innecesarios.

3.1.1. Requisitos funcionales

Los requisitos funcionales recopilados para El Camino del Alquimista se engloban en diferentes categorías dependiendo a qué funcionalidad hacen referencia.

Partida

La partida del jugador y sus datos de guardado

- Part1 - El progreso de la partida se guardará cada vez que se salga de una mazmorra.
- Part2 - No se podrá guardar el progreso dentro de una mazmorra.
- Part3 - Los ajustes de usuario, como el idioma o nivel de volumen se guardarán.
- Part4 - Los datos del jugador que se guardan son: la vida máxima, el arma equipada, los ítems que posee tanto en la bolsa como en el baúl.

Equipamiento

Referente a las armas y objetos útiles (bombas o de curación) que se puede equipar el jugador.

- Equip1 - El arma que llevará el jugador será una lanza. Esta se podrá mejorar, pero nunca desequipar o empeorar.
- Equip2 - Se podrá equipar una sola lanza a la vez. No ocupará hueco en la bolsa.
- Equip3 - Se podrán equipar hasta 3 ítems. No pueden ser ítems sin función.
- Equip4 - Se podrán usar los ítems equipados de forma rápida sin entrar en el inventario.
- Equip5 - Se podrán reordenar los ítems equipados.

Bolsa

La bolsa es donde el jugador guardará los ítems que vaya recogiendo, así como los que decida llevarse para estar preparado en su aventura.

- Bolsa1 - El jugador podrá acceder en todo momento a la bolsa.
- Bolsa2 - La bolsa tendrá un espacio limitado (10-20) ítems.
- Bolsa3 - La bolsa tendrá un bolsillo seguro para meter ítems que no se quieran perder.
- Bolsa4 - El bolsillo seguro tendrá un tamaño reducido (3 ítems).
- Bolsa5 - Los ítems que hay en la bolsa podrán ser redistribuidos a gusto del jugador.

Baúl

El baúl es el inventario que el jugador tiene en la casa. Este es de mayor tamaño que la bolsa y no puede perder los ítems guardados aquí.

- Baul1 - El baúl tendrá un espacio muy amplio (30-60) ítems.
- Baul2 - Los ítems guardados aquí no desaparecerán ni se perderán.
- Baul3 - Sólo se podrá acceder al baúl dentro de la casa.
- Baul4 - Los ítems que hay en el baúl podrán ser redistribuidos a gusto del jugador.

Alquimia

El sistema de artesanía del juego, con el que se puede fabricar ítems a partir de otros

- Alq1 - Sólo se podrá alquimizar en el caldero de la casa.
- Alq2 - No se podrá fabricar un ítem sin poseer su receta.
- Alq3 - Las recetas se obtendrán en las mazmorras.
- Alq4 - Se podrá fabricar más de un ítem igual a la vez, si se dispone de materiales suficientes.

Item

Es un objeto, ingrediente o material que se puede obtener en el juego.

- Item1 - Los ítems se podrán obtener fabricándolos con su respectiva receta, en cofres dentro de las mazmorras, comprándolos a la vendedora ambulante o al vencer a un monstruo.
- Item2 - Habrá distintos tipos de ítems: sin función, de cura, de ataque, lanza, de mejora y especiales.
- Item3 - Los ítems serán acumulables hasta un límite que puede variar de uno a otro.
- Item4 - Los ítems no deseados se pueden tirar o vender a la mercader ambulante.

Receta

Una receta explica cómo fabricar un ítem, indicando cuáles son sus ingredientes y en qué cantidades.

- Rece1 - Las recetas sólo podrán tener como resultado un único ítem.
- Rece2 - Una receta no podrá tener más de cuatro ingredientes diferentes.
- Rece3Rece1 - Una receta deberá tener al menos un ingrediente.
- Rece4 - Las recetas se podrán conseguir solamente dentro de los cofres del tesoro.
- Rece5 - Las recetas podrán ser consultadas en cualquier momento en el libro de recetas.

Libro de recetas

Colección de recetas.

- Libro1 - El libro de recetas tiene solo las recetas que conoce el jugador.
- Libro2 - El libro de recetas comienza vacío.
- Libro3 - Las recetas se mostrarán en orden alfabético.
- Libro4 - Las recetas se podrán filtrar en base al tipo de ítem que fabrican, por ejemplo recetas de curación.

Monedero

El dinero del jugador.

- Moned1 - La mínima cantidad de dinero que puede tener es cero.
- Moned2 - El jugador no podrá tener más de cien monedas.
- Moned3 - El jugador podrá consultar cuánto dinero tiene en todo momento.
- Moned4 - A la hora de comprar o vender, la cantidad de dinero actual será visible para el jugador.

Combate

El sistema de batalla del jugador.

- Com1 - El arma de combate será una lanza, que se podrá ir mejorando mediante recetas.
- Com2 - El combate será uno contra muchos, yendo todos los monstruos a por el jugador a la vez.
- Com3 - Se podrán asignar hasta 3 objetos útiles diferentes a la vez para utilizarlos rápidamente en el combate.
- Com4 - Se podrán utilizar objetos no equipados desde la bolsa de inventario.
- Com5 - La lanza tendrá un ataque y un bloqueo.
- Com6 - Se podrá atacar con cuatro tipos de bombas diferentes, cada una de uno de los cuatro elementos. Habrá tres bombas de cada tipo: una débil, una normal y otra fuerte.
- Com7 - Los monstruos nunca huyen del combate, luchan hasta el final.
- Com8 - El combate será de acción, no a turnos.
- Com9 - El jugador podrá recuperar vida en medio del combate.
- Com10 - El jugador puede huir de los monstruos si no quiere combatir.

Mazmorra

Laberinto donde el jugador se adentra.

- Maz1 - Las mazmorras serán generadas proceduralmente.
- Maz2 - Las mazmorras están compuestas por varios pisos. Habrá que encontrar una puerta mágica para pasar de piso.
- Maz3 - En las mazmorras habrán monstruos, cofres del tesoro y trampas, todo generado de manea aleatoria.
- Maz4 - Para poder acceder a una mazmorra se tendrán que haberte pasado la anterior.
- Maz5 - Para pasarse una mazmorra habrá que llegar al último piso y derrotar al jefe final.
- Maz6 - Cada mazmorra tendrá unos monstruos distintos, adaptados a su clima.
- Maz7 - Si se muere en la mazmorra se perderá todo lo recolectado, menos lo guardado en el bolsillo seguro.
- Maz8 - Si se muere en la mazmorra, se perderá la mitad del dinero que se tiene.
- Maz9 - En todo momento se podrá volver a un piso anterior usando la puerta mágica de bajada.
- Maz10 - En las mazmorras se podrá encontrar a la vendedora ambulante.
- Maz11 - Se podrá huir de una mazmorra sin perder lo obtenido en esta con ayuda de un ítem.

Monstruo

Criatura que atacarán al jugador.

- Mon1 - Todo monstruo será de uno de los 4 elementos.
- Mon2 - Todo monstruo pertenece a un nivel 1, 2, 3 o Jefe, que indica cuán fuerte es.
- Mon3 - Todo monstruo tendrá las características de: Puntos de vida, potencia de ataque, velocidad.
- Mon4 - Al ser derrotado todo monstruo podrá (no siempre) soltar la esencia del elemento al que pertenece. Cuanto más nivel, más cantidad soltará.
- Mon5 - Todo monstruo detectará, al jugador cuando esté a cierta distancia.
- Mon6 - Todo monstruo perseguirá y atacará al jugador cuando lo haya detectado, evitando muros y agujeros.
- Mon7 - El monstruo tendrá una barra de vida encima de su cabeza, que indicará la cantidad de puntos de vida restantes.
- Mon8 - Los monstruos podrán atacar cuerpo a cuerpo o a distancia.
- Mon9 - Los ataques a distancia de los monstruos podrán dañar a otros monstruos que sean de un elemento distinto al suyo.
- Mon10 - Los monstruos que atacan cuerpo a cuerpo podrán quedar en estado de conmoción si son bloqueados en mitad de un ataque.

Vendedora ambulante

Mujer dedicada a la compra-venta de ítems.

- Ven1 - El jugador podrá comprar o vender ítems a la vendedora ambulante.
- Ven2 - La vendedora ambulante se encontrará sólo dentro de las mazmorras.
- Ven3 - Habrá una o ninguna vendedora ambulante por cada piso de la mazmorra.
- Ven4 - Los ítems que vende la vendedora podrán variar de un encuentro a otro.
- Ven5 - La mercader sólo puede vender 4 ítems distintos.
- Ven6 - El precio de venta de los objetos será la mitad del precio de compra.
- Ven7 - Hay ítems exclusivos que sólo se pueden obtener comprándolos a la vendedora.

Cofres del tesoro

Cofres con algo en su interior.

- Tes1 - Los cofres del tesoro se podrán encontrar en las mazmorras.
- Tes2 - Cada cofre se podrá abrir una sola vez.
- Tes3 - En un cofre se podrá encontrar monedas, ítems o recetas.
- Tes4 - Un cofre del tesoro no podrá dar al jugador una receta que ya conozca.
- Tes5 - Podrá haber más de un cofre en el mismo piso de la mazmorra.

Pozo de los deseos

Pozo donde puedes tirar una ofrenda.

- Pozo1 - Podrás arrojar cualquier cosa dentro (ítem o monedas) pero sólo 1.
- Pozo2 - El pozo te devolverá un ítem, monedas o nada.
- Pozo3 - El pozo será de solo un uso.
- Pozo4 - Podrás encontrar más de un pozo en cada mazmorra.
- Pozo5 - Sólo podrá haber uno o ningún pozo por piso de mazmorra.

Excursionista

Persona perdida en las mazmorras.

- Excur1 - Propondrán un trueque aleatorio de un ítem por otro.
- Excur2 - Podrán cambiar hasta 3 ítems.
- Excur3 - El trueque no cambiará aunque vuelvas a hablar con él/ella.

3.1.2. Requisitos no funcionales

A continuación, se enumeran los requisitos no funcionales del juego.

- RNF1 - La interfaz deberá ser sencilla e intuitiva. Sólo mostrará la información necesaria. El jugador debe aprender a navegar por ella nada más verla.
- RNF2 - La curva de aprendizaje del juego será muy rápida. En cada partida, jugador debe ser capaz de llegar más lejos en la mazmorra sin morir.
- RNF3 - El juego debe ser divertido y tener buen ritmo, no puede ser frustrante.
- RNF4 - Los controles deben ser intuitivos y sencillos de recordar. Un jugador medio debe ser capaz de recordarlos tras jugar dos o tres veces.
- RNF5 - El juego podrá ser ejecutado en un ordenador con sistema operativo Windows que sea al menos un procesador i3 y 4GB de RAM.

3.2 Modelado conceptual

Una vez detallados los requisitos, es hora de especificar de una forma más visual cómo será el sistema. Para ello, lo más común es utilizar el estándar UML (Lenguaje Unificado de Modelado) ya que es el lenguaje de modelado de software más conocido y utilizado en la actualidad. Con él definiremos los casos de uso, que son el conjunto de acciones realizadas por el sistema que dan lugar a un resultado observable. Estas acciones dependen de un actor para ser inicializadas, en nuestro caso el jugador. En la Figura 3.1 se define de una manera general los casos de uso de este proyecto. Se han agrupado en tres sistemas diferentes: el sistema de artesanía, el sistema de interacción con el mundo y manejo del inventario, y por último el sistema de compra-venta, con este diagrama se puede ver el contexto general de los casos de uso, además de cómo se relacionan entre

sí, y los actores implicados. Estos sistemas se pueden ver en mayor detalle, con todos sus casos de uso especificados en la Figura 3.2.

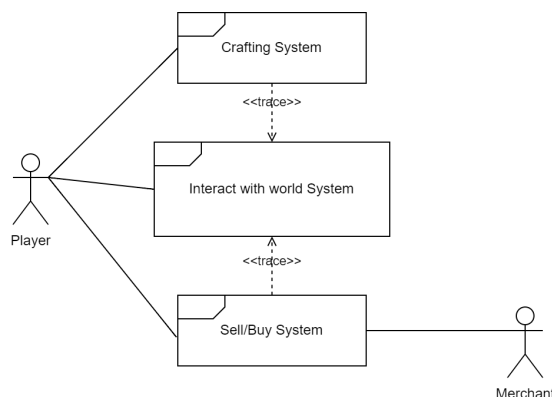


Figura 3.1: Diagrama de contexto

En el capítulo siguiente, se realizará un modelado de las clases a partir de los casos de uso y requisitos de este capítulo. Este modelado seguirá también el estándar UML y describirá con más detalle el sistema.

3.3 Análisis del marco legal

La propiedad intelectual considera aspectos como qué tipo de licencia se ha utilizado en el software utilizado para el desarrollo de la aplicación. El motor de videojuegos utilizado para el desarrollo del proyecto ha sido Unity Personal, la versión gratuita de Unity para usuarios individuales y aficionados.

Otros aspectos a tener en cuenta al hablar de la propiedad intelectual de un producto son las imágenes utilizadas en este. Como ya se ha mencionado en la introducción de esta memoria, toda la parte artística de este videojuego, ya sean *sprites* de los personajes, los mapas del juego, elementos de los menús como botones o fondos, incluso los elementos de las interfaces de usuario, han sido realizados exclusivamente para este proyecto por parte del artista y compañero Tomás Puertas Caballero, y por lo tanto usado propiedad intelectual de terceros. Otro aspecto importante es la música y efectos de sonido. Ambas han sido descargadas de [OpenGameArt](#), una web cuyo contenido es arte y música para videojuegos, donde los autores ceden sus obras a cambio de darles crédito en los juegos donde se utilizan. Para la música de fondo del juego he utilizado 'A small fire will do' de Cal McEachern, 'A long way from home' de HitCtrl, 'Woodland fantasy' de Matthew Pablo, 'Mage theme' de Maou. Para los efectos de sonido (SFX) he utilizado bibliotecas de los siguientes autores: Cleyton Kauffman, Kenney Vleugels, Tuomo Untinen, rubberduck, artisticdude, David Mckee, Little Robot Sound Factory, Leohpaz. Todos estos autores, al igual que mi compañero artista, aparecen en los créditos de El Camino del Alquimista.

3.4 Solución propuesta

El Camino del Alquimista es el nombre de la solución propuesta en este trabajo. Es un videojuego 2D donde el jugador encarna a un joven alquimista novato llamado Nico. El joven protagonista tiene un libro de recetas alquímicas que al principio de la partida está vacío, por lo que para llenarlo deberá adentrarse en peligrosas mazmorras armado con

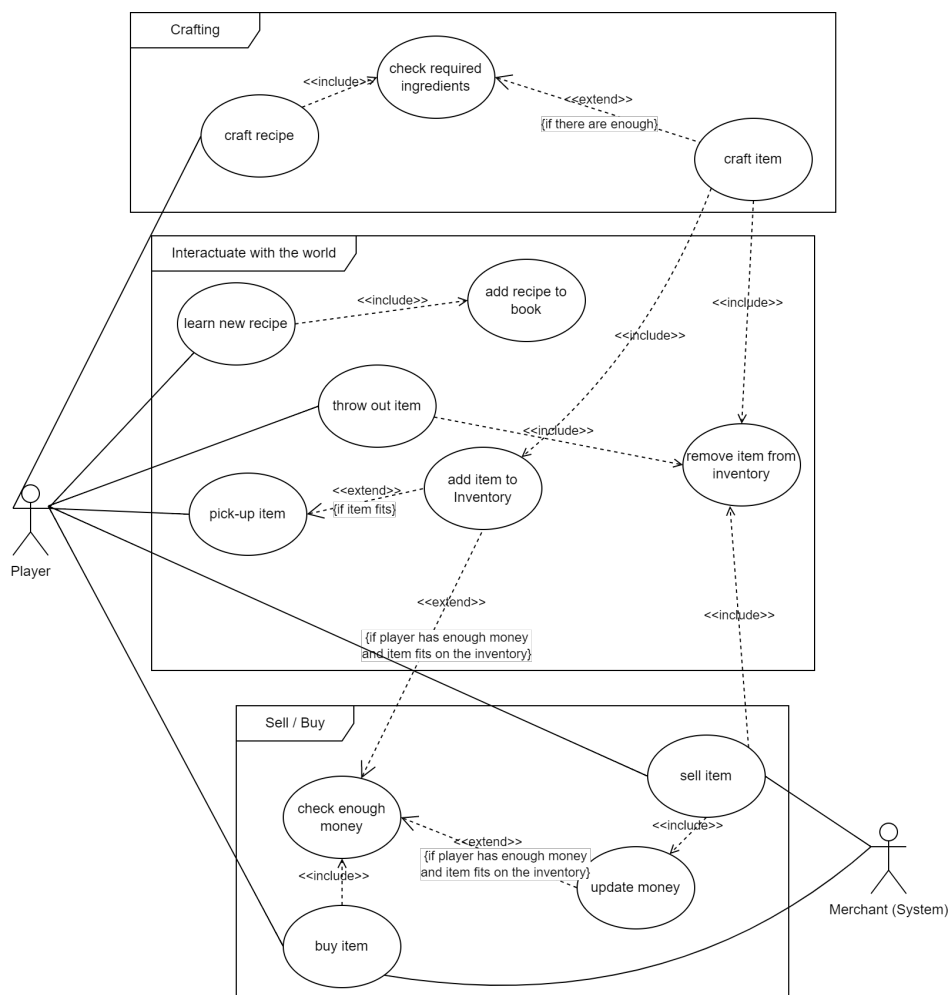


Figura 3.2: Diagrama casos de uso

su lanza, y buscar en los cofres del tesoro para encontrar nuevas recetas. En las mazmorras también se encontrará todos los ingredientes necesarios para fabricar los ítems que indican las recetas, además de agresivos monstruos que le atacarán sin dudar una vez lo detecten. Su objetivo es encontrar la receta del elixir de vida y fabricarlo, para así curar a su hermana enferma. Este juego utiliza las mazmorras como medio para centrarse en la otra mecánica principal que es el alquimia, ya que el jugador las explora para encontrar recetas e ítems.

Una vez el juego tenga sus mecánicas básicas implementadas (las mazmorras se generen correctamente, hayan monstruos inteligentes en ellas, el jugador pueda manipular correctamente su inventario, además de poder fabricar ítems), se realizará una prueba con gente de confianza para conseguir *feedback* de estos primeros jugadores y así mejorar el producto todo lo posible. Una vez hecho esto, el juego se liberará en internet donde todo el mundo podrá jugar.

3.5 Plan de trabajo

Recogidos ya los requisitos funcionales y no funcionales, se redactaron las tareas que conformarían el proyecto. Una vez se realicen todas y cada una de ellas, será cuando el proyecto esté finalizado. Las tareas más costosas se dividieron en subtareas y se les

asignó una prioridad, dependiendo de cómo de urgentes e importantes fueran para conseguir un juego viable. En la Figura 3.3 se pueden ver algunas de las tareas grandes que componen este videojuego. Se puede ver que están ordenadas por prioridad, de más prioritaria (más arriba con una bandera roja) a menos (más abajo con una bandera azul). Además las tareas se etiquetaron para saber a grandes rasgos qué suponían. Estas etiquetas eran: diseño (artístico), realización de *sprites*, dibujo de animaciones, programación, implementación uniendo el arte con la programación para obtener el producto final, mecánicas cuando se crea nuevos funcionamientos, UI referente a las interfaces de usuario y documentación, cuando había que investigar y realizar tareas menos prácticas, cómo podía ser inventar las recetas que serían parte del juego.

Estas tareas se asignarían a uno de los miembros del equipo, realizando mi compañero Tomás todas las tareas que incluyan el diseño artístico de personajes, la realización de *sprites* y sus animaciones. A mí me tocó realizar el resto de tareas referentes tanto a la programación e implementación de nuevas mecánicas como lo relacionado con el diseño del juego, como el diseño de niveles, la invención de los ítems que el juego incluye, además de las recetas para obtenerlos entre otras cosas.

Para poder llevar un seguimiento adecuado del trabajo, se decidió utilizar una aplicación web, donde poder dividir las tareas en diversos sprints de dos semanas. En esas dos semanas el objetivo era realizar las tareas asignadas, de este modo se podía repartir más equitativamente el trabajo y podíamos tener controlado el trabajo tanto nuestro como del equipo entero. Cada sprint tiene tres categorías distintas: 'por hacer', 'en progreso' y 'hechas' como se puede ver en la Figura 3.4. En estas categorías se reparten las tareas según su estado, y así se puede ver en todo momento qué está hecho y qué falta.

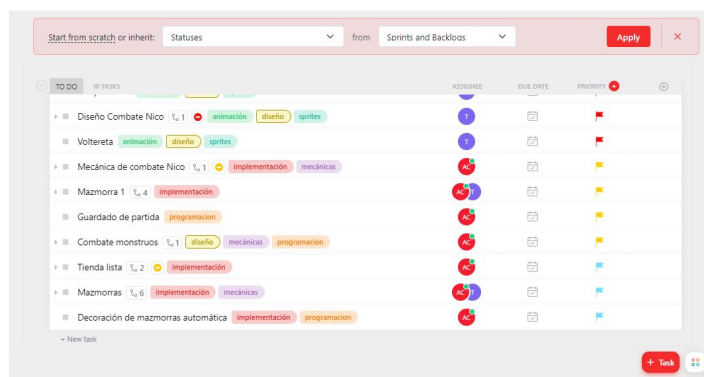


Figura 3.3: Lista de tareas del proyecto

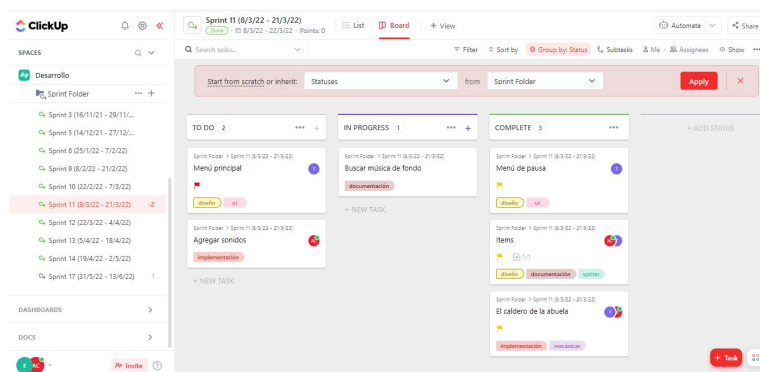


Figura 3.4: Sprint semana 11

CAPÍTULO 4

Diseño

En el análisis hemos detallado cómo se va a llevar a cabo la solución propuesta. A continuación, en este capítulo se detallará el diseño del videojuego en distintos subapartados, cada uno con más nivel de detalle que el anterior.

4.1 Arquitectura del Sistema

En esta sección detallaremos la arquitectura del sistema. El programa está compuesto por distintos módulos que interactúan entre sí como se puede ver en la Figura 4.1. Estos son principalmente dos: la mazmorra y todos los sub-componentes que esta engloba, como el calabozo en sí, los monstruos que lo habitan y la mercader que encierra el sistema de compra-venta del juego; el otro componente principal es el jugador junto a sus distintos inventarios. Estos dos componentes están conectados con la base de datos, está compuesta principalmente de objetos *ScriptableObjects*, estos son unos contenedores de datos del motor Unity. Los datos se almacenan como instancias de clases derivadas de la clase por defecto *ScriptableObject*. Por ejemplo, tenemos la clase *Item* que hereda de *ScriptableObject*, así todos los ítems del juego son instancias de esta clase. De esta manera se puede ahorrar memoria, pues no hay información duplicada, multiples objetos pueden referirse al mismo *ScriptableObject* y así sus datos se almacenan una vez en memoria, independientemente de las veces que sean utilizados.

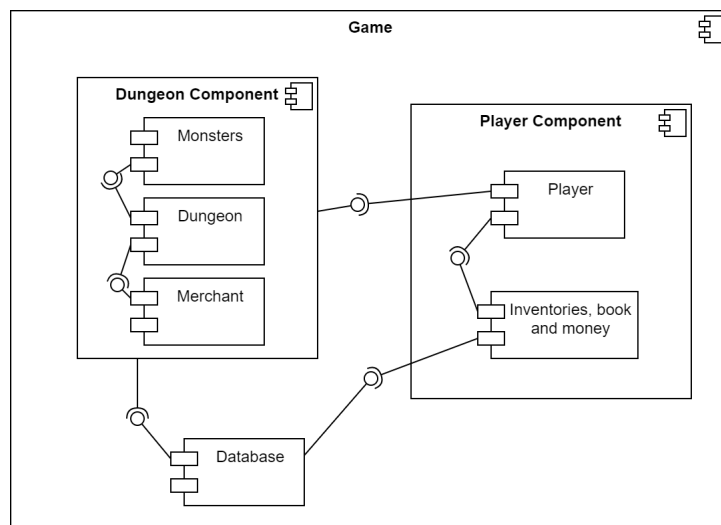


Figura 4.1: Diagrama de componentes UML

Como se puede ver en la Figura 4.1, el componente de la mazmorra se comunica con la base de datos, para consultar los datos de su creación, ya que como hemos dicho en capítulos anteriores se crea de forma procedural a partir de unos parámetros dados. Los parámetros que dirigen la construcción de una mazmorra son: las dimensiones de los pisos de la mazmorra, la cantidad de pisos que tiene, los ítems que se pueden comprar en dicha mazmorra (los cuales venderá la mercader ambulante), las recetas e ítems que se podrán encontrar en los cofres, además del número máximo de estos por piso y qué clase de monstruos habitarán este calabozo. Este es el componente que engloba el mundo del videojuego en sí, se ve en más detalle en la Figura 4.14 donde se especifican las clases que contiene.

El otro componente principal es el jugador, que contiene el sub-componente jugador que engloba los valores del personaje en sí y sus atributos, como sus puntos de vida, su velocidad, su arma y objetos equipados. Este se comunica con la mazmorra, durante el juego. Por otro lado, está los inventarios del jugador, que engloban su inventario de ítems, el inventario de recetas (libro) y su dinero. Todo esto es un mismo componente que se comunica con la base de datos, ya que tiene que consultar los ítems y recetas. Este componente se ve en más detalle en la Figura 4.13 en la siguiente sección.

4.2 Diseño detallado

En esta sección detallaremos el diseño de los distintos componentes del videojuego.

4.2.1. Diseño del juego

En la Figura 4.2 se muestra cómo se relacionan entre sí los distintos niveles o pantallas que componen el videojuego. Como se puede ver, se comienza en la pantalla de menú principal, desde la que se puede acceder a la pantalla de créditos, y desde la que solamente se puede volver al menú. Desde el menú también se puede entrar en el juego, lo que nos lleva a la pantalla del nivel cero, que es la casa del jugador y sus alrededores. Desde ahí, se puede acceder a la primera mazmorra. Si el jugador muere o utiliza un objeto para huir de la mazmorra, este volverá al nivel cero de nuevo. Esto será así en cualquiera de los niveles de las diferentes mazmorras. Si por el contrario, el jugador llega al final de esta sin morir, se encontrará un descansillo, donde podrá elegir si volver al nivel cero o continuar en la siguiente mazmorra. En la siguiente sucedería lo mismo; si el jugador sobrevive hasta el final de la mazmorra dos, habrá un descansillo, donde, si decide continuar, en lugar de volver al nivel cero, llegará a la sala del monstruo jefe. En esta última pantalla del jefe, el jugador sólo puede volver al nivel cero o por el contrario volver al menú principal.

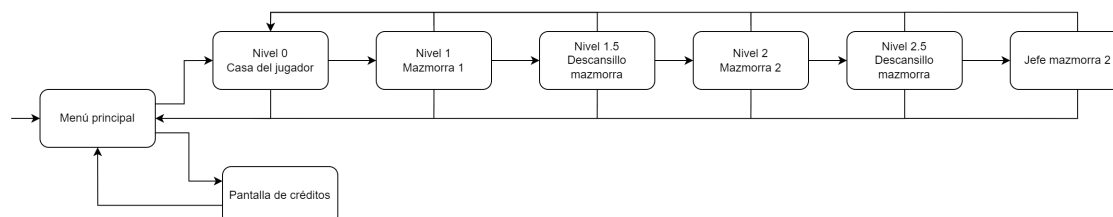


Figura 4.2: Diagrama de navegación de pantallas

El Camino del Alquimista está fuertemente inspirado por títulos de consola portátil en especial juegos RPG para GameBoy Advance (Nintendo, 2001) y Nintendo DS (Nin-

tendo, 2004), pues no se puede usar el ratón del ordenador. Por el contrario, la navegación por los menús y el control del personaje se realiza por teclado o botones si se juega con un mando. Se ha implementado esta interfaz para emular los videojuegos de consolas. También se ve la inspiración en la estética píxel y vista cenital del proyecto, como en títulos de GameBoy Advance como Golden Sun (Camelot Software Planning, 2001) en la Figura 4.3 o de Nintendo DS como Inazuma Eleven (Level-5, 2008) en la Figura 4.4. En esta consola los escenarios comenzaban a ser modelos 3D, pero los *sprites* de los personajes continuaban siendo *pixelart*. El movimiento por el mundo es en los ejes x e y, como en gran número de RPGs.



Figura 4.3: Golden Sun, Camelot Software Planning, 2001



Figura 4.4: Inazuma Eleven, Level-5, 2008

4.2.2. Diseño del personaje

Para diseñar el personaje tomamos inspiración de The Legend of Zelda: The Minish Cap (Flagship y Capcom, 2004). Como se puede ver en la Figura 4.5, en este título el sistema de puntos de vida del jugador se mide en corazones, donde cada corazón está compuesto de cuatro fragmentos. Cuando el jugador pierde todos sus corazones, muere.

El manejo del personaje también está inspirado en este título. Las acciones que puede realizar el protagonista son atacar con su lanza, bloquear con la lanza ataques de los enemigos y rodar para esquivar los monstruos, ataques. Este movimiento también se

puede utilizar para ir más deprisa. Esta última es una mecánica característica de los juegos de The Legend of Zelda. Otras acciones que puede hacer son manejar el inventario, pero esto ya no es parte del *Gameplay* sino de la UI.



Figura 4.5: The Legend of Zelda: The Minish Cap, 2004

El jugador podrá equipar objetos para usarlos de forma rápida, con un atajo. Estos objetos equipados se limitarán a tres diferentes y se podrá ver en pantalla en todo momento cuáles están equipados y cuál de todos está seleccionado. Como se puede ver en la Figura 4.6, se indica qué teclas se utilizan para usar el objeto seleccionado y para cambiarlo por otro de los equipados. Se puede ver que al igual que en el título de Zelda, en El Camino del Alquimista está indicado en todo momento en pantalla qué botón hace cada cosa, lo que facilita al usuario el aprender a jugar.

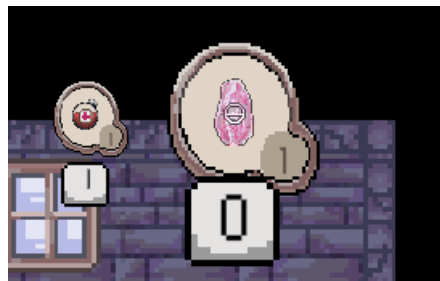


Figura 4.6: Objetos equipados, imagen del Camino del Alquimista

4.2.3. Diseño de la mazmorra

El elemento clave dentro del juego son las mazmorras, pues son las que explora el jugador, y donde transcurre la mayor parte del juego, por lo que el diseño de estas es crucial para el resultado final del proyecto. A la hora de diseñarlas, se barajaron distintas opciones. Por una parte, se pensó en crear habitaciones prediseñadas con posibles puertas en las cuatro direcciones cardinales: norte, sur, este, oeste. Este tipo de mazmorras se puede ver en videojuegos como The Binding of Isaac (Edmund McMillen, 2011) y tiene dos variaciones: sin pasillos, donde cada puerta se une con otra puerta como se ve en la Figura 4.7. De esta manera, para que dos habitaciones encajen, tienen que tener una puerta en el lado contrario, por ejemplo, una habitación con una puerta en el este sólo puede tener a su lado derecho una sala con una puerta en el oeste. La otra variante es la que se puede ver en la Figura 4.8, donde en las puertas se coloca un pasillo, horizontal o vertical dependiendo de la orientación de la puerta, que puede ser un callejón sin salida o

incluso conectar con otro pasillo en lugar de con una habitación. El tener las habitaciones ya prediseñadas permite agregarles una decoración mejor, pues se diseña con la sala, en lugar de ser colocarse por un algoritmo aleatorio.

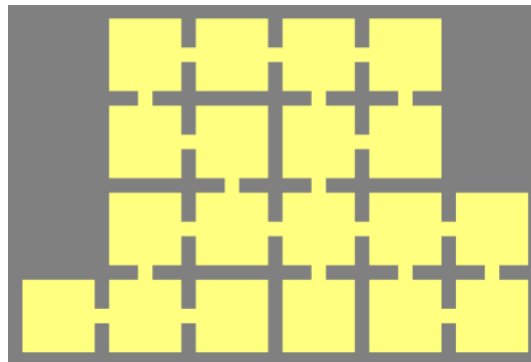


Figura 4.7: Mazmorra sin pasillos

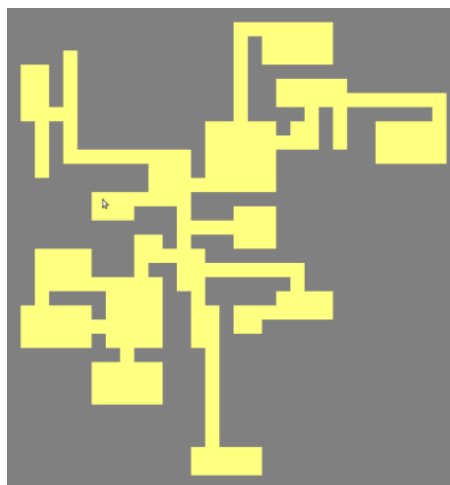


Figura 4.8: Mazmorra con pasillos

La otra opción que había para generar las mazmorras, la opción por la que nos decantamos finalmente, es utilizar un algoritmo conocido como el algoritmo de la 'Caminata aleatoria', en concreto la caminata aleatoria en cuadrículas. Una caminata aleatoria es un proceso aleatorio donde la posición de una partícula en un cierto instante depende solo de su posición en el instante previo y algún parámetro aleatorio que determina la longitud y dirección de su siguiente paso. En la variante de la cuadrícula, las posibles direcciones se ven restringida a los movimientos de la cuadrícula, como se ve en la Figura 4.9. Se puede tener en cuenta cuatro u ocho diferentes vecinos, dependiendo se quiere tener la posibilidad de moverse en diagonal.

Se ha escogido este algoritmo para la generación de las mazmorras porque es un algoritmo que genera resultados más naturales, que es lo que se buscaba para el proyecto, ya que las mazmorras en El Camino del Alquimista son paisajes naturales como bosques, montañas... Así que se optó por este algoritmo, utilizando más de un caminante a la vez para hacer los calabozos un poco más laberínticos. En la Figura 4.10 se puede ver cómo sería una mazmorra generada con este algoritmo. Al ser una generación totalmente aleatoria, la decoración y el resto de elementos de esta, son también generadas al azar, por lo que este tipo de mazmorras suelen ser menos atractivas visualmente. El algoritmo del

paseo aleatorio se verá con más detalle en el capítulos siguiente.

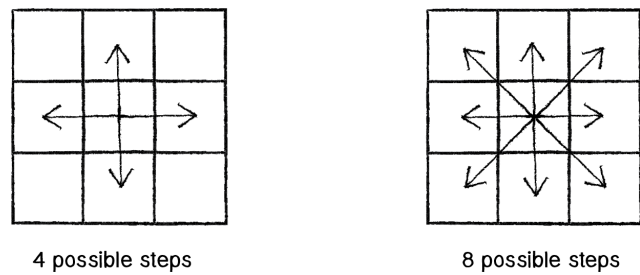


Figura 4.9: Pasos disponibles en una cuadrícula

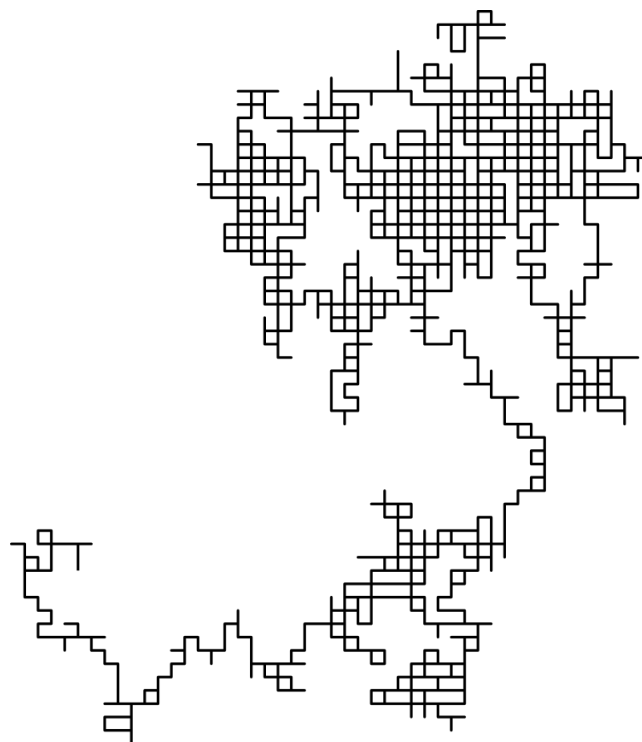


Figura 4.10: Mazmorra generada con el algoritmo de la caminata aleatoria, imagen obtenida de la página de la Wikipedia en inglés [Random Walk](#)

4.2.4. Diseño de la IA de los enemigos

Para diseñar la inteligencia artificial de los monstruos del juego se ha implementado una máquina de estados. En el videojuego se pueden encontrar dos comportamientos de enemigos distintos.

Por una parte, están los enemigos que atacan cuerpo a cuerpo. El comportamiento de estos se puede ver en la máquina de estados de la Figura 4.11. Estos monstruos deambulan hasta que detectan al jugador, entonces lo persiguen para atacarlo cuando estén a una distancia adecuada.

Por otra parte, están los enemigos que atacan a distancia, disparando proyectiles al jugador. El comportamiento de estos se muestra en la Figura 4.12. Básicamente estos mons-

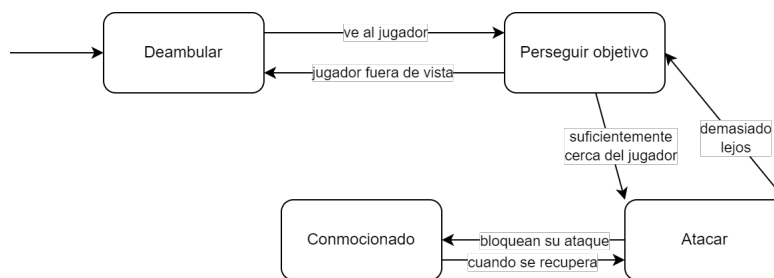


Figura 4.11: Máquina de estados, IA enemigos cuerpo a cuerpo

truos deambulan hasta encontrar al jugador. Una vez detectado, si están la distancia adecuada, comienzan a lanzarle proyectiles. Si el jugador se aleja de su campo de visión, paran de disparar y comienzan a deambular de nuevo. Si, por el contrario, el jugador se acerca demasiado a estos monstruos, entran en pánico y comienzan a disparar proyectiles con mucha mayor frecuencia.

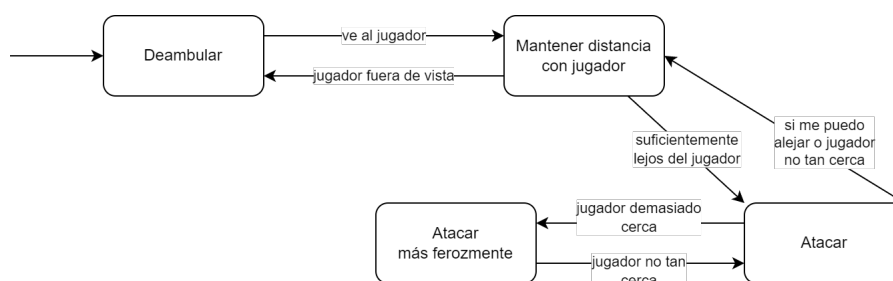


Figura 4.12: Máquina de estados, IA enemigos a distancia

4.2.5. Diseño de clases

Una vez definida la arquitectura del sistema, lo siguiente es detallar especificando las clases que tendrán, usando los diagramas de clases UML. En la Figura 4.14 se puede ver el diagrama de clases del componente de la mazmorra o *dungeon* en inglés. Se puede ver que está compuesta de niveles o pisos de mazmorra, y estos son los que están relacionados con el resto de clases del diagrama. Por otra parte, en la Figura 4.13 está representado el jugador, sus atributos e inventarios.

4.3 Tecnologías utilizadas

En último lugar, hablaremos de las tecnologías utilizadas para el desarrollo del proyecto de este TFG.

4.3.1. Unity

Comenzamos la sección con el motor de videojuegos utilizado, Unity. Este motor es ampliamente usado tanto por la industria de videojuegos, principalmente en proyectos de empresas independientes, al igual que por particulares y no profesionales. Muchos videojuegos han sido desarrollados con este motor, algunos de los más populares son: Cuphead (Studio MDHR, 2017), Hearthstone (Blizzard Entertainment, 2014) o el fenó-

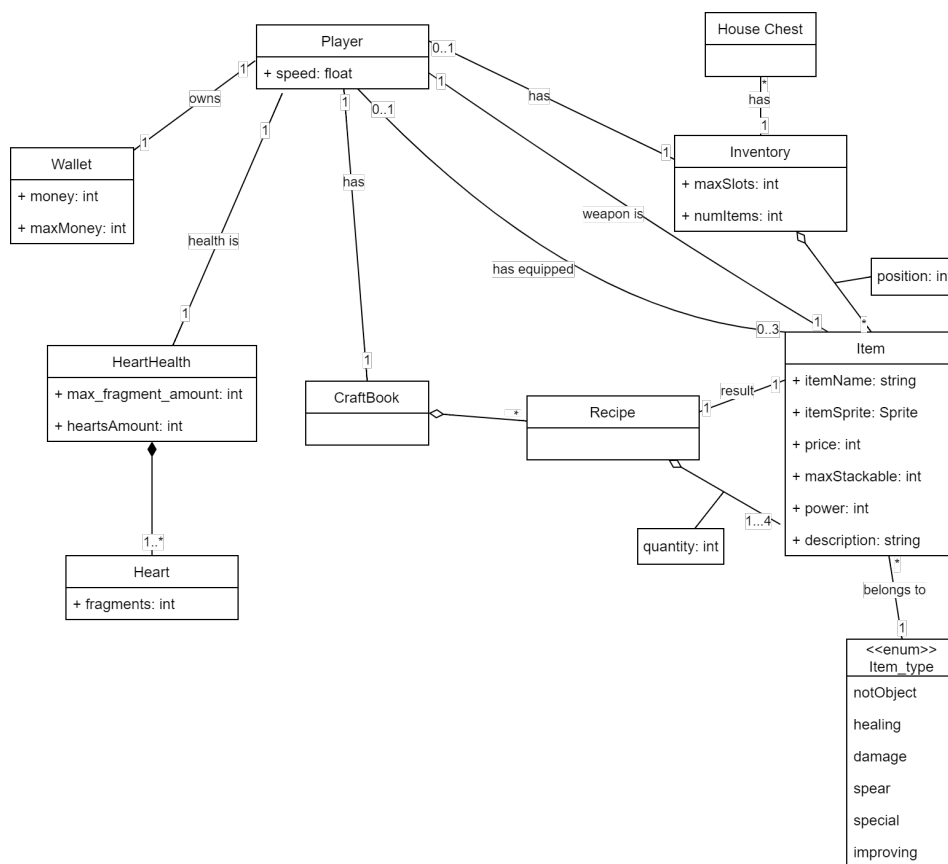


Figura 4.13: Diagrama de clases UML del jugador

meno Pokémon GO (Niantic, 2016).

El lenguaje de programación utilizado en Unity es C# que es un lenguaje multiparadigma, pues es estructurado, imperativo, dirigido por eventos, orientado a objetos, funcional, genérico y reflexivo. Este lenguaje fue desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. [11]

Una de las ventajas de Unity es que engloba muchos motores, como un motor de renderizado de imágenes, un motor de físicas 2D y otro de físicas 3D, motor de audio, motor de animaciones, además de herramientas para soportar multijugador tanto local como en línea, herramientas de navegación para inteligencia artificial y más. Por lo que con Unity tienes todo lo necesario para hacer un videojuego. Asimismo, Unity dispone de una documentación muy amplia tanto en su [manual de usuario](#) como en sus [APIs](#), esto es realmente útil a la hora de desarrollar un proyecto, ya que ayuda a solucionar los posibles problemas con mayor facilidad.

El editor de Unity se puede ver en la Figura 4.15. Está dividido en diferentes ventanas, cada una con una utilidad. A continuación explicaremos las más importantes para el desarrollo de un proyecto.

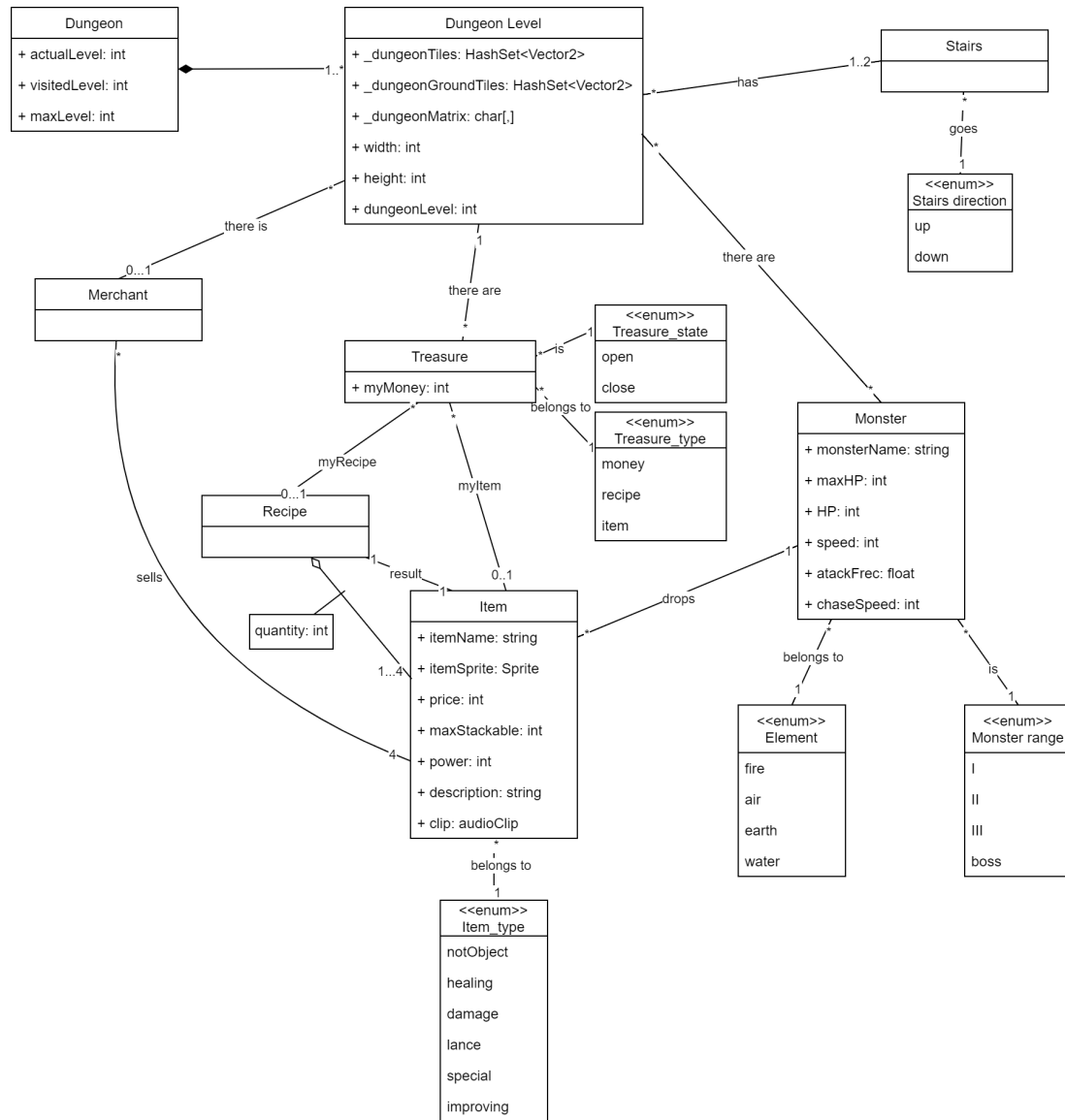


Figura 4.14: Diagrama de clases UML de las mazmorras

Ventana de Jerarquía

Primero tenemos la jerarquía (*Hierarchy*) en la parte izquierda de la pantalla que se puede ver en la Figura 4.16. En esta se puede ver, como indica su nombre, los elementos y relaciones de los *GameObjects*, que son como se conoce en Unity a los distintos elementos que están en el juego, como puede ser el jugador, la cámara o el escenario. Aquí se representa el parentesco de los distintos elementos. Así, un *GameObject* puede ser padre o hijo de otro. esta relación se muestra con las sangrías de los nombres de los diferentes objetos. Por ejemplo, en la Figura se ve el objeto 'Points' que tiene tres objetos hijos 'TelePoint (1)', 'TelePoint' y 'SpawnPoint'. A su vez, todos los objetos son hijos de la escena 'House'. Desde esta ventana se pueden crear y eliminar objetos, además de editar su relación.

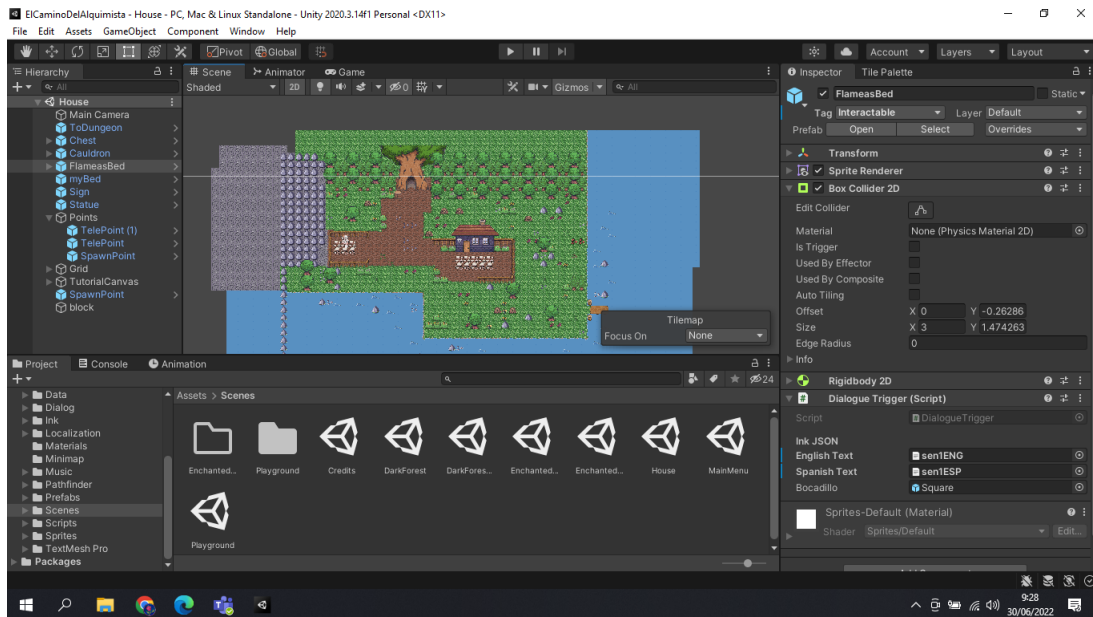


Figura 4.15: Editor de Unity

Vista de Escena

En segunda instancia está la vista escena (*Scene*), que se puede ver en la Figura 4.17. Esta vista permite editar la escena, además de que todos los elementos de la jerarquía aparecen en la escena y viceversa. Se puede seleccionar cualquier elemento del juego ya sea la cámara, el jugador u otro *GameObject* para poder manipularlo, moverlo... esta es una de las funciones más importantes de Unity, por lo que esta ventana es esencial.

Ventana de Proyecto

En tercera instancia tenemos la ventana de proyecto (*Project*). Como se aprecia en la Figura 4.18, en esta vista se puede acceder y gestionar los distintos *assets* del proyecto como materiales, modelos, *sprites*, audios, *scripts*... Los *assets* se representan con un icono con su tipo. En el panel de la izquierda se ven los distintos directorios del proyecto en una vista jerárquica. Con click derecho/*Create* se puede crear cualquier *asset* dentro de la capeta que queramos.

Ventana del inspector

En cuarta instancia tenemos la ventana del inspector, uno de los elementos más importantes del editor de Unity. Con el inspector se puede ver y editar las propiedades, preferencias u otros ajustes de un *GameObject*. Cuando se selecciona un *GameObject* ya sea en la jerarquía como en la escena, el inspector muestra sus propiedades de todos sus componentes y permite editarlas, así como los materiales del objeto. Cuando un *Script* personalizado es componente del elemento seleccionado, las variables públicas de este se podrán ver y cambiar desde esta ventana. Esto permite cambiar los valores de las variables sin necesidad de cambiar el código. El inspector también se puede utilizar para inspeccionar los *assets*.

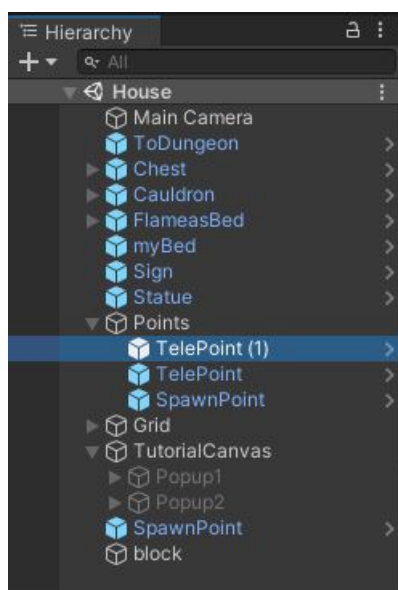


Figura 4.16: Ventana de jerarquía de Unity

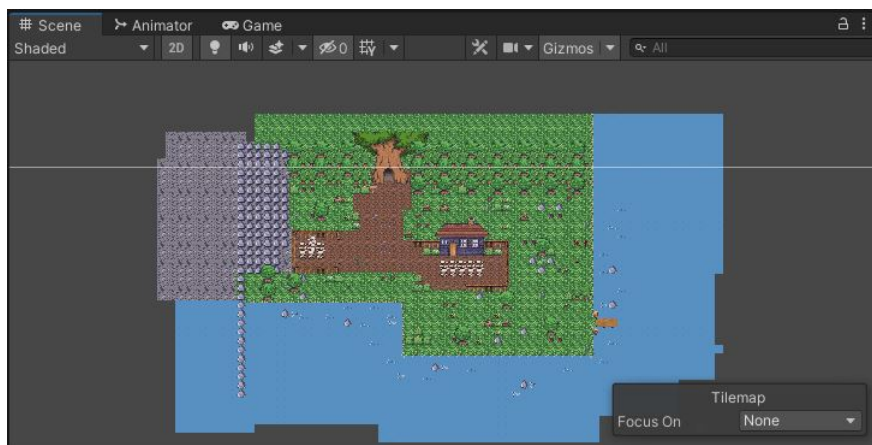


Figura 4.17: Vista de escena de Unity

Vista de Juego

En último lugar, tenemos la vista del juego, que se renderiza por la(s) cámara(s) del juego. Esta es la representación del juego finalizado, aquí se puede jugar y ver cómo será una vez exportado. Para comenzar a jugar en esta vista, es necesario pulsar el botón *play* de la barra de herramientas. Durante la ejecución se puede pausar el juego con el botón pertinente, y así inspeccionar los distintos elementos o ver distintos ángulos desde la ventana de escena.

Scripting

Una vez explicados todos los elementos esenciales del editor de Unity, ahora toca hablar de la programación de *scripts*, que son los ficheros C#. Estos son una parte fundamental de cualquier juego, pues incluso el juego más simple los necesita. Los *scripts* se utilizan para todo, ya sea conseguir controlar los comportamientos físicos de un objeto, hacer que el juego reaccione a las instrucciones del jugador, cambiar de escena, imple-

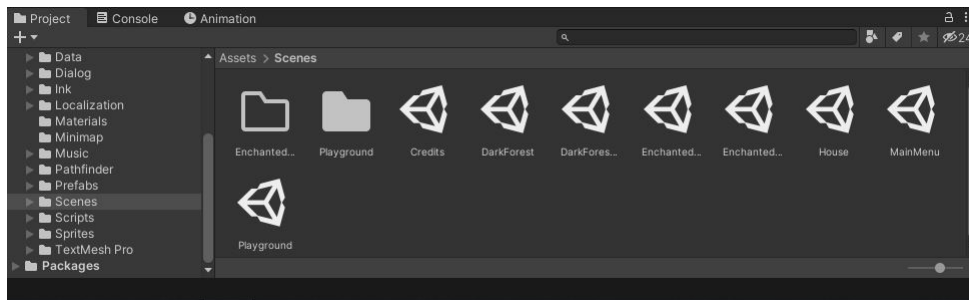


Figura 4.18: Ventana de proyecto de Unity

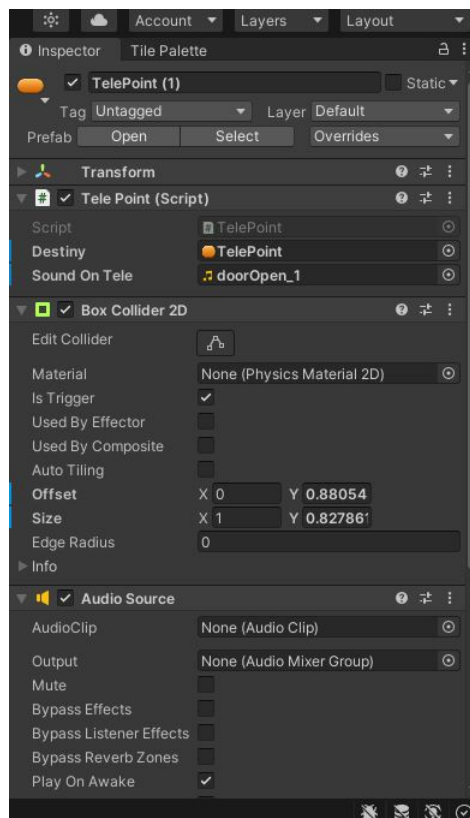


Figura 4.19: Ventana de inspector de Unity

mentar la inteligencia artificial de los personajes, controlar los menús... Básicamente la programación de estos son la base de cualquier juego. [12]

Prefabs

Es importante comentar un elemento muy importante en Unity, los *Prefabs*. Estos sirven para reutilizar un *GameObject* varias veces en una o más escenas. En lugar de copiar y pegar varias veces el objeto deseado, se crea un *Prefab*, que es una plantilla del objeto. La ventaja de estas plantillas es que cualquier cambio realizado en ella, se verá reflejado en todas sus instancias. Por el contrario, un cambio realizado en una instancia no cambia la plantilla.



Figura 4.20: Ventana de juego de Unity

4.3.2. GitHub

GitHub es un sistema de control de versiones de Git.

Un programa de control de versiones registra todos los cambios realizados en el código, por lo que se puede ver de un vistazo cómo era el código en una versión anterior. Muy útil en caso de que algo que antes funcionaba, de repente en una versión más reciente ya no va. Así se puede ver los cambios y saber cuál a sido el problema. Además, también es bueno para llevar un seguimiento del proyecto, pues en la aplicación web se puede ver un gráfico con la cantidad de versiones realizadas a lo largo de una línea temporal.

Si el proyecto se realiza en grupo, esta herramienta es esencial, pues permite un mayor control en los cambios realizados por distintas personas, para lograr evitar en la medida de lo posible que el trabajo de uno sobrescriba el del otro. Sin olvidar la posibilidad de creación de ramas de un mismo proyecto a partir de una versión base. Cada rama podría tener unos cambios y trabajar independientemente. Una vez terminado el trabajo, la rama se puede volver a unir al proyecto principal siempre con el control de unión (*merge*) que proporciona el programa para evitar problemas. Esto también puede ser útil en un proyecto individual, para trabajar en distintas partes del proyecto en cada rama sin miedo a estropear el proyecto principal. [13]

CAPÍTULO 5

Desarrollo de la solución

5.1 Comienzo del proyecto

La idea del proyecto surgió en el comienzo del curso 2021/2022. Se quería adaptar el juego de mesa Alquimia, mencionado en anteriores capítulos, a un videojuego 2D desarrollado en Unity. Durante el primer semestre, cursando la asignatura Desarrollo de videojuegos 3D (11645), hablé con el profesor del proyecto, quién, al ser el proyecto un videojuego 2D, me recomendó ponerme en contacto con Paco Abad, profesor de Desarrollo de videojuegos 2D (14101). Paco aceptó tener una reunión en la que le presenté la idea del proyecto y le propuse ser mi tutor de TFG, y él aceptó amablemente.

Le comenté a mi tutor que me interesaba tener arte original para mi videojuego, y él me puso en contacto con un alumno del grado de Tecnologías Creativas. Desgraciadamente, el alumno y yo no conseguimos llegar a un acuerdo, así que decidimos no trabajar juntos. Decidida a encontrar un artista, realicé unos carteles que pegué por la Facultad de Bellas Artes para buscar un artista para desarrollar el videojuego. Además, me puse en contacto con una profesora de la Facultad de Bellas Artes, que impartía asignaturas relacionadas con la animación 2D y le expliqué mi situación y qué buscaba. Esto último dio sus frutos, pues ella preguntó a sus alumnos si había alguno interesado en colaborar en el proyecto. Uno de ellos era Tomás, con quién acabé realizando este videojuego.

Una vez creado el equipo de trabajo, comenzamos a planear cómo se desarrollaría el juego, cuantos niveles tendría, cómo funcionaría el sistema de combate... Se concretaron todos estos aspectos en el GDD. También mi compañero diseñó los primeros bocetos de nuestro protagonista del videojuego: Nico.

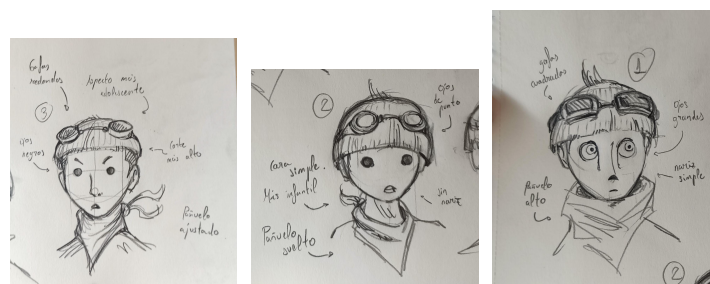


Figura 5.1: Bocetos de Nico, realizados por mi compañero Tomás

5.2 Desarrollo enfocado al usuario

El Camino del Alquimista, fue inicialmente planteado como un juego solo disponible en español. Posteriormente, se decidió desarrollarlo con distintos idiomas, dando al usuario la posibilidad de jugar tanto en castellano como en inglés, para llegar a una mayor cantidad de público. Así, se añadió la posibilidad de cambiar el idioma en el menú principal antes de jugar, cambiando así el texto de los elementos del juego. Además, por comodidad para el jugador, el guardado de partida es automático. Este nunca se realizará dentro de las mazmorras.

Por otra parte también se introdujo la posibilidad de elegir el control del juego, ya sea usando el teclado o conectando un mando al ordenador para poder disfrutar de un control más cómodo. La posibilidad de utilizar un mando es algo que se pensó desde las primeras ideas del juego, ya que se buscaba un juego que recordara a los títulos de GBA y NDS, consolas que se controlaban principalmente por botones pues sólo la segunda disponía de pantalla táctil. Es necesario resaltar, que el control por mando está diseñado de tal forma que se parezca a los juegos de Nintendo, ya que estos han sido la principal inspiración. Se puede ver que el botón de aceptar al usar el mando es el este (A), y el de cancelar es el sur (B). Algo que en los juegos de otras plataformas como PlayStation o XBOX, estos controles están intercambiados. Los requisitos cumplidos en el desarrollo de este subapartado se pueden ver en la Tabla 5.1.



Figura 5.2: Mando de Super Nintendo Entertainment System

Part1
Part2
Part3
Part4

Tabla 5.1: Requisitos cumplidos en el desarrollo enfocado al usuario

5.3 Desarrollo del personaje jugador

El jugador se desarrolló teniendo en mente que portaría una lanza como arma en todo momento. Se decidió que se pudiera mejorar la lanza, a través de la alquimia y la creación de nuevas lanzas. Así, se optó por hacer de la lanza un *GameObject* separado del objeto

jugador, pudiendo cambiar así su *sprite* cuando fuera necesario sin necesidad de cambiar el del jugador.

Se creó una clase lanza donde se almacenarían los valores de la lanza, como su aspecto y su poder de ataque. Estos atributos se almacenan en el objeto ítem correspondiente a cada lanza. De esta manera, se puede controlar de manera más precisa los distintos movimientos del jugador realizados con la lanza, como atacar y defenderse, pues los bloqueos también se realizan con dicha arma.

Además de la lanza, se pensó que el jugador pudiera equipar ítems con función para utilizarlos rápidamente en los combates. Pues la forma normal de utilizarlos es desde el inventario, lo que no es lo más cómodo en los combates de acción que tiene nuestro videojuego. Así que se implantó un sistema de equipamiento. Para ello, desde el menú de inventario se implementó la opción “equipar”. Con ella, se pueden equipar hasta tres ítems distintos. Estos se visualizarán en la pantalla del jugador indicando cuál de los ítems equipados está seleccionado, pudiendo rotarse. El ítem seleccionado es el que se usará cuando el jugador aprente el botón indicado. Este menú de ítems equipados se puede ver en la Figura 5.3.

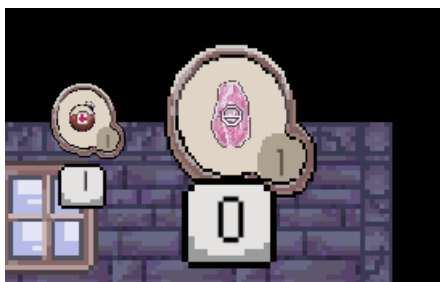


Figura 5.3: Objetos equipados, imagen del Camino del Alquimista

Por otra parte, el comportamiento del jugador se realizó mediante estados, permitiendo así con mejor control de sus movimientos. Los estados del jugador son tres: normal, atacando e inmune. Atacando es el estado al que pasa cuando realiza los movimientos con el arma, ya sea atacar o defender. Cuando el jugador está inmune no puede recibir daño, y entra en dicho estado cuando rueda por el suelo, logrando esquivar de esta manera los ataques de los enemigos o sortear trampas. También es inmune durante un breve momento después de recibir daño. El control de estados permite, por ejemplo, que el jugador sólo se pueda mover cuando esta en el estado normal, impidiendo que ataque y se mueva a la vez. Los requisitos cumplidos en el desarrollo de este subapartado se pueden ver en la Tabla 5.2.

| |
|--------|
| Equip1 |
| Equip2 |
| Equip3 |
| Equip4 |
| Equip5 |
| Com1 |
| Com5 |
| Com8 |

Tabla 5.2: Requisitos cumplidos en el desarrollo del jugador

5.4 Implementación de la alquimia e inventario

5.4.1. Lógica

Lo primero que se implementó en el proyecto, fue el sistema de ítems, recetas e inventarios. Los ítems que dispongan de algún uso más allá de ser un ingrediente de otra receta, pueden ser equipados para que el jugador los use en cualquier momento, o bien ser usados desde el inventario. Las recetas contienen las cantidades e ingredientes necesarios para crear el ítem de dicha receta además dicho ítem resultado.

Se realizó la lógica del inventario de ítems que funciona por huecos y tiene una capacidad máxima. En cada hueco sólo puede haber un tipo de ítem, ya sea una unidad o una pila de estos, hasta un máximo que varía dependiendo del ítem. Las pilas de ítems tienen asignadas una posición en el inventario, que se puede cambiar si se desea. Cuando el inventario esté lleno, todos sus huecos están ocupados por un montón o pila de un ítem (ya sea un montón de una sola unidad), no se podrá añadir más ítems al inventario. Sin embargo, si el tipo de ítem que se desea añadir ya está en el inventario, este se podrá añadir si los montones de este no están llenos, no llegan al máximo de ítems apilables. También es posible eliminar ítems del inventario, además de juntar las pilas del mismo ítem si se desea, siempre y cuando no se supere el máximo apilable.

Después se realizó la lógica del inventario de recetas, conocido como libro de recetas. Aquí sólo se puede añadir una receta siempre y cuando esta no se encuentre ya en el libro. No es posible eliminar recetas del libro. Se pueden ordenar las recetas tanto por orden alfabético como por la clase del ítem que fabrican, ya sea un ítem de curación, uno de material o uno de bomba. También se puede pedir la lista de todas las recetas como solamente de las recetas que se tengan de una clase específica.

Una vez se tenía esto, se desarrolló el sistema de alquimia, que necesitaba del sistema de inventario, para ver si se poseen los ingredientes necesarios, así como del libro de recetas, donde consultar la receta deseada. Este sistema se implementó como parte de la clase del libro de recetas. Se añadió un método a dicha clase para consultar si la receta deseada se puede fabricar. Este método consulta el inventario y comprueba la cantidad disponible de los ingredientes de la receta, devolviendo el máximo de ítems que se pueden fabricar de dicha receta. Devuelve cero si no hay ingredientes suficientes en el inventario. También se añadió otro método con el que fabricar la receta, que elimina los ingredientes usados del inventario y le añade el ítem creado en la cantidad decidida por el jugador. Los requisitos cumplidos en el desarrollo de este subapartado se pueden ver en la Tabla 5.3.

| |
|-------|
| Item1 |
| Item2 |
| Item3 |
| Item4 |
| Rece1 |
| Rece2 |
| Rece3 |

Tabla 5.3: Requisitos cumplidos en el desarrollo de la lógica de ítems, recetas e inventarios

5.4.2. Inventario de ítems

Una vez definida la lógica, se implementó los controladores para interactuar con ella mediante interfaces de usuario. Se creó dos interfaces para los dos inventarios de ítems. Una para la bolsa, este inventario es de un tamaño reducido. El jugador puede acceder a ella en cualquier momento. Y otra para el baúl, este inventario tiene una capacidad mayor a la bolsa, pero sólo se puede acceder a este desde la casa del jugador. Los ítems que estén en esta bolsa y tengan alguna función, ya sea una poción que restaura puntos de vida o una bomba, podrán ser equipados. Con un máximo de tres ítems equipados al mismo tiempo. Cuando el jugador muere, sólo se pierden los ítems de la bolsa que estén fuera del llamado bolsillo seguro, que son los huecos de un color más oscuro que el resto, estos son los que componen dicho bolsillo.

El controlador del inventario dispone de una variable “tipo”, que le indica si es la bolsa o el baúl. El baúl dispone de algunas funcionalidades exclusivas. Ambas interfaces muestran el nombre y descripción del objeto seleccionado. La descripción es especialmente útil cuando el objeto seleccionado tiene algún uso como puede ser un objeto de curación, ya que aquí se indica la cantidad de puntos de vida que restaura dicho ítem. La Figura 5.5 muestra la descripción de la “Sal rosa”, que restaura un corazón. Como se puede ver en la Figura 5.5, la interfaz del baúl muestra además el contenido de la bolsa, esto es para que el jugador pueda transferir objetos de un inventario al otro. Los requisitos cumplidos en el desarrollo de este subapartado se pueden ver en la Tabla 5.4.



Figura 5.4: Interfaz de la bolsa en el videojuego

| |
|--------|
| Bolsa1 |
| Rece2 |
| Rece3 |
| Rece4 |
| Rece5 |
| Baul1 |
| Baul2 |
| Baul3 |
| Baul4 |

Tabla 5.4: Requisitos cumplidos en el desarrollo de las UI de los inventarios de ítems



Figura 5.5: Interfaz del baúl en el videojuego

5.4.3. Libro de recetas y sistema de alquimia

El controlador del libro de recetas se reutilizaría para el sistema de alquimia añadiéndole la posibilidad de crear la receta seleccionada. Se decidió usar la metáfora de un libro de recetas y un caldero, para hacer este menú. Se realizaron distintos bocetos de distribución de los componentes necesarios a mostrar al jugador. Primero se pensó el boceto de la Figura 5.6 donde al abrir el menú se seleccionarían la pestaña que se deseara, ya sea para visualizar todas las recetas o sólo una sección de ellas. Entonces, se abriría una ventana con todas las recetas, y aquí se seleccionaría una receta para visualizar sus ingredientes y fabricarla. Este proceso resultó ser un poco largo y complicado, así que se simplificó en el segundo boceto que se puede ver en la Figura 5.7.

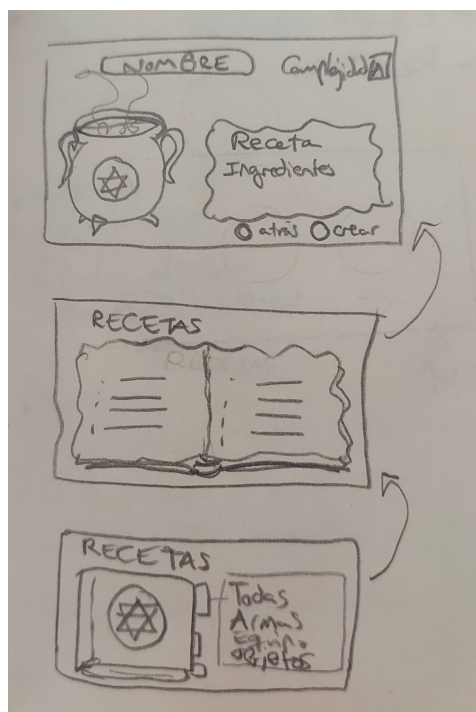


Figura 5.6: Primer boceto para el libro de recetas

En este diseño, se pensó utilizar una página del libro para visualizar las recetas y la otra para ver los ingredientes necesarios. Se mostrarían los ingredientes de la receta donde este el navegador, evitando así tener que seleccionarla para visualizarla como ocurría

en el otro boceto. Además, para pasar de una pestaña a otra no haría falta ir a otra ventana. Este boceto es el que se ha seguido para la implementación final que se puede ver en la Figura 5.8.

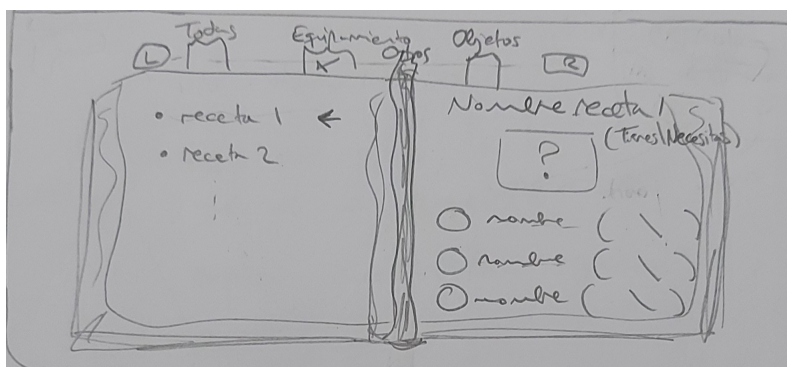


Figura 5.7: Segundo boceto para el libro de recetas

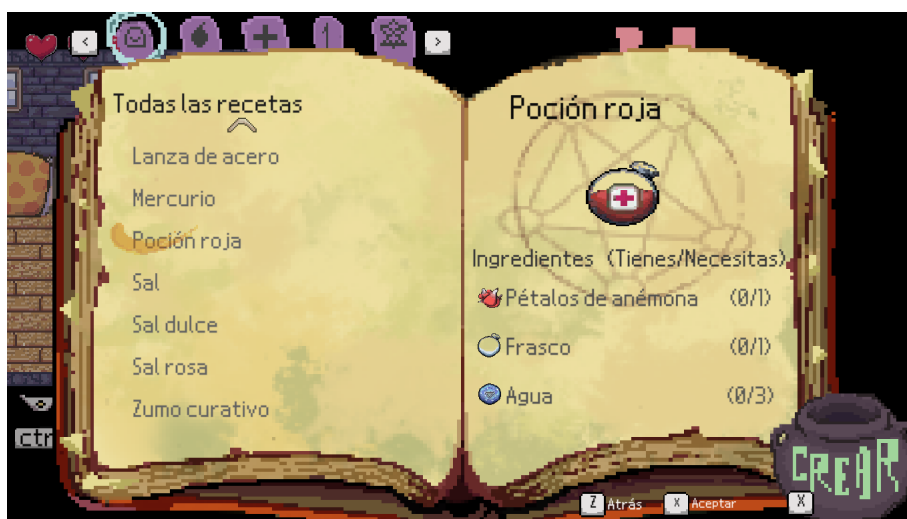


Figura 5.8: Libro de recetas implementado en el juego

Como se ve en la Figura 5.8, hay una imagen de un caldero con la palabra “crear”. Este elemento de la interfaz se muestra solamente si el jugador interactúa con el objeto caldero de su casa, que es la estación de fabricación que utilizará para poder fabricar los objetos y pociones que desee. Cuando el jugador se encuentre en las mazmorras podrá consultar el libro de recetas para ver qué recetas dispone, pero no podrá crear ninguna. Para indicar que no se puede fabricar nada, esta imagen se desactiva.

En la Figura 5.9 se ve cómo funciona el sistema de alquimia. Cuando se tienen los ingredientes necesarios para crear la receta seleccionada, el icono de crear cambia. Este se ilumina, como se puede ver en la imagen de la derecha de la Figura 5.9 indicando que se puede fabricar. Si el jugador quiere crear dicha receta, solo tiene que apretar la tecla indicada y aparece un *pop-up* donde el jugador indica la cantidad que quiere fabricar. Una vez elige la cantidad, aparecerá un mensaje de confirmación del ítem que se ha creado. Otra cosa a destacar en la interfaz del libro de recetas es que sólo se puede ver el aspecto de un ítem que se haya fabricado antes. Como se puede ver en la Figura 5.9, la imagen de la ‘Granada ígnea’ aparece en la página derecha sólo una vez se ha fabricado. Los requisitos cumplidos en el desarrollo de este subapartado se pueden ver en la Tabla 5.5.



Figura 5.9: Sistema de alquimia del videojuego

Rece5
Alq1
Alq2
Alq4
Libro1
Libro2
Libro3
Libro4

Tabla 5.5: Requisitos cumplidos en el desarrollo del libro de recetas y el sistema de alquimia

5.5 Desarrollo del sistema compra-venta

El sistema compra-venta del videojuego está gobernado por la vendedora ambulante. No es posible comprar ni vender ítems sin esa PNJ. Ella aparece de manera aleatoria dentro de las mazmorras y siempre vende cuatro ítems diferentes. Los ítems pueden cambiar de un encuentro a otro, pues tiene varios productos que va cambiando en cada aparición. Algunos de los ítems que vende, son sólo obtenibles comprándoselos a ella. El conjunto de posibles ítems disponibles en la tienda cambian de una mazmorra a otra.

La interfaz del sistema de compra se puede ver en la Figura 5.10, donde se muestran los cuatro ítems que vende la PNJ. Para cada artículo, se muestra el precio, el nombre y el aspecto. Además, también se ve el dinero que el jugador tiene disponible para comprar, que nunca podrá ser menor a cero o mayor a cien. Al seleccionar el ítem deseado, aparece un *pop-up* donde el jugador puede seleccionar la cantidad que desea comprar, permitiendo así comprar varias unidades al mismo tiempo, siempre y cuando se disponga de monedas suficientes.

Por otra parte, el sistema de venta se ha realizado como una función más del controlador del inventario. Al querer vender, se abre la interfaz de la bolsa que hemos visto más arriba. La diferencia es que en cada hueco de la bolsa, aparece el precio de venta del ítem que está en dicho hueco, además de que al seleccionar un ítem aparece la nueva opción de vender. Este precio es siempre menor al precio de compra. Los requisitos cumplidos en el desarrollo de este subapartado se pueden ver en la Tabla 5.6.

5.6 Desarrollo de los enemigos

Los enemigos son importantes en cualquier videojuego, pues son los seres que el jugador debe derrotar para continuar su aventura. En El Camino del Alquimista, los enemi-



Figura 5.10: Interfaz de la tienda de la vendedora ambulante

| |
|--------|
| Moned1 |
| Moned2 |
| Moned4 |
| Ven1 |
| Ven2 |
| Ven3 |
| Ven4 |
| Ven5 |
| Ven6 |
| Ven7 |

Tabla 5.6: Requisitos cumplidos en el desarrollo del sistema compra-venta

gos son monstruos pertenecientes a uno de los cuatro elementos. Para indicar esto, en la clase creada para controlar el comportamiento de los enemigos, se ha creado un tipo de variable “elemento”. Los monstruos también tienen un nivel asignado, que indica cómo de fuertes son.

El elemento de cada enemigo es algo que se tuvo en cuenta para comenzar a diseñarlos. A pesar de ser algo similares, cada uno de los cuatro enemigos implementados en el juego, tiene características del elemento al cual pertenecen como se puede ver en el boceto realizado por mi compañero Tomás Puertas en la Figura 5.11. Un ejemplo de esto sería el monstruo de tierra, que posee unos cuernos hechos de roca mucho más grandes que los del resto de enemigos.

El elemento de un monstruo puede evitar que este sea afectado por una bomba o ataque de su mismo elemento. Por ejemplo, una bomba explosiva o proyectil de fuego, no dañará a un monstruo de dicho elemento, pero sí a uno de otro. De esta manera, los monstruos de distintos elementos se pueden dañar entre sí. El jugador debe tener en cuenta esto en la batalla. El elemento de un monstruo también decidirá que ítem deja el monstruo al morir, ya que un monstruo de aire dejará aire al morir.

Los monstruos del videojuego disponen de una inteligencia artificial que ha sido implementada mediante una máquina de estados, explicada en el capítulo anterior. Su IA también les permite sortear obstáculos con un algoritmo de búsqueda de caminos. Para implementar esto, se ha utilizado el componente *NavMeshAgent* disponible en la librería de IA de Unity. Con este componente, se puede determinar un objetivo a seguir para el

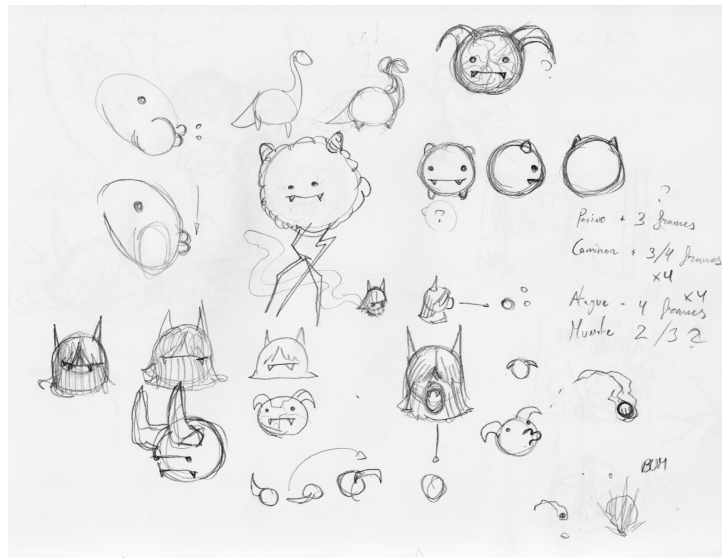


Figura 5.11: Boceto de los enemigos



Figura 5.12: Imagen de distintos enemigos

monstruo. Además, se pueden indicar las áreas que son caminables y cuales no utilizando *meshes* de navegación en los escenarios y paredes como se puede ver en la Figura 5.13. Incluso es posible añadir un indicador de obstáculos a un *GameObject* como pueden ser el resto de enemigos o la vendedora por ejemplo, con el componente *NavMeshObstacle*. Los requisitos cumplidos en el desarrollo de este subapartado se pueden ver en la Tabla 5.7.

5.7 Desarrollo de niveles

Los niveles del videojuego están compuestos de diversas mazmorras, cada una con distinta ambientación como un bosque o una cueva. En cada mazmorra o nivel, se podrán encontrar diferentes monstruos, recetas e ingredientes. Dichos monstruos se adaptarán al hábitat dónde se encuentren, siendo más fácil encontrar monstruos del elemento tierra en la cueva por ejemplo. Estas mazmorras, como hemos dicho en capítulos anteriores, son generadas de forma procedural (ver el Algoritmo 5.1).

Se planearon cuatro niveles distintos en el juego, con cuatro mazmorras distintas, una por cada elemento. Algunas de estas estarían compuestas de dos mazmorras diferentes, accediendo a la segunda al llegar al final de la primera. Se comenzaría por el elemento



Figura 5.13: NavMesh de una mazmorra

| |
|-------|
| Com2 |
| Com7 |
| Com10 |
| Mon1 |
| Mon2 |
| Mon3 |
| Mon4 |
| Mon5 |
| Mon6 |
| Mon7 |
| Mon8 |
| Mon9 |

Tabla 5.7: Requisitos cumplidos en el desarrollo de los monstruos

de tierra y al final de este primer nivel, habría un enemigo jefe de dicho elemento. El siguiente nivel sería agua, el tercer elemento sería aire y se acabaría con fuego. Cada nivel tendría un jefe del elemento de la mazmorra al final. Este diseño de niveles remarca la importancia de los cuatro elementos en el juego. En la Figura 5.14 se puede ver la primera de las mazmorras implementada “El bosque encantado”.

Al derrotar a cada uno de los jefes, se obtendrá un pedazo de la receta del elixir de vida. Como dijimos en anteriores capítulos, fabricar este elixir es el objetivo de este juego. De esta manera, es necesario vencer a los cuatro jefes así como explorar los cuatro niveles para acabar el juego.

5.7.1. Generación de mazmorras

Para la generación de las mazmorras del Camino del Alquimista, se ha utilizado el algoritmo del caminante aleatorio. En concreto, se ha utilizado la versión que cada caminante tiene sólo cuatro posibles movimientos: arriba, abajo, izquierda y derecha, impidiendo así el movimiento en diagonal. Para generar cada mazmorra, se indica el número de caminantes que se quiere utilizar y el número máximo de pasos que pueden dar antes de detenerse. Contra más caminantes se utilicen, más dispersa estará la mazmorra, y contra más grande sea el número de pasos, más largos serán los pasillos.



Figura 5.14: Imagen de una mazmorra del videojuego

Algorithm 5.1 Algoritmo del caminante aleatorio

Input: $M \leftarrow [\text{posiblemoves}], n \leftarrow \text{steps}$

Output: walk

- 1: $walk \leftarrow []$
 - 2: $pos \leftarrow (0,0)$
 - 3: $posLast \leftarrow pos$
 - 4: $walk.add(pos)$
 - 5: **while** $n > 0$ **do**
 - 6: $i \leftarrow \text{random}[0, \text{len}(M)]$
 - 7: $pos \leftarrow posLast + M[i]$
 - 8: $walk.add(pos)$
 - 9: $n --$
 - 10: **end while**
 - 11: **return** walk
-

Esta parametrización en la creación de las mazmorras permite generar mazmorras más sencillas y pequeñas para los primeros niveles, y crear calabozos grandes y laberínticos en los últimos, aportando así un mayor reto en los niveles más avanzados. [14]

Una vez generados los pasillos de la mazmorra con el algoritmo mencionado, el siguiente paso es poblarla de monstruos, tesoros, añadir escaleras y decorarla para darles un mejor aspecto. Para ello se traduce la mazmorra a una matriz de caracteres, donde cada carácter indica qué hay en esa casilla de la mazmorra, como se ve en la Figura 5.15. El significado de los caracteres que conforman esta representación se ve en la Tabla 5.8.

De esta forma es más fácil controlar qué elementos poner y dónde hacerlo. Así, se evita de manera más eficaz el colocar decoración o tesoros que entorpezcan el paso del jugador y, por ejemplo, bloqueen un pasillo estrecho, impidiendo al jugador atravesarlo. También permite controlar mejor la generación de las paredes de forma correcta en la mazmorras. Además, la colocación de las escaleras para cambiar de un piso al otro, necesitan de dicha matriz para comprobar eficientemente que no estén demasiado cerca de las paredes por ejemplo, pues esto podría generar problemas al transportar al jugador de

| | |
|---|--------------------------------|
| x | vacío |
| w | muro |
| . | baldosa de suelo |
| m | monstruo |
| S | escalera salida |
| A | escalera entrada |
| d | decoración sin <i>collider</i> |
| D | decoración con <i>collider</i> |
| i | tesoro con ítem |
| r | tesoro con receta |
| o | tesoro con dinero |
| M | vendedora ambulante |

Tabla 5.8: Significados de los caracteres de la representación ASCII de una mazmorra

| |
|-------|
| Maz1 |
| Maz2 |
| Maz3 |
| Maz4 |
| Maz5 |
| Maz6 |
| Maz7 |
| Maz8 |
| Maz9 |
| Maz10 |

Tabla 5.9: Requisitos cumplidos en el desarrollo de la generación de la mazmorra

menos el primero, que solo tiene escalera de subida.

Los cofres pueden contener tres tipos de tesoro en su interior: ítem, receta o dinero. Una vez abiertos, no se pueden volver a abrir. Hay ítems que sólo se pueden obtener de esta manera. Las recetas sólo se encuentran dentro de los cofres, no hay otra manera de aprender recetas nuevas. Un cofre nunca dará al jugador una receta que ya conozca. Haciendo así más eficiente la obtención de recetas. Los ítems, recetas y cantidades de dinero que se pueden obtener en los cofres cambian de una mazmorra a otra. Los requisitos cumplidos en el desarrollo de este subapartado se pueden ver en la [Tabla 5.10](#).

| |
|-------|
| Tes1 |
| Tes2 |
| Tes3 |
| Tes4 |
| Tes5 |
| Alq3 |
| Rece4 |

Tabla 5.10: Requisitos cumplidos en el desarrollo de los elementos de la mazmorra

CAPÍTULO 6

Implantación y pruebas

6.1 Implantación

El videojuego, una vez convertido en un producto viable, ha sido publicado en la página web itch.io que es una plataforma destinada a la publicación de juegos independientes. Durante la primera versión de prueba, se publicó de forma privada con una contraseña, que se repartió a personas cercanas a los desarrolladores del producto. A partir de estas primeras pruebas, se consiguieron detectar algunos fallos y *bugs* que se corrigieron en la siguiente versión.

Después de corregir estos problemas, se añadió una segunda mazmorra, más monstruos y armas. Se publicó en dicha plataforma de forma pública una versión más definitiva. El aspecto final de la página del [Camino del alquimista](#) donde descargar el juego se puede ver en la Figura 6.1.

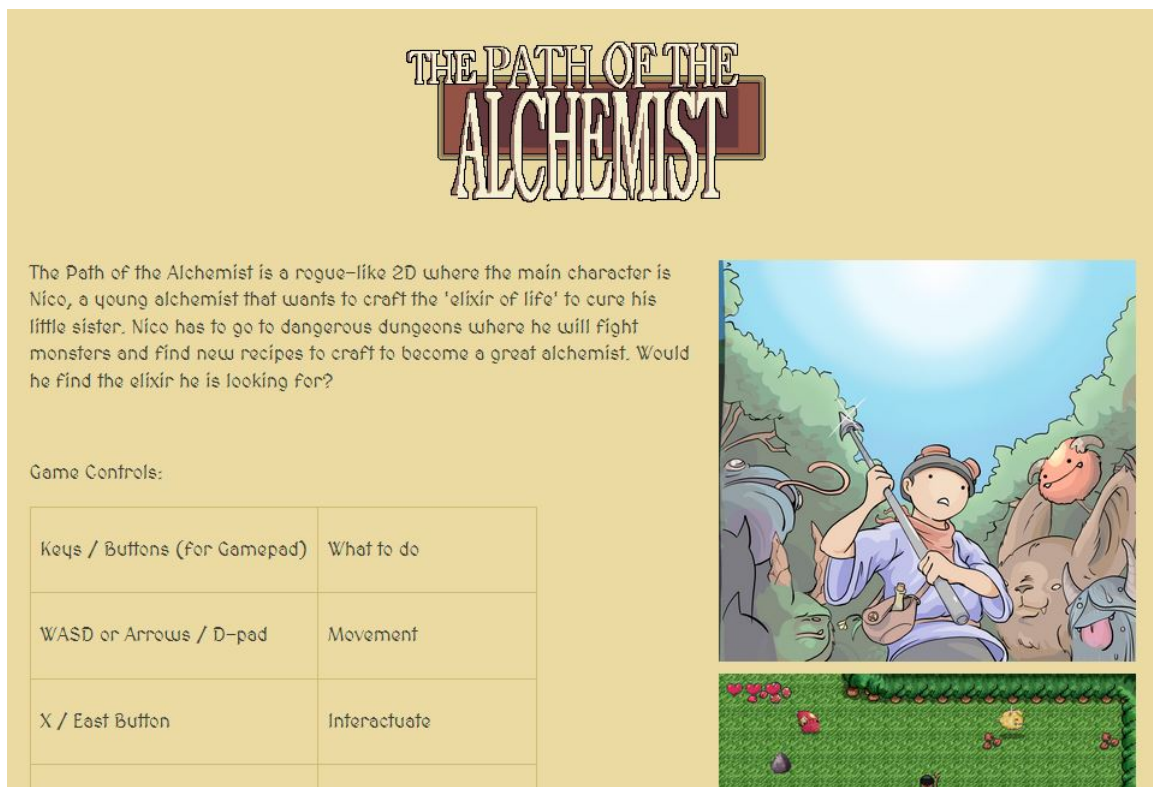


Figura 6.1: Página del Camino del Alquimista en itch.io

Junto a la publicación del videojuego en internet, se creó un formulario de Google Forms donde se encuestaba a los usuarios sobre los diferentes aspectos del juego y así conocer su opinión para saber en qué hay que mejorar, además de cuáles son los aspectos más fuertes del Camino del Alquimista. A continuación, analizaremos los resultados de este cuestionario donde se han obtenido doce respuestas de usuarios.

En la Figura 6.2 se ve el resultado a la afirmación sobre la dificultad del videojuego. Se puede ver que los resultados están un poco dispares, pero la mayoría, más de un 60 % piensa que tiene una dificultad media o media-fácil. Nadie piensa que es un juego muy fácil y sólo un 16 % piensan que es un juego muy difícil. Al ser un juego *rogue-like* se quería que tuviera cierto nivel de dificultad, sin ser demasiado complicado. Con estos resultados, se puede creer que se ha cumplido.



Figura 6.2: Resultados del formulario, dificultad del juego

La Figura 6.3 muestra la opinión de los usuarios a si sabes qué hay que hacer en el juego, o por el contrario están perdidos cuando comienzan a jugar. Aquí se quería medir cómo de entendible es el juego, y si hace falta algún tutorial introductorio para comenzar a jugar, ya que el juego enseña al jugador cómo jugar de forma sutil. Por los resultados de esta pregunta, más de un 60 % de los encuestados saben qué hacer, por lo que podemos concluir que el sistema enseña correctamente al jugador qué hacer, aunque sí se podría mejorar, pues hay algún usuario que no tenía claro del todo qué hacer.

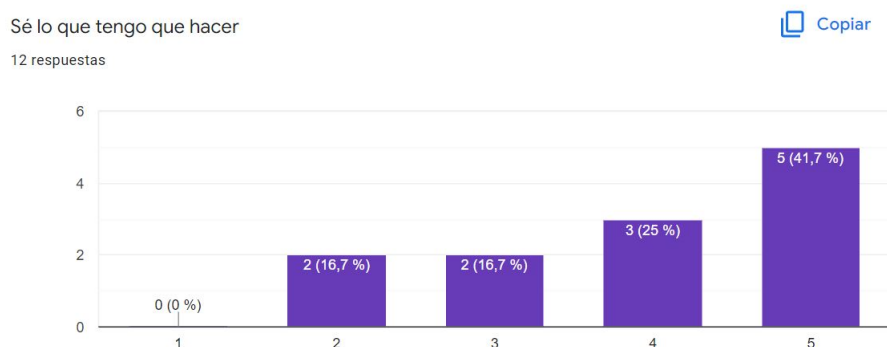


Figura 6.3: Resultados del formulario, se entiende qué hacer

La siguiente pregunta del cuestionario era si el juego es frustrante. Como se ve en la Figura 6.4, la mayoría de los encuestados, más de un 70 % piensan que no es frustrante o lo es solo un poco con la opinión neutra siendo la más votada. Por lo que se puede ver, se

ha conseguido un juego que evita en que medida de lo posible lo frustrante que suelen ser los videojuegos del género de mazmorras. En parte, esto puede deberse a la decisión de no añadir la muerte permanente.



Figura 6.4: Resultados del formulario, si es frustrante el juego

A continuación se preguntó por los controles del videojuego, más concretamente si eran intuitivos. Como se aprecia en la Figura 6.5 más de un 70 % de las respuestas son positivas. Aunque algunas respuestas son muy negativas, por lo que sería importante revisar los controles para poder satisfacer a todos los usuarios, haciendo un juego más disfrutable.

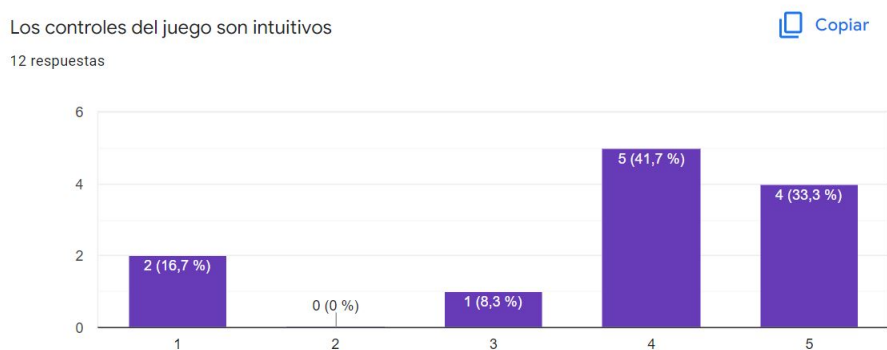


Figura 6.5: Resultados del formulario, controles intuitivos

En la Figura 6.6 se puede ver qué mecánicas del videojuego son las preferidas por los usuarios. Se puede ver que un 50 % de los encuestados prefieren intentar llegar lo más lejos posible dentro de la mazmorra, siendo la exploración lo más popular del juego. Un tercio de los usuarios prefieren matar monstruos, encontrando divertido el sistema de combate implementado. Sólo una pequeña parte de los encuestados prefieren alquimizar las nuevas recetas encontradas. Esto sería un aspecto a mejorar en los trabajos futuros.

A continuación se preguntó sobre el sistema de alquimia del juego, y así ver qué mejorar para hacerlo más popular en los jugadores. En la Figura 6.7 se ve que el sistema de fabricación implementado es sencillo, cómo se quería desde un primer momento. No se ha obtenido ninguna respuesta negativa, y más de un 80 % de los encuestados creen que este sistema es fácil de entender y usar.

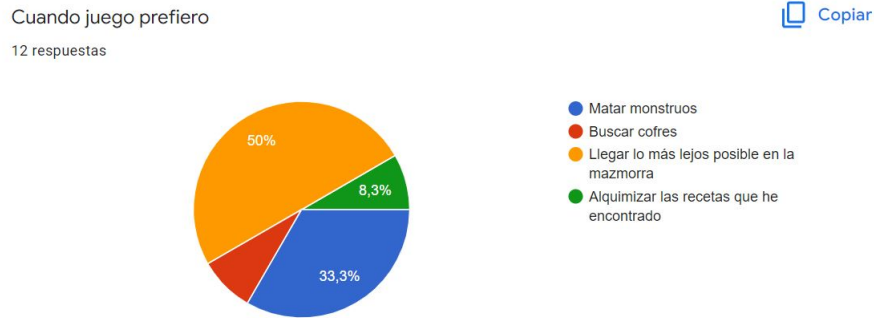


Figura 6.6: Resultados del formulario, mecánicas preferidas del juego

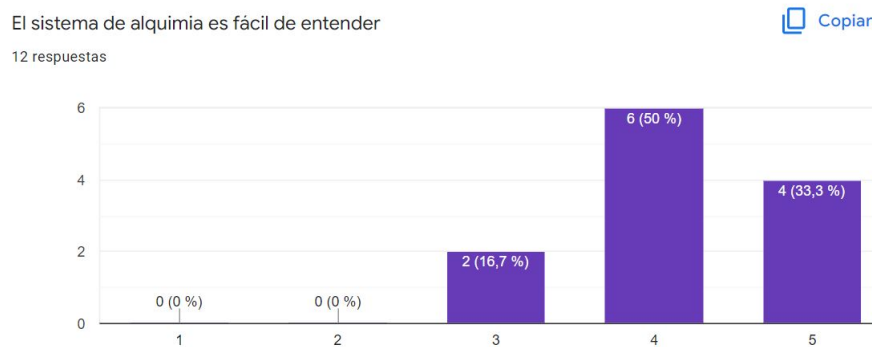


Figura 6.7: Resultados del formulario, sistema de alquimia del juego

Siguiendo con el sistema de alquimia, se preguntó sobre la obtención de recetas para poder fabricar ítems, cuyos resultados se ven en la Figura 6.8. Dos tercios de los encuestados averiguaron cómo conseguir las recetas, un resultado bastante positivo. Sin embargo, el otro tercio no lo averiguó o no sabían de qué recetas se estaba hablando, por lo que en versiones futuras se plantea implementar un pequeño tutorial sobre esta mecánica, ya que es uno de los ejes del juego.

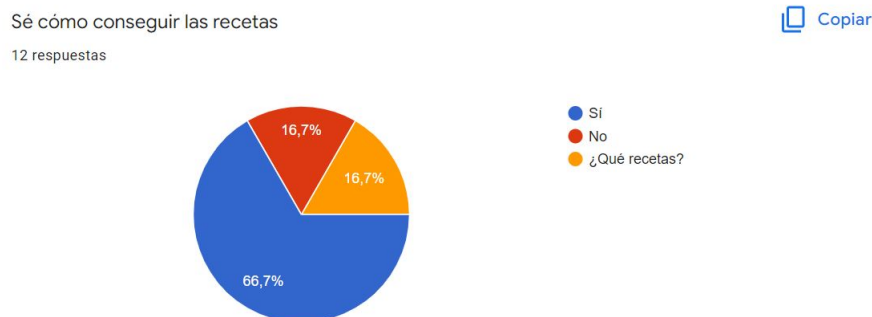
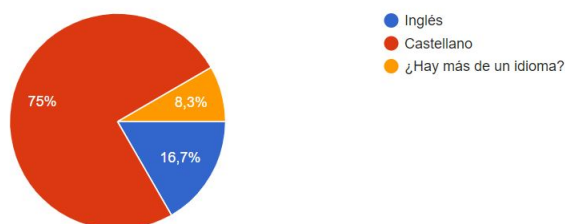


Figura 6.8: Resultados del formulario, obtención de recetas en el juego

A continuación se preguntó a los usuarios por las preferencias respecto al idioma. Como se ve en la Figura 6.9 la gran mayoría de los encuestados, un 75% eligieron el idioma castellano para jugar. También hubo una pequeña parte de los encuestados que prefirieron el inglés a la hora de jugar. Un porcentaje más pequeño no llegó a entrar en los ajustes para ver la posibilidad del cambio de idioma.

El idioma que he elegido ha sido

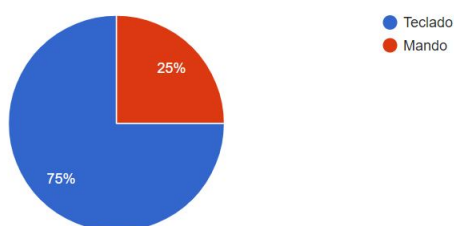
12 respuestas

 Copiar**Figura 6.9:** Resultados del formulario, idioma preferido del juego

Respecto a preferencias, también se encuestó a los usuarios respecto al tipo de controles que prefería usar, el teclado del ordenador o un mando conectado a este. Los resultados se pueden ver en la Figura 6.10. Una gran parte, un 75 % eligió usar el teclado, esto era de esperar, ya que no todo el mundo posee de un mando. Sin embargo, un 25 % jugaron con mando. En la sección de comentarios del juego, un usuario anónimo remarcó que la utilización del mando hace el juego más disfrutable y sencillo de manejar.

He jugado usando

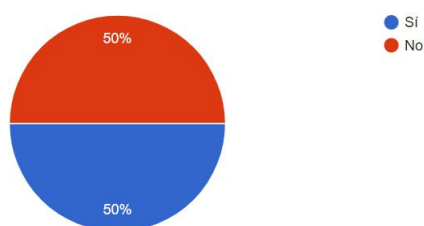
12 respuestas

 Copiar**Figura 6.10:** Resultados del formulario, controles preferidos en el juego

Finalmente, se preguntó a los usuario si fueron capaces de llegar al final de la mazmorra. Como se ve en la Figura 6.11 un 50 % de los encuestados lograron tal hazaña. Esto demuestra que es un juego con una dificultad adecuada, como ya se vio en la pregunta sobre la dificultad.

He llegado al final del bosque

12 respuestas

 Copiar**Figura 6.11:** Resultados del formulario, acabar la mazmorra

Conclusiones y trabajos futuros

7.1 Conclusiones

Con la realización del proyecto de este TFG se ha logrado cumplir los objetivos propuestos que se plantearon al principio del mismo. Primero de todo, se ha logrado convertir la idea original en un videojuego viable, de manera eficaz en estos meses de trabajo. Como se ha explicado en esta memoria, se ha tratado el proyecto como un proyecto de software y ha pasado con éxito por todas las fases del desarrollo hasta obtener el resultado actual. Esto se ha conseguido gracias a mi compañero Tomás Puertas con el que se ha formado un equipo multidisciplinar, él como artista y yo como programadora y diseñadora del videojuego, cumpliendo así otro de los objetivos planteados inicialmente. Trabajando en un equipo como este, he aprendido qué desacuerdos pueden surgir entre personas de disciplinas tan distintas, además de entender más el trabajo de los artistas en los proyectos informáticos, y sobre todo a no subestimar el tiempo requerido para realizar los diseños, *sprites*...

Hemos conseguido desarrollar el videojuego a tiempo con las principales mecánicas planteadas, aunque no se han podido implementar todos los niveles, enemigos deseados en un principio y alguna mecánica pensada, pues surgió alguna dificultad en la creación de los *tilesets* que retrasaron el proyecto, además de que se planearon más cosas de las que un equipo de dos personas podía realizar en este periodo de tiempo, pero conseguimos adaptar las tareas a unas más realistas y así conseguir un producto final viable, eliminando las mecánicas y elementos más prescindibles para centrarnos en lo importante. Esto no habría sido posible de no utilizar metodologías ágiles para planear el desarrollo del proyecto.

Con la realización de este proyecto he conseguido aprender a utilizar la vertiente 2D del motor de videojuegos Unity. He aprendido a utilizar el sistema de físicas 2D, el sistema de animaciones, la máquina de estados del animador para controlarlas y a crear y manejar interfaces de usuario, a manejar el sistema de audio de Unity, el sistema de input para poder conectar un mando al juego además de usar el teclado del ordenador, también he aprendido a utilizar el paquete de localización de Unity que permite hacer un videojuego con más de un idioma. Aparte de usar el motor de Unity, he aprendido el lenguaje de programación C#, el cual no habíamos visto con demasiada profundidad durante el grado.

En definitiva, se puede decir que los objetivos han sido cumplidos con éxito, por lo que después de este proyecto podemos estar satisfechos con el trabajo.

7.2 Trabajos futuros

Como se ha mencionado en las conclusiones, en un principio se plantearon más niveles y mecánicas que al final no fueron posibles de desarrollar a tiempo para la entrega de este proyecto. Como trabajos futuros lo primero que nos gustaría implementar sería el monstruo jefe final de la mazmorra que tenemos. El diseño de este jefe se puede ver en el boceto de la figura 7.1, donde el jefe es la cara gigante de piedra que pertenecería al elemento de tierra. Este atacaría con sus manos flotantes de piedra y al ser ambas derrotadas se caería la cara y habría que atacar a su corazón de cristal que se aprecia también en la figura. A pesar de estar diseñado, tanto el comportamiento como el aspecto, no llegó a implementarse ni a realizar sus *sprites* y animaciones por falta de tiempo.

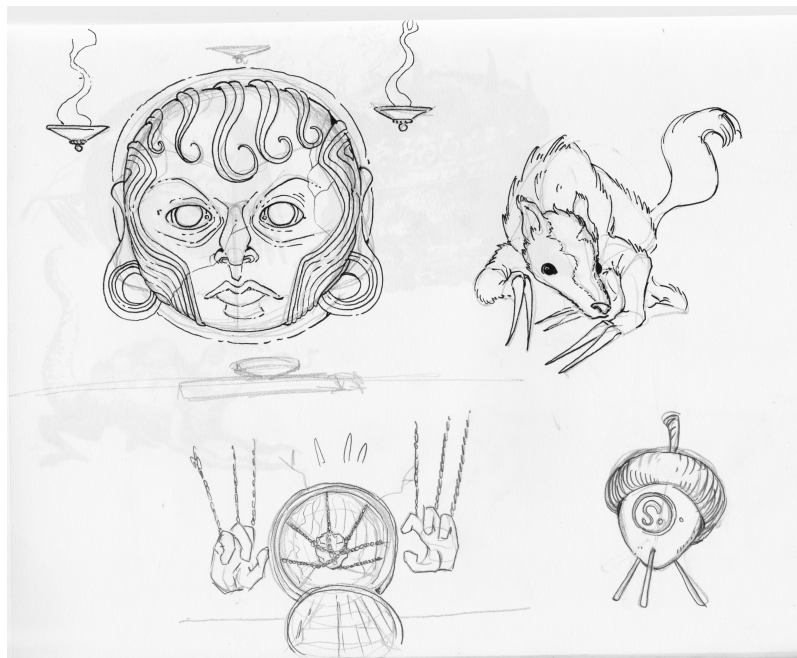


Figura 7.1: Boceto del jefe de la mazmorra

A parte de este jefe, nos gustaría ampliar el juego a largo plazo, implementando las mazmorras que aparecen en el GDD. Se planearon cuatro niveles con una o dos mazmorras distintas cada uno y sólo se implementó finalmente la primera. Con estos nuevos niveles, se implementarían nuevos monstruos y jefes más fuertes que los vistos en la primera. También nos gustaría implementar alguna de las mecánicas que se quedaron en el tintero y que aparecen en el GDD, como eran los excursionistas perdidos. El jugador podría hacer trueques con estos excursionistas para conseguir ítems de mayor rareza a cambio de los suyos, ellos aparecerían en las mazmorras de manera aleatoria. Como se puede ver, se planearon en el GDD varias cosas que no llegaron al producto final, y nos gustaría poder implementarlas en el futuro.

Bibliografía

- [1] Donovan, Tristan *Replay, The History of Video Games*. East Sussex, England, primera edición, 2010.
- [2] Juego de mazmorras - Wikipedia la enciclopedia libre. Consultado en https://es.wikipedia.org/wiki/Juego_de_mazmorras
- [3] Alquimia - Wikipedia la enciclopedia libre. Consultado en <https://es.wikipedia.org/wiki/Alquimia>
- [4] Lindsay, Jack *The Origins of Alchemy in Graeco-Roman Egypt*. Barnes y Noble, Nueva York, 1970.
- [5] Berlin Interpretation. Consultado en http://www.roguebasin.com/index.php?title=Berlin_Interpretation
- [6] Glenn R. Wichman *A BRIEF HISTORY OF ROGUE*. copyright Glenn R. Wichman, 1997
- [7] A. K. Dewdney *AN EXPERT SYSTEM OUTPERFORMS MERE MORTALS AS IT CONQUERS THE FEARED DUNGEONS OF DOOM*. Scientific American, volume 252, issue 2, February 1985, pp. 18-21
- [8] Mystery Dungeon. Consultado en https://en.wikipedia.org/wiki/Mystery_Dungeon
- [9] Russ Frushtick, Hades blends God of War with Binding of Isaac in marvelous ways. *Polygon*, Dec 7, 2018
- [10] April Grow, Melanie Dickinson, Johnathan Pagnutti, Noah Wardrip-Fruin, Michael Mateas. Crafting in Games. *DHQ: Digital Humanities Quarterly*, volumen 11, número 4, 2017
- [11] C# - Wikipedia, la enciclopedia libre. Consultado en https://es.wikipedia.org/wiki/C_Sharp
- [12] Unity Documentation. Consultado en <https://docs.unity3d.com/es/530/Manual/LearningtheInterface.html>
- [13] ¿Qué es GitHub? - Control de versiones de un vistazo. Consultado en <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/github/>
- [14] Tanya X. Short y Tarn Adams. *Procedural Generation in Game Design*. CRC Press, Taylor Francis Group

APÉNDICE A

Objetivos desarrollo sostenible

ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

| Objetivos de Desarrollo Sostenible | Alto | Medio | Bajo | No procede |
|---|------|-------|------|---------------|
| ODS 1. Fin de la pobreza. | | | | x |
| ODS 2. Hambre cero. | | | | x |
| ODS 3. Salud y bienestar. | | | | x |
| ODS 4. Educación de calidad. | | | | x |
| ODS 5. Igualdad de género. | x | | | |
| ODS 6. Agua limpia y saneamiento. | | | | x |
| ODS 7. Energía asequible y no contaminante. | | | | x |
| ODS 8. Trabajo decente y crecimiento económico. | | | x | |
| ODS 9. Industria, innovación e infraestructuras. | | | | x |
| ODS 10. Reducción de las desigualdades. | | x | | |
| ODS 11. Ciudades y comunidades sostenibles. | | | | x |
| ODS 12. Producción y consumo responsables. | | | | x |
| ODS 13. Acción por el clima. | | | | x |
| ODS 14. Vida submarina. | | | | x |
| ODS 15. Vida de ecosistemas terrestres. | | x | | |
| ODS 16. Paz, justicia e instituciones sólidas. | | | | x |
| ODS 17. Alianzas para lograr objetivos. | | | | x |



Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

Para este proyecto se deseaba cumplir tantos objetivos de desarrollo sostenible como fuera posible, pero al tratarse de un videojuego, esto no era tan fácil. Muchos de estos objetivos son muy complicados de cumplir en un videojuego, ya que no están nada relacionados con este campo. Por ello muchos de los ODS están marcados como ‘no procede’, que como es obvio, un videojuego no puede acabar con la pobreza o el hambre en el mundo.

Este videojuego tiene relación con el objetivo de igualdad de género, pues se ha buscado que la cantidad de personajes masculinos y femeninos que aparecen sea similar. Esto se ha conseguido pues de los tres personajes que aparecen, Nico el personaje del jugador es un chico, y su hermana y la vendedora ambulante son una niña y una mujer. Por lo que están igualados ambos géneros. También se planeó introducir un cuarto personaje, que sería masculino, pero no llegó a implementarse por falta de tiempo.

También está relacionado en menor medida con la búsqueda de un trabajo decente y crecimiento económico. El proyecto se ha publicado en internet para buscar propinas de los usuarios, y así conseguir ganar algo de dinero con todo el trabajo que hemos realizado durante estos meses. Además, se valora el trabajo de todos los colaboradores de este proyecto, apareciendo todos en los créditos y no utilizando *assets* de terceros sin consentimiento.

Además, este trabajo está relacionado con el ODS 10: ‘Reducción de las desigualdades’ pues la historia del juego trata de que el protagonista y su hermana son dos huérfanos sin mucho dinero. El hermano mayor, el personaje jugador, intenta buscarse la vida convirtiéndose en un alquimista además de conseguir curar a su hermana enferma. Para reforzar el cumplimiento de este objetivo, el juego se ha desarrollado en dos idiomas: castellano e inglés, para conseguir llegar a un mayor público y no dejar de lado a las personas que no sepan leer castellano.

Este trabajo también está relacionado con el ODS: ‘ Vida en ecosistemas terrestres’ pues las mazmorras que aparecen en el juego están pobladas llenas de formas de vida: los monstruos. Estos representan los cuatro elementos de la naturaleza, y cada una de las mazmorras representa un tipo de paisaje natural. Además, no es obligatorio derrotarlos para llegar al final de la mazmorra, por lo que se respeta en mayor o menor medida la vida de estos. Aunque hay que tener en cuenta que se trata de un videojuego y derrotar enemigos es parte de la mayoría de los juegos, y es difícil de evitar.

En resumen, se ha intentado implantar los ODS a este proyecto en la medida de lo posible, aunque fuera complicado. Se ha podido encontrar alguna relación con ellos, al menos con algunos.

APÉNDICE B

Alquimia - manual de juego

Alquimia: manual de juego

| | |
|----------------------------------|----------|
| El juego | 1 |
| Componentes del juego | 1 |
| Preparación | 1 |
| Cómo jugar | 1 |
| Qué hacer en tu turno | 1 |
| Cómo ganar | 2 |
| Turno de ejemplo | 2 |
| Explicación de las cartas | 3 |
| Cartas de monstruo | 3 |
| Duendecillo | 3 |
| Cartas de lugar | 4 |
| Lugares con monstruos | 4 |
| Tienda | 4 |
| Mercader | 4 |
| Fuentes | 4 |
| Ganar y perder monedas | 4 |
| Negativas | 5 |
| Pozo de los deseos | 5 |
| Iglesia | 5 |
| Objetos útiles | 5 |
| Elementos mejorados | 5 |
| Pociones | 6 |
| Otros objetos | 6 |
| Ingredientes | 6 |
| Alquimizaciones | 7 |
| Precios de venta | 7 |

El juego

Alquimia es un juego en el que tu objetivo será alcanzar la vida eterna gracias a la fabricación del mítico Elixir de vida (y ganar así la partida). Para ello tendrás que alquimizar todo tipo de ingredientes, siendo los 4 elementos los más básicos de todos.

Componentes del juego

- Mazo de cartas de lugar (azules)
- Mazo de cartas de monstruo (fucsias)
- Mazo de cartas de ingredientes y objetos
- Fichas de los 4 elementos (agua, tierra, aire y fuego)
- Monedas de plata
- Monedas de oro
- Recetarios
- Dado de 20 caras
- Dado de 6 caras
- Dado de 4 caras



Preparación

Baraja los mazos de cartas de lugar y de monstruo, cada uno por separado. Una vez barajados se colocan boca abajo en la mesa uno al lado del otro.

Separa las cartas de ingredientes y objetos teniendo un montón por cada uno de ellos, para poder tener todos a la vista formando así la banca de ingredientes y objetos.

Separa las fichas de elementos y las monedas formando 6 montones en el centro de la mesa, 4 de los elementos, y 2 para las distintas monedas de oro y plata.

Una vez tenemos listos los componentes del juego, reparte 3 monedas de plata a cada jugador. También reparte los recetarios a los jugadores.

Tirar todos el dado de 6 caras, quién saque el número más alto será el jugador inicial. Luego se seguirá el turno en dirección contraria a las agujas del reloj.

Cómo jugar

Qué hacer en tu turno

El turno comienza levantando una carta del mazo de lugares, colocándola boca arriba al lado del mazo.

- Si en la carta hay algo escrito en la parte inferior, esto indicará el efecto de la carta, para más información, mirar la [explicación de las cartas](#)

- Si por el contrario, es sólo el dibujo del lugar y no hay nada escrito, levanta una carta del mazo de monstruos, y la colocas al lado de este boca arriba. Tira el dado de 20 caras para enfrentarte a él

Después, puedes alquimizar uno o varios objetos siempre que tengas los ingredientes necesarios en tu posesión, para esto mira qué necesitas en el recetario.

También es posible realizar un trueque con otro jugador para intercambiar ingredientes, objetos o elementos.

Finalmente, si posees un objeto puedes utilizarlo en tu turno. Para más información ver objetos útiles.

Cómo ganar

El juego acaba cuando un jugador consigue fabricar el **elixir de vida** proclamándose ganador de la partida.

Turno de ejemplo

Tenemos 4 jugadores: Laura, Andrea, Alejandro y Sandra.

Empieza el turno Laura, que levanta una carta azul de lugar. Le sale la cascada barracuda, entonces tiene que levantar una carta fucsia de monstruo. Levanta una carta de monstruo y le sale un monstruo de fuego de rango I llamado 'Musaraña de fuego', como es de rango I necesita un 3 para vencerlo. Tira el dado de 20 caras, y saca un 7. Ha vencido al monstruo y gana 3 elementos de fuego.

Le pasa el turno a la jugadora de su derecha, Andrea. Levanta una carta de lugar, y le sale el lago, coge 5 fichas del elemento de agua. Con estas 5 aguas, las gasta y alquimiza un agua mágica.

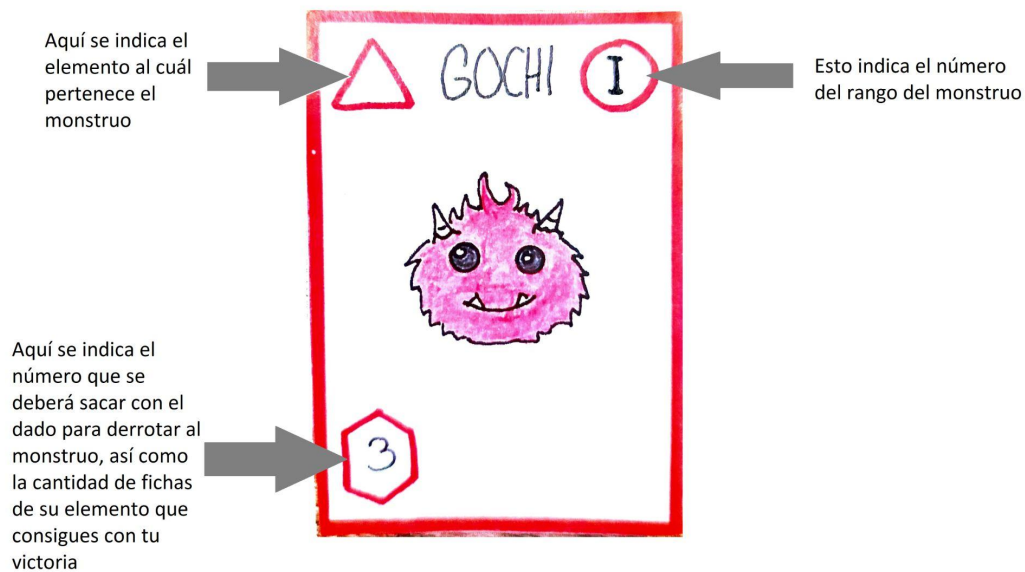
Ahora le toca al jugador de su derecha, Alejandro. Levanta una carta de lugar, le sale la tienda. Como tenía 2 monedas de plata se las gasta en comprarse un veneno.

Le pasa el turno a Sandra, quien levanta una carta de lugar y le sale el pozo de la suerte, como no tenía ninguna moneda, no puede tirarla al pozo. Sandra aprovecha su turno para proponer un trueque a Andrea, le quiere cambiar su tela, por el agua mágica que Andrea se acababa de fabricar. A Andrea le parece un buen trato y acepta, se intercambian los ingredientes. Así acabaría la ronda y empezaría de nuevo Laura.



Explicación de las cartas

Cartas de monstruo



Los monstruos pueden ser uno de los 4 elementos (agua, tierra, aire y fuego).



Hay 3 rangos de monstruos (I, II, III).

Para **enfrentarse a un monstruo** habrá que usar el dado de 20 caras:

- Si sacas un número igual o superior al indicado consigues derrotarlo, y como recompensa coge dicho número de fichas del elemento del monstruo y te las quedas. Si el monstruo era de rango III, coge una moneda de plata también.
- Si sacas un número inferior pierdes, no conseguirás ninguna ficha.

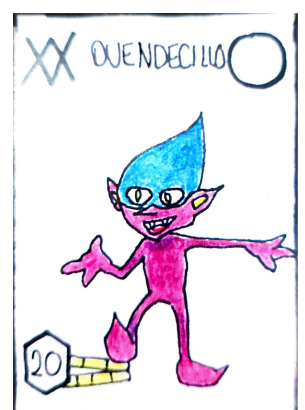
En tu enfrentamiento podrás ayudarte de objetos como las *botas pegaso* o el *brazalete audaz*. Explicadas más adelante en objetos útiles.

Duendecillo

Existe un monstruo especial, el duendecillo. No es de ningún elemento, tiene el tipo neutro. Será necesario sacar un 20 para vencerlo.

- Si sacas un 20, recibirás una moneda de **oro**.
- Si sacas menos, perderás todas tus monedas de plata.

No podrás utilizar ningún objeto de ayuda para enfrentarte a él.



Cartas de lugar

En esta parte de la carta se indica su efecto



Lugares con monstruos

Estas son las cartas que no tienen ningún texto en su parte inferior. Hay 7 cartas de este tipo, todas funcionan de la misma manera:

- Cascada barracuda
- Páramo géiser
- Glaciar errante
- Bosque encantado
- Ruinas abandonadas
- Cueva de cristal
- Picos afilados

Cuando levantes una carta de este tipo, a continuación deberás levantar una carta del mazo de monstruos y este será el monstruo contra el que deberás enfrentarte.

Tienda

Podrás comprar los ingredientes que están a la venta con tus monedas de plata:

- *Raíz de mandrágora*
- *Veneno*
- *Tela*
- *Frasco*

Estos ingredientes y sus precios están indicados en la carta.

También podrás vender los ingredientes u objetos que ya no quieras, para más información consultar precios de venta.

Mercader

El mercader es una tienda especial, vende objetos muy útiles a cambio de monedas de plata:

- *Capa de invisibilidad*
- *Poción de protección*

Estos ingredientes y sus precios están indicados en la carta.

Para conseguir uno de estos productos **gratis**, podrás enamorar al mercader y que te lo regale, para ello deberás utilizar una *poción de amor*.

Aquí también es posible vender lo que ya no quieras.

Fuentes

En las fuentes conseguirás elementos sin necesidad de ningún enfrentamiento. Hay 8 fuentes distintas, 2 por cada elemento:

- Agua: Lago (+5) y fuente (+3)
- Tierra: Duna de arena (+5) y mina abandonada (+3)
- Aire: Tornado (+5) y niebla (+3)
- Fuego: Volcán (+5) y hoguera (+3)

Ganar y perder monedas

Hay 3 cartas donde recibes monedas de plata, estas indican el número que ganas:

- Caza de brujas (+4)
- Hada traviesa (+3)
- Tombola (+2)

Hay 2 cartas donde pierdes monedas de plata:

- Puente de los trols (-2)
- Timo barato (-1)

Por otra parte, están las tabernas, hay 2 distintas:

- Taberna 1: pierdes el 50% de tus monedas de plata
- Taberna 2: tira el dado de 6 caras, según el resultado ganarás o perderás monedas de plata.
 - 1: -3 monedas
 - 2: -2 monedas
 - 3: -1 moneda
 - 4: +1 moneda
 - 5: +2 monedas
 - 6: +3 monedas

En la parte inferior de la carta indica qué hace cada una.

Negativas

Tenemos 2 cartas de este tipo:

- Rapto, te hará perder el turno. No podrás alquimizar nada, ni realizar ningún trueque.
- Tropiezo en la plaza, perderás 2 fichas de tu elemento más abundante. Si tienes la misma cantidad en varios elementos, elige cual quieres perder.

Pozo de los deseos

Si estás ante el pozo de los deseos, lo primero será tirar 1 moneda de plata al pozo. A continuación tira el dado de 4 caras. Si no tienes ninguna moneda, no lo tires.

Tal del número que saques ocurrirá una cosa:

- 1: nada
- 2: ganas 2 fichas de un elemento a tu elección (deben ser del mismo elemento)
- 3: ganas 3 monedas de plata
- 4: nada

Los posibles resultados del dado están indicados en la parte inferior de la carta.

Iglesia

En la iglesia podrás conseguir *agua bendita*, para ello necesitarás tener fichas del elemento agua. Podrás cambiar 1 o 2 aguas normales por el mismo número de *agua bendita*.

Si no tienes ninguna ficha de agua no podrás bendecirla.

Objetos útiles

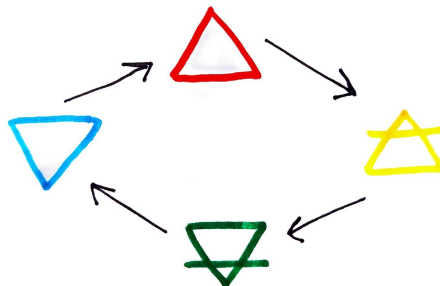
Todos estos objetos se gastan tras su uso.

Elementos mejorados

Hay 4 ingredientes, con los que se puede robar todas las fichas de elemento a otro jugador, estos son:

- *Agua mágica*: roba el fuego
- *Tierra seca*: roba el agua
- *Aire cambiante*: roba la tierra
- *Fuego fatuo*: roba el aire

Estos roban el elemento débil a ellos. Según la rueda:



La flecha indica a cuál vencen.

Pociones

Otros objetos que te puedes fabricar son las pociones, cada una tiene una utilidad:

- *Poción de amor*: enamora al señor mercader, y este te regala uno de los objetos de su tienda
- *Poción de protección*: con ella te puedes proteger de un robo
- *Poción de mal de ojo*: destruye las *pociones de protección* del resto de jugadores

Otros objetos

Objetos para **robar** a otros jugadores:

- *Guantes pícaros*: roba cualquier objeto que no tenga rango S a otro jugador
- *Capa de invisibilidad*: roba cualquier objeto a otro jugador

Estos objetos no son robables.

Objetos para enfrentarse a **monstruos**:

- *Brazalete audaz*: suma +6 a la tirada del dado, si consigues la victoria ganas 3 monedas de plata

- *Botas pegaso*: vence a cualquier monstruo sin necesidad de enfrentarse a este. Consigues 5 monedas de plata

Estos objetos no se pueden utilizar contra el duendecillo.

El último objeto:

- *Piedra de duplicación*: duplica cualquier objeto salvo los de rango S

Ingredientes

Los ingredientes y objetos de **rango S** están indicados en el recetario, y son los siguientes:

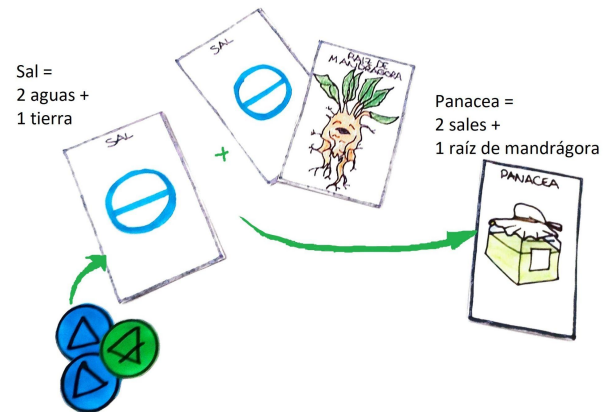
- Elixir de vida
- Capa de invisibilidad
- Poción de amor
- Poción de protección
- Oro

En la carta de dichos ingredientes hay una S dorada indicando su rango.

Alquimizaciones

Dentro de tu turno podrás alquimizar cuantos objetos quieras, pero no en un turno ajeno. Las recetas de estos vienen en el recetario que cada jugador tendrá siempre para consultar.

Cuando fabriques algo, los ingredientes usados para su alquimización se gastarán y serán devueltos a la banca de ingredientes. Excepto la *piedra filosofal*, pues no se gasta tras su uso y se puede utilizar un número ilimitado de veces.



Precios de venta

| | | | | | | | |
|--------------------|---|-------------------|---|-----------------------|---|-----------------------|----|
| Tela | 1 | Aire cambiante | 2 | Poción de mal de ojo | 5 | Éter | 7 |
| Frasco | 1 | Agua mágica | 2 | Botas pegaso | 5 | Capa de invisibilidad | 8 |
| Azufre | 1 | Fuego fatuo | 2 | Guantes pícaros | 5 | Agua bendita | NV |
| Mercurio | 1 | Tierra seca | 2 | Rocío | 5 | Piedra filosofal | NV |
| Sal | 1 | Brazaletes audaz | 3 | Poción de protección | 5 | Oro | NV |
| Hierro | 1 | Panacea | 3 | Poción de amor | 6 | Elixir de vida | NV |
| Veneno | 1 | Pirita | 3 | Piedra de duplicación | 6 | | |
| Raíz de mandrágora | 1 | Jarabe escarchado | 3 | Ácido tartárico | 7 | | |

NV: No se vende

APÉNDICE C

El Camino del Alquimista -
Documento de diseño de juego

El camino del Alquimista

| | |
|---|-----------|
| Análisis general | 2 |
| Declaración de intenciones | 2 |
| Género | 2 |
| Plataformas | 3 |
| Audiencia objetiva | 3 |
| Personajes | 3 |
| Historia | 5 |
| Gameplay | 5 |
| Vistazo al gameplay | 5 |
| Experiencia jugable | 5 |
| Claves Gameplay | 5 |
| Objetivos y recompensas | 6 |
| Mecánicas | 6 |
| Alquimia | 6 |
| Mazmorras | 6 |
| Enfrentamientos | 6 |
| Monstruos | 7 |
| Diseño de niveles | 7 |
| Nivel 0 - Tu casa | 7 |
| Nivel 1 - Bosque encantado + Cueva encantada (Tierra) | 7 |
| Nivel 2 - Lagos celestes (Agua) | 7 |
| Nivel 3 - Montañas afiladas con castillo siniestro (Aire) | 8 |
| Nivel 4 - Bosque quemado + subida al volcán (Fuego) | 8 |
| Eventos de las mazmorras | 8 |
| Vendedor ambulante | 8 |
| Padre Federico | 8 |
| Cofres del tesoro | 8 |
| Pozo de los deseos | 9 |
| Excursionista | 9 |
| Objetos utilizables | 9 |
| Curativos | 9 |
| De ataque | 9 |
| Elementos del juego | 10 |
| Recetas | 10 |
| Esquema de control | 10 |
| Mando | 10 |

Análisis general

El camino del alquimista es un juego donde el jugador recorrerá mazmorras cambiantes donde encontrará recetas nuevas e ingredientes, e irá en busca de la receta del *elixir de vida*.

Disfruta de la exploración de mazmorras, el descubrimiento de nuevos ingredientes y recetas para crearlos, empieza desde los 4 elementos hasta fabricar increíbles y complejos ítems. Enfrentate también a diferentes monstruos elementales a lo largo del recorrido e intenta llegar hasta la salida de la mazmorra sin perderse o morir.



Declaración de intenciones

Se busca explotar la mecánica de la creación de objetos centrando la historia en un alquimista principiante y cómo llega a fabricar el *elixir de vida*, el objetivo de los antiguos alquimistas.

También se busca disfrutar al máximo de la exploración de mazmorras, siendo aleatorias para que así siempre sea una nueva aventura dando una mayor rejugabilidad.

Género

- Roguelike (Roguelite)
- RPG
- Juego de artesanía (crafting)
- Tercera persona
- 2D

Plataformas

El juego es desarrollado para PC, enfocado especialmente en Windows, pero también habrá una versión para Linux y otra para Mac.


Audiencia objetiva

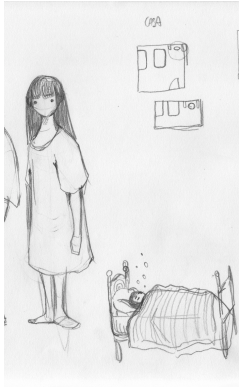

Juego orientado al público joven y adulto, en especial los que disfruten de los juegos roguelike, apto también a los que no se hayan iniciado en dicho género.

Es un juego con una dificultad media - fácil, apta para todo tipo de público desde principiantes hasta experimentados, que pueden hacer retos de velocidad o acabar el juego sin mejoras para probar su habilidad.

Sería un juego PEGI 7, pues la violencia sería de fantasía. El estilo pixelart hace los combates menos realistas, además habrá ausencia total de sangre en las batallas.

Personajes

| Personaje | Descripción | Características | Atributos | Más Info |
|---|--|---|---|----------|
| Nico  | Protagonista , personaje manejado por el jugador | Nico es un joven de 21 años, es bueno, inteligente, gracioso y tranquilo. No es el típico héroe sin miedo a lo desconocido, emprende la aventura de convertirse en un alquimista porque se preocupa mucho por su hermana y quiere curarla. Es un chico valiente cuando debe de serlo. Le gusta hacer bromas y chistes, sobre todo para hacer reír a su hermana. No sabe pelear muy bien, y tiene un don para la alquimia. | PV: puntos de vida Ataque: cuanto daño hace Defensa: cuanto defensa tiene | |

| | | | | |
|---|--------------------------------------|---|---|--|
| <p>Flamea</p>  | <p>NPC, hermana del protagonista</p> | <p>Flamea, una niña de 9 años. Hermana pequeña del protagonista. Está enferma y no se mueve de la cama.</p> <p>Flamea es bastante lista para su edad, y muy perspicaz. Sabe que no va a vivir mucho más, pero a pesar de eso es una niña muy positiva. Quiere mucho a su hermano, le anima en su aventura y le da consejos.</p> | | <p>Es una niña muy burlona, y le enseñará a Nico las cosas básicas (tutorial).</p> |
| <p>Padre Federico</p> | <p>NPC, sacerdote</p> | <p>Hombre de fe, delgado y alto vestido con una túnica.</p> | | <p>Estará en las mazmorras y si le ayudas te dará <i>agua bendita</i>.</p> |
| <p>Mercader ambulante, tía Maruja</p>  | <p>NPC</p> | <p>El mercader es una mujer gitana anciana, sabe mucho de alquimia, y cómo encontrar los ingredientes más difíciles. Lleva un saco con toda la mercancía que vende.</p> | | <p>Estará en la mazmorras y podrás realizar compra-venta de ítems hablando con ella.</p> |
| <p>Monstruos</p> | <p>NPC con IA</p> | <p>Habrán 4 rangos de monstruos. Siendo 1 los más débiles y 4 los jefes.</p> | <p>PV: puntos de vida</p> <p>Ataque: cuanto daño hace</p> <p>Rango: cómo de fuerte es</p> <p>Elemento: A qué elemento pertenece</p> | |

Historia

(A grandes rasgos)

Nico es un muchacho normal de una pequeña aldea. Su hermanita Flamea es una niña muy enfermiza, que siempre está en cama. Recordando las viejas historias de la abuela Ambrosia, recuerdas que en sus tiempos de alquimista buscó fabricar el *elixir de vida*, capaz de dar años de vida a quién no le quedan. Eso da a Nico una esperanza de curar a su hermana.

Buscando entre las cosas de la abuela, encuentra su viejo caldero, y un libro de recetas con lo más básico.

Nico sabe que en los diferentes lugares de los alrededores los alquimistas más expertos escondieron sus recetas, así que se dispone a encontrar la del *elixir de vida*. Armado con una vara de madera, se dispone a convertirse en el mejor alquimista y así salvar a Flamea.

Gameplay

Vistazo al gameplay

Coge las mecánicas del roguelike de la exploración de mazmorras aleatorias en busca de tesoros, en este caso serían ingredientes y recetas, y la muerte permanente, lo que te hará perder casi todos los ítems obtenidos en la mazmorra. Pero no perderás todo el progreso del juego, pues volverás a tu casa donde estarán todos los ítems que hayas guardado allí anteriormente.

Combina el roguelike con la creación de ítems, donde tendrás que recorrer un gran árbol de recetas de ítems hasta llegar a la receta final.

Experiencia jugable

En este juego busca enfocarse en la alquimia como eje de la historia y no solo un recurso para conseguir mejores armas o trajes. El juego tendrá una sensación de aventura gracias a la exploración, animando al jugador a recorrer toda la mazmorra en busca de sus recetas.

Claves Gameplay

Cuando el jugador muera, volverá a su casa donde habrá perdido todos los ítems obtenidos en su aventura por la mazmorra, salvo aquellos que

estuvieran en el bolsillo seguro de la bolsa, y los equipados (arma y traje).

Cuando el jugador llegue al final de la mazmorra y acabe con el jefe, obtendrá un pedazo de la receta del *elixir de vida* y podrá salir de la mazmorra con todos los ítems de su bolsa.

El jugador buscará recetas y sus ingredientes en las mazmorras para posteriormente poder fabricarlos en el caldero de su casa.

Objetivos y recompensas

- El juego tendrá pequeñas **recompensas**, en forma de la fabricación de recetas cada vez más complicadas, acercándonos a la receta definitiva poco a poco.
- El **objetivo** es llegar a esta receta final: el *elixir de vida* demostrando ser un alquimista experto y salvando a tu hermana.
- La **dificultad** del juego irá aumentando en cada mazmorra, siendo estas cada vez más largas y laberínticas, así como los monstruos que te encuentres en ellas, más fuertes.

Mecánicas

Alquimia

La alquimia es la ciencia de la creación de objetos. Para ello tendrás que usar el Caldero de la abuela, en tu casa. Podrás crear objetos a partir de una receta que hayas obtenido (o bien probar suerte con los ingredientes que tengas).

Mazmorras

Las mazmorras están compuestas de varios pisos. En ellas te encontrarás monstruos a los que te deberás enfrentar, así como cofres del tesoro que encierran recetas nuevas.

Cada mazmorra tendrá ingredientes exclusivos. Algunas mazmorras necesitaran de un objeto para moverse por la mazmorra y poder llegar al final de esta.

Si te matan antes de llegar al final, pierdes casi todos los ingredientes recogidos en está.

Enfrentamientos

En las mazmorras te encontrarás monstruos que deberás derrotar. Cada monstruo pertenece a uno de los cuatro elementos (agua, fuego, aire o tierra) y al vencerlo obtendrás la esencia de dicho elemento.

Para esto, usarás una lanza.

Monstruos

En cada mazmorra, te podrás encontrar monstruos diferentes dependiendo del hábitat donde más cómodos se sientan.

Los monstruos se pueden dividir entre los 4 elementos: fuego, agua, tierra y aire. También se separan en su rango de fortaleza: I, II, III y jefe.

Acaba con ellos con ayuda de tu lanza. Al vencerlos, obtendrás la esencia de su elemento. El ingrediente básico de todas las recetas.

Diseño de niveles

Nivel 0 - Tu casa

Tu casa es la base donde harás todo. Vives con tu hermana en la vieja casa de tu abuela Ambrosia. Aquí encontrarás:

- El caldero de la abuela: En él podrás alquimizar todo tipo de objetos con las recetas que vayas encontrando.
- Tu cama: Aquí despertarás cuando seas derrotado en una mazmorra, habrás perdido todos los ingredientes recolectados.
- Tu baúl: Aquí podrás guardar los diferentes objetos que encuentres en tus aventuras.
- Flamea, tu hermana: ella siempre estará en casa, podrás hablar con ella.

Los niveles están divididos en diversas mazmorras.

Mazmorras aleatorias, cambian cada vez que entras en ella. Tienen cofres del tesoro. Y están repletas de monstruos. En el último piso habrá un jefe final.

Nivel 1 - Bosque encantado + Cueva encantada (Tierra)

Primer lugar en el que te adentras. Un bosque oscuro que parece sacado de un cuento de hadas. Aquí podrás encontrarte con el vendedor ambulante. En los pisos más adelantados te adentrarás en una cueva donde estará el jefe final de tierra.

- No es necesario nada para entrar aquí.
- Ingredientes especiales del lugar: *Madera encantada* y *Polvo de hadas*
- Ingrediente del jefe:

Nivel 2 - Lagos celestes (Agua)

Lugar lleno de ríos, lagos y cascadas. Será difícil avanzar por aquí sin mojarse.

- Será necesario una lanza firme para usarla de pértiga.
- Ingredientes especiales del lugar: *Rocío* y *granada azul*
- Ingrediente del jefe:

Nivel 3 - Montañas afiladas con castillo siniestro (Aire)

En las montañas altas donde hay una tormenta permanente, siempre hay un viento fuerte y hay que tener cuidado para que no te caiga un rayo. En lo alto está el castillo de un viejo alquimista muy valorado. Aquí encontrarás la receta del elixir de vida.

- Será necesaria una llave para pasar las verjas del castillo y adentrarte en su interior.
- Ingredientes especiales del lugar:
- Ingrediente del jefe:

Nivel 4 - Bosque quemado + subida al volcán (Fuego)

Parte del bosque encantado ha sido arrasado por las llamas cuando el volcán vecino entró en erupción, a causa del despertar del monstruo que dormía en su interior.

- No será necesario nada para entrar.
- Ingredientes especiales del lugar: *diamante en bruto* y
- Ingrediente del jefe: *Lágrima de dragón*

Eventos de las mazmorras

Vendedor ambulante

Te lo puedes encontrar en todas las mazmorras, vende objetos exclusivos:

- Veneno
- Raíz de mandrágora
- Pétalos de anémona
- Ingrediente aleatorio de los cofres

También puedes vender cosas y así ganar monedas de plata.

Padre Federico

Ronda los pisos superiores de las mazmorras. Suele ser víctima de un ataque de monstruos, ayúdalo.

Te pedirá un poco de *agua* o *agua mágica*, si se la das te dará a cambio *agua bendita*.

Cofres del tesoro

En las mazmorras encontrarás cofres del tesoro, ábrelos. Podrás encontrarte:

- Recetas nuevas
- Monedas de plata
- Ingredientes exclusivos:
 - Los especiales de cada mazmorra
 - Azúcar lunar
 - Clavo espinoso

- Aceite negro
- Ácido
- Un duendecillo travieso: quién te robará todos tus ingredientes y dinero.

Algunos cofres estarán cerrados, necesitando una *Llave de Ladrón* para abrirlos.

Pozo de los deseos

En las mazmorras puedes encontrar un pozo de los deseos. Arroja un ítem o monedas dentro y puede que te devuelva algo especial (o nada).

Excursionista

En las mazmorras hay excursionistas que se han aventurado dentro, te propondrán un trueque.

Objetos utilizables

Curativos

- *Sal rosa*: 0,5 corazón
- *Poción roja*: 1 corazón
- *Zumo curativo*: 1 corazón
- *Panacea*: 2,5 corazones
- *Bocanada de aire*: 4 corazones
- *Panacea perfecta*: 6 corazones
- *Grito de auxilio*: huyes de la mazmorra sin perder nada

De ataque

Fuego:

- Granada ígnea
- Bomba
- Explosión máxima

Tierra:

- Púas
- Rocas afiladas
- Trampa venenosa

Aire:

- Tornado
- Vendaval
- Rayo

Agua:

- Frío embotellado
- Ventisca de bolsillo

- Frío polar

Elementos del juego

Recetas

Las recetas se pueden ver en el documento excel adjunto como anexo.

Esquema de control

Mando



Teclado

| Tecla | Función |
|----------------|--------------------------------------|
| WASD o Flechas | Moverse |
| X | Interactuar |
| Espacio | Atacar |
| O | Usar objeto equipado |
| I | Cambiar objeto equipado seleccionado |
| P | Bloqueo |

| | |
|---------|------------|
| J | Voltereta |
| Escape | Pausa |
| Control | Inventario |

Estética e interfaz de usuario

La estética del juego es pixel art, recreando los juegos antiguos de 16 bits. El mundo tiene una ambientación Edad Media fantástica, con toques de brujería y alquimia. Esto último se puede ver en la casa del protagonista así como en el libro de recetas que usa.

Las interfaces de usuario recrean los objetos que el jugador usa, teniendo el inventario aspecto de zurrón como el que el jugador lleva colgando de su hombro. Además el libro de recetas tiene el aspecto de un libro de verdad. El otro inventario que tiene el jugador, que es el baúl de su casa tiene el aspecto de un baúl viejo de madera. La interfaz de usuario está cuidada para dar un aspecto más inmersivo al jugador.

APÉNDICE D
Recetario

| | TOTAL | ELIXIR | ATAQUE | CURACION |
|--------------------|-------|--------|--------|----------|
| AGUA | 318 | 240 | 53 | 25 |
| AIRE | 270 | 190 | 68 | 12 |
| FUEGO | 255 | 190 | 59 | 6 |
| TIERRA | 340 | 264 | 58 | 18 |
| RAÍZ DE MANDRÁGORA | 24 | 18 | | 6 |
| VENENO | 23 | 12 | 8 | 3 |
| PÉTALOS DE ANÉMOMA | 48 | 36 | | 12 |
| ACEITE NEGRO | 54 | 36 | 12 | 6 |
| CLAVOS ESPINOSOS | 22 | 18 | 4 | |
| ÁCIDO | 24 | 21 | 3 | |
| AZÚCAR LUNAR | 12 | 4 | | 8 |
| MADERA | 21 | 18 | 3 | |
| POLVO DE HADAS | 46 | 36 | 8 | 2 |
| GRANADA AZUL | 25 | 18 | 4 | 3 |
| ROCÍO | 12 | 6 | 4 | 2 |
| MAGNETITA | 10 | 9 | 1 | |
| DIAMANTE EN BRUTO | 4 | 4 | | |

| Elixir de vida | | | |
|----------------|---|--|-----|
| Oro puro | 1 | | 240 |
| Éter | 2 | | 190 |
| Agua bendita | 2 | | 190 |
| | | | 264 |

| Sal | | Sal dulce | |
|--------|---|-----------|---|
| Tierra | 2 | 2 | 2 |
| Agua | 2 | 3 | 0 |
| | | | 0 |
| | | | 2 |

| Mercurio | | Mercurio nocturno | |
|----------|---|-------------------|---|
| Aire | 3 | 2 | 2 |
| Agua | 2 | 3 | 3 |
| | | | |
| | | | |

| Hierro | | Pirita | |
|--------|---|--------|---|
| Tierra | 3 | 1 | |
| Fuego | 2 | 2 | 2 |
| | | | 2 |
| | | | 3 |

| Pirita colorada | | Frasco de cristal | |
|-----------------|---|-------------------|---|
| Pirita | 2 | 6 | |
| Sal dulce | 3 | 1 | |
| Magnetita | 1 | 3 | 3 |
| | | | 1 |

| Azufre | | Azufre corrosivo | |
|--------|---|------------------|---|
| Fuego | 3 | 1 | |
| Aire | 3 | 3 | 3 |
| | | | 3 |
| | | | |

| Jarabe escarchado | | Sal rosa | |
|-------------------|---|----------|---|
| Mercurio | 2 | 4 | 4 |
| Azúcar lunar | 2 | 6 | 0 |
| Veneno | 2 | 0 | 4 |

| Jarabe ácido | | Poción roja | |
|-------------------|---|-------------|---|
| Jarabe escarchado | 1 | 4 | 3 |
| Azufre corrosivo | 2 | 12 | 0 |
| | | 6 | 3 |
| | | 0 | 1 |

| Panacea | | Panacea perfecta | |
|--------------------|---|------------------|----|
| Sal dulce | 2 | 4 | 12 |
| Raíz de mandrágora | 2 | | 6 |
| Pétalos de anís | 3 | | 0 |
| | | 4 | 8 |

| Zummo curativo | | Bocanada de aire | |
|----------------|---|------------------|---|
| Agua | 2 | 2 | 0 |
| Azúcar lunar | 2 | | 6 |
| Granada azul | 1 | | 3 |
| | | | 1 |

| Zummo zagro | | Rocío sereno | |
|------------------|---|--------------|----|
| Madera | 2 | 4 | 13 |
| Azufre corrosivo | 1 | 3 | 6 |
| Zummo curativo | 2 | 3 | 0 |
| Clavos de espino | 2 | | 4 |

| Colmillo de plata | | | Diamante | | |
|-------------------|----|----|---------------|---|----|
| Mercurio noct | 4 | 8 | Diamante en l | 4 | |
| Pirita colorada | 2 | 12 | Lágrima de dr | 1 | 15 |
| Fuego | 12 | 20 | Azufre | 5 | 15 |
| | | 32 | | | |

| Recetas elementos | | | Eter | | |
|-------------------|--|---|----------------|---|----|
| Agua mágica | | 5 | Agua mágica | 4 | 20 |
| Aire cambiante | | 5 | Aire cambiante | 4 | 20 |
| Fuego fatuo | | 5 | Fuego fatuo | 4 | 20 |
| Tierra seca | | 5 | Tierra seca | 4 | 20 |

| Mandrágora feérica | | | Ácido tartárico | | |
|--------------------|---|---|-----------------|---|----|
| Sal rosa | 2 | 8 | Zumo zagro | 3 | 44 |
| Raiz de mandr | 2 | | Mandrágora f | 3 | 33 |
| Polvo de hada | 4 | | Jarabe ácido | 2 | 21 |
| | | 8 | | | 24 |

| Piedra filosofal | | | Oro puro | | |
|------------------|---|-----|-----------------|---|-----|
| Rocío sereno | 2 | 176 | Piedra filosofa | 1 | 200 |
| Acido tartarico | 3 | 99 | Colmillo de pl | 3 | 150 |
| Pirita colorada | 3 | 75 | Diamante | 1 | 150 |
| | | 128 | | | 224 |

| Llave del castillo | | |
|--------------------|---|----|
| Magnetita | 2 | |
| Clavo espinoso | 4 | 6 |
| Azufre corrosi | 2 | 16 |
| Fuego fatuo | 2 | |

| Lanza de acero | | Alabarda | |
|----------------|----|--------------|----|
| Hierro | 2 | Madera | 4 |
| Madera | 6 | Pirita | 2 |
| Fuego | 6 | Jarabe ácido | 1 |
| | 10 | | 10 |
| | 6 | | 10 |

| Granada ígnea | | Frío embotellado | |
|---------------|---|------------------|---|
| Fuego | 4 | Jarabe escarct | 1 |
| Granada azul | 1 | Agua | 4 |
| Hierro | 1 | Frasco de crist | 1 |
| | 6 | | 3 |
| | 3 | | 1 |

| Bomba | | Ventisca de bolsillo | |
|---------------|----|----------------------|---|
| Granada ígnea | 1 | Frío embotella | 1 |
| Aceite negro | 1 | Rocío | 2 |
| Fuego fatuo | 1 | Agua mágica | 1 |
| | 11 | | 3 |
| | 3 | | 1 |

| Gran falla | | Frío polar | |
|----------------|----|----------------|---|
| Bomba | 2 | Ventisca de br | 1 |
| Azufre corrosi | 1 | Tornado | 1 |
| Madera | 3 | Agua mágica | 1 |
| | 25 | | 3 |
| | 6 | | 1 |

| Grito de auxilio | | Poción de protección | |
|------------------|---|----------------------|----|
| Mandrágora fi | 1 | Frasco de crist | 2 |
| Aire cambiant | 2 | Zumo zagro | 2 |
| Frasco de crist | 1 | Azúcar lunar | 3 |
| Clavos espino: | 2 | | 12 |
| | 9 | | 2 |

| Tridente | | Matalobos | |
|-------------|---|-----------|----|
| Magnetita | 2 | 6 | 8 |
| Madera | 7 | 36 | 12 |
| Vendaval | 3 | 15 | 40 |
| Fuego Fatuo | 3 | | 44 |

| Púas | | Tornado | |
|----------------|---|---------|---|
| Tierra | 4 | 4 | 2 |
| Clavo espinoso | 1 | 7 | |
| Hierro | 1 | 2 | |
| | | 7 | |

| Rocas afiladas | | Vendaval | |
|----------------|---|----------|--|
| Púas | 1 | 2 | |
| Pirita | 1 | 12 | |
| Tierra seca | 1 | 2 | |
| | | 12 | |

| Trampa tóxica | | Rayo | |
|----------------|---|------|--|
| Rocas afiladas | 2 | 8 | |
| Veneno | 2 | 21 | |
| Ácido | 2 | 4 | |
| | | 24 | |

| Elixir encantado | | Candil | |
|------------------|---|--------|---|
| Mandrágora f | 1 | 33 | |
| Azufre corrosi | 3 | 21 | |
| Rocío sereno | 1 | 9 | |
| Panacea perfe | 1 | 20 | |
| | | | 2 |

APÉNDICE E

Glosario

- AI o IA: Inteligencia Artificial
- GBA: GameBoy Advance
- GDD: Documento de diseño de juego (*Game Desing Document* en inglés)
- NDS: Nintendo DS
- NPC: Personaje no Jugador (*Non Playable Character* en inglés)
- ODS: Objetivos de Desarrollo Sostenible
- PNJ: Personaje no Jugador
- TFG: Trabajo Fin de Grado
- RPG: Videojuego de Rol (*Role-Playing Game* en inglés)
- UI: Interfaz de Usuario (*User Interface* en inglés)
- UML: Lenguaje Unificado de Modelado (*Unified Modeling Language* en inglés)
- UPV: Universidad Politécnica de Valencia