



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

Tecnologías de la información

TRABAJO FIN DE GRADO

Videojuego para el estudio de elementos de diseño motivadores
para aumentar la adherencia

Tiberiu Cristian Stavar

septiembre, 2024



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

Tecnologías y Sistemas de Información

Tecnologías de la información

TRABAJO FIN DE GRADO

**Videojuego para el estudio de elementos de diseño
motivadores para aumentar la adherencia**

Autor: Tiberiu Cristian Stavar

Tutor(a): Ramón Hervás Lucas

Co-tutor(a): Luis Cabañero Gómez

septiembre, 2024

La adherencia y motivación en videojuegos
© Tiberiu Cristian Stavar, 2024

Este documento se distribuye con licencia CC BY-NC-SA 4.0. El texto completo de la licencia se puede obtener en <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

La copia y distribución de esta obra está permitida en todo el mundo, sin regalías y por cualquier medio, siempre que esta nota sea preservada. Se concede permiso para copiar y distribuir traducciones de este libro desde el español original a otro idioma, siempre que la traducción sea aprobada por el autor del libro y tanto el aviso de copyright como esta nota de permiso, sean preservados en todas las copias.

Este texto ha sido preparado con la plantilla $\LaTeX$ para Trabajo Fin de Estudios en Ingeniería Informática para la UCLM publicada por [Jesús Salido](#) en el repositorio público Zenodo, DOI: [10.5281/zenodo.4561708](https://doi.org/10.5281/zenodo.4561708), como parte del curso « $\LaTeX$ esencial para preparación de TFG, tesis y otros documentos académicos» impartido en la Escuela Superior de Informática de la Universidad de Castilla-La Mancha. Si la empleas para preparar tu TFG, te agradeceré que la cites [18] e incluyas en tus referencias como se indica en Zenodo con el DOI suministrado para todas las versiones.



TRIBUNAL:

Presidente: _____

Vocal: _____

Secretario(a): _____

FECHA DE DEFENSA: _____

CALIFICACIÓN: _____

PRESIDENTE

VOCAL

SECRETARIO(A)

Fdo.:

Fdo.:

Fdo.:

*A Ana por el apoyo
y a mi madre por todo.*

La adherencia y motivación en videojuegos

Tiberiu Cristian Stavar
Ciudad Real, septiembre 2024

Resumen

Con un mercado de videojuegos cada vez más saturado, es especialmente importante incrementar el interés y aumentar la adherencia de los videojuegos cuando salen al público. Los desarrolladores de videojuegos tienen que ingeniárselas cada vez más para conseguir que la gente no solo pruebe su juego si no que decida seguir jugándolo después.

Con este proyecto se pretende estudiar la eficacia de algunos de los elementos que los diseñadores incorporan a su juego para mantener la adherencia de los jugadores. Esto se hará desde el punto de vista de la motivación, observando las diferencias entre un juego con múltiples elementos que fomentan la motivación extrínseca y otro carente de estas características.

Para ello se desarrollarán dos versiones distintas de un mismo videojuego. Una versión provista de elementos de motivación extrínseca como pueden ser los logros o un ranking y otra completamente libre de estos. Ambas versiones compartirán los componentes que fomentan la motivación intrínseca como las mecánicas de juego o la narrativa. Posteriormente, en una fase de experimentación, se analizarán y compararán los datos de uso de los jugadores de ambas versiones para ver la efectividad de las técnicas empleadas.

Palabras clave: motivación, adherencia, diseño de videojuegos .

Engagement and motivation in video games

Tiberiu Cristian Stavar
Ciudad Real, September 2024

Abstract

Due to the overcrowding of the video game market, it is harder and harder for game designers to keep the focus of the players. Because of that the developers need to constantly search for new ways for players not only to try their games but to get them to keep playing it.

The goal of this project is to study the effectiveness of the strategies the designers use to keep the players engaged. This will be achieved by focusing on the motivation of the players. Observing the difference between a game with many elements that are intended to provide extrinsic motivation and a game with no such elements whatsoever.

For that purpose two versions of the same video game will be developed. One version with extrinsic motivation elements like achievements or a players ranking. The other version has none of those elements. Both versions will share intrinsic motivation elements like the narrative or game mechanics. After the development of the two versions an experimental phase will take place where game metrics will be collected from both games to analyze them to see how effective the included extrinsic characteristics are.

Keywords: motivation, engagement video game design

Contents

Resumen	v
Abstract	vii
List of Figures	xi
1. Introducción	1
1.1. Motivación	1
1.2. Solución propuesta	4
1.3. Estructura del documento	4
2. Objetivos	5
2.1. Objetivo general	5
2.2. Objetivos específicos	5
2.3. Herramientas utilizadas	5
3. Estado del arte	7
3.1. Marco psicológico	7
3.2. Clasificación de los elementos de diseño	11
3.3. Marco tecnológico	13
4. Metodología	17
4.1. Metodología ágil	17
4.2. Planificación	17
4.3. Reuniones	18
5. Resultados	21
5.1. Iteraciones	22
6. Experimento y análisis de datos	39
6.1. Protocolo	39
6.2. Métricas de juego	41
6.3. Datos de encuestas	45
7. Conclusiones	49
7.1. Trabajos derivados y futuros	49
7.2. Objetivos alcanzados	50
7.3. Justificación de competencias adquiridas	50
7.4. Valoración personal	51
Bibliografía	53

A. Anexo: Simulación de Partida**57**

List of Figures

1.1.	Total de juegos publicados por año en <i>Steam</i>	2
1.2.	Aplicaciones publicadas en <i>GooglePlay</i> al mes	2
1.3.	Logros del juego <i>Dark Souls 3</i> en <i>Steam</i>	3
3.1.	Imagen del videojuego Don't Starve donde los elementos más importantes para la supervivencia del jugador están más arriba en el menú de la izquierda.	8
3.2.	Imagen del videojuego serio MINING-VIRTUAL con una pregunta para comprobar el aprendizaje del jugador.	11
3.3.	Logo de <i>Unreal</i>	13
3.4.	Logo de <i>Unity</i>	14
3.5.	Logo de <i>Godot</i>	14
3.6.	Tabla comparativa de las características principales de los motores.	15
4.1.	Diagrama de Gantt del progreso de los objetivos del proyecto.	18
5.1.	Ejemplo de jerarquía de nodos.	21
5.2.	El personaje con su ataque básico y el primer enemigo.	23
5.3.	Los ataques obtenibles por el jugador.	25
5.4.	La apariencia de los enemigos en el juego.	26
5.5.	Diagrama de todas las mejoras obtenibles.	28
5.6.	La pantalla de estadísticas del juego.	30
5.7.	La pantalla de logros del juego	30
5.8.	La configuración de la función ruido para el terreno.	31
5.9.	Extracto del código que determina la celda a colocar en el escenario según el valor del ruido.	32
5.10.	La configuración de la función ruido para los objetos y edificios.	32
5.11.	El edificio con el secreto y unos barriles cerca.	33
5.12.	Ataque básico del jefe.	34
5.13.	Ataque especial del jefe.	35
5.14.	Diagrama de la estructura de la base de datos <i>Firestore</i>	36
5.15.	Versión final del menú de inicio de sesión.	37
5.16.	Versión final del menú principal.	38
6.1.	Tabla con todas las preguntas del cuestionario final.	39
6.2.	Tabla con todas las preguntas del cuestionario final.	41
6.4.	Distribución de las puntuaciones de las partidas en los dos grupos.	42
6.3.	Gráfico con la distribución de los géneros de los participantes.	42
6.5.	Distribución del número de partidas por jugador.	43
6.6.	Comparación de las duraciones de partida de los participantes que solo jugaron una partida.	44
6.7.	Puntuaciones de los participantes que solo jugaron una partida.	44

6.8. Distribución de los enfrentamientos con el jefe final del grupo extrínseco.	44
6.9. Distribución de los enfrentamientos con el jefe final del grupo intrínseco.	45
6.10. Respuestas a las preguntas sobre lo que han sentido jugando del cuestionario según el grupo. Las líneas verdes representan la mediana.	46
6.11. Respuestas a las preguntas del cuestionario sobre su satisfacción del juego y la recomendación de este. Las líneas verdes representan la mediana.	47
A.1. Pantalla de inicio de sesión y creación de cuenta.	57
A.2. Pantalla del menú principal de la versión extrínseca.	57
A.3. Pantalla del menú principal de la versión intrínseca.	58
A.4. Pantalla de los logros con todos ellos bloqueados.	58
A.5. Pantalla con los diferentes ataques y una explicación de los mismos.	58
A.6. Pantalla de las estadísticas del jugador.	59
A.7. Pantalla que aparece al inicio de las partidas y que cuenta la historia del mundo del videojuego.	59
A.8. Visualización de uno de los escenarios aleatorios justo al comenzar la partida. . . .	59
A.9. El comienzo de una oleada de enemigos básicos rodeando a un jugador en apuros. .	60
A.10. Pantallazo que muestra varios tipos de enemigos del juego en gran cantidad. . . .	60
A.11. Pantalla en la que se ven las opciones de mejora al subir de nivel.	60
A.12. Pantalla de los logros pero con varios de ellos ya desbloqueados.	61
A.13. Pantalla del final de una partida con el daño causado y la puntuación obtenida. . .	61
A.14. Pantalla que muestra la disposición del ranking del juego al terminar la fase de experimentación.	61
A.15. Pantalla de los coleccionables con algunos desbloqueados.	62

Introducción

El objetivo es estudiar cómo de eficaces son algunas de las herramientas empleadas por los desarrolladores para aumentar la **adherencia** de los usuarios y lo que fomenta la motivación de los jugadores.

1.1. MOTIVACIÓN

En su aparición los videojuegos eran principalmente vistos como un entretenimiento infantil, un pasatiempo para los niños. Con el tiempo, estos han ido conquistando a personas de todas las edades y cada vez más juegan a videojuegos. Millones de personas juegan todos los días en el mundo.

Uno de los factores más importantes que los desarrolladores deben tener en cuenta a la hora de crear los juegos es la **adherencia** o engagement en inglés. Este término se refiere tanto al sentimiento de seguir queriendo jugar como a lo involucrado o inmerso que se está con el juego. Esto además es cada vez más un problema mayor debido a varios factores externos a los propios jugadores.

Los hechos que se muestran a continuación son algunos de los que muestran que la adherencia sea algo a lo que los estudios de videojuegos deben prestarle atención si quieren que sus juegos tengan éxito.

Uno de los más notables es el enorme aumento en la cantidad de juegos publicados cada año. Claramente visible en *Steam*, la mayor plataforma de videojuegos para ordenador (Figura 1.1), pero también en la aparición de otras como *Itch.io* donde cualquier persona puede subir su videojuego para el público de manera gratuita. Pero quizás el mayor aumento se haya dado en las plataformas móviles, en la Figura 1.2 se puede ver un mercado que se ha inundado por completo de creaciones de desarrolladores de todo el mundo gracias a las facilidades para programar para móvil y sobretodo al enorme mercado al que se accede. El 49% de la población mundial tiene un smartphone ¹ lo que convierte a las aplicaciones en el mayor mercado al que se puede acceder como desarrollador.

Al saturarse el mercado de manera tan visible, cualquier persona promedio que quiera jugar un videojuego, si no está completamente satisfecha con lo que le ofrece un producto puede simplemente buscar otro entre los miles de juegos disponibles. Una forma de ver esto es mediante la base de datos de la plataforma *Steam*, donde se puede estudiar el porcentaje de personas que consiguen los logros de los juegos. Entrando en cualquier juego que tenga logros que indiquen el haber llegado a ciertas partes de la historia o haber completado el juego se ve un gran decremento en el porcentaje de personas que completan un juego o si quiera llegan a la mitad de este comparado con las que lo comienzan. La media de personas [1] que acaba un juego en *Steam* es de 14%, con el juego más completado, *The Walking dead: Temporada dos*, teniendo un 56.1% y el menos completado, *Poly Bridge*, un 0.2%. En el mismo estudio también hacen una recopilación de la mediana de horas dedicadas a los juegos publicados en *Steam* y se observa que el 80% de ellos no superan las 5 horas. Teniendo en cuenta que la mayoría de juegos publicados tienen una duración mayor a esas 5 horas de media,

¹<https://www.gsma.com/newsroom/press-release/smartphone-owners-are-now-the-global-majority-new-gsma-report-reveals/>

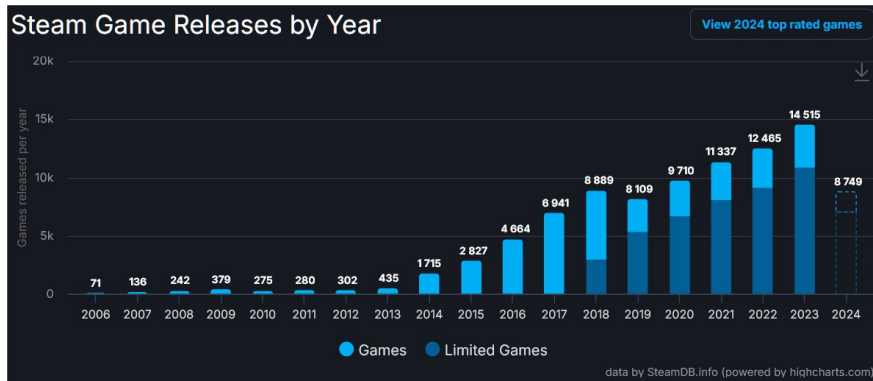


Figure 1.1: Total de juegos publicados por año en *Steam*

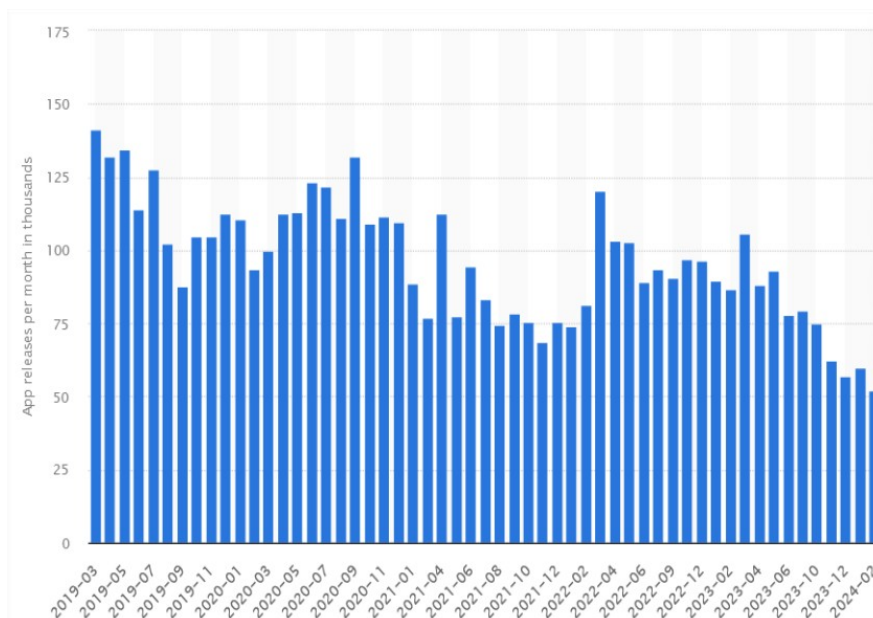


Figure 1.2: Aplicaciones publicadas en *GooglePlay* al mes

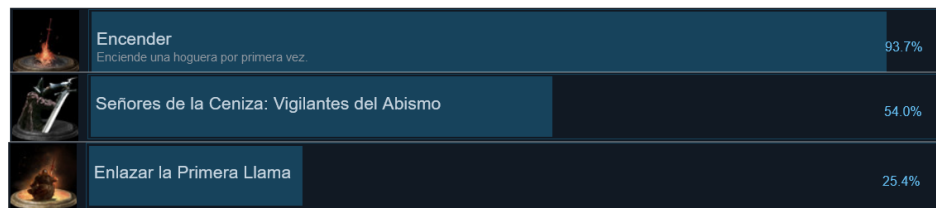


Figure 1.3: Logros del juego *Dark Souls 3* en Steam

queda claro que la mayoría de personas que comienzan un juego le dedican solo el tiempo suficiente para sopesar si es de su gusto o si pasan al siguiente en sus listas.

A continuación, se pueden observar algunas de las estadísticas del juego *Dark Souls 3*. Se trata de un juego de rol y acción de fantasía conocido principalmente por su dificultad, tanto, que en la primera instancia de la saga, *DarkSouls*, el juego llevaba como subtítulo: "*Prepárate para morir*". La saga está diseñada y pensada para ser un reto a los jugadores. Esta filosofía junto a su apartado artístico y jugable le ha hecho a la saga consagrarse como una de las más queridas por el público.

Lo primero que puede llamar la atención de la Figura 1.3 es que ni el 100% de las personas que poseen este juego en la plataforma *Steam*² han dado los primeros pasos para jugarlo. Del 93.7% de personas que han empezado una partida, bajamos al 54% de jugadores en llegar a lo que representa alrededor de un cuarto de la progresión total en el juego. En el último logro de la figura 1.3 se observa que solo un 25.4% de jugadores han completado el juego.

Seguramente no todas las personas que hayan comprado y comenzado una partida en *Dark Souls 3* eran conscientes del concepto principal del juego y sea un gran motivo para la diferencia entre los dos primeros porcentajes. Según palabras del propio creador[12], dar la oportunidad y las herramientas a los jugadores para sobreponerse a dificultades y superar retos, era la prioridad durante el desarrollo ya que eso les proporciona una gran satisfacción. Esa filosofía les ha llevado a tener éxito y reconocimiento internacional aunque como se ha observado más arriba, no todo el mundo comparte esa visión o esta dispuesto a invertir el tiempo necesario para superar los retos que implica.

Si un juego no es lo que el jugador busca, este solo tiene que buscar otro que se adapte más a sus gustos o le proporcione la satisfacción que quiere. Para intentar evitar esto, los diseñadores de videojuegos hacen uso de múltiples técnicas y casi todas ellas están centradas en la búsqueda de motivación para los usuarios. Como se discutirá más en profundidad en capítulos posteriores, la motivación humana se puede dividir en dos categorías:

1. La **motivación intrínseca** siendo aquella innata a las personas, la que empuja a hacer cosas por el mero placer de ello.
2. La **motivación extrínseca** es la influencia externa que se recibe para llevar a cabo algo, ya sea en forma de recompensa o castigo.

Los diseñadores deben tener en cuenta ambos tipos de motivación para el desarrollo de su juego ya que estos están interconectados.

Cuando se habla de adherencia, también hay que hablar de los problemas que puede implicar. El mayor de ellos, el abuso por parte de algunas empresas, de prácticas agresivas para enganchar a los jugadores, más allá del disfrute y transformando a usuarios en adictos a su producto. En [2] hablan de como una gran cantidad de estudios que desarrollan juegos con micro transacciones llegan a basar más de la mitad de sus ingresos en solo 5% de sus usuarios y que además no suele tratarse de jugadores con muchos ingresos sino de personas con problemas de adicción. Esto suele conseguirse con factores que estimulan la motivación extrínseca como nuevos personajes o elementos cosméticos de algún tipo.

²<https://steamcommunity.com/stats/374320/achievements>

Este trabajo se centrará en la búsqueda de una motivación sana y que no conlleve factores adictivos.

1.2. SOLUCIÓN PROPUESTA

Para intentar definir y medir los elementos que determinan la adherencia de los jugadores, se propone desarrollar dos versiones de un mismo videojuego: una versión base y otra donde se añadirán mecánicas que tienen como objetivo fomentar la adherencia mediante la motivación extrínseca. Este trabajo se centra en la búsqueda de una motivación sana por lo que no se implementa ninguna de las prácticas más agresivas de la industria de los videojuegos relacionadas con la búsqueda de adherencia de los jugadores.

Posteriormente en una fase de experimentación con usuarios en la que habrá dos grupos, uno por cada versión del juego, se medirán datos de uso de cada versión para determinar con métricas objetivas, qué ha motivado a cada usuario. También se realizará una encuesta final a los participantes para conseguir información acerca de los sentimientos y sensaciones que han tenido respecto a su versión de juego.

Una vez obtenidos los datos tanto numéricos como subjetivos por parte de los usuarios se procederá a analizarlos y a comparar las diferencias entre las versiones.

Para ver la evolución que ha seguido el desarrollo del videojuego, así como el resultado final, está disponible el siguiente enlace al repositorio del proyecto: <https://github.com/CristianStavar/TFG>

1.3. ESTRUCTURA DEL DOCUMENTO

Por último se muestra la estructura del documento que consta de los siguientes capítulos:

1. **Introducción.** Donde se trata la motivación y se justifica la pertinencia del trabajo.
2. **Objetivo.** En el que se discute el objetivo principal y los específicos que se buscan con este trabajo, además de detallar las herramientas usadas.
3. **Estado del arte.** En este capítulo se analizan las técnicas empleadas por los desarrolladores de videojuegos para aumentar la adherencia de los jugadores y los estudios relacionados con la motivación en las personas, tanto en el ámbito de los videojuegos como fuera de este.
4. **Metodología.** Aquí se explica la metodología y herramientas empleadas para llevar a cabo el proyecto.
5. **Resultados.** Donde se detallan los pasos dados en el desarrollo del videojuego así como las decisiones sobre el diseño del mismo.
6. **Conclusiones.** En el que se realiza una discusión sobre los resultados obtenidos y la justificación de la aplicación de las competencias adquiridas.
7. **Bibliografía.** Lista de las referencias bibliográficas citadas en el texto.
8. **Anexos.** Contenidos auxiliares que complementan del trabajo.

CHAPTER 2

Objetivos

Los objetivos general y específicos que se buscan en este trabajo, serán lo primero a tratar en el capítulo, seguido de las herramientas secundarias utilizadas, pues la más importante, el motor gráfico usado, tiene su propia sección en el siguiente capítulo.

2.1. OBJETIVO GENERAL

El objetivo principal (OP) de este trabajo es diseñar y desarrollar un videojuego que sirva de base para estudiar la influencia y efectividad de los elementos de motivación extrínsecos e intrínsecos en los videojuegos. El videojuego se usará en una fase de experimentación con usuarios para medir las diferencias entre la motivación de los usuarios a seguir jugando cuando están presentes los elementos extrínsecos y cuando no.

2.2. OBJETIVOS ESPECÍFICOS

Se divide el objetivo generales varios específicos para facilitar y delimitar el desarrollo del proyecto.

1. **Realizar una revisión del estado del arte sobre el diseño de videojuegos, la adherencia y la motivación de los jugadores.** Este es el primer paso para entender los mecanismos que emplean los desarrolladores para mantener el interés de los jugadores y emplearlos de forma correcta. Se buscarán tanto artículos científicos relacionados con el diseño de videojuegos y la motivación, así como entrevistas y declaraciones de diseñadores sobre las técnicas y recursos que mejor funcionan con sus objetivos.
2. **Estudiar, definir y clasificar los elementos de diseño pertenecientes a cada una de las motivaciones para su implementación.** La motivación de los jugadores está dividida en dos tipos diferentes, la intrínseca y la extrínseca, este objetivo busca la correcta identificación de técnicas de diseño de videojuegos que pertenecen a cada uno de esos tipos.
3. **Desarrollar los elementos comunes a las dos versiones del videojuego.** Diseñar e implementar el juego base teniendo en cuenta las diferencias decididas en el objetivo anterior.
4. **Implementar los cambios a cada una de las versiones para fomentar cada tipo de motivación a los jugadores.** Añadir todas las características asociadas a la motivación extrínseca en su correspondiente versión.
5. **Determinar la influencia de cada elemento en la adherencia de los jugadores mediante la recogida de datos y estudio posterior de los resultados a través del videojuego.** Este último objetivo consiste en la utilización del videojuego para la recogida de datos y estudio posterior de los resultados para determinar cuál versión consigue mantener mejor el la adherencia a corto plazo de los jugadores.

2.3. HERRAMIENTAS UTILIZADAS

De la herramienta principal utilizada para el desarrollo del juego, el motor gráfico, se hablará en detalle en los siguientes capítulos. Aquí el punto de atención esta en medios software auxiliares.

- **Microsoft Teams:** Usado tanto para la comunicación entre alumno y directores como para la coordinación y compartición de archivos.
- **Overleaf:** Empleado para la elaboración de la memoria partiendo de la plantilla creada por *Jesús Salido*.
- **Gimp:** Programa de edición de imágenes usado en la modificación y creación de elementos gráficos para el videojuego desarrollado.
- **Milanote:** Aplicación utilizada para la organización de tareas en el desarrollo.
- **Github:** Empleado para alojar los distintos prototipos realizados
- **Zotero:** Gestor de referencias utilizado para la fácil adición de estas a Overleaf.
- **Notepad++:** Editor de textos empleado para la modificación de algunos archivos.
- **Microsoft Excel:** Utilizado para realizar análisis de los datos obtenidos y la realización de gráficos.
- **Google Drive:** Empleado en el alojamiento y la sincronización de archivos entre dispositivos.

En cuanto a los recursos externos usados en el videojuego, tenemos visuales y audio.

- Los assets visuales son todos creados por el artista Kenney¹ que tiene a disposición de los usuarios y de uso libre una gran cantidad de elementos tanto gráficos como sonoros.
- En cuanto al audio se ha obtenido de un artista, Peritune², que pone sus obras para el uso libre y gratuito.

¹<https://www.kenney.nl/>

²<https://peritune.com>

Estado del arte

Antes de seguir, debemos hablar del concepto clave de este trabajo, la motivación. Según la American Psychological Association APA [13], la motivación se define como la fuerza que dota de intención o sentido al comportamiento y que está presente a nivel consciente o subconsciente. En este proyecto se estudia la motivación como factor que determina las ganas de seguir jugando un videojuego y las razones para que a un usuario le apetezca hacerlo.

3.1. MARCO PSICOLÓGICO

Para entender realmente el funcionamiento de la motivación y la forma en la que afecta a las personas hay que distinguir dos tipos principales.

- La **motivación intrínseca** que implica la realización de una actividad por la satisfacción que la propia actividad otorga. Cuando se lee un libro o ve una película, se hace por gusto, porque un sentimiento ha nacido en la persona que la ha impulsado a ello.
- La **motivación extrínseca** es la incentivación externa de participar en una actividad por la expectación de una recompensa o castigo. Cuando se está en el instituto y mandan leer un libro porque habrá un examen sobre el, una fuerza externa ha sido la que ha iniciado la acción aunque posteriormente a la persona si le guste el libro. Lo mismo pasa con el trabajo, aun siendo una actividad que resulte agradable o incluso se disfrute, el motivo para llevarla a cabo es la expectativa de un pago.

3.1.1. Teoría de la auto-determinación

Una de las teorías más extendidas y estudiadas relacionadas con la motivación en ámbitos general y también en los videojuegos, es la llamada **Teoría de la Auto-Determinación** TAD [4]. Esta teoría sostiene que la psique humana está basada en tres necesidades básicas que en conjunto cumplen con la función de hacer sentir a uno mismo que se está consiguiendo un crecimiento personal o que se tiene un propósito.

La TAD se basa en la necesidad de la motivación intrínseca para lograr esos sentimientos de crecimiento y cumplimiento de metas propias. Esto viene dado por tres elementos principales:

- **Autonomía.** Sentir que se tiene libertad y opción real de tomar decisiones y hacer cambios.
- **Competencia.** Saber y ser consciente de las capacidades y conocimientos que se tiene. Esto no solo proporciona validación hacia uno mismo si no también seguridad para actuar.
- **Vinculación** o pertenencia a un grupo. Tener relaciones sociales o personales satisfactorias.

En [19] comentan que varios estudios encontraron que tanto en ámbitos académicos como profesionales las personas que cumplían satisfactoriamente los tres puntos anteriores desempeñaban mejor sus labores y conseguían satisfacción en ello. Mientras que en caso contrario las personas que les faltaba alguno o todos los elementos que definen la auto determinación, tenían peores



Figure 3.1: Imagen del videojuego Don't Starve donde los elementos más importantes para la supervivencia del jugador están más arriba en el menú de la izquierda.

resultados a nivel académico, perdían antes la motivación con las actividades deportivas y personales y presentaban un mayor malestar físico o mental.

A raíz de conclusiones como la anterior se han ido realizando múltiples estudios enfocados a analizar el TAD en el mundo de los videojuegos, como este [8] en el que el cumplimiento de los tres pilares por parte de la experiencia jugable significa un mayor tiempo de juego. En concreto la **vinculación**, ha sido determinada como el factor más importante, dejando claro que la socialización es muy importante para las personas.

Con esos datos en mente y sabiendo la importancia que tienen, los diseñadores de videojuegos siempre deben tener en cuenta los dos tipos de motivación que hay para mantener el interés de los jugadores. En una entrevista a los desarrolladores del videojuego *Don't Starve* [11], estos decían que, al darles total libertad a sus jugadores, estos no sabían qué hacer y se perdían, por lo que desaparece su interés por el juego. Al tratarse la exploración del pilar fundamental del juego, era algo que debían subsanar si querían tener éxito. Para intentar corregir el problema, añadieron una pequeña lista de misiones para guiar a los jugadores en sus primeros pasos. Para su desgracia los resultados de esto fueron como los que recopila Alfie Kohn¹ sobre la inclusión de elementos de motivación extrínseca a corto plazo. El mero hecho de añadir elementos de motivación extrínseca como recompensas es contraproducente muchas veces ya que puede llevar a efectos indeseados como la pérdida de interés una vez acabadas las recompensas. Esto es así incluso en actividades que resultan interesantes para las personas antes de añadirles una motivación externa.

En el caso de *Don't Starve*, al ver el listado de misiones, los jugadores que probaban el juego se centraban únicamente en eso, obviando todo lo demás e ignorando los aspectos que los desarrolladores consideraban más importantes como la exploración libre del mundo y el descubrimiento de nuevos objetos y mecánicas. Además, una vez completadas todas las misiones, las personas volvían a sentirse perdidas. La forma de conseguir que los jugadores disfrutarán de verdad del juego fue mediante el diseño de juego y de la interfaz. En lugar de dar un listado de objetivos y requisitos se dejan pequeñas pistas visuales y jugables de cómo hacer algunas y cuáles deberían ser los siguientes pasos de los jugadores (Figura 3.1). Esto les lleva de una motivación extrínseca encubierta a una completamente intrínseca pues sienten la satisfacción de descubrir las cosas por si mismos.

Ese tipo de relación entre las dos motivaciones llevan al diseñador de videojuegos y profesor de universidad Jesse Schell² a afirmar que las motivaciones extrínsecas e intrínsecas están muy

¹<https://web.archive.org/web/20120414070043/http://www.csse.uwa.edu.au/pk/motivation.html>

²gamedeveloper.com

conectadas. Simplemente añadir un elemento extrínseco a un juego no va a hacer que el jugador mantenga el interés a largo plazo, solo a corto y mientras ese elemento externo siga existiendo. En cambio si el jugador asocia ese elemento extrínseco a algo personal, eso puede convertirse en un ancla para la motivación intrínseca. El ejemplo más fácil de entender y además el más común es mediante el factor social. Si un grupo de amigos empieza a jugar a un videojuego con una clasificación de puntuaciones, lo que en un comienzo es intentar quedar alto en la clasificación puede transformarse en una actividad social donde el ranking ya no importa si no la interacciones que crea en el grupo .

También basado en la TAD hay estudios que buscan las motivaciones de jugadores competitivos, tanto profesionales como aficionados.

Deci y Reeve [17] llegaron en su estudio a la conclusión de que a las personas competitivas la motivación extrínseca relacionada con competir les era más efectiva a largo plazo, pero seguía sin ser comparable con el interés genuino de la motivación intrínseca.

En el estudio [7], centrado en la búsqueda de las motivaciones de jugadores profesionales de eSports, llegan a la conclusión de que la motivación intrínseca de los participantes era inversamente proporcional con su *MMR* (medida numérica que indica lo bueno que es un jugador) y a cuánto habían jugado a lo largo de su carrera. Esto es debido a que llegaba un momento en el que dejaban de ver el juego como un juego y se convertía únicamente en su trabajo o aspiración. También encontraron que los factores más importantes en la motivación de los profesionales era sentir una libertad de decisión y sobre todo la sensación de ir mejorando y venciendo a los oponentes dejando clara su habilidad en el juego coincidiendo esto con varios puntos del **GameFlow** la otra gran teoría que busca entender la motivación.

3.1.2. GameFlow

Otra forma de estudiar la motivación de las personas es mediante la teoría del *Flow* o **fluir** de Csikszentmihalyi [3]. Estando enfocado más en la obtención de la felicidad sigue siendo una teoría válida en el ámbito de este proyecto ya que habla del disfrute de las personas. Su validez y amplia aceptación en el mundo de los videojuegos se demuestra en [20] donde desarrollan el **GameFlow** como medida del disfrute de los usuarios mientras juegan a un videojuego.

El GameFlow se basa en ocho indicadores o elementos que determinan si el usuario disfrutará del juego.

- **Concentración.** Cuanto más haya que concentrarse en un tarea, significa que más atención deberá prestarle el jugador. Si no se da motivos para emplear toda la atención ya sea mediante elementos jugables, narrativos o visuales, los jugadores pueden distraerse con fuerzas externas al juego.
- **Desafío.** La dificultad del juego debe ajustarse adecuadamente a la habilidad de los usuarios o la experiencia de estos será negativa. Si un juego resulta demasiado difícil para alguien, se traducirá en un aumento de la frustración. En cambio si la dificultad es demasiado baja el resultado será la indiferencia del jugador o incluso el aburrimiento. Muy relacionado con esto está el concepto de ritmo o *pacing*[23] en inglés. Si se exige el máximo de la habilidad de los jugadores constantemente pueden llegar a sufrir fatiga mental. Para evitarlo, los diseñadores deben intercalar esas fases que requieren de toda la habilidad de los jugadores, con otras más tranquilas, para no exigir demasiado a los usuarios y que estos tengan descansos mentales.
- **Habilidad del jugador.** Muy relacionada con el punto anterior está la sensación de los usuarios de mejorar y aprender a medida que van jugando. Para ello una vez más el ritmo de juego es importante, la forma en la que se presentan los desafíos y posibles tutoriales a los jugadores influye en su experiencia. Los jugadores deben aprender de forma natural mientras juegan, no mediante enormes tutoriales en forma de texto que pueden llegar cansar o incluso frustrar [14]. De la misma manera, no se debe plantear un desafío en el que las herramientas necesarias para superarlo no han sido debidamente explicadas. Otro factor importante son las recompensas,

estas deben ser acorde con la habilidad exigida al jugador para ser satisfactorias. Si después de una sección de mucha intensidad o dificultad se premia a los jugadores con algo que no consideran suficiente, implicará un aumento de frustración por parte de los usuarios.

- **Control.** Este punto es similar a la autonomía de la TAD, los jugadores deben sentir que lo que intentan o quieren realizar en el juego es posible. Si parece que se puede hacer algo en el juego, se debe poder hacer. Por ejemplo, la habilidad de salto y un obstáculo con una altura menor a la del salto dará al jugador la sensación de que es un objeto saltable, pero si hay un muro invisible que impide el salto, se romperán las expectativas del jugador.
- **Objetivos claros.** Un juego no solo debe tener objetivos, sino que además deben ser claros para el usuario. Esto puede ser o bien un objetivo principal a largo plazo acompañado de otros secundarios, o una cadena de objetivos más pequeños que lleven a un gran final.
- **Retroalimentación.** Para concentrarse en una tarea es vital saber si se está haciendo bien, si se progresa, y eso se consigue mediante la retroalimentación que recibe el usuario. Tanto cuando algo se hace bien o se hace mal, el jugador debe ser informado adecuadamente de ello.
- **Inmersión.** Consiste en sentirse involucrado con el videojuego sin tener que esforzarse para ello. Que los usuarios pierdan la noción de lo que hay a su alrededor y tengan puesta su atención en la experiencia que les proporciona el juego.
- **Interacción social.** La interacción social es una de las grandes fuerzas que mueven a los seres humanos y no es menos en el ámbito que se trata en este proyecto. Ya sea fomentando la cooperación, competición o conexión, los juegos que promueven la interacción con otras personas son más disfrutables.

3.1.3. Gamificación y juegos serios

Dos ámbitos muy relacionados donde también se habla sobre la motivación de los jugadores es la gamificación y los juegos serios. Esto es así en parte porque es más fácil aprender algo cuando se está motivado ya sea por el tema estudiado o por los métodos empleados. Esto segundo además siendo más importante como se ha demostrado por múltiples estudios mencionados donde se ha visto un mayor rendimiento e interés por parte de los participantes expuestos a métodos de aprendizaje basados en juegos [5, 19].

Tuzun [22] demostró que los estudiantes que aprendieron con un videojuego, además de obtener mejores resultados, estaban más motivados a seguir aprendiendo posteriormente por su cuenta que los que lo hicieron con métodos tradicionales.

En [9] definen **gamificación** como la aplicación de elementos de juegos a sistemas o actividades para mejorar la participación y motivación de los usuarios. Esto se puede conseguir con simples cambios en la interfaz de un programa para que parezca más un juego o añadiendo sistemas y mecánicas enteros provenientes del diseño de videojuegos. En la misma revisión sistemática encuentran que el uso principal de la gamificación es la enseñanza en línea. Gamificar una herramienta de aprendizaje no implica que vaya a funcionar mejor por que sí, en [10] advierten de que estos elementos de juego han de estar correctamente implementados e integrados para formar una experiencia disfrutable. Al igual que ocurre con los videojuegos para el entretenimiento, debe haber un diseño concienzudo por parte de los desarrolladores. Si se añaden elementos de gamificación a un sistema ya existente sin hacer ninguna adaptación, estos acabarán por formar una capa extra de trabajo a los usuarios en lugar de hacer la experiencia más agradable.

Por su lado, los **juegos serios** son aquellos juegos completos que se utilizan con un fin diferente al entretenimiento, como puede ser entrenamiento o simulación [10]. La principal diferencia entre gamificación y juegos serios es que estos últimos son videojuegos propiamente dichos mientras que la gamificación es un concepto que abarca cualquier técnica, programa o instrumento que incorpore elementos originarios de los videojuegos. En [9] encuentran que la mayoría de juegos serios están

dirigidos a la rama de la salud, tanto mental como física, pero también se usan ampliamente en educación y formación para profesionales.

En [16] se encuentra que el uso de un juego serio para la formación de personal de enfermería pediátrica es más efectivo que métodos tradicionales. Habiendo una mayor retención de conocimientos y mejora en la toma de decisión por parte del grupo que uso el juego serio en su entrenamiento.

En el artículo [6] se creó un videojuego serio para el entrenamiento de mineros mediante la realidad virtual Figura 3.2. En su estudio se sostiene que la mayoría de accidentes del sector minero vienen por la falta de entrenamiento de los trabajadores, relacionado tanto con la toma de decisión, la correcta identificación y evaluación de peligros y el uso de maquinaria pesada o peligrosa. En las pruebas con el videojuego que se desarrolló, *MINING-VIRTUAL*, se observó que las personas que jugaban a videojuegos como pasatiempo estaban más cómodas con el manejo del juego serio pero los resultados finales de la experiencia eran similares entre jugadores habituales y no jugadores. Al finalizar la fase de entrenamiento con el juego serio, se hizo un cuestionario para calificar su experiencia, en la que ambos grupos concluyeron que era una actividad útil y que creían debía ser implementada para la formación de personal minero. Además el experimento se consideró exitoso, pues en las preguntas sobre lo aprendido en la simulación los resultados fueron muy buenos para los dos grupos.

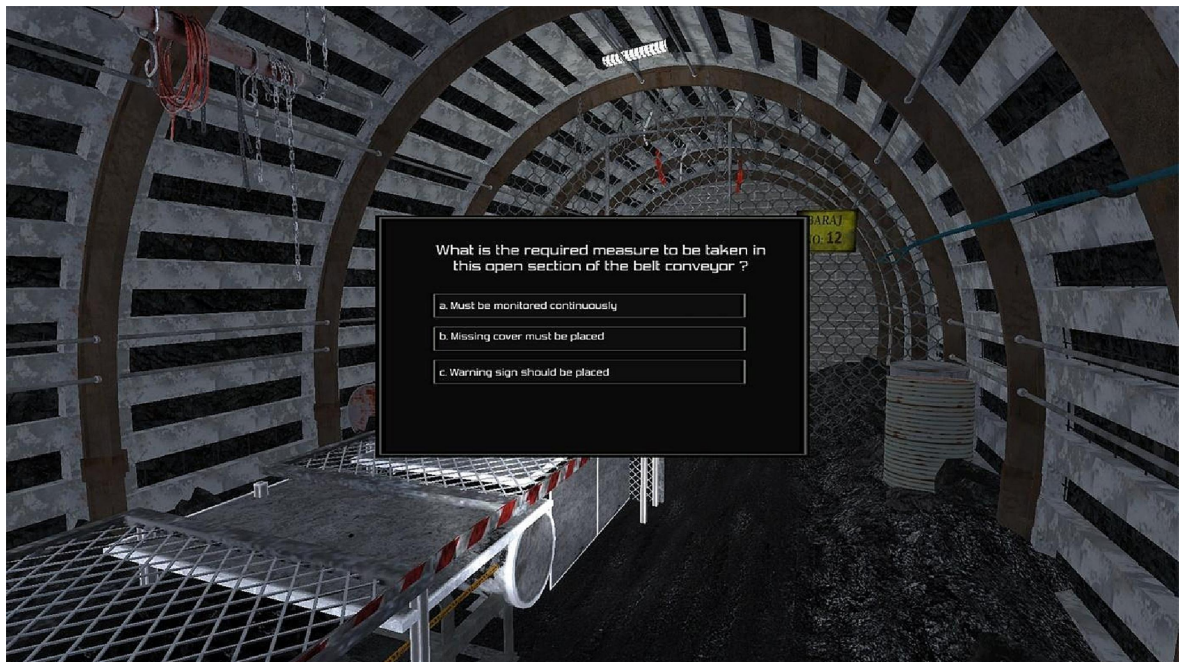


Figure 3.2: Imagen del videojuego serio MINING-VIRTUAL con una pregunta para comprobar el aprendizaje del jugador.

Como se puede observar, los juegos serios son un tema en constante estudio y desarrollo gracias a que se está demostrando que realmente funcionan como herramienta para el aprendizaje y entrenamiento.

3.2. CLASIFICACIÓN DE LOS ELEMENTOS DE DISEÑO

Después de estudiar las tendencias y consensos de la motivación en el diseño de videojuegos, llega la hora de clasificar los elementos pertenecientes a cada tipo de motivación. Algunas características no gozan de un consenso absoluto por parte de la comunidad de desarrolladores y diseñadores en cuanto a donde pertenecen. Este hecho se comentará en los casos en los que corresponda y se explicará en detalle que ha hecho que se coloque en un tipo u otro de motivación.

Motivación intrínseca

Se ha incluido todo aquel elemento que no interrumpa de una forma u otra la experiencia de juego o no sea necesario interrumpirla para verlo y que se pueda considerar una recompensa en sí misma.

- **Mecánicas de juego.** Son algo innato al videojuego y otorgan satisfacción por sí mismas, sobretodo si se trata de un tipo de juego que gusta al usuario.
- **Narrativa.** La historia contada tanto de manera directa con texto, vídeo o audio, como con elementos visuales en el mundo del videojuego como pueden ser edificios, estatuas u objetos colocados de tal manera que dejen ver que ahí ha ocurrido algo.
- **Exploración.** Relacionado tanto con la TAD como el GameFlow, da a los jugadores satisfacción mediante la búsqueda y el descubrimiento de lugares y objetos. Es la manera más eficaz de generar curiosidad a los usuarios, dar la posibilidad de explorar el mundo creado.
- **Arte.** Aunque quizás este se podría incluir en la narrativa, pues puede llegar a ser una forma única de contar una historia, se ha separado porque no siempre tiene la misma importancia. Hay juegos completamente visuales donde lo que se ve en pantalla es lo más importante y otros en los que es simplemente un medio para conseguir otro objetivo, además no se trata de un gusto universal. No hay un estilo definido que los diseñadores puedan elegir para su juego y que sepan que va a provocar que los jugadores quieran jugarlo pues cada uno tiene sus gustos propios.
- **Aleatoriedad.** Esta característica ha sido una de las más discutidas en cuanto a su pertenencia. Al poder usarse para crear una escasez artificial de algo que el jugador pueda querer y verse obligado a jugar repetidamente una misma sección o nivel si lo quiere conseguir, se puede ver como una característica de motivación extrínseca, pues hay una fuerza externa entre el usuario y su objetivo. Ese tipo de enfoque suele ser utilizado en juegos de rol de acción, donde una pieza de equipo u objeto especial solo es obtenible por un enemigo o en un lugar concreto. Por otro lado la aleatoriedad puede ser vista como una mecánica de juego en sí misma, tanto para la creación de escenarios procedurales o por que se usen elementos como dados. Al final por el uso que se le da en el videojuego del proyecto se ha decidido incluirlo en los elementos de motivación intrínseca.

Motivación extrínseca

Aquí están incluidas las características funcionan como recompensa por parte de los diseñadores o que a su vez otorgan una como se explica a continuación.

- **Logros.** Una de las características de motivación extrínseca más usadas en los videojuegos. La mayoría de juegos publicados hoy día tiene un sistema de logros interno o en la propia plataforma en la que se publican. Los logros sirven de motivación a dos grupos distintos de personas, las coleccionistas que buscan obtener todos los logros posibles y las competitivas que pueden querer un logro que indica que han conseguido superar una hazaña difícil en el juego.
- **Desafíos.** Son similares a los logros en cuanto a que buscan la parte competitiva de las personas, pero estos además, normalmente tienen asociada una recompensa dentro del juego. Completar un desafío concreto puede significar conseguir un objeto especial y único o algún tipo de moneda dentro del juego. En el videojuego desarrollado para este proyecto, los desafíos están incluidos en forma de logros una vez superados.
- **Ranking.** Las clasificaciones en los videojuegos son una manera muy efectiva de generar competitividad entre los participantes y empujarlos a seguir jugando para superar las mejores puntuaciones. También suelen implicar un componente social del que se hablará más adelante.
- **Estadísticas.** Dar información extra a los jugadores en forma de estadísticas puede estimular el afán coleccionista de varias maneras. Viendo las estadísticas, los usuarios pueden darse cuenta de que hay algo que todavía no han conseguido en el juego o superar una marca personal

ya sea de puntuación o de algún objetivo interno del juego. Además se consigue satisfacer la curiosidad de los jugadores por saber como están avanzando en el juego o simplemente ver sus resultados a lo largo del tiempo.

- **Coleccionables.** Objetos añadidos por los diseñadores para dar otro objetivo a los jugadores. Si un jugador da por finalizada su experiencia con un juego pero ve que hay algún coleccionable que no ha conseguido puede hacer que siga jugando para encontrarlo. Estos pueden estar escondidos físicamente en el mundo del videojuego o detrás de alguna barrera de habilidad a superar por los jugadores.

En último lugar hay que hacer una mención especial al **factor social**. Este puede fomentarse con elementos de motivación extrínseca como los desafíos o las clasificaciones de mejores puntuaciones, pero también puede aparecer sin necesidad de estos. Al ser un elemento de difícil clasificación por estar interconectado con el resto de características, se incluye aparte.

El factor social no es únicamente algo que se ve fomentado por los otros elementos discutidos, sino también una fuente propia de motivación interconectada a los factores anteriores. Al verse un jugador en un entorno en el que se hable de un videojuego concreto, si este no lo ha probado se verá influido a ello por su círculo social y, si sí lo ha jugado, los usuarios se retroalimentarán entre ellos para seguir jugándolo y explorar su mundo o conseguir mayores puntuaciones. Esto es lo que hace al factor social tan influyente, su relación con todas las demás características y sobretodo su capacidad de cambiar un elemento de motivación extrínseca y convertirlo en algo intrínseco gracias a la influencia positiva de otras personas.

3.3. MARCO TECNOLÓGICO

A continuación, se va a profundizar sobre la herramienta más importante que se ha de usar en la realización de este proyecto, el motor o framework para el desarrollo del videojuego. Para ello se realizó un estudio de las capacidades y dificultades de uso de las plataformas de propósito general más usadas y accesibles.

UNREAL ENGINE



Figure 3.3: Logo de Unreal

Motor gráfico desarrollado por la empresa Epic Games y ampliamente usado en la industria de los videojuegos y del cine por su gran capacidad de crear y soportar gráficos ultra realistas. Destaca en su uso para el 3D, entre otras cosas, gracias a su potente sistema de iluminación, que como contraparte implica que los juegos necesitan de una máquina con bastante potencia para funcionar correctamente. Las dificultades que presenta son principalmente la curva de aprendizaje que tiene debido a su alta complejidad.

- **Licencia.** Uso gratuito siempre y cuando los ingresos que aporte el juego o programa desarrollado sean inferiores a un millón de euros anual.
- **Lenguaje.** Usa C++, así como una herramienta de programación visual llamada *Blueprints*. C++ es un lenguaje de programación basado en C que lo amplía y añade características para la manipulación de objetos. Son destacables su alto rendimiento y la gran cantidad de funcionalidades que posee. Como contrapunto, está su complejidad de uso y dificultad de aprendizaje.
- **Complejidad.** Alta complejidad debido a la gran cantidad de características de las que dispone sobretodo si se decide usar con C++ en lugar de *Blueprints*
- **Comunidad.** Gran cantidad de tutoriales tanto propios como de los usuarios.
- **Tamaño.** Ocupa mas de 40GB en disco duro sin ningún plugin ni herramienta extra.

Unity



Figure 3.4: Logo de Unity

Motor gráfico de uso muy extendido entre desarrolladores independientes y personas que crean juegos por afición. Relativamente sencillo de usar y gran cantidad de documentación y ayuda disponible en internet. Usado tanto para videojuegos en 3D como en 2D.

- **Licencia.** Uso gratuito para entidades que tengan ingresos anuales menores a 100.000\$. Si se superan esos ingresos hay que pagar una licencia de 2.000\$ al año para individuos y más si se es una empresa.
- **Lenguaje.** Usa C# como principal lenguaje aunque también soporta el uso de C++. C# es un lenguaje orientado a objetos creado por Microsoft. Su sintaxis es parecida a la de Java y tiene una curva de aprendizaje media.
- **Complejidad.** Tiene una complejidad y dificultad de uso media.
- **Comunidad.** Dispone de Amplia documentación y multitud de ayudas y tutoriales en internet. Tiene una tienda de assets con gran cantidad de herramientas y elementos visuales.
- **Tamaño.** Tiene un tamaño en disco relativamente bajo, ocupando unos 4GB.

Godot



Figure 3.5: Logo de Godot

Motor open source y de uso libre principalmente usado para 2D, pero con capacidad para el 3D. Ofrece soporte oficial para C#, C++ y otros lenguajes están soportados usando addons creados por la

comunidad pero tiene su propio lenguaje de programación optimizado para las funcionalidades que ofrece y es el que recomiendan, GDScript.

- **Licencia.** Completamente gratuito y open source.
- **Lenguaje.** Usa uno propio llamado *GDScript*. Lenguaje orientado a objetos y con sintaxis similar a Python. Es de fácil aprendizaje y está en constante mejora siguiendo la evolución del motor.
- **Complejidad.** Gran facilidad de uso debido a su filosofía de mantener simples las funciones más básicas y frecuentemente utilizadas.
- **Documentación.** Menos tutoriales y ayuda en internet, pero amplia documentación propia. Dispone de múltiples addons pero en menor cantidad que las otras dos alternativas.
- **Tamaño.** Es el que menos ocupa de los tres con menos de 150MB en disco.

También estaba la opción de desarrollar el juego directamente en HTML y Javascript, pero no fue considerada debido que el autor está menos familiarizado con su uso y requería de un mayor estudio previo.

	Unreal	Unity	Godot
Licencia gratuita	Si	Si	Si
Lenguaje	C++/ Blueprints	C# / C++	GDScript/C#/C++
Complejidad	Alta	Media	Baja
Documentación	Amplia	Amplia	Media

Figure 3.6: Tabla comparativa de las características principales de los motores.

Las principales características que se buscan del motor gráfico son la posibilidad de exportar los juegos a Web para que sean más fácilmente accesibles por todo el mundo y la sencillez de su uso. La primera parte la cumplían todos los motores mencionados pero la facilidad de uso y el previo conocimiento de sus capacidades y lenguaje propio hicieron que el motor elegido fuese Godot.

La segunda decisión importante fue la elección del método para conseguir que los participantes del experimento pudiesen acceder fácil y directamente al juego, las posibilidades eran alojarlo en un servidor o web propia o algún servicio de terceros. Se consideraron las siguientes opciones:

- Itch.io una plataforma online destinada a la publicación y venta de todo tipo de bienes digitales. Su uso más extendido es como tienda para videojuegos independientes. Ofrece herramientas para publicar juegos en formato *HTML* y que sean directamente jugables en la propia página. Es de uso completamente gratuito.
- La plataforma GitHub también presenta una opción haciendo uso de su servicio *Pages*, que permite a los usuarios crear sus propias páginas web de manera directa, sencilla y gratuita.
- Por último se estudió *Neocities*, un servicio de hosting de código abierto y uso gratuito que permite la ejecución de juegos en HTML.

En este paso la decisión fue más rápida y clara. Se optó por usar la plataforma Itch.io como host para el juego, ya que es la única que maneja de forma automática todas las opciones de configuración más complejas. Otra función que también es muy útil y será utilizada en la fase de experimentación, es que permite ocultar el juego y que sea accesible solo mediante un enlace especial. Con esto se podrá disponer de un entorno más controlado.

Metodología

Capítulo dedicado a describir la metodología empleada en el desarrollo del proyecto. Al ser una decisión importante, fue abordada en una de las primeras reuniones y se optó por implementar una metodología ágil.

4.1. METODOLOGÍA ÁGIL

Este término engloba a una gran cantidad de formas de desarrollo pero que se basan en cuatro pilares.

1. **Personas e interacciones.** Primero en aparecer por su importancia, sin las personas correctas no se puede llevar a cabo un proyecto.
2. **Software que funciona antes que documentación extensa.** Prima la creación del programa sobre la definición y redacción de la documentación.
3. **Colaboración con los clientes.** La adecuada comunicación con las personas es fundamental para el cumplimiento de los requisitos del cliente.
4. **Respuesta a los cambios.** Es más importante ser capaz y poder adaptarse a problemas inesperados que seguir un plan abocado al fracaso. La flexibilidad envuelve a los tres puntos anteriores, todo debe planificarse y hacerse teniendo en mente que seguramente haya modificaciones por el camino.

Muchas de las ideas y buenas prácticas de las metodologías ágiles están enfocadas a equipos pero eso no significa que sus fundamentos no sean aplicables a desarrollos en solitario. Basado en las prácticas ágiles está el **desarrollo de prototipos incrementales**. Este consiste en la creación del proyecto a base de pequeñas iteraciones en las que se trabaja sobre los resultados anteriores. Se comienza con las funcionalidades básicas y más importantes y se desarrollan los futuros prototipos partiendo de ahí. Tiene la gran ventaja de poder modificar elementos rápidamente pues una vez se tiene un prototipo y se prueba los cambios que haya que realizar serán sobre las funcionalidades nuevas. De esta manera se tiene estabilidad en el proyecto sin la necesidad de una extensa y precisa planificación.

4.2. PLANIFICACIÓN

Se ha dividido el proyecto en las 9 iteraciones siguientes que en el próximo capítulo serán explicadas en profundidad:

- **Iteración 0:** Dedicada a la elección del género del videojuego y a todas las decisiones de diseño necesarias para iniciar el desarrollo.
- **Iteración 1:** Se implementa el primer prototipo con el movimiento del personaje del jugador, un enemigo y el ataque básico del usuario.
- **Iteración 2:** Se incrementa el arsenal del jugador añadiendo más ataques y se añade el sistema para que el jugador suba de niveles venciendo enemigos.

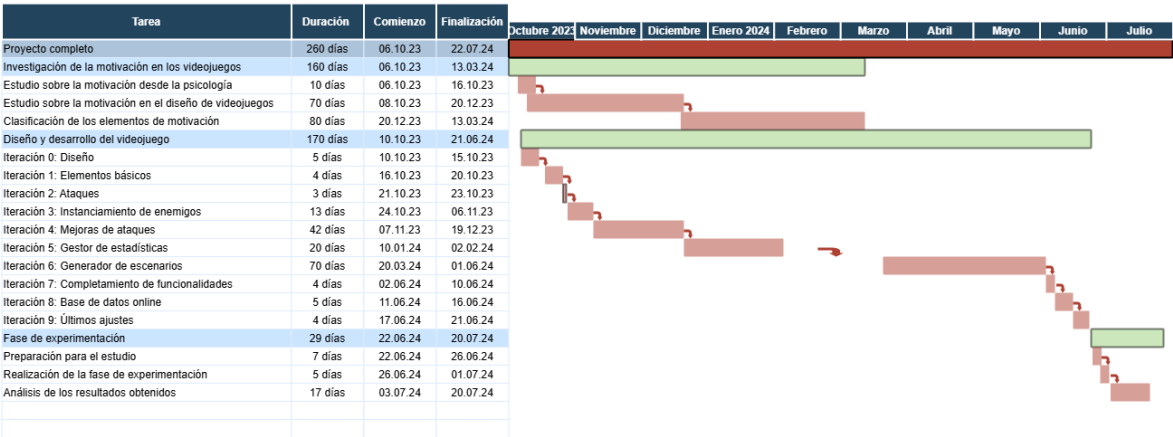


Figure 4.1: Diagrama de Gantt del progreso de los objetivos del proyecto.

- **Iteración 3:** Se implementan nuevos enemigos y un nuevo tipo de ataque de estos: los proyectiles.
- **Iteración 4:** Se añaden mejoras a las armas y personaje, algunas de ellas únicas y poderosas para dar más variedad de juego.
- **Iteración 5:** Con el fin de recoger información se crea el gestor de estadísticas que se encarga de guardar los datos de la partida.
- **Iteración 6:** Para crear los escenarios de las partidas se recurre a la implementación de un generador de mundos procedural.
- **Iteración 7:** Se añade al jefe final, sonidos y música y se completan sistemas en el prototipo.
- **Iteración 8:** Se implementa el uso de una base de datos en línea para la creación de cuentas para los usuarios y para guardar las métricas de partidas de los mismos.
- **Iteración 9:** Con la creación de una interfaz orientada a los usuarios y los últimos ajustes necesarios a sistemas y elementos pasados se da por finalizado el desarrollo del videojuego.

Por último se deja un diagrama de Gantt visible en la Figura 4.1 en el que observar el avance del proyecto.

4.3. REUNIONES

Al principio del proyecto estas se realizaron de forma más frecuente pues había una gran cantidad de decisiones que tomar y funcionalidades del juego que implementar y de las que comprobar el correcto funcionamiento. En el capítulo 5 se van detallando las reuniones relacionadas con decisiones de diseño del videojuego, aquí se hablará de otro aspectos de éstas:

- En la **primera reunión** uno de los puntos tratados más importantes fue la propia selección de metodología a usar a lo largo del proyecto. Recordando las palabras de múltiples profesores a lo largo de la carrera, **Scrum** no es una de las opciones presentadas pues no se trata de un proyecto en equipo. Basándose en las sugerencias de los tutores para este tipo de proyectos y que además en otros desarrollos de videojuegos el autor ha empleado este método de forma natural, se decidió por los prototipos incrementales. También se establecen como tareas para la próxima reunión el desarrollo del primer prototipo y la búsqueda de los elementos y características que emplean los diseñadores profesionales para intentar fomentar la motivación extrínseca de los jugadores.
- Para la **segunda reunión** ya se tenía el primer prototipo en el que comprobar las sensaciones de las mecánicas básicas como mover y atacar. Al quedar satisfechos con los resultados se determinaron las tareas para la siguiente reunión: seguir con la búsqueda de material relacionado con la motivación en los videojuegos y la creación de más ataques.

- Se llegó a la **tercera reunión** con las tareas cumplidas. Se habló de la necesidad de tener mejoras únicas para esos nuevos ataques para tener más opciones de juego y se empezó a discutir la pertenencia en cada categoría de los elementos que fomentan la motivación en los videojuegos. Se establece como tareas la implementación de más enemigos y la creación de mejoras para los ataques y el personaje.
- La **cuarta reunión** fue principalmente de seguimiento de las tareas pues estas requieren de la creación de sistemas que se comunican entre sí. Aquí se mencionaron algunas de las estadísticas que podrían resultar interesantes de guardar para los jugadores. Al estar las tareas anteriores cerca de su finalización se dejó propuesta una nueva: la implementación de la recogida de datos más básica, el daño causado y los enemigos vencidos.
- Con las tareas completadas e integradas en el prototipo, en la **quinta reunión** se comprueba la satisfacción de los resultados obtenidos y se fija la vista en el siguiente objetivo. Uno de los elementos de motivación que quedó sin categoría asignada por la dificultad de interpretarlo, volvió a ser el centro de atención. Se discutió sobre la aleatoriedad en el juego y sobretodo en su implementación en cuanto a la creación del escenario del proyecto. La conclusión a la que se llegó queda reflejada en el capítulo anterior en la sección de "*Clasificación de los elementos de diseño*". Como tarea se fija la implementación de un sistema de creación de mundos procedural.
- Se llegó a la **sexta reunión** con el prototipo de creación de escenarios funcionando de forma correcta. El siguiente paso a dar fue la creación de un sistema de comunicación con una base de datos en la que guardar las métricas de las partidas de los usuarios, cosa que se dejó como tarea junto a la creación de una interfaz gráfica definitiva.
- En la **octava reunión** y el prototipo con comunicación con la base de datos implementado, se decidió realizar la división de las dos versiones y dar por completados los prototipos finales. Después se discutieron los detalles de la fase de experimentación como son: la duración de la misma, las preguntas a realizar a los jugadores y el mensaje enviado a estos para unirse al experimento.
- A partir de este punto las reuniones fueron menores pues se trataba de correcciones e ideas para la escritura de la memoria. Mencionar que a lo largo del proyecto ha habido más reuniones que no se mentan por ser de menor importancia al tratarse únicamente de seguimiento de las tareas y posibles dudas del autor.

Resultados

En este capítulo se habla de los pasos dados para desarrollar el videojuego en sus dos versiones.

Antes de entrar en detalle, es importante entender varios conceptos inherentes a la arquitectura de *Godot*. Todos los elementos que van a componer el juego son **nodos**, algunos de ellos tienen sus propias características y funciones dependiendo de su tipo, pero para hacer uso de todas sus posibilidades hay que asignarles un script.

Los nodos son las piezas que conforman los objetos del motor y que están estructurados en forma de **árbol**. Siempre se tiene un nodo padre como raíz y luego se le pueden agregar uno o más nodos hijo que añaden más características y funcionalidades al objeto. Como se puede observar en la Figura 5.1 los nodos hijos pueden a su vez tener otros hijos. El objeto de la figura posee nodos que le otorgan una imagen visible al jugador gracias a *Sprite2D*, texto, otorgado por el nodo *Label* y un área de colisión a cargo de *Area2D* y su nodo hijo *CollisionShape2D*. Cuando se componen nodos de esta forma, se consigue una **escena**. Estas se podrían considerar nodos personalizados pues al fin y al cabo son nodos a los que se les ha añadido las funcionalidades deseadas y al guardarlos se pueden usar como nodos hijo en otros nodos o escenas. Tanto el personaje del jugador, los ataques de este, los enemigos como el mapa en que ocurre una partida son escenas, es decir, colecciones de nodos que cumplen una función específica.

Como se ha mencionado en el apartado anterior, Godot ofrece soporte para múltiples lenguajes, pero recomienda el suyo propio, *GDScript*, especialmente pensado para sacar partido de todo lo que ofrece el motor gráfico y es el que se usará en el proyecto. Otro punto a favor del uso de *GDScript* es que algunas de las funcionalidades del motor funcionan únicamente con ese lenguaje. La más importante limitación de los otros lenguajes y que afectaría de manera directa al desarrollo de este proyecto es que en la versión actual de Godot no hay soporte para exportarlos a web.

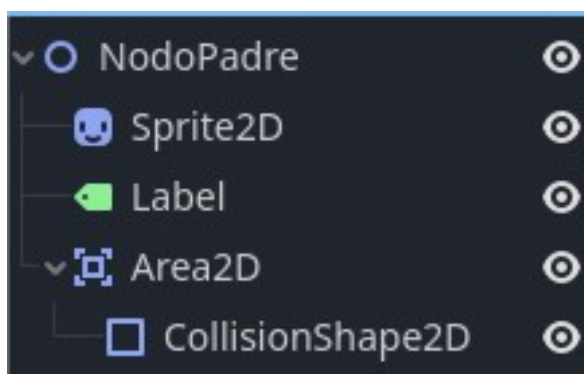


Figure 5.1: Ejemplo de jerarquía de nodos.

5.1. ITERACIONES

Se ha considerado relevante discutir las primeras decisiones de diseño como una iteración cero ya que contiene información relevante al desarrollo del proyecto.

5.1.1. Iteración 0: Diseño

El primer paso por dar es la elección del propio juego. Las características a tener en cuenta para la decisión final fueron las siguientes:

- Simpleza en jugabilidad para una mayor accesibilidad, que ayudará a conseguir usuarios para la parte de experimentación.
- Que pertenezca a un género donde la incorporación de elementos de motivación extrínseca no sea solo posible si no que resulte natural.

Teniendo esto en cuenta, se consideraron dos potenciales géneros. Uno del género **RogueLike** y **SurvivorsLike** / **HordeSurvivors** pues todavía no hay un consenso en la comunidad de que nombre darle.

El primer género consiste en partidas en las que siempre se empieza desde el mismo punto, es decir, no se guardan los avances de las anteriores partidas que se hayan jugado. Estos generalmente suelen ofrecer mejoras desbloqueables según los logros o hitos que se consigan en las partidas. Si has llegado a vencer al jefe final se desbloquea un nuevo poder o mejora, consiguiendo así que cuanto más se juegue más opciones y posibilidades ofrezca el juego.

El segundo se trata realmente de un subgénero de los *RogueLike* ya que plantea una idea similar, pero enfoca su atención en otro punto. Mientras que los *RogueLike* pueden combinarse con otros géneros de videojuegos fuera de la acción como pueden ser la gestión o incluso los puzzles, los *Survivor* se basan completamente en la supervivencia de oleadas. "*Tú solo contra el mundo*" es su filosofía, normalmente además tratándose de oleadas interminables donde el objetivo es ver cuán lejos puedes llegar en la partida.

Comparando estás dos opciones se seleccionó al final un juego con mecánicas de *SurvivorsLike*, ya que ofrece, además de los puntos mencionados anteriormente, unos controles más sencillos y una implementación más directa.

Otro punto del diseño a mencionar es la **narrativa** de la que se va a dotar al juego. La mayor parte de juegos que pertenecen al género elegido no tienen una historia como tal pero si incluyen píldoras de información en distintos formatos. Estos pueden estar en descripciones de objetos o logros. En el videojuego a desarrollar en este proyecto sí se compartirá la historia que está aconteciendo en las partidas. Se ha creado un mundo mágico con dos facciones enfrentadas, por una parte los magos de la naturaleza y por otra los **hierromantes**, siendo esta última a la que pertenece el jugador. Estos son capaces de crear y usar el hierro a su voluntad. En esta característica basaremos todos los ataques y poderes que el jugador podrá obtener a lo largo de las partidas.

Antes de empezar con la primera iteración se llevó a cabo una reunión donde se tomaron varias decisiones de diseño del juego. Las más destacables son las siguientes:

1. **Simplificar los controles lo máximo posible.** Algunos juegos del género, además del movimiento del personaje, dan la posibilidad de controlar directamente todos o casi todos sus ataques. Aunque esto pueda ofrecer más profundidad jugable también limita en cierta medida el público posible del juego. Para evitar eso se decidió limitar los controles únicamente al movimiento del personaje y dejar que los ataques se lancen automáticamente siguiendo un temporizador personalizado para cada uno.
2. **Usar un estilo gráfico sencillo.** Usando assets gráficos creados por alguien con experiencia, como es el caso de Kenney ¹, además de un aspecto visual más placentero, habría una claridad

¹<https://www.kenney.nl/>



Figure 5.2: El personaje con su ataque básico y el primer enemigo.

visual necesaria para este tipo de juegos. Saber en todo momento donde está el personaje del jugador y qué está pasando en pantalla, siendo capaz de identificar todos los elementos presentes en esta, es vital para el disfrute de la experiencia.

Una vez tomadas las decisiones de diseño más importantes, se procedió a desarrollar el primer prototipo de juego.

5.1.2. Iteración 1: Elementos básicos

En la **primera iteración** se implementa el movimiento del jugador, así como su ataque básico. El nodo principal que forma al jugador es de tipo *CharacterBody2D* que sirve para definir objetos que se moverán, ya sea por un script o controlados por el jugador como es nuestro caso. Por defecto los scripts que heredan de esta clase tienen una implementación de movimiento pero es para movimiento 2D visto desde el lateral estilo *Mario Bros*, así que se modifica para poder moverse en todas las direcciones y no solo izquierda y derecha. Para asignar el movimiento se ha usado el sistema de *Godot* de controles y accesibilidad, que permite conectar la pulsación de teclas con señales propias y asignarles una función. Esto se ha implementado de forma que al pulsar las flechas direccionales del teclado, el personaje se moverá en esa dirección. Si se pulsa más de una, mientras no sean direcciones contrarias se moverá en la combinación de ambas direcciones. Para tener en cuenta a los jugadores de móvil los controles se adaptan para funcionar tocando en la pantalla la dirección en la que se deseen mover.

Al personaje se le han añadido también unos atributos básicos como la vida y la velocidad de movimiento para la modificación directa de sus valores. Mediante el uso de la anotación de *Godot* **@export** en esas variables, que permite su modificación desde el editor sin tener que entrar al código. Esto otorga una gran flexibilidad a la hora de hacer ajustes sobre el balanceo de la dificultad y posibles futuros cambios en el diseño. Además, se ha le ha acoplado un nodo de cámara al personaje del jugador para que esta le siga allá donde vaya.

También se ha creado el primer tipo de enemigo usando también un nodo *CharacterBody2D*. En este caso al contrario que con el personaje no será movido por el jugador si no que se hará uso de una de las funciones que ofrece, que haciendo uso de un vector de dos valores que indica la dirección y un atributo velocidad lo desplazan automáticamente. Este enemigo básico busca la posición del jugador e intenta acercarse a este para hacerle daño al contacto.

Para el movimiento del enemigo se usan las herramientas de **pathfinding** implementadas en *Godot*. Esta ofrece múltiples opciones de personalización además de garantizar la correcta circulación de los enemigos esquivando obstáculos y encontrando el camino más rápido hacia su objetivo.

En cuanto al ataque básico del jugador, se ha optado por un proyectil que sale disparado en la dirección en la que el personaje está mirando y que se mueve hasta impactar con un objetivo o se acaba su tiempo dado de vida. Este ataque al se le ha dado el aspecto de una daga, tiene como base un *Node2D* básico que por si mismo no hace nada, más allá de tener una posición en el escenario, representada

por una transformación bidimensional. Para dotarlo de funcionalidad, se le ha agregado un nodo hijo de tipo *Area2D* que detecta de forma automática cuando un nodo de tipo *CharacterBody2D* entra en un área definida. Se consigue así mediante una simple comprobación para ver si el cuerpo que ha entrado en el área de ataque es un enemigo un objeto que daña a los enemigos e ignora al resto de actores en la escena.

Por último, se crea una ***PackedScene*** del ataque básico, que no es más que una escena empaquetada y optimizada para ser instanciada en tiempo real en el juego.

5.1.3. Iteración 2: Ataques

Con la **segunda iteración** se añaden más tipos de ataques para el jugador y un nuevo tipo de enemigo. También se desarrolla el sistema de experiencia y niveles.

Para el nuevo tipo de enemigo se ha hecho uso de la herencia de la clase básica de enemigo. Así añadimos directamente las funcionalidades nuevas sin perder las básicas. La principal característica del nuevo tipo es su capacidad de disparar proyectiles al jugador una vez se ha acercado lo suficiente. Esto se consigue mediante la adición de un área al enemigo que va a detectar cuando el jugador entra. En ese momento se dispara un proyectil a la posición que tenga el personaje del jugador.

Nuevos ataques

En cuanto a los nuevos ataques, algunos se han creado heredando del ataque básico creado en la anterior iteración otros fueron creados desde cero debido a diferencias sustanciales en su naturaleza.

Para la implementación de los ataques Lanzamiento de martillo y Lanzamiento de lanza, se ha hecho uso de herencia del ataque básico ya que también se trata de ataques de proyectil con un comportamiento similar.

- El **lanzamiento de martillo** consiste en un ataque de proyectil directamente dirigido a un enemigo cercano. Su funcionamiento es igual al del ataque básico, un área que detecta cuando algún cuerpo entra en contacto y le hace daño si se trata de un enemigo. Este ataque tiene un daño y frecuencia de disparo menor que la daga ya que al contrario que esta no hay que apuntar a los enemigos manualmente sino que es automático.
- Con el **tiro de lanza** el procedimiento es el mismo salvo por un detalle. Los ataques anteriores desaparecen al impactar con un enemigo pero la lanza permanece. Esta atraviesa y daña a todos los enemigos con los que impacte hasta desaparecer después de un tiempo determinado. Otra particularidad es que mientras los dos primeros ataques son lanzamientos individuales, este es múltiple: varios proyectiles son lanzados con cada ataque. En cuanto a la elección de objetivos o dirección de lanzamiento esta se realiza de forma aleatoria. Se crea un círculo invisible alrededor del jugador y se elige un punto de ese círculo para cada proyectil y luego se lanzan todos al mismo tiempo en su correspondiente dirección. El círculo es de un radio determinado para que los puntos elegidos aleatoriamente no estén demasiado juntos y el ataque se disperse en múltiples direcciones.

Los ataques para los que no se ha hecho uso de la herencia son Tormenta de hachas y Escudo orbital. El primer tipo corresponde a un ataque que consiste en crear un área de daño continuo en un punto determinado. El segundo se trata de un objeto que va dando vueltas alrededor del personaje del jugador y dañando a los enemigos que toca.

- La **Tormenta de hachas** está formada por un *Node2D* básico y un *Area2D* como hijo, al igual que el ataque básico, pero con forma circular y una implementación diferente. En lugar de dañar un enemigo y desaparecer, esta se queda en el punto en el que se ha invocado y va dañando a todos los enemigos que se encuentren en su área según un temporizador asignado. Este ataque es invocado siempre encima de un enemigo que esté cercano al jugador. Para conseguir el efecto visual de tener una gran cantidad de hachas dando vueltas en un área se ha hecho uso del sistema de **partículas** de *Godot*. Se añade una imagen de hacha para representar



Figure 5.3: Los ataques obtenibles por el jugador.

cada partícula y se define que su movimiento sea circular a la vez que giran sobre sí mismas, para dar una sensación visual de ataque caótico y poderoso.

- Para crear el **Escudo orbital** hacemos uso del mismo *Node2D* base que el resto de ataques. Su diferencia radica en el comportamiento implementado. Se crea un círculo de un determinado radio con el centro encima del jugador y se añade una función que le hace dar vueltas alrededor de ese círculo a una determinada velocidad de forma continua.

En este punto se puede hablar de las decisiones de diseño que han llevado a elegir estos ataques. Cada uno cumple una función distinta.

1. El ataque básico o daga al ser dirigido da libertad al jugador para elegir sus propios objetivos. Este es uno de los tres puntos fundamentales de la **TAD**.
2. El martillo funciona como un seguro, aunque el personaje no este apuntando a nadie, sabes que al menos uno de tus ataques será exitoso. Esto resulta especialmente útil en una situación de huida.
3. Las lanzas por su lado sirven como mecanismo de limpieza de enemigos más débiles. No son controlables pero al ser numerosas si se está rodeado se sabe que van a impactar con todo lo que se tenga alrededor. Cuantos más enemigos haya cerca más efectivas serán.
4. La tormenta tiene un objetivo similar a las lanzas, pero al igual que con el martillo, aporta la seguridad de saber que siempre aparecerá una posición en la que haya enemigos.
5. Para el escudo se tuvo en mente el crear algo con un mecanismo que aporte seguridad a los jugadores. Al ser algo que se mantiene cerca del personaje este funciona como un mecanismo de defensa, un último recurso frente a los peligros inminentes.

Se ha mencionado en dos ataques que su objetivo se elige en base a enemigos cercanos al jugador, lo cual se explica a continuación. Se ha creado un sistema basado en un nodo *Area2D* que detecta a los enemigos que toque y los añade a un array de enemigos. Éstos sirven como posibles objetivos de nuestras habilidades. Al instanciarse un ataque que necesita objetivo, se elige aleatoriamente un enemigo del array y se utiliza su posición bien para definir la dirección del ataque o para establecer el punto de aparición. Cuando un enemigo muere o sale del área se elimina del array para evitar seleccionar objetos que no existen o están muy lejos.

Siguiendo la iteración llegamos a la implementación de niveles y experiencia. Cada vez que un enemigo es vencido aparece en la que era su posición un orbe de experiencia que los jugadores deberán tocar para recoger y aumentar así su nivel. Los niveles representan el aumento de poder del personaje pues con cada subida aumenta su vida máxima total y se le presentará la oportunidad de mejorar alguna de sus armas o estadísticas. Cada vez que el personaje suba de nivel, la experiencia

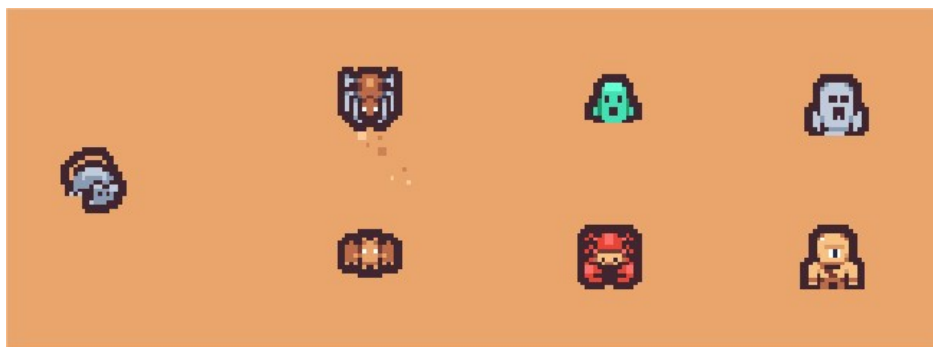


Figure 5.4: La apariencia de los enemigos en el juego.

necesaria para el siguiente aumento de forma lineal. Este tipo de aumento implica que los primeros niveles se alcanzarán rápidamente pero cada vez será mas costoso llegando a un punto que será casi imposible volver a aumentar de nivel. De esta forma no es necesario definir un nivel máximo para el jugador sino que llegará un momento en el que le costará mucho mas esfuerzo conseguir nuevos niveles.

También se prepara lo que en el futuro será el sistema que gestione las mejoras del jugador. En esta iteración se deja listo el código que avisa de una subida de nivel y la pantalla de pausa en la que se podrá elegir la mejora deseada por conseguir un nuevo nivel.

5.1.4. Iteración 3: Instanciamiento de enemigos

Antes de la tercera iteración hubo otra reunión de diseño en la que la idea más importante a destacar es el uso de las mejoras del jugador para aumentar la sensación de descubrimiento y satisfacción. No añadir únicamente mejoras numéricas como pueden ser el daño y la vida si no también algunas que cambien el comportamiento de un ataque.

También se habló del apartado gráfico y se consideró que los assets que se consideraron inicialmente como temporales serían seguramente definitivos ya que el artista que los crea y publica tiene múltiples paquetes de elementos con el mismo estilo visual que son combinables de forma indistinguible consiguiendo más variedad visual.

En la **tercera iteración** se ha procedido en primer lugar a la **refactorización** del código según las guías de estilo del lenguaje propio de *Godot*, *GDScript*, para facilitar el uso y la comprensión del código previamente escrito, así como el del futuro.

Posteriormente se han añadido el resto de enemigos basados en los dos tipos básicos, los que disparan proyectiles y los que se intentan golpear al jugador a cuerpo a cuerpo. Se ha determinado y diseñado a los enemigos para que cada uno pertenezca a distintos niveles de dificultad para el jugador, indicados por su **clase** y poder así usarlos posteriormente en un sistema de aparición automática de los mismos en función del tiempo de partida.

Una vez creados los nuevos enemigos, se ha desarrollado el gestor de aparición de los mismos. A medida que pasa el tiempo, se van creando enemigos a una distancia prudente alrededor del jugador siempre fuera de pantalla para tener tiempo suficiente de reacción. El número y tipo de enemigos van determinados por cuánto tiempo de partida ha pasado. A mayor tiempo sobreviviendo, distintas y más poderosas criaturas aparecerán así como más frecuentemente hasta llegar a tener una frecuencia de aparición de menos de un segundo. Para darle más variedad y picos a la dificultad hay establecido un temporizador que invoca una oleada entera de enemigos rodeando al jugador. El número de enemigos invocados por oleada no incrementa para evitar la posibilidad de bloquear al jugador y obligarlo a recibir daño, lo que si incrementa es la frecuencia, que va aumentando progresivamente hasta la aparición del jefe final, momento en el que se queda fija el resto de la partida.

A continuación se explica más detalladamente cada enemigo que se puede ver en la Figura 5.4:

- **Rata**, clase 0. El enemigo más básico y débil de todos. Poca vida, poco daño y velocidad de movimiento normal. Se eligió como primer enemigo ya que no es una criatura muy amenazadora.
- **Murciélago**, clase 1. Más vida y velocidad de movimiento que la rata aunque sigue sin presentar un gran desafío a los jugadores.
- **Araña**, clase 1. Posee mas vida y daño que las criaturas anteriores y la misma velocidad que la rata.
- **Cangrejo**, clase 2. Al igual que la araña tiene un aumento en las estadísticas de vida y daño pero también en la velocidad. A partir de este punto la representación visual de los enemigos está elegida para inspirar cada vez más poder.
- **Fantasma pequeño**, clase 2. Este es el primer enemigo con ataque a distancia que se encontrarán los jugadores. Para compensar esa dificultad extra posee una vida y un daño relativamente bajo.
- **Fantasma grande**, clase 3. Mucho más poderoso que su contrapartida pequeña en todos los atributos. Presenta una gran amenaza a los jugadores gracias a su potente ataque a distancia. Como se puede observar los dos enemigos normales que poseen ataque a distancia, son de naturaleza similar para que los jugadores lo asocien rápidamente.
- **Cíclope**, clase 3. El enemigo más poderoso de todos obviando al jefe final. Gran cantidad de vida y daño con una velocidad ligeramente aumentada. Aunque algunos enemigos como este se indique que tienen más velocidad, es siempre en comparación con el enemigo básico, la rata, ya que todos los enemigos son mas lentos que el jugador para darle siempre una opción de escape. La única excepción a esta regla es el jefe final que perseguirá al jugador allá donde vaya sin posibilidad de escapar.

Relacionada con los enemigos y la dificultad de estos está la adición de una variable que determina cuánto mejora cada tipo de enemigo con el paso del tiempo. Cada cierta cantidad de tiempo los enemigos también suben de nivel aumentando sus estadísticas de vida y daño. Esto es para que una criatura en el minuto 5 de partida no tenga el mismo poder que en el minuto 10. Así conseguimos un escalado más controlado de la dificultad a lo largo de la partida y evitamos que el juego sea demasiado fácil provocando una posible pérdida del interés por parte de los jugadores.

También cambiamos el orbe de experiencia para que este otorgue una cantidad acorde con la fuerza del enemigo que lo suelta. Escalando este tanto con la clase de enemigo como con su nivel según el tiempo de partida. Así, los enemigos más poderosos darán más experiencia que los más débiles.

En esta iteración aparece la **primera dificultad** considerada seria, y es el rendimiento. Haciendo pruebas con el gestor de aparición de enemigos se vio que, en cuanto se acumulaban muchos enemigos, el juego empezaba a funcionar visiblemente peor, acentuado aún más tras exportarlo para navegador web. Esto era debido al uso del pathfinding de Godot. Aunque se hicieron optimizaciones, reduciendo la frecuencia con la que se consultaba la posición del jugador y se actualizaba el camino de los enemigos, el rendimiento seguía sin ser adecuado. También se probó a eliminar la funcionalidad que permitía que los enemigos se esquivasen entre si cuando estuviesen a punto de chocar pero seguía sin ser suficiente, así que se decidió cambiar por completo su funcionamiento.

En lugar de estar dotados de un movimiento inteligente, se cambió a los enemigos para que simplemente se moviesen en la dirección del jugador. Se crea un vector dirección entre el enemigo y el jugador y estos se mueven en línea recta sin importar los posibles objetos que hubiese en medio de su ruta. Esto podía provocar que los enemigos se acumulasen detrás de obstáculos, pero por la naturaleza del juego, en la que se está constantemente en movimiento, no se consideró que pudiese llegar a ser un problema real. Más adelante, teniendo esto último en cuenta y para evitarlo, se desarrollará una función que elimine de memoria a los enemigos que se han quedado atascados lejos del jugador.

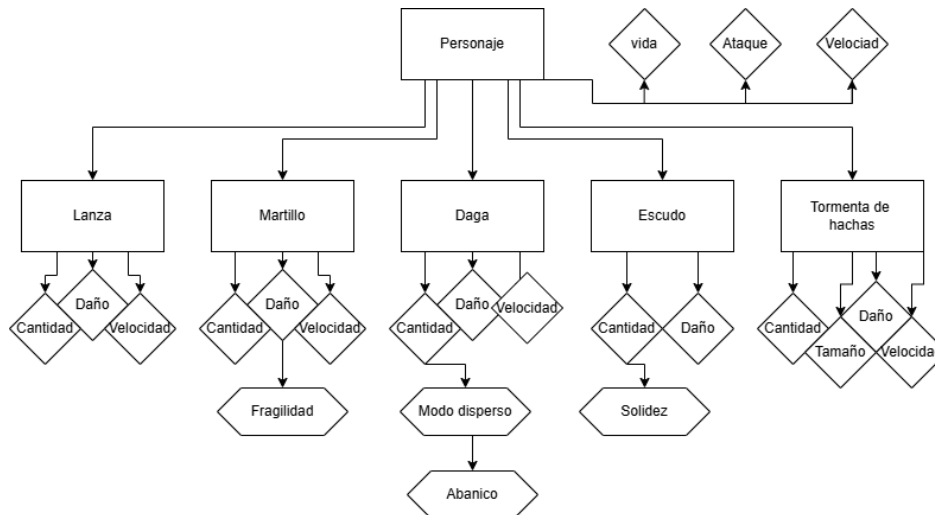


Figure 5.5: Diagrama de todas las mejoras obtenibles.

5.1.5. Iteración 4: Mejoras de ataques

En la cuarta iteración seguimos con la implementación del gestor de mejoras y subidas de nivel del jugador, así como la creación de mejoras para cada arma.

En este paso también procedemos a crear un **bus de señales** para facilitar la comunicación entre clases y elementos del juego. Una de las herramientas más versátiles y poderosas que ofrece Godot es la posibilidad de crear señales propias personalizadas. Aprovechando esto, se crea una clase global que contenga todas nuestras señales para tenerlas localizadas, clasificadas y en un único lugar de fácil acceso para todas las demás clases.

Las señales en Godot funcionan como mensajes que se pueden mandar entre nodos. Estas se pueden usar para informar a un nodo de que ha ocurrido un evento o incluso mandar información en forma de variables.

Una vez creado el bus de señales, se procede a hacer uso del mismo en el gestor de mejoras para una clara y directa comunicación entre el mismo y la clase del jugador, así como los elementos de interfaz relacionados con la elección de mejoras al subir de nivel. Cada vez que el jugador consigue un nuevo nivel, el gestor de mejoras busca las que están disponibles y elige aleatoriamente 3 para mostrar al jugador. La clase que define las mejoras tiene entre sus atributos una variable de tipo *booleano* que indica si están desbloqueadas y disponibles o no. Algunos ataques están desbloqueados desde el inicio de la partida pero los más poderosos solo se desbloquean después de cierto tiempo sobrevivido. Cuando se adquiere un ataque nuevo al subir de nivel, se desbloquean las mejoras básicas asociadas a esa habilidad. Para las mejoras únicas, hay que cumplir ciertos requisitos en la partida para poder desbloquearlas, que se desglosan más adelante. Estas mejoras pueden ser tanto nuevos ataques, como subidas de estadísticas para el personaje o alguna de sus otras habilidades.

Mejoras únicas

Teniendo el sistema completo creado, el siguiente paso es la creación de todas las mejoras de los ataques y estadísticas base del jugador así como las únicas, Figura 5.5. Como se comentó en la anterior iteración, se busca crear mejoras que no sean solo un aumento de números sino que también impliquen cambios en la funcionalidad.

Éstas son las mejoras únicas creadas:

- **Ataque disperso.** Mejora única para la daga que cambia su modo de disparo además de añadir dos dagas extra al ataque. Una vez obtenida, las dagas pasan de dispararse en línea recta desde la posición del jugador a lanzarse en ángulo. Cada daga lanzada elegirá aleatoriamente un

ángulo entre 15° y -15° y será lanzada en esa dirección. Esto facilita el apuntado ya que al tener varias dagas en direcciones ligeramente distintas no se requerirá tanta precisión por parte del jugador. Para desbloquear esta mejora hace falta haber elegido tres veces la mejora que añade dagas extra a cada lanzamiento.

- **Abanico de dagas.** Cada daga se sustituye por un abanico de dagas que al impactar se romperá en cuatro dagas más débiles que se verán lanzadas en las 4 direcciones cardinales. Las dagas lanzadas por el abanico no son abanicos, solo dagas normales pues lo contrario provocaría una reacción en cadena que acabaría con todos los enemigos en pantalla. Esta mejora se desbloquea automáticamente cuando pasan siete minutos de partida.
- **Escudo sólido.** Mejora del escudo que provoca que este destruya los proyectiles enemigos con los que impacte volviéndolo una poderosa herramienta defensiva. Esto se desbloquea cuando el jugador ha seleccionado dos mejoras de *escudo extra*.
- **Martillo frágil.** Provoca que cuando el martillo impacte con un enemigo se rompa en trozos que salen volando direcciones aleatorias. Estas hacen menos daño que el martillo normal. Para controlar que los fragmentos no se crearan encima del enemigo con los que impactó el martillo se ha creado una función que revisa el ángulo en el que se moverá y mueve el trozo al cuadrante correspondiente. Para desbloquear esta mejora el jugador debe elegir tres mejoras de daño para el martillo.

Los requisitos para desbloquear las mejoras únicas se crean de esta manera para que sea complicado para una persona conseguir todas las mejoras únicas en una sola partida para intentar fomentar que se jueguen múltiples partidas para descubrir y probar todo lo que ofrece el juego. Si todas las mejoras únicas se desbloqueasen por tiempo los jugadores esperarían hasta desbloquear la que les guste, pero estando detrás de condiciones distintas, estos tendrán más motivos para jugar y ver que otros poderes encuentran.

5.1.6. Iteración 5: Gestor de estadísticas

Para la **quinta iteración** comenzamos con uno de los sistemas vitales tanto del juego como del proyecto: el gestor de estadísticas.

Aquí mediante el uso del bus de señales se crean señales específicas para la recogida de datos. En primer lugar, se procede con el daño individual de cada arma del usuario y del número de enemigos vencidos de cada tipo, posteriormente se implementan más estadísticas como son el tiempo total sobrevivido o el número de partidas jugadas. Cada vez que un ataque impacta, se manda una señal con su valor de daño para añadirlo al total de la partida. Similar es el caso para los enemigos eliminados, que envían una señal con su tipo cuando estos desaparecen.

También se crean las funciones necesarias para guardar después de cada partida las estadísticas de esta y sumarmas al total de todas las partidas del jugador que posteriormente se mantienen en un fichero para asegurar la persistencia de los datos. La mayoría de estos datos son estadísticas que solo el propio jugador verá y que estarán presentes en elementos de motivación extrínseca.

Colecciones

Relacionada con las estadísticas está la creación de un gestor de **colecciones** que se usa para la comunicación con el gestor de estadísticas y mostrar los datos de este al usuario en una interfaz creada con ese fin. Interfaz que se comienza a implementar en este punto de forma más extendida ya que hasta ahora el foco estaba en las funcionalidades básicas del juego. Lo más importante de la interfaz es el menú de colecciones. Ahí no solo se mostrarán las estadísticas del jugador Figura 5.6, como el daño causado por cada arma y los enemigos vencidos de cada tipos, sino también un listado de logros que se desbloquean con determinadas acciones en la partida, los posibles ataques que se pueden desbloquear en una partida, así como sus mejoras y los objetos coleccionables que se implementarán más adelante.



Figure 5.6: La pantalla de estadísticas del juego.

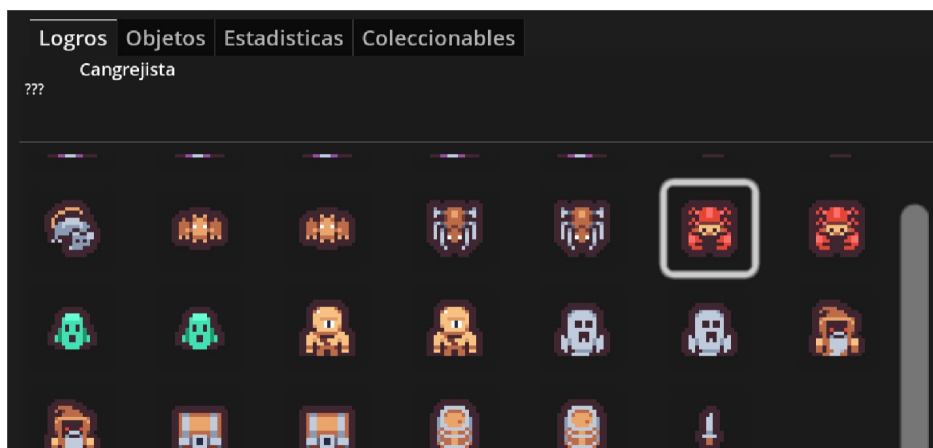


Figure 5.7: La pantalla de logros del juego

Los logros procedemos a crearlos en este punto. Para la creación de los logros se hace uso de un elemento de *Godot* llamado **Recursos** que permite la creación de clases a modo de cajón de información y que se almacenan en el disco. Creamos un recurso de tipo logro que luego podremos usar para crear multitud de logros de manera directa, con las mismas características, pero con valores distintos. Los recursos son fácilmente accesibles y modificables, así se podrá cambiar sus valores para indicar cuando se han desbloqueado.

La mayoría de los logros consistirán en eliminar un número de enemigos de cada tipo o sobrevivir un tiempo. Pero también de recogida de elementos como secretos o pociones, que implementaremos más adelante. Los secretos serán objetos que a priori parecen inaccesibles o están escondidos, pero a los que se puede llegar. También se implementa en el gestor de estadísticas, la función que verificará al final de cada partida si se ha dado la condición necesaria para desbloquear alguno de los logros. Esta consiste en una serie de comprobaciones de las condiciones de cada logro creado, aunque no sea una solución muy escalable, debido a que la modificación de cumplimiento de un logro implica también la modificación manual de la comprobación, al tener un número reducido en el proyecto no implica una gran carga de trabajo.

5.1.7. Iteración 6: Generador de escenarios

En la reunión anterior a la sexta iteración se discutió el tema principal de la iteración: la creación del mundo. Se planteó si esto también debiera ser o no un punto distintivo entre las dos versiones de

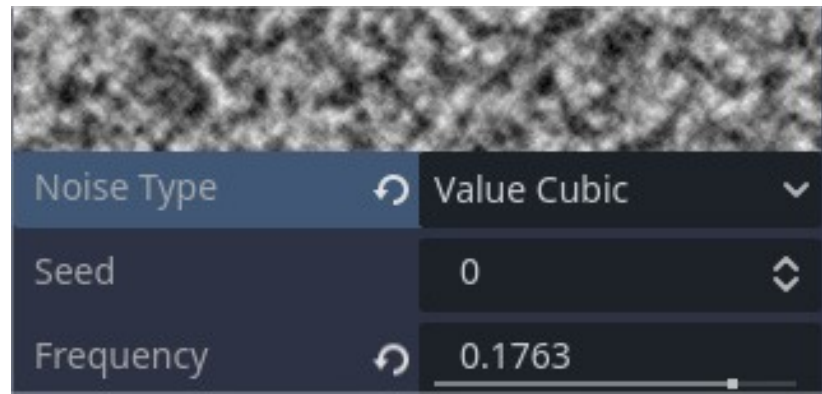


Figure 5.8: La configuración de la función ruido para el terreno.

juego. Esto nos llevaría a que la versión puramente intrínseca no tuviese un mundo cambiante, si no que se definiría una semilla de generación. Esto significa que siempre que se comience una partida, se jugaría en el mismo escenario con todos los elementos colocados de la misma manera. Al final esta posibilidad fue descartada. Debido a la dificultad de clasificación de algunos elementos en un tipo u otro, se concluyó que era mejor que la aleatoriedad estuviese presente en ambas versiones.

Además, convertiría la experiencia jugable de esa versión en algo más aburrido que si fuese un mundo cambiante y es importante tener claro que el juego debe ser el mismo. Una versión privada de elementos extrínsecos no significa una versión más aburrida y exenta de características jugables.

En la **sexta iteración** el punto central es la creación de un **generador de mundos** para hacer que en cada partida el escenario sea distinto.

Esto se lleva a cabo con la herramienta de creación de *TileMaps* que implementa *Godot*. Esta herramienta permite importar imágenes y dividirlas en baldosas de igual tamaño, separando así todos los distintos elementos visuales que esta contenga. Luego, desde la imagen, posteriormente podemos usar todos los objetos de la plantilla y dibujar con ellos directamente en el mundo de juego para la creación del escenario. Para evitar la colosal tarea de crear múltiples escenarios, se ha creado un script que mediante el uso de distintas funciones de ruido y una selección de baldosas del *TileMap*, se crea automáticamente un mundo para el jugador antes de comenzar la partida.

Los parámetros de la función ruido se pueden ver en la Figura 5.8. Esta nos muestra tres características importantes, el tipo de ruido, la semilla y la frecuencia. Existen más parámetros que pueden ser modificados pero no fue necesario para el uso dado a esta función.

- El **tipo de ruido** es *Value Cubic* [21]. Se trata de una función de ruido con bordes suaves, perfecta para usos como el terreno. En la página web del autor está disponible una explicación detallada de los cálculos que se realizan dentro de la función [21].
- La **semilla** determina la pseudo aleatoriedad de la función. La misma semilla va a dar siempre el mismo resultado. En este caso es cero pero se cambia mediante código cada vez que se crea el mapa para que este no se repita.
- La **frecuencia** se puede entender como el zoom que se le hace a la textura generada por la función de ruido. A mayor frecuencia más alejado el punto de vista. De manera empírica se concluyó que, esta frecuencia daba resultados satisfactorios.

Las texturas de ruido consisten en la interpretación visual de valores entre los rangos de la función de ruido. La función asigna a los píxeles de su textura, un valor que esta comprendido entre unos rangos mínimos y máximos según su configuración. Una vez averiguados esos rangos se procede a definir un valor que indicarán que casilla del *TileMap* colocar en cada punto del escenario según la textura de ruido. En este caso y por limitaciones de la paleta artísticas se define únicamente un valor, que determinará si la celda a colocar será de tipo hierba o tierra. El **valor elegido** es -0.1 que provoca que haya más hierba y asegura que los trozos de mapa con tierra sean consistentes. En la

```

if noise_value>=grass_ceil:
    tile_map.set_cell(floor_layer,Vector2(x,y),source_id,grass_atlas.pick_random(),randi_range(0,1))
elif noise_value<grass_ceil:
    tile_map.set_cell(floor_layer,Vector2(x,y),source_id,ground_atlas.pick_random(),randi_range(0,3))

```

Figure 5.9: Extracto del código que determina la celda a colocar en el escenario según el valor del ruido.

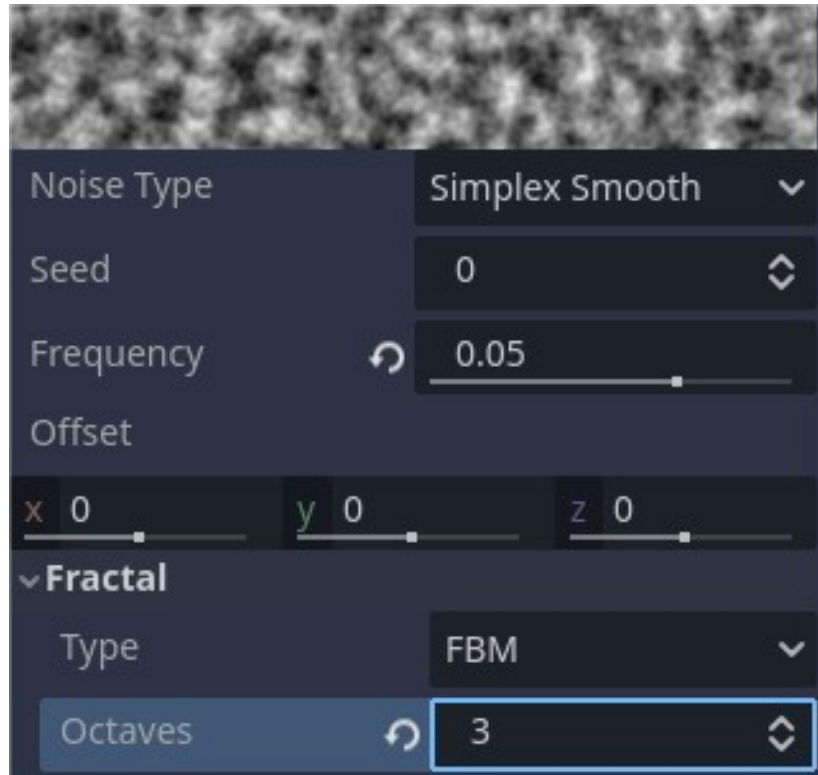


Figure 5.10: La configuración de la función ruido para los objetos y edificios.

Figura 5.9 se puede ver código que se encarga de colocar las casillas en el escenario según el valor de la textura de ruido en ese punto.

Para que los escenarios no sean demasiado repetitivos ni parezcan completamente vacíos, se han tomado varias medidas. La primera de ellas es crear variaciones sobre la selección de terreno realizada anteriormente para tener más opciones visuales y que el mundo no sea simplemente cuatro baldosas repetidas constantemente. En lugar de tener solo dos tipos de hierba y dos de tierra, se crea un subgrupo de casillas en las que se modifican atributos como la orientación y el color de las mismas. Así se consigue más variedad visual y no se repiten demasiado las texturas.

También se han realizado, esta vez sí a mano, una serie de edificaciones que se procederá a colocar en el mundo utilizando otra función de ruido para una distribución diferente en cada partida. Junto a estos se ha añadido la posibilidad de aparición de pequeños bosques o pequeñas agrupaciones de árboles para aumentar la variación de elementos visuales.

En la Figura 5.10 se puede observar que hay más opciones que en la función utilizada para el terreno.

- El **tipo de ruido** es *Simplex Smooth* que tiene mayor eficiencia computacional y una apariencia similar al usado anteriormente pero con picos más suaves que reparten más uniformemente los edificios por el escenario. En la publicación original [15] se encuentra una explicación en profundidad de su funcionamiento.
- La **semilla** se asigna aleatoriamente en cada partida por código, al igual que en el caso anterior.
- Para la **frecuencia** se ha bajado el valor, ya que no queremos que el escenario esté lleno de



Figure 5.11: El edificio con el secreto y unos barriles cerca.

edificios, sino que aparezcan de manera esporádica.

- El **tipo de fractal** es *FBM* Fractal Brownian Motion, el mismo que en la función de ruido del terreno, que no se mencionó anteriormente debido a que es el tipo por defecto. Este consiste, de manera simplificada, en crear una textura combinando tanto otras texturas como ondas de distinta frecuencia para conseguir el aspecto pseudoaleatorio. El cambio respecto a la anterior función de ruido viene en las Octavas, estas determinan lo suaves que son las transiciones entre una zona y otra. Poniendo el valor a 3 se consigue que los objetos no aparezcan demasiado apelotonados y sobretodo que haya cortes. Evitamos así la existencias de bosques demasiado grandes, edificios muy pegados o áreas llenas de barriles, de los cuales se hablará a continuación.

El último punto de trabajo de esta iteración ha sido la creación de objetos que aparecen en el mundo aleatoriamente o están presentes de forma predefinida en algunos de los edificios. Se trata de barriles y pociones. Los barriles sueltan una poción al ser destruidos y estas curan al jugador cierta cantidad de vida. Esto se hace para que la pérdida de puntos de vida por parte del jugador no sea tan punitiva. Dando la posibilidad de recuperar vida los jugadores, estos tendrán más libertad de decisión, pudiendo realizar acciones más arriesgadas sin que signifique condenar la partida al fracaso.

Con las reglas del mundo ya definidas se decidió que los secretos mencionados en la anterior iteración se añadiesen finalmente a los edificios recién creados. Estos serán colocados en uno de los tejados donde los jugadores podrán acceder atravesando una de las paredes que tiene una apariencia distinta. Poniendo los secretos en un edificio y estando a plena vista es más fácil que llamen la atención a los jugadores y que intenten obtenerlos que si estuviesen completamente ocultos y dependiese únicamente de la suerte el verlos o no. Se han añadido dos pistas diferentes a los jugadores sobre los secretos, una es la pared diferente en el edificio, que asemeja a las paredes secretas de distinto color usadas en los videojuegos clásicos de rol y disparos para indicar que son destruibles y que detrás hay regalos. La otra es que en la pantalla de los logros, el llamado "*Secreto*" tiene como imagen el cofre, el mismo *Sprite* que se usa en los secretos. Con esto se intenta que los jugadores sientan que deben poder llegar ahí y que busquen el método correcto para ello.

5.1.8. Iteración 7: Completamiento de funcionalidades

Con la **séptima iteración** volvemos la vista atrás para completar algunos de los sistemas y elementos que están aún sin finalizar.

Se comenzó con la creación de un **jefe final**. Se trata de un mago de la facción enemiga que se comentó al describir la narrativa del juego, los magos del bosque. Este aparecerá cuando cierto tiempo de partida pase, para presentar un desafío al jugador. El jefe final no solo tiene un poder muy superior a cualquiera de los enemigos normales, sino que presenta ataques únicos que pondrán a prueba las habilidades de los jugadores, así como su elección de mejoras a lo largo de la partida.

En la Figura 5.12 se puede ver el ataque básico del jefe que lanza cada pocos segundos y que



Figure 5.12: Ataque básico del jefe.

consiste en tres proyectiles que harán al jugador cambiar de trayectoria. Para aumentar su arsenal, también se le añadió un ataque especial que tarda más en lanzar y que consiste en un círculo de proyectiles a su alrededor 5.13. Además, en la búsqueda de crear un combate más único, se le ha dotado de un comportamiento especial. Si el jugador se acerca mucho al jefe final, este saldrá corriendo para mantener una distancia mínima.

El siguiente paso es hacer un ajuste general de la dificultad del juego ya teniendo en cuenta todos los elementos que presentará la versión final del juego. Entre ellos hay que destacar un cambio en la experiencia y subida de nivel. Para crear un ritmo de juego más satisfactorio se añade una variable en los orbes de experiencia que otorga una cantidad fija de experiencia además de la determinada por el tipo de enemigo. Con esto se consigue que en los primeros niveles el jugador suba antes de nivel y note un mayor incremento de poder. En cambio cuando se tenga niveles más altos costará más y cada mejora elegida será vista como una decisión más importante.

En este momento mientras se hacían las pruebas de la dificultad se encontró un **problema** en la aparición de los enemigos. El escenario de juego es finito, los usuarios tienen un área delimitada para las partidas. La forma de establecer los bordes del mapa es mediante la creación de enormes e infranqueables bosques, tanto para el usuario como para los enemigos, en los extremos del escenario. El problema fue que los enemigos aparecían dentro de dicho bosque y se quedaban atascados al ser elementos sólidos. Esto no fue visible hasta ahora ya que en anteriormente no hubo mucho acercamiento a los bordes por parte del autor.

Godot detecta automáticamente cuando se intenta crear un objeto encima de otro y mueve al que va a ser creado para que no haya colisión. Pero esto no funciona con los árboles ya que no son objetos sino celdas inmóviles en el *TileMap*. Esto provoca que los enemigos aparezcan encima de los bosques y quedándose atascados ya que son casillas solidas en la que no se pueden mover ni ellos ni el jugador. Para solucionar esto hubo que hacer una comprobación antes de crear a los enemigos para ver si estaban encima de una celda sólida o no. Esto se hace convirtiendo la posición global del enemigo a la posición de la casilla en el *TileMap*. Después hay que comprobar que en esa posición la casilla presente no es sólida. Así, si se trata de una celda sólida como un árbol cambiamos de posición al enemigo. Esto también implica una decisión de diseño. Al no ser descartados los enemigos que fuesen a aparecer en el bosque sino que son recolocados, la amenaza hacia el jugador es mayor en los bordes del mapa debido a que es más fácil ser rodeado. Esto provoca de manera orgánica que los



Figure 5.13: Ataque especial del jefe.

usuarios al verse rápidamente acorralados, se vuelvan a dirigir al centro del mapa, donde están los secretos y barriles llenos de pociones.

Para dotar de más **retroalimentación** al juego se han añadido múltiples efectos de sonido en función de las acciones del jugador o diferentes sucesos, como son los ataques, impactos y recogida de objetos. También se ha añadido música de fondo en la partida, consiguiendo así una experiencia de juego más satisfactoria.

Por último, se han ajustado y corregido algunos logros basados en el balance de dificultad anteriormente mencionado, no solo en dificultad u objetivo, sino también en la imagen o nombre mostrado. Para dar pequeñas pistas a los jugadores de lo que es necesario hacer para conseguirlos, se muestra un nombre significativo y, una vez desbloqueado, aparece una pequeña descripción del objetivo logrado.

5.1.9. Iteración 8: Base de datos online

La **octava iteración** está completamente dedicada a la implementación de un sistema de creación de cuenta e inicio de sesión. También incluye la conexión del juego con una **base de datos** para guardar estadísticas de uso de los jugadores para su posterior estudio y análisis.

Antes de proceder a ello se hizo un estudio previo de las opciones disponibles. Entre ellas destacaron 3 servicios externos. *LootLocker*² y *SilentWolf*³, plataformas creadas para ofrecer este servicio concreto a videojuegos y *Firebase*⁴ de Google de uso más general. A primera vista y viendo lo que ofrecía cada opción, la primera elección fue *LootLocker* debido a su sencillez y lo directo de su implementación. Pero algunas de las características que se necesitaban para la recogida de datos, solo están disponibles en la versión de pago así que fue descartada.

De las dos restantes, *SilentWolf*, servicio creado explícitamente para *Godot*, ofrece menos flexibilidad que *Firebase* para algunos usos de base de datos que se van a necesitar para la recogida de datos en la fase de experimentación. Por lo tanto, finalmente se decidió por usar *Firebase*, que se incorporó al videojuego mediante un plugin, *GodotFirebase*⁵, disponible para *Godot* que facilita su uso y oculta las funciones más complejas.

Como se ha comentado, instalamos el plugin para *Firebase* y se procede su configuración con el

²[LootLocker](#)

³[SilentWolf](#)

⁴[Firebase](#)

⁵<https://github.com/GodotNuts/GodotFirebase>

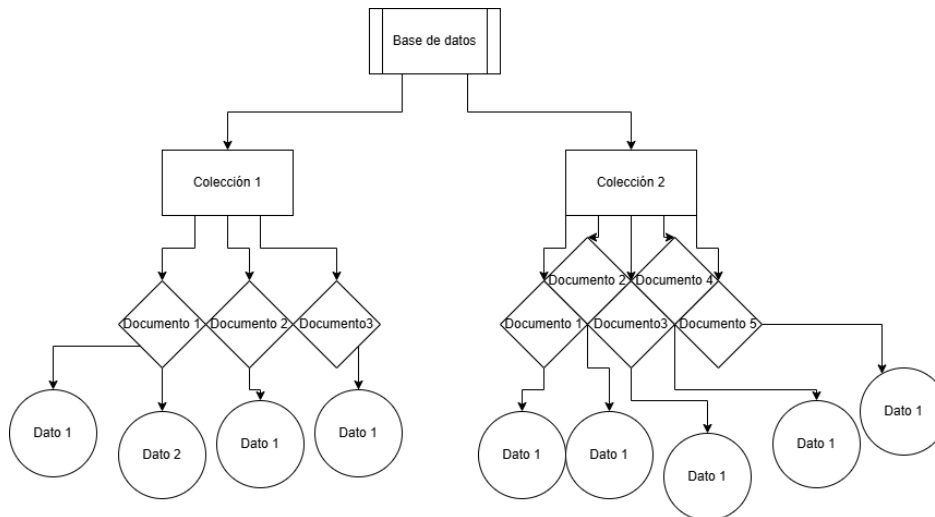


Figure 5.14: Diagrama de la estructura de la base de datos *Firestore*.

servicio de Google. Se hace uso de **Firestore** uno de los tipos de base de datos que ofrece *Firebase*. Se trata de una base de datos NoSQL cuya estructura es observable en la Figura 5.14 que está dividida en colecciones en la que se pueden guardar documentos con sus propios datos en un formato JSON. La versión gratuita ofrece 1GB de almacenamiento y 70.000 operaciones de lectura/escritura al día. Posteriormente se diseñó el menú de crear cuenta e iniciar sesión y comenzamos su implementación.

Haciendo lo mismo con el ranking, se detectó un fallo mientras se implementaba. Este se encontraba en la consulta para pedir los datos de la base de datos. Se desconoce si el fallo estaba en alguna operación interna del plugin o en la base de datos *Firestore* ya que el código usado venía dado de ejemplo en la wiki de *GodotFirebase* y se supuso correcto al no dar ningún error. Por suerte, estaba localizado únicamente en la función de auto ordenación de los datos devueltos, que al ser utilizada devolvía siempre un array vacío, y no en la consulta completa a la base de datos, así que fue fácilmente evitable. En lugar de pedir la información ya ordenada, se piden todas las entradas del ranking y se ordenan con un método propio para luego poder elegir las 10 puntuaciones más altas y mostrarlas.

Para hablar del ranking hay que hablar de puntuaciones, pues es lo que va a determinar la posición de los jugadores en el mismo. En este momento es cuando se crea el sistema de puntuación del juego haciendo uso de la información de partida recopilada por el gestor de estadísticas. Se crea una pequeña fórmula matemática 5.1 que en función de varios datos da una puntuación final de partida. La decisión de los valores empleados es para darle un valor similar al tiempo sobrevivido que a los enemigos vencidos. En cuanto a la vida perdida es para recompensar a los jugadores capaces de esquivar con éxito sin penalizar demasiado a quien le cueste más dominar el movimiento del personaje.

$$s * 4 + \sum p - v/2 \quad (5.1)$$

Los símbolos de la ecuación anterior significan los siguiente:

1. **s**= Segundos sobrevividos
2. **p**= Puntuación individual de cada enemigo vencido
3. **v**= Cantidad de vida total perdida

En último lugar, se crearon varias colecciones más en *Firestore* donde se guardará información de las estadísticas de uso de los usuarios en cada partida.

Tras comprobar el correcto funcionamiento de todas las características recién implementadas, se

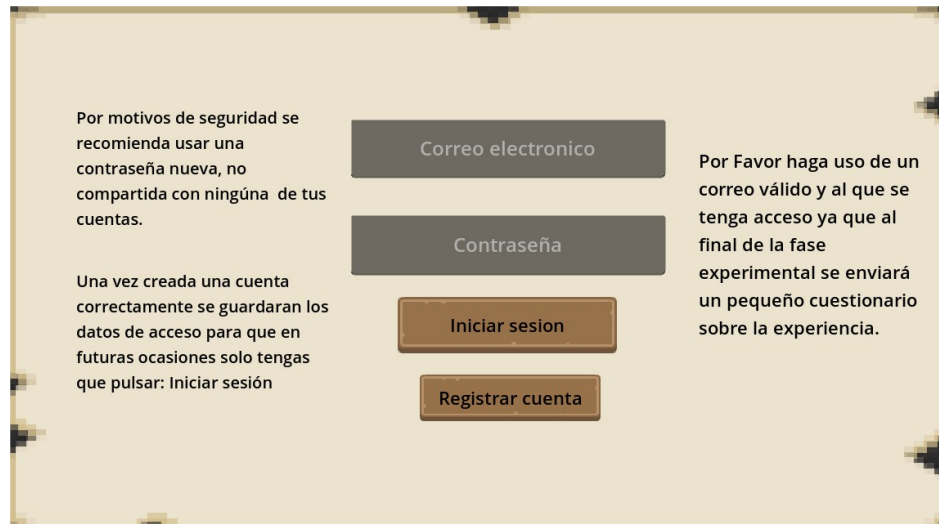


Figure 5.15: Versión final del menú de inicio de sesión.

exportó el proyecto a web para comprobar los sistemas en vivo. Ahí se encontró un problema que no fue posible localizar ni arreglar. Desde el navegador el juego no se comunicaba correctamente con la base de datos aunque desde el editor de *Godot* si lo hacía. Esto se solucionó finalmente cambiando a la última versión del motor, la Beta 4.3, que mejora la exportación a web. El uso de la beta sin embargo, no viene exento de riesgo y siempre está la posibilidad de introducir problemas nuevos e inesperados, como fue en este caso.

La herramienta Tilemaps usada para la creación del mundo es la afectada. En este caso una de sus opciones, la encargada de garantizar que las celdas se fusionen de forma invisible al usuario, no funciona correctamente, lo que provoca que al moverse el jugador aparezcan líneas negras entre las casillas que forman el mundo de juego. La única posible solución y que no garantizaba su eficacia era modificar la imagen usada en el TileMap de forma extensa, cosa que llevaría bastante tiempo y al no garantizar el resultado esperado se decidió dejar el error gráfico.

5.1.10. Iteración 9: Últimos ajustes

En la **novena y última iteración** se agregaron los últimos elementos al juego y se procedió a arreglar errores y a mejorar el aspecto de los menús e interfaz de juego.

El último elemento añadido son los coleccionables. Estos se obtienen al vencer al jefe final en cada partida y proporcionan al usuario información extra de la historia del mundo del videojuego, así como huevos de pascua, término que se usa para definir pequeños secretos que hacen referencia a cosas externas al propio videojuego, tales como pueden ser elementos de la cultura pop o referencias directas a otros videojuegos.

También se hacen ajustes finales de dificultad y de errores tanto en código como en elementos visuales y de textos.

Lo último que queda por hacer es terminar de definir la separación de versiones. A lo largo del desarrollo todos los elementos que están presentes en solo una de las versiones se han codificado teniendo eso en mente y preparando el terreno para la separación. Todas las características que se encuentran en solo una de las versiones tienen un condicional según la versión del juego que le corresponda, así si algo solo está en la versión que medirá la motivación extrínseca, comprobará una variable global que determinará su activación. Se implementa además un sistema para que el juego automáticamente al crearse el usuario una cuenta, le asigne una de las dos versiones. Así, cuando se conecte cada usuario el juego se adaptará a la versión que le corresponde a cada uno en el experimento.



Figure 5.16: Versión final del menú principal.

Por último se deja disponible una versión del juego ligeramente modificada para poder ser probada por cualquier lector de este documento o interesado en el juego: <https://segra.itch.io/hierromante-public?secret=w68PXN9UjBF0vjmPCo2S1g4Ac4>.

Experimento y análisis de datos

Este capítulo está dedicado al análisis de los datos obtenidos de métricas del juego y de las encuestas finales a los usuarios. Aquí se verá la eficacia de los elementos de motivación extrínseca para inducir una adherencia a corto plazo en los jugadores, es decir, el interés de seguir jugando más allá de las primeras partidas.

Antes de entrar en materia hay que advertir de un problema encontrado en esta fase del proyecto. Se ha detectado tanto por análisis de datos como por conversaciones con participantes de la fase de experimentación que algunas de las partidas jugadas no mandaron sus métricas al servidor. Gracias a la comunicación con algunos de los afectados se ha visto que casi todas las partidas que no mandaron datos al servidor fueron jugadas en dispositivos móviles por lo que se deduce que quizás se deba a pérdidas de conexión temporales que no recuperaban la sesión o alguna configuración de *Android* que no permitía la comunicación adecuada con el servidor. Con esto en mente, al hablar de datos y realizar cálculos se hará con los recogidos, pese a la posibilidad de que este error pueda provocar un pequeño sesgo.

6.1. PROTOCOLO

Se van a detallar los pasos dados para llevar a cabo la fase experimental del proyecto divididos en varias partes. Estos están visibles en la Figura 6.1

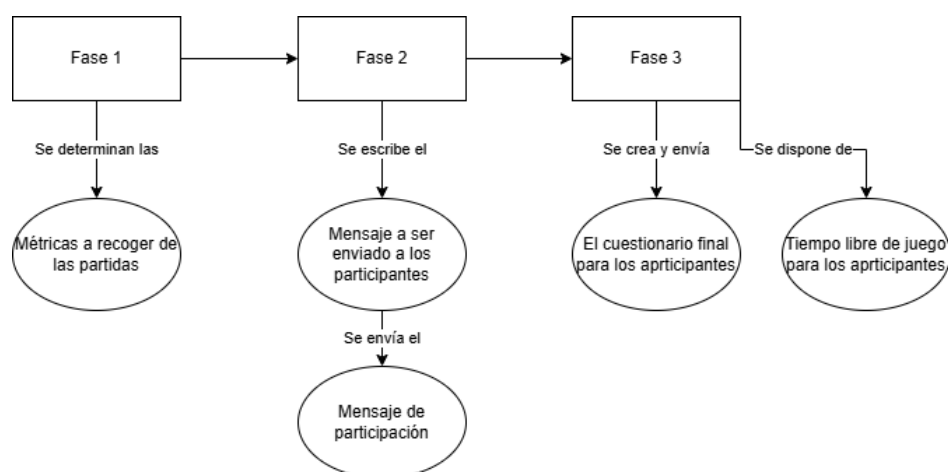


Figure 6.1: Tabla con todas las preguntas del cuestionario final.

6.1.1. Fase 1

Se definen los datos a recoger de las partidas de los usuarios para que sean relevantes y útiles en la evaluación de los resultados. Las métricas elegidas para ser recogidas de las partidas son las siguientes:

- **Tiempo de partida.** La duración que tiene cada partida jugada. Esto también da información relacionada con los eventos que funcionan con temporizador como la aparición de nuevas oleadas y tipos de enemigos o el desbloqueo de algunas armas y mejoras. Según cuanto haya durado la partida, también se puede saber si el usuario sobrevivió lo suficiente como para llegar al enfrentamiento con el jefe final.
- **Puntuación.** Determinada por lo bien que ha jugado una partida el usuario. Viene definida por el tiempo de partida como por los enemigos vencidos en esta. Formula usada visible en la *Iteración 8* del capítulo *Resultados*.
- **Derrotar al jefe.** Indica si se consiguió o no vencer al jefe final.
- **Dispositivo móvil.** Valor booleano que indica si se jugó en un dispositivo móvil o no.
- **Grupo.** En cada partida se guarda el grupo al que se pertenece para tenerlo más visible a la hora del análisis. También usado para asegurar que el usuario está jugando a la versión correcta asignada al crear la cuenta.

6.1.2. Fase 2

Esta comienza con el envío del enlace privado para poder acceder al videojuego. El enlace se envía acompañado de un texto explicativo que indica algunos detalles que los usuarios deben tener en cuenta. Se avisa de que es necesario crear una cuenta antes de empezar a jugar y que se enviará una encuesta al final del periodo de juego, que consistía de un total de cinco días en los que jugar de forma libre. Al buscarse medir la adherencia a corto plazo no hay necesidad de dejar a los jugadores un gran número de días para el experimento pues se sale del objetivo del proyecto. Además, a pesar de que exista rejugabilidad en el proyecto, en menos de diez partidas un usuario puede ver todo lo que ofrece el juego en cuanto a armas, mejoras, coleccionables e incluso logros. Parte de las instrucciones eran sobre el objetivo del proyecto: medir la motivación. Sin embargo, no se divulgaba que había dos grupos distintos de jugadores, para evitar que hubiese algún sesgo entre los jugadores y que no haya sentimientos de insatisfacción por creer que están jugando a "una versión incompleta del juego" por estar en el grupo intrínseco. Al tratar de ver la eficacia de los elementos de motivación extrínseca, se aclaraba que no había ninguna obligación de jugar cierto número de partidas, que tenían que jugar la primera y a partir de ahí, eran libres de jugar cuando les apeteciese o no hacerlo más si no tenían interés en ello.

Otro punto a destacar en las instrucciones enviadas es el aviso de compatibilidad con navegadores y dispositivos. A lo largo de este año se han actualizando los navegadores y esto causó un problema con uno de ellos. Por problemas de incompatibilidad con algunas opciones de *Godot*, el navegador Mozilla Firefox dejó de ser compatible con el juego. De manera similar algunos dispositivos de *Apple* podían presentar problemas a la hora de ejecutarse.

A continuación el mensaje íntegro enviado a los participantes del experimento:

"¡Muy buenas! Este mensaje es para pedir la participación en la fase experimental de un TFG. Esta consiste en jugar a un videojuego en navegador para medir la adherencia de las mecánicas en videojuegos. Para ello solo debéis seguir el enlace adjunto, crear una cuenta y jugar al juego. Juega tanto como quieras y deja de jugar cuando no te apetezca, no lo hagas por obligación. El domingo cerraremos el experimento por lo que podrás jugar durante estos próximos días. Para poder jugar hay que usar alguno de los siguientes navegadores: -Opera, -Chromium, -Thorium, -Microsoft Edge, -Vivaldi, -Chrome, -Brave. El juego NO funciona en Mozilla Firefox ni en dispositivos Apple. Por Favor hágase uso de un correo válido y al que se tenga acceso ya que al final de la fase experimental que durará hasta este lunes día 1, se enviará un pequeño cuestionario sobre la experiencia."

6.1.3. Fase 3

Consiste en el periodo de juego libre de los usuarios y en los cuestionarios finales enviados al acabar los cinco días de los que disponían para jugar. A continuación se muestran más detalles sobre los

cuestionarios finales. Para las encuestas se usó el servicio *Forms* de Microsoft, donde se crearon dos encuestas diferentes aunque con las mismas preguntas, ya que los jugadores no sabían de la existencia de los dos grupos.

En la Figura 6.2 se muestran las preguntas del cuestionario enviado al final del periodo de juego y que en apartados siguientes se desarrollaran más en profundidad. Este comienza con una parte general para incluir información del usuario como el género y la edad y una tercera pregunta sobre la frecuencia a la que juegan a videojuegos con una escala Likert que limita las respuestas a 5 opciones distintas para indicar lo de acuerdo o desacuerdo que están con la pregunta. En la segunda parte del cuestionario están las preguntas relacionadas con la experiencia de juego de los usuarios y las partes que les fomentaron el interés en seguir jugando.

Pregunta	Tipo de pregunta	Tipo de respuesta
¿Cual es tu género?	Personal	Selección limitada
¿Cual es tu edad?	Personal	Selección de rango
Frecuencia con la que juegas normalmente a videojuegos	Personal	Escala Likert
Sentiste que tenías motivos para jugar más después de la primera partida	Sobre el juego	Escala Likert
Cuando acababas una partida tenías ganas de echar otra.	Sobre el juego	Escala Likert
Al subir de nivel y conseguir mejoras y ataques nuevos hacía que quisieras seguir jugando	Sobre el juego	Escala Likert
Si llegaste a ganarle al jefe final, después, perdiste la motivación de seguir jugando	Sobre el juego	Escala Likert
Disfrutaste del juego	Sobre el juego	Escala Likert
Recomendarías el juego	Sobre el juego	Escala Likert
¿Qué cambiarías o añadirías para hacer el juego más a tu gusto?	Sobre el juego	Respuesta libre
¿Que elemento o elementos te han impulsado a querer seguir jugando? (Si es que los hay)	Sobre el juego	Respuesta libre

Figure 6.2: Tabla con todas las preguntas del cuestionario final.

6.2. MÉTRICAS DE JUEGO

En primer lugar se van a estudiar los datos numéricos obtenidos de las partidas jugadas pero antes, aquí se dispone el acceso a un repositorio público donde se pueden ver todos los datos recopilados incluidos los resultados de las encuestas <https://zenodo.org/uploads/12804966>.

En total 27 personas se registraron en el juego y 26 llegaron a jugar al menos una partida. De esas 27 personas, el juego les asignó de forma aleatoria la versión base de juego, la puramente **intrínseca**, a 13 jugadores y la versión con elementos extrínsecos, **versión extrínseca**, a 14. Al no tener partidas registradas de uno de los participantes, todos los cálculos realizados se hacen sin tenerle en cuenta, lo que implica un total de 13 jugadores para cada versión. Se tienen los siguientes datos demográficos de los participantes según las respuestas a las encuestas finales donde contestaron 24 de los 26 usuarios. Hubo 5 participantes del género femenino, 18 masculino y 1 no binario, representando 20,8%, 75% y 4.2% respectivamente (Figura 6.3). De la edad de los usuarios se tiene que el rango más frecuente es el de 26 a 32 años con 16 participantes, se puede destacar también la presencia de 1 persona menor a 18 años y 3 entre 41 y 50 años.

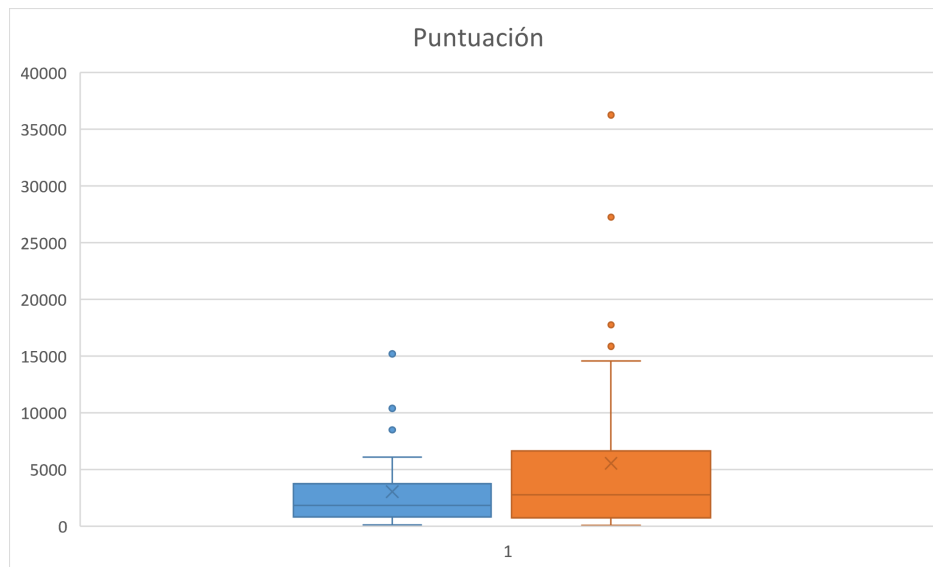


Figure 6.4: Distribución de las puntuaciones de las partidas en los dos grupos.

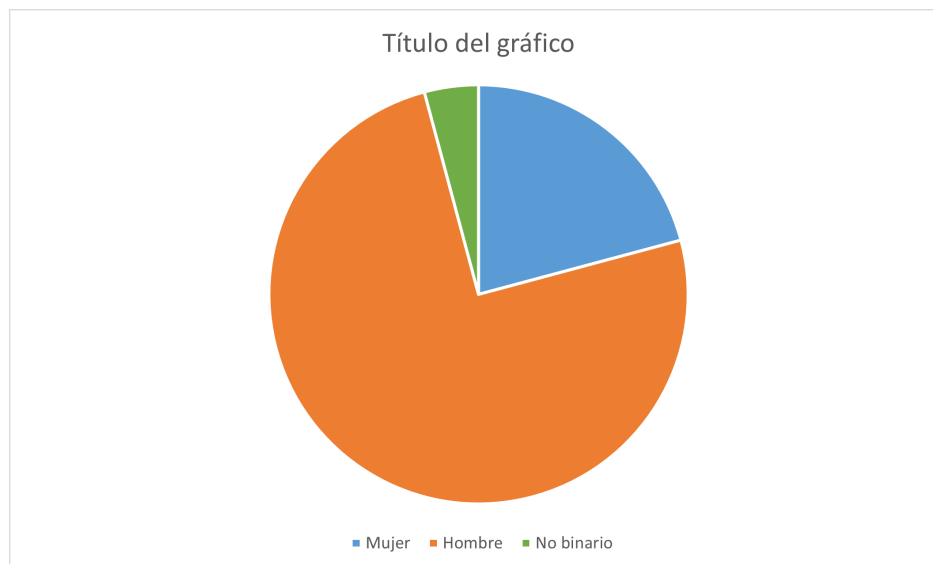


Figure 6.3: Gráfico con la distribución de los géneros de los participantes.

Esos 26 usuarios jugaron un total de 65 partidas, 31 en la versión intrínseca y 34 en la extrínseca. Antes de seguir con el análisis de las métricas se ha realizado un proceso de limpieza de datos en los que se ha observado la existencia de 3 partidas con un tiempo de juego inferior a dos segundos. Al tratarse de partidas en las que realmente no se ha jugado se ha determinado necesaria su eliminación de los cálculos realizados quedando por lo tanto 29 partidas en el grupo intrínseco y 33 en el extrínseco.

Con estos datos se puede calcular una media de 2,23 partidas por usuario en la versión base y 2,53 en la extrínseca. A priori no hay mucha diferencia entre los dos grupos pero si se analizan otros datos de juego, estas sí son apreciables. Mientras que en la versión base la puntuación media por partida es de 3034,10 puntos, para el grupo extrínseco se alcanza una media de 5548,52. Con estos datos si se ve una mayor dedicación por parte del grupo con la versión extrínseca. La puntuación más alta alcanzada por los jugadores ha sido de 36263 puntos y corresponde a un participante del grupo extrínseco, de hecho, 4 de las 5 mayores puntuaciones son de jugadores del grupo extrínseco como se puede observar en la Figura 6.4.

En cuanto al tiempo, se mostrarán los datos recopilados en un formato **minutos:segundos** para

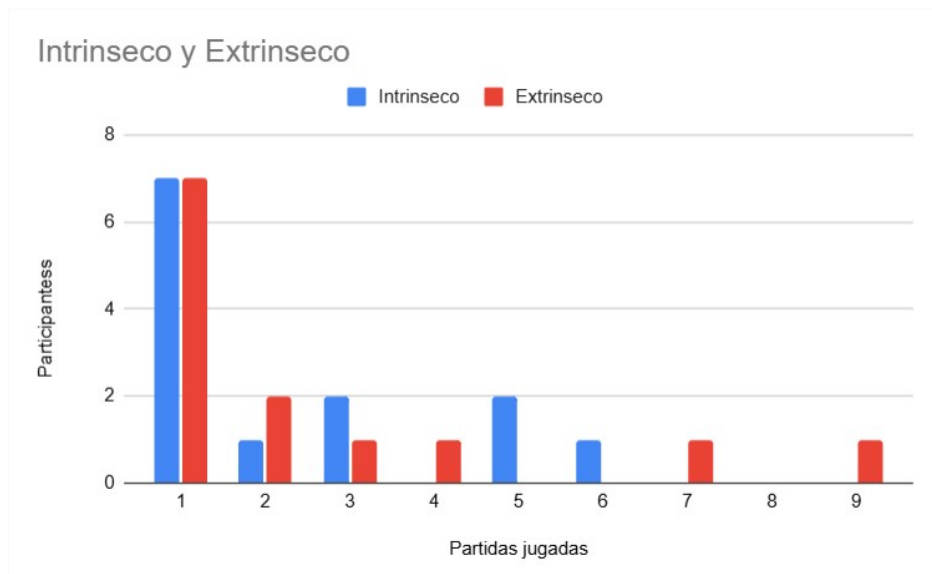


Figure 6.5: Distribución del número de partidas por jugador.

ser fácilmente interpretados. Comenzamos con cuanto se han pasado jugando los usuarios, se tiene una media de 07:56 minutos para el grupo intrínseco y 13:00 minutos para el extrínseco, una vez más una diferencia notable. Mirando en los dos grupos, el tiempo máximo dedicado a una partida ha sido de una hora dos minutos y veintidós segundos.

En la Figura 6.5 se puede observar la distribución de partidas jugadas por participantes. Destacando el hecho de que la mitad de usuarios de cada grupo, siete, ha jugado únicamente una partida. También es interesante hablar del hecho de que en el grupo extrínseco se encuentran las personas que más partidas han jugado.

Analizando las partidas de esos usuarios que solo llegaron a jugar una, se pueden ver varios datos interesantes. En la Figura 6.6 está la representación de la duración media de las partidas, donde se aprecia una gran diferencia entre los dos grupos. La media del grupo extrínseco es más del doble que en el grupo intrínseco. La partida más larga del grupo extrínseco supera la media hora, mientras que en el grupo intrínseco no llega a los diez minutos. Observando a fondo el diagrama de bigotes se ve que la mediana de duración en el caso extrínseco es bastante baja comparada con el valor más alto por lo que se deduce que ese jugador está bastante por encima del resto.

En cuanto a la puntuación de estos jugadores se puede ver en la Figura 6.7 que los resultados son similares a la duración de las partidas como cabría esperar. La media de puntuación del grupo extrínseco alcanza los casi 6000 puntos mientras que en el grupo intrínseco supera por poco los 1000. Si se miran las puntuaciones máximas, en los participantes extrínsecos alcanza un pico muy lejos del resto de usuarios que solo jugaron una partida mientras que en los usuarios del grupo intrínseco los datos están mucho más agrupados.

6.2.1. Jefe final

En el grupo extrínseco observable en la Figura 6.8, se aprecia que un total de 16 partidas de 34 alcanzaron el jefe final y en 9 de estas consiguieron abatirle representando el 26,47% de las partidas del grupo extrínseco. En un 56,25% de las partidas que alcanzaron al jefe se le venció en este grupo. En cuanto a la cantidad de jugadores distintos que llegaron al jefe en este grupo tenemos 6 de los cuales 5 consiguieron vencer.

Las métricas del **jefe final** en el grupo intrínseco se pueden observar en la Figura 6.9. En 11 de las 31 partidas jugadas se llega al enfrentamiento final y de éstas, en 7 partidas se le vence siendo esto el 22,58% de las partidas de dicho grupo. En el 63,64% de las partidas que alcanzaron al jefe final

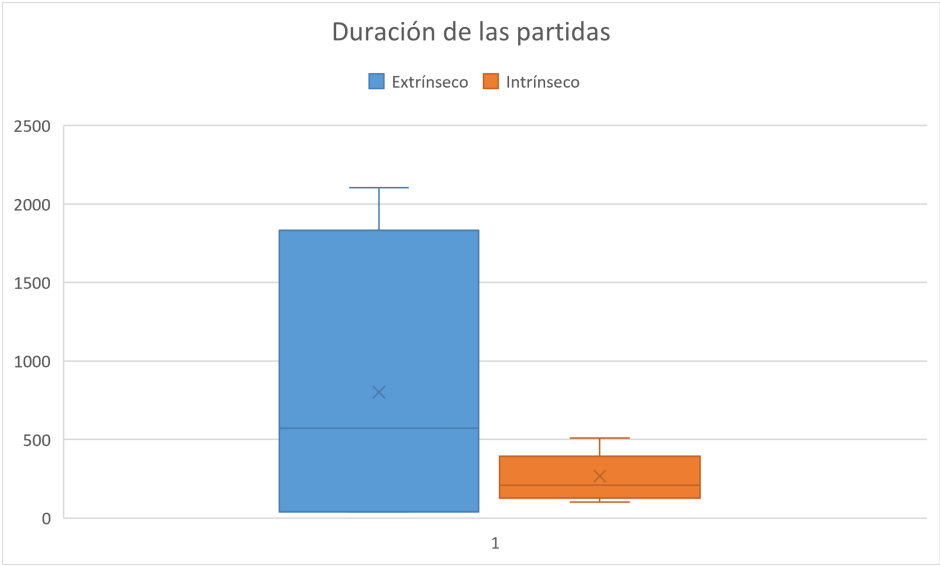


Figure 6.6: Comparación de las duraciones de partida de los participantes que solo jugaron una partida.

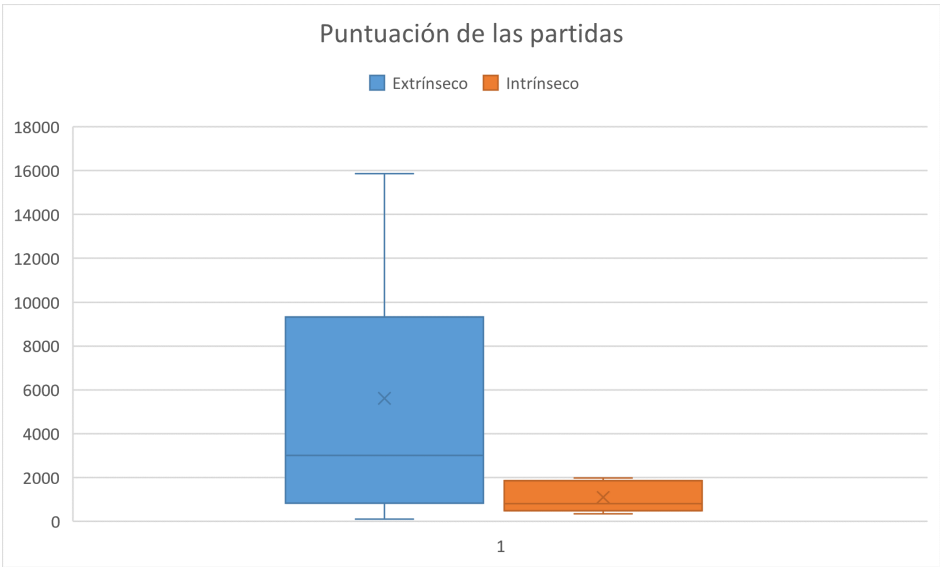


Figure 6.7: Puntuaciones de los participantes que solo jugaron una partida.

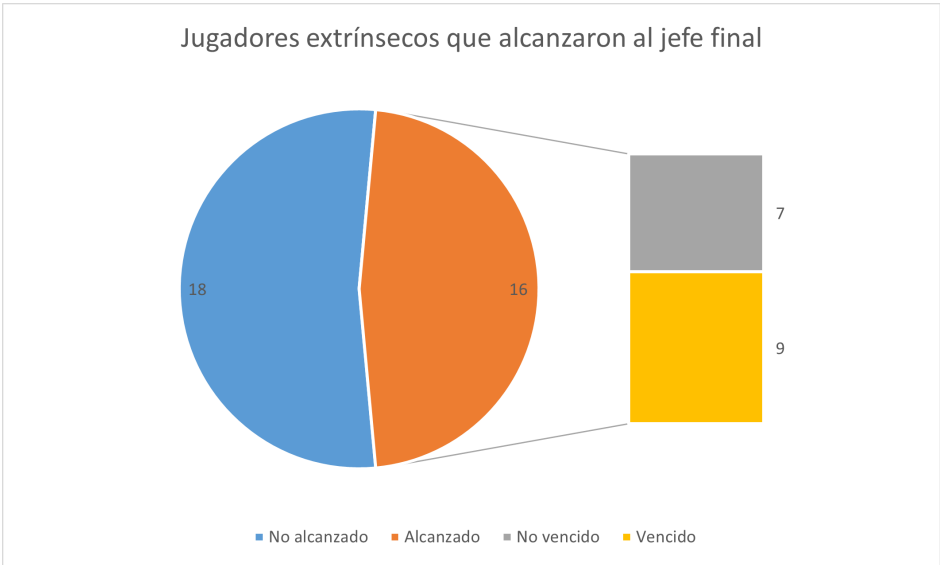


Figure 6.8: Distribución de los enfrentamientos con el jefe final del grupo extrínseco.

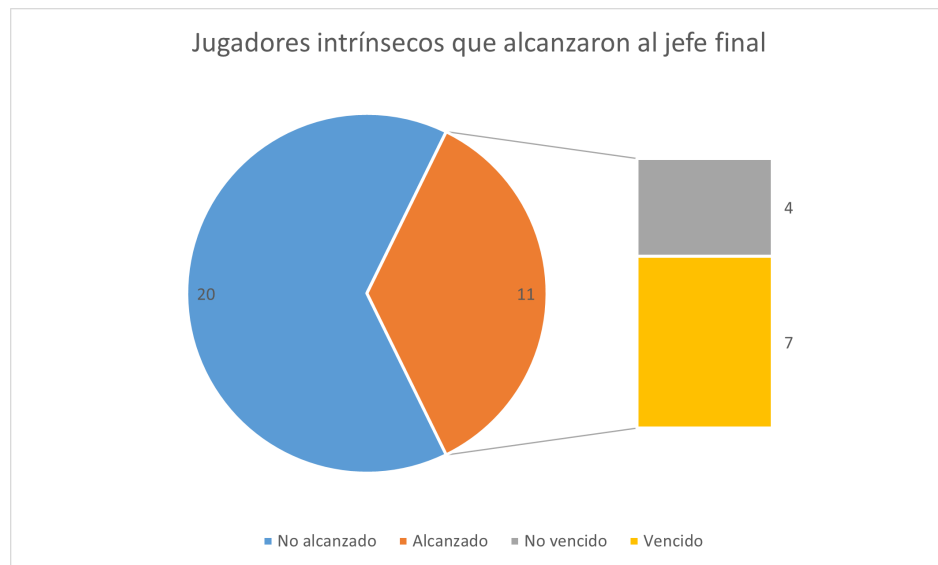


Figure 6.9: Distribución de los enfrentamientos con el jefe final del grupo intrínseco.

en la versión intrínseca lo vencieron. Hay que destacar que esos números están distribuidos entre 7 jugadores que llegaron a enfrentarse al jefe final y 5 que consiguieron vencerle, por tanto, menos de la mitad del grupo llegó a ganar una partida.

Como se ha podido observar comparando los datos, se ha llegado el enfrentamiento con el jefe final un mayor número de veces en el grupo extrínseco pero se le ha vencido un menor porcentaje de veces.

En cuanto a estadísticas específicas de dispositivos móviles se destaca que han supuesto un total de 15 partidas, 8 de ellas en el grupo intrínseco y 7 en el extrínseco. En el primer grupo hay una media de 3:56 minutos de tiempo de partida y 1109,25 puntos, unas métricas bastante por debajo de la media de la versión intrínseca. En el segundo se observa un tiempo medio de partida de 2:55 minutos y 917,57 puntos de media. Unos número no solo más bajos que la media extrínseca si no que también más bajos que las medias de dispositivos móviles del grupo intrínseco.

6.3. DATOS DE ENCUESTAS

A continuación se hablará de las preguntas presentes en la encuesta enviada a los jugadores. Habiendo hablado de las preguntas de carácter general en la sección anterior, ahora vienen las específicas a la experiencia con el juego. De las 26 personas que jugaron al juego, 24 respondieron a la encuesta, 11 del grupo intrínseco y los 13 del grupo extrínseco.

La primera pregunta es *"Cuando acababas una partida tenías ganas de echar otra"* a lo que los participantes respondieron lo visible en la Figura 6.10. Como se puede observar hay una clara diferencia en los porcentaje entre las dos versiones. En la versión intrínseca ningún usuario dio una respuesta negativa y muy pocos fueron indiferentes mientras que en la extrínseca si hay usuarios que no expresaron deseo de seguir jugando después de acabar una partida y un mayor número de indiferentes. Esto a priori choca con los datos recogido en el grupo intrínseco pues la mitad de los jugadores solo jugaron una partida a pesar de haber mostrado interés por jugar más.

La siguiente pregunta está relacionada con la anterior pero siendo más específica, *"Sentiste que tenías motivos para jugar más después de la primera partida"*. En la "Pregunta 2" de la Figura 6.10 se observa que casi un 40% de los participantes de los dos grupos respondieron de manera negativa o indiferente. Si que hay una diferencia en las respuestas muy positivas, mientras que casi el 40% dijo estar "muy de acuerdo" en el grupo extrínseco, en el intrínseco ese número solo llega al 25%. Observando esto si tienen más sentido las métricas recopiladas donde casi el 50% de personas jugaron

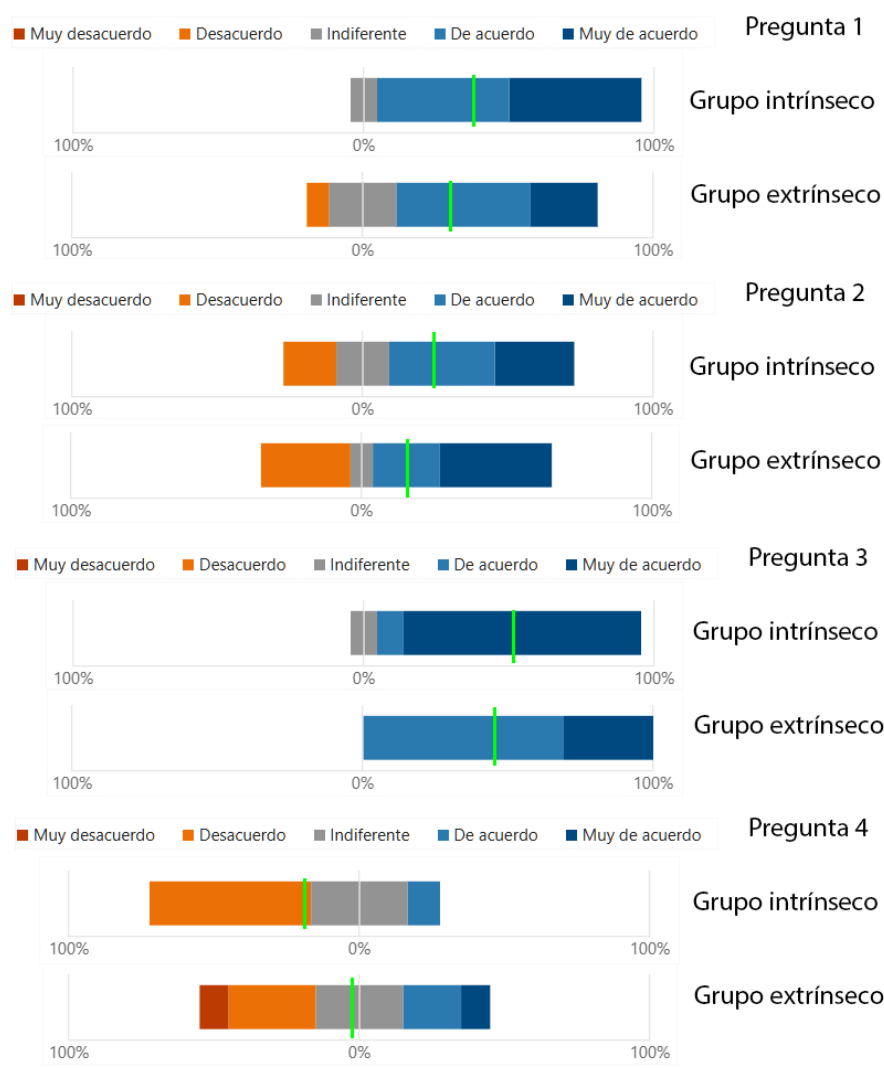


Figure 6.10: Respuestas a las preguntas sobre lo que han sentido jugando del cuestionario según el grupo. Las líneas verdes representan la mediana.

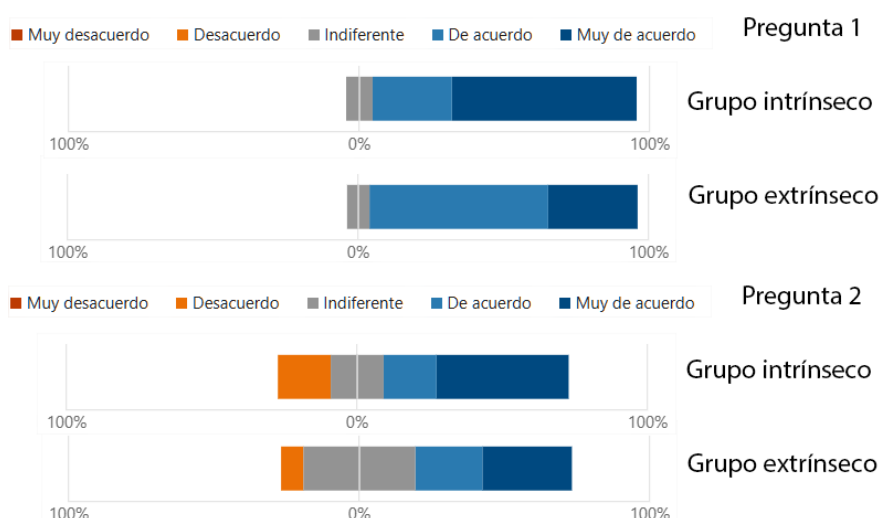


Figure 6.11: Respuestas a las preguntas del cuestionario sobre su satisfacción del juego y la recomendación de este. Las líneas verdes representan la mediana.

una única partida. Esto teniendo en cuenta siempre la pérdida de datos mencionadas al inicio del capítulo.

Sobre la pregunta *"Al subir de nivel y conseguir mejoras y ataques nuevos hacía que quisieras seguir jugando"* ambos grupos respondieron de forma positiva y el grupo intrínseco dio una respuesta extremadamente positiva observable en la *"Pregunta 3"* de la Figura 6.10. Esto probablemente este relacionado con el hecho de que al no tener otros elementos de juego que explorar como los logros o el ranking, se centrase casi toda su atención en la obtención de nuevos ataques y mejoras únicas para intentar probarlas todas.

En cuanto a la pregunta sobre el jefe final *"Si llegaste a ganarle al jefe final, después, perdiste la motivación de seguir jugando"* las respuestas son parecidos pero con una diferencia llamativa. En el grupo extrínseco el 30% de los usuarios afirman haber perdido la motivación de jugar después de ganarle al jefe, frente a un 10% de usuarios del grupo intrínseco, datos visibles en en la *"Pregunta 4"* de la Figura 6.10. En primera instancia uno pensaría que estos datos deberían estar al revés pues los jugadores extrínsecos disponen de más objetivos claros y visibles que no dependen del jefe.

Los resultados de la pregunta *"Disfrutaste del juego"* indican que la mayoría de los participantes disfrutaron en mayor o menor medida del juego, con solo un bajo porcentaje de usuarios indiferentes. Datos observables en la *"Pregunta 1"* de la Figura 6.10

En la pregunta *"Recomendarías el juego"*, el 50% de los jugadores del grupo extrínseco respondieron de forma positiva mientras que en el intrínseco superó el 60% de participantes. Del 50% restante de los usuarios extrínsecos, 40% respondieron indiferente y los restante dijeron que no. En cuanto al grupo intrínseco, el porcentaje restante está dividido de forma igualitaria con un 20% indiferente otro 20% con una respuesta negativa. Datos observables en la *"Pregunta 2"* de la Figura 6.10.

A continuación se revisarán las dos preguntas de respuesta libre. La primera de ellas *"¿Qué cambiarías o añadirías para hacer el juego más a tu gusto?"* tiene una gran variedad de sugerencias, algunas más cortas y otras con ideas muy detalladas provenientes de usuarios que juegan frecuentemente videojuegos. Se va a hablar de las respuestas más comunes o consideradas más informativas.

- Mejorar el **control en móvil** es una propuesta de mejora presente en las respuestas de ambos grupos. Siendo cierto que los controles en dispositivos móviles es mas complicado que con un teclado y teniendo en cuenta que un reducido número de usuarios ha utilizado un móvil para jugar, se considera que es un punto importante para los jugadores.

- Dos usuarios han sugerido que se pueda **interactuar** con los edificios concretamente y un gran número ha dicho que les gustaría más mecánicas jugables y contenido en general. Algunas de esas sugerencias de contenido vienen del grupo intrínseco y hablan específicamente de elementos que sí están presentes en la versión intrínseca como estadísticas y coleccionables.
- El **balance de dificultad** ha sido otro punto bastante mencionado. Mientras que los jugadores habituales han encontrado el juego fácil, la gente con menos experiencia en videojuegos se ha visto ante un reto bastante grande.

La ultima pregunta, "**Que elemento o elementos te han impulsado a querer seguir jugando? (Si es que los hay)**" también ha tenido algunas respuestas comunes entre los dos grupos.

- La respuesta más común ha sido **encontrar la combinación de ataque más poderosa**, la búsqueda de satisfacción al ver aumentar su fuerza.
- En el grupo extrínseco ha destacado el **ranking**. La mitad de las respuestas han sido relacionadas con superar al resto de participantes en la clasificación de puntos. Teniendo esto en cuenta y la dedicación visible en las métricas, de algunos participantes del grupo extrínseco por hacer buenas partidas y conseguir muchos puntos, no sería erróneo afirmar que el ranking ha sido el elemento de motivación extrínseca más exitoso.

Conclusiones

Después de desarrollar un videojuego con dos versiones con la intención de medir la eficacia de algunas técnicas empleadas por los diseñadores de videojuegos para fomentar la motivación extrínseca de los jugadores y conseguir así una mayor adherencia, se realizó un estudio midiendo las métricas de juego de los participantes. En ese estudio se consiguieron datos que indican que los elementos de motivación extrínseca sí funcionan pues los usuarios de ese grupo aparte de jugar más partidas, se veían motivados para intentar sacar una mejor puntuación en estas.

Este último capítulo se dedica a revisar el cumplimiento de los objetivos propuestos así como posibles trabajos futuros y mejoras y por último un repaso a las competencias adquiridas.

7.1. TRABAJOS DERIVADOS Y FUTUROS

Algunos puntos a mejorar según los usuarios encuestados:

- Destacando las sugerencias de mejora de los participantes está en primer lugar el añadir **nuevas mecánicas**. Algo que varios usuarios dijeron es añadir armas o mejoras únicas que combinasen entre ellas para crear nuevos ataques especiales.
- **Ajustar más la dificultad**. Como esto ha sido un punto que ha recibido peticiones tanto para aumentar como para disminuir la dificultad, se proponen dos posibles soluciones. Una de ellas es la adición de un selector de dificultad para las partidas, de esta forma los jugadores que necesiten más tiempo para aprender y adaptarse podrán tener la libertad de hacerlo y los que busquen un reto podrán encontrarlo desde el principio. La otra, quizás más orgánica e imperceptible a los usuarios es haciendo uso de otra de las propuestas recibidas. Implementar múltiples niveles con escenarios completamente distintos y que se vayan desbloqueando a medida que se completan los anteriores, yendo de los niveles más fáciles a los más difíciles. También hubo una propuesta relacionada con la dificultad después de vencer al jefe final, y es aumentar drásticamente la dificultad para que cualquier punto conseguido en ese momento sea un desafío.

Las sugerencias de los participantes no son la única fuente de posibles mejoras. También se tienen en mente consideraciones propias como las siguientes:

1. Dotar a los enemigos de **comportamientos distintos**. Actualmente solo hay dos tipos de enemigos los que se acercan para golpearte y los que disparan cuando están a cierta distancia del jugador. Dotando de formas nuevas y distintas de moverse o atacar a algunos enemigos le dará más variedad al juego.
2. Desde un punto de vista más técnico, se podría implementar el **guardado de partidas en la base de datos** utilizada para la recogida de métricas, de esa forma los jugadores tendrán sus estadísticas y desbloques disponibles en cualquier dispositivo.

Otra posibilidad es la expansión del propio proyecto. En lugar de centrarse en la adherencia inmediata, estudiar también los efectos de los elementos de motivación extrínseca a largo plazo. Para eso se necesitaría más tiempo para los jugadores en la fase de experimentación y también una

ampliación del juego con más mecánicas y contenido en general para aumentar la rejugabilidad y tener de esta forma un juego más largo y por tanto más capacitado para la medición de adherencia a largo plazo. Se podría incluso desarrollar más versiones donde cada una tenga el enfoque en distintos elementos para medir de forma precisa la eficacia de cada uno pues el proyecto actual calcula la eficacia de todas las características que fomentan la motivación extrínseca de forma conjunta.

7.2. OBJETIVOS ALCANZADOS

A continuación una revisión de los objetivos propuestos y como se han resuelto.

- **Realizar una revisión del estado del arte sobre el diseño de videojuegos, la adherencia y la motivación de los jugadores.** Se realizó una investigación en la materia tanto desde el punto de vista psicológico como del diseño de videojuegos como puede verse en el capítulo 3.
- **Estudiar, definir y clasificar los elementos de diseño pertenecientes a cada una de las motivaciones para su implementación.** En base a las prácticas de la industria de videojuegos y a las dos teorías sobre la motivación mas influyente en esta se determinaron las características de diseño a tener en cuenta para el juego del proyecto. Esto queda reflejado en el segundo apartado del capítulo 3, *Clasificación de los elementos de diseño*.
- **Desarrollar los elementos comunes a las dos versiones del videojuego.** Se desarrolló un videojuego completo del género *SurvivorsLike* con mecánicas como la subida de nivel, la selección de mejoras y una batalla con un jefe final.
- **Implementar los cambios a cada una de las versiones para fomentar cada tipo de motivación a los jugadores.** Se agregaron a la versión base las características que fomentan la motivación extrínseca obteniendo así las dos versiones de juego.
- **Determinar la influencia de cada elemento en la adherencia de los jugadores mediante la recogida de datos y estudio posterior de los resultados a través del videojuego .** Mediante la etapa de experimentación, que se ve relegada en el capítulo 6 y su correspondiente recogida de información se realizó luego un análisis de esos datos para concluir la eficacia de las características que fomentan la motivación extrínseca.

7.3. JUSTIFICACIÓN DE COMPETENCIAS ADQUIRIDAS

A lo largo de la realización de este TFG se han aplicado las competencias correspondientes a la Tecnología Específica de *[Tecnologías de la información]*:

Tecnologías de la información 3: *[Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas]*. Mediante el uso de la metodología incremental iterativa basada en prototipos se ha mantenido un desarrollo con el usuario final siempre en mente. Sumándole además las características usadas para la recopilación de métricas se consigue la competencia.

Tecnologías de la información 2: *Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.* En la búsqueda de las herramientas de trabajo, en su uso extensivo para la creación y publicación del videojuego.

Tecnologías de la información 6: *[Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil]*. Haciendo uso del servicio de creación de cuentas y la base datos tanto para las métricas de las partidas como para el ranking mostrado en el juego, se cumplen los requisitos de la competencia.

7.4. VALORACIÓN PERSONAL

Aunque haya desarrollado algún videojuego anteriormente, el creado para este trabajo es el más grande de todos. Se han usado sistemas y características que no han sido empleadas con anterioridad. La mayor dificultad junto al aprendizaje de *Godot* y *GDScript* fue el proceso de documentarse, tanto sobre el factor psicológico general como en su papel en los videojuegos buscando y leyendo gran cantidad de entrevistas a profesionales del mundo del diseño como artículos científicos sobre videojuegos y sobre motivación. También mencionar la importancia de las pruebas de las funcionalidades a medida que se van implementando. Si no se hiciesen pruebas de lo añadido en cada prototipo estos irían escalando de manera problemática.

Personalmente estoy satisfecho con el trabajo realizado, con el tiempo he ido aprendiendo a controlar el sentimiento que se tiene al desarrollar un videojuego de decir *"pues ahora estaría bien meterle esto o lo otro"*, el control del alcance de un proyecto es una habilidad importante en el mundo del software. Por otra parte, eso no significa que haya un *Hierromante 2* tanto con las características y mejoras que propusieron los participantes como las que se hayan quedado en el tintero para el proyecto.

Bibliografía

- [1] Eric Bailey and Kazunori Miyata. Improving video game project scope decisions with data: An analysis of achievements and game completion rates. *Entertainment Computing*, 31:100299, August 2019.
- [2] James Close, Stuart Gordon Spicer, Laura Louise Nicklin, Maria Uther, Joanne Lloyd, and Helen Lloyd. Secondary analysis of loot box data: Are high-spending “whales” wealthy gamers or problem gamblers? *Addictive Behaviors*, 117:106851, June 2021.
- [3] Mihaly Csikszentmihalyi. Flow: The Psychology of Optimal Experience. In *Flow: The Psychology of Optimal Experience*. New York: Harper & Row, January 1990.
- [4] Edward L. Deci and Richard M. Ryan. *Intrinsic Motivation and Self-Determination in Human Behavior*. Springer US, Boston, MA, 1985.
- [5] Fulya Eyupoglu and John Nietfeld. Intrinsic Motivation in Game-Based Learning Environments. In *Game-Based Assessment Revisited*, pages 85–102. Springer Cham, November 2019.
- [6] Servet Gürer, Elif Surer, and Mustafa Erkayaoğlu. MINING-VIRTUAL: A comprehensive virtual reality-based serious game for occupational health and safety training in underground mines. *Safety Science*, 166:106226, October 2023.
- [7] Rame Hulaj, Markus B. T. Nyström, Daniel E. Sörman, Christian Backlund, Sebastian Röhlcke, and Bert Jonsson. A Motivational Model Explaining Performance in Video Games. *Frontiers in Psychology*, 11, July 2020.
- [8] Daniel Johnson, John Gardner, and Penelope Sweetser. Motivations for videogame play: Predictors of time spent playing. *Computers in Human Behavior*, 63:805–812, October 2016.
- [9] Jussi Kasurinen and Antti Knutas. Publication trends in gamification: A systematic mapping study. *Computer Science Review*, 27:33–44, February 2018.
- [10] Jeanine Krath, Linda Schürmann, and Harald F. O. von Korflesch. Revealing the theoretical basis of gamification: A systematic review and analysis of theory in research on gamification, serious games and game-based learning. *Computers in Human Behavior*, 125:106963, December 2021.
- [11] Ben Kuchera. The PA Report - Intrinsic vs. Extrinsic rewards in Klei’s latest game: Don’t Starve. <https://web.archive.org/web/20130117075748/http://penny-arcade.com/report/editorial-article/intrinsic-vs.-extrinsic-rewards-in-kleis-latest-game-dont-starve>, January 2013.
- [12] Keza MacDonald. The Mind Behind Dark Souls - IGN. <https://www.ign.com/articles/2011/11/08/the-mind-behind-dark-souls>.
- [13] Betsy McCoach and Jessica Kay Flake. The role of motivation. In *APA Handbook of Giftedness and Talent*, APA Handbooks in Psychology®, pages 201–213. American Psychological Association, Washington, DC, US, 2018.
- [14] Randy Pagulayan, Kevin Keeker, Dennis Wixon, Ramon Romero, and Thomas Fuller. User-Centered Design in Games. *Human-Computer Interact. Handb*, pages 883–906, January 2003.

- [15] Ken Perlin. Chapter 2: Noise Hardware. *ACM SIGGRAPH Computer Graphics, Volume 19, Issue 3*, 1985.
- [16] Ali Razaghpoor, Zahra Taheri-Ezbarami, Fateme Jafaraghaee, Saman Maroufizadeh, and Atefeh Falakdami. The effect of serious game and problem-based learning on nursing students' knowledge and clinical decision-making skill regarding the application of transfusion medicine in pediatric nursing. *Journal of Pediatric Nursing*, 76:e1–e8, May 2024.
- [17] Johnmarshall Reeve and Edward Deci. Elements within the competitive situation that affect intrinsic motivation. *Personality and Social Psychology Bulletin*, 22:24–33., January 1996.
- [18] Jesús Salido. Plantilla LaTeX para Trabajo Fin de Estudios en Ingeniería Informática - UCLM (España). Zenodo, DOI: <https://doi.org/10.5281/zenodo.4561708>, 2019.
- [19] Juliana Beatriz Stover, Flavia Eugenia Bruno, Fabiana Edith Uriel, and Mercedes Fernández Liporace. Teoría de la Autodeterminación: una revisión teórica. *Perspectivas en Psicología: Revista de Psicología y Ciencias Afines*, 14(2):105–115, 2017.
- [20] Penelope Sweetser and Peta Wyeth. GameFlow: A Model for Evaluating Player Enjoyment in Games. *Computers in Entertainment*, 3:3, July 2005.
- [21] Job Talle. Cubic noise. https://jobtalle.com/cubic_noise.html.
- [22] Hakan Tuzun, Meryem Yilmaz Soylu, Turkan Karakus Yilmaz, Yavuz Inal, and Gonca Cumaoğlu. The effects of computer games on primary school students' achievement and motivation in geography learning. *Computers & Education*, 52:68–77, January 2009.
- [23] Robert Yang. Pacing | The Level Design Book. <https://book.leveldesignbook.com/process/preproduction/pacing>, March 2023.

ANEXOS

Anexo: Simulación de Partida

En este anexo se incluyen una serie de capturas de pantalla que muestran el proceso de jugar una partida.



Figure A.1: Pantalla de inicio de sesión y creación de cuenta.



Figure A.2: Pantalla del menú principal de la versión extrínseca.



Figure A.3: Pantalla del menú principal de la versión intrínseca.

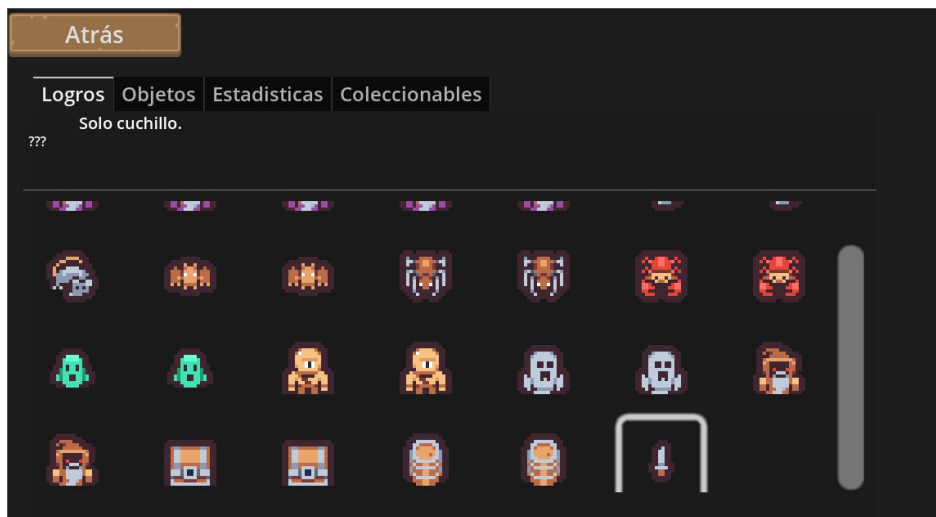


Figure A.4: Pantalla de los logros con todos ellos bloqueados.



Figure A.5: Pantalla con los diferentes ataques y una explicación de los mismos.



Figure A.6: Pantalla de las estadísticas del jugador.

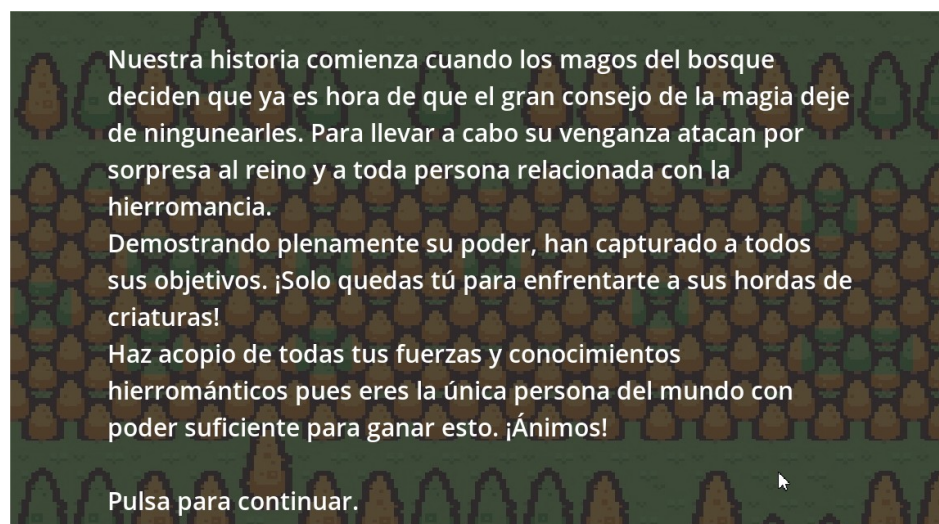


Figure A.7: Pantalla que aparece al inicio de las partidas y que cuenta la historia del mundo del videojuego.



Figure A.8: Visualización de uno de los escenarios aleatorios justo al comenzar la partida.



Figure A.9: El comienzo de una oleada de enemigos básicos rodeando a un jugador en apuros.



Figure A.10: Pantallazo que muestra varios tipos de enemigos del juego en gran cantidad.



Figure A.11: Pantalla en la que se ven las opciones de mejora al subir de nivel.

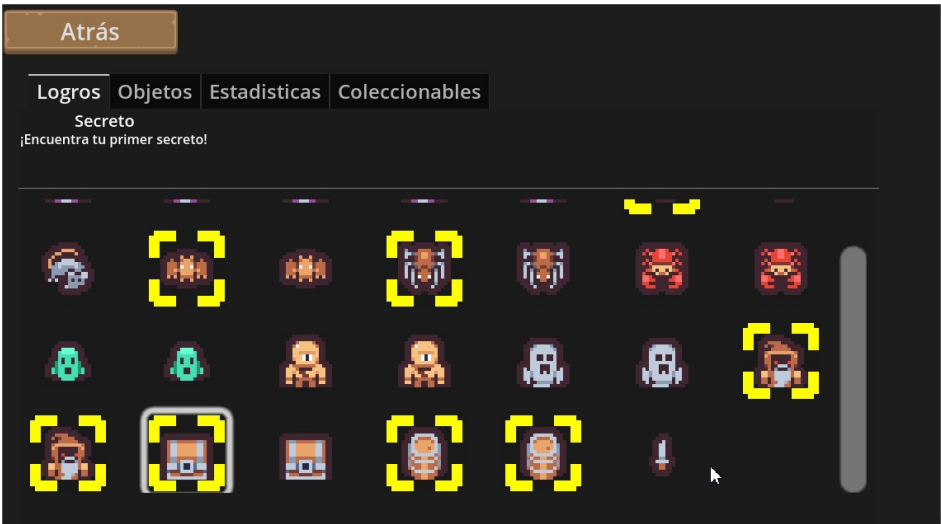


Figure A.12: Pantalla de los logros pero con varios de ellos ya desbloqueados.



Figure A.13: Pantalla del final de una partida con el daño causado y la puntuación obtenida.



Figure A.14: Pantalla que muestra la disposición del ranking del juego al terminar la fase de experimentación.

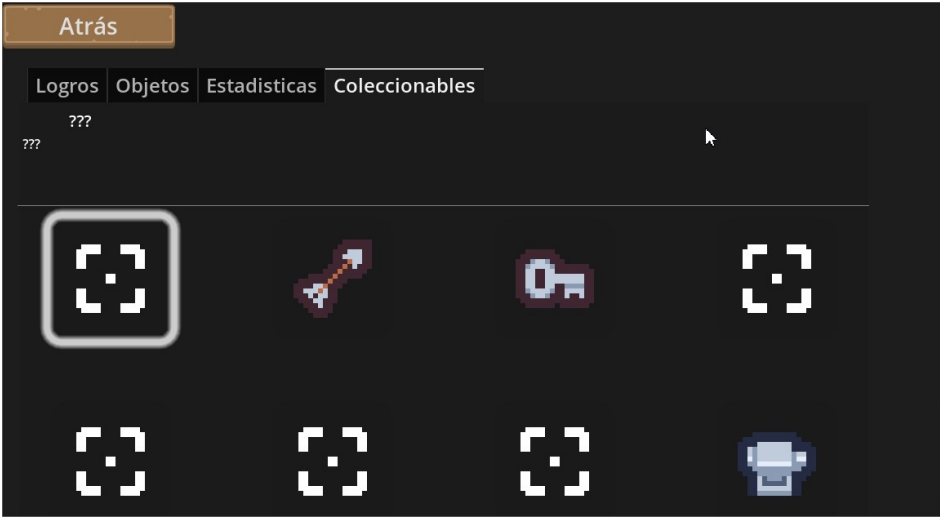


Figure A.15: Pantalla de los coleccionables con algunos desbloqueados.