



Universitat Oberta
de Catalunya

Inferno 's Court

Autor: Jorge Pulido Lozano

Tutor: Nwdd García Romero & Xabier Pallicera Malivern.

Profesor: Joan Arnedo Moreno

Grado en Ingeniería Informática

Área de Computación

27/02/2022



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

Copyright © 2022 Jorge Pulido Lozano.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

© Jorge Pulido lozano

Reservado todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante el alquiler y/o préstamo, sin la autorización escrita del autor o de los límites que autorice la ley de propiedad intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Inferno's Court</i>
Nombre del autor:	<i>Jorge Pulido Lozano</i>
Nombre del colaborador/a docente :	<i>Nwdd García Romero & Xabier Pallicera Malivern</i>
Nombre del PRA:	<i>Joan Arnedo Moreno</i>
Fecha de entrega (mm/aaaa):	<i>06/2022</i>
Titulación o programa:	<i>Ingeniería Informática</i>
Área del Trabajo Final:	<i>Videojuegos</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Aventura, Metroidvania, Unity, 2D, Search-action, plataformas</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo</i>	
<p>El objetivo de este Trabajo de fin de grado (TFG) es la creación de un videojuego con el motor grafico Unity 3D. Para ello haremos uso de las metodologías y técnicas adquiridas a lo largo del grado junto con herramientas de uso público para la elaboración de un software de manera aceptable para los estándares del ámbito donde se ubica.</p> <p>Inferno's Court es un juego 2D de desplazamiento lateral de aspecto retro de la generación de 16-32 bits encuadrado dentro del género search-action utilizando assest de licencia libre para la elaboración de un mundo donde los jugadores puedan divertirse</p> <p>Nuestro personaje, un caballero anónimo deberá embarcarse en una épica epopeya para librar a un poblado recóndito del asedio al que se enfrentan por la amenaza de las huestes provenientes del círculo más salvaje del infierno que intentarían por todos los medios impedirselo.</p> <p>El proyecto ha sido realizado sirviéndose del motor grafico Unity e implementado las clases, mecánicas y físicas en el entorno de programación Visual Studio en lenguaje C# y se han seguido todas las fases de desarrollo establecidas desde la planificación inicial con la idea hasta la obtención de un producto final.</p> <p>En conclusión, puedo determinar que el desarrollo dentro de unos plazos de tiempo tan ajustados ha supuesto un desafío, pero ver el resultado final corriendo es una grata satisfacción, divirtiéndome mucho en el proceso y que me permitirá abordar con otra perspectiva el análisis de los desarrolladores profesionales.</p> <p>Enlace repositorio: https://bitbucket.org/jorgepul/infernos-court/src/master/</p>	
Abstract (in English, 250 words or less):	
The objective of this Final Degree Project (TFG) is the creation of a video game with the Unity 3D graphics engine. For this we will make use of the methodologies and techniques	

acquired throughout the degree along with public use tools for the development of software in an acceptable manner for the standards of the field where it is located.

Inferno's Court is a 2D side-scrolling game of retro look of the 16-32 bits generation framed within the search-action genre using free license assest for the development of a world where players can have fun.

Our character, an anonymous knight must embark on an epic quest to free a remote village from the siege they face by the threat of the hosts from the wildest circle of hell that will try by all means to prevent it.

The project has been carried out using the Unity graphic engine and implemented the classes, mechanics and physics in the Visual Studio programming environment in C# language and all the established phases of development have been followed from the initial planning with the idea until obtaining a final product.

In conclusion, I can determine that the development within such tight deadlines has been a challenge, but seeing the final result running is a great satisfaction, having a lot of fun in the process and that will allow me to approach with another perspective the analysis of professional developers.

Repository link: <https://bitbucket.org/jorgepul/infernos-court/src/master/>

El coraje no necesita ser recordado, porque nunca es olvidado
Princesa Zelda
Breath of the Wild, 2017

La vida se trata de determinación. El resultado es secundario
Waka
Okami, 2006

Índice

1. Introducción.....	11
1.1. Contexto y justificación del trabajo	11
1.1.1. Motivación personal para el desarrollo del proyecto.....	11
1.1.2. Justificación del interés y la relevancia de la propuesta	12
1.2. Descripción/Definición	16
1.3. Objetivos generales	18
1.3.1. Objetivos principales	18
1.3.2. Objetivos secundarios	18
1.4. Metodología y proceso de trabajo.....	19
1.5. Planificación.....	21
1.6. Breve resumen de los productos obtenidos.....	24
1.7. Estructura del resto del documento	24
2. Estado del arte y Análisis de mercado	26
2.1. Género del juego	26
2.2. Estado del arte	28
2.2.1. Unity.....	29
2.3. Público objetivo (i.e. <i>target audience</i>) y perfiles de usuario.....	29
2.4. Análisis de mercado.....	30
2.5. Competencia/Antecedentes (o marco teórico/estado del arte)	31
3. Propuesta del Juego.....	35
3.1. Definición del producto	35
3.1.1. Descripción del juego.....	35
3.1.2. Historia, ambientación y/o trama.....	35
3.1.3. Definición de los personajes/elementos	36
3.1.4. Interacción entre los actores del juego	37
3.1.5. Objetivos planteados	37
3.1.6. Concepts Arts.....	38
3.1.7. Plataforma de destino	39
3.2. Definición de objetivos/especificaciones del producto	40

3.3. Modelo de negocio	40
3.4. Estrategia de marketing.....	40
4. Diseño.....	42
4.1. Entorno y requisitos.....	43
4.1.1. Entorno.....	43
4.1.2. Requisitos técnicos	44
4.2. Herramientas utilizadas	44
4.3. Recursos del videojuego.....	45
4.3.1. Assets.....	45
4.3.2. Imágenes	46
4.3.3. Sonidos	51
4.3.4. Fuentes	51
4.3.5. Listado de Assets	51
4.4. Camara del juego.....	55
4.5. Escenas	55
4.6. Scripts.....	56
4.7. Arquitectura del juego	57
4.8. Inteligencia artificial	57
5. Diseño de niveles.....	62
5.1. Diseño del mundo	62
5.1.1. El ultimo poblado	62
5.1.2. Bosque	62
5.1.3. Cementerio.....	63
5.1.4. Castillo.....	63
5.2. Diseño de combate.....	64
5.3. Instrucciones de instalación.....	64
6. Implementación y Demostración	65
6.1. Requisitos mínimos	65
6.2. Formas de iniciar el juego	65
6.3. Instrucciones de uso.....	65
6.3.1. Menús.....	65
6.3.2. Controles.....	66

6.4. Prototipos	66
6.4.1. Prototipos Lo-Fi.....	66
6.4.2. Prototipos Hi-Fi	67
6.5. Test	68
6.6. Ejemplos de uso del producto (o manual de usuario)	71
7. Conclusiones y líneas de futuro	72
7.1. Conclusiones	72
7.2. Líneas de futuro.....	75
Bibliografía	77
Anexos	82

Figuras y tablas

Índice de figuras

Figura 1: Salón recreativo	11
Figura 2: Diferentes perspectivas del modelo MDA según el rol.....	13
Figura 3: Final Fantasy IX (Square-Enix, 2000). Vivi buscaba dar sentido a su vida.	14
Figura 5: Metroid dread. Samus escapando de un EMMI.	¡Error! Marcador no definido.
Figura 6: BloodStained y sus diversas áreas. Capturas de Nintendo Switch, de mi partida personal.	16
Figura 7: Subclases de dragones en Bloodstained.....	17
Figura 8: Castlevania Mirror of Fate. Trevor Belmont en el cementerio del castillo	¡Error! Marcador no definido.
Figura 9: Soy Guybrush Threepwood y quiero ser pirata.....	¡Error! Marcador no definido.
Figura 10: Desglose de las tareas del proyecto	22
Figura 11: Diagrama de Gantt Inferno's Court (GanttPRO web)	23
Figura 12: El origen de todo. Samus explorando Zebes	¡Error! Marcador no definido.
Figura 13: Comparativa Castlevania III en SNES (Izq.) vs Mega Drive (Dcha.).....	27
Figura 14: SOTN, la mezcla perfecta de dos sagas.....	¡Error! Marcador no definido.
Figura 15: El penitente debe expiar la culpa para alcanzar la madre de madres....	¡Error! Marcador no definido.
Figura 16: Interface Unity.....	¡Error! Marcador no definido.
Figura 17: Cuadrícula DAFO. Obtenido de https://www.mastermarketing-valencia.com/ventas-y-gestion-comercial/blog/que-es-dafo/	32
Figura 18: Personaje principal del juego, el caballero.....	¡Error! Marcador no definido.
Figura 19: Enemigos	¡Error! Marcador no definido.
Figura 20: Ítems y coleccionables.....	¡Error! Marcador no definido.
Figura 21: La estatua de la diosa funciona como Check Point y Save Point	¡Error! Marcador no definido.
Figura 22: Estructura del reino de Idraldor.....	38
Figura 23: Pueblo, donde se inicia la aventura	¡Error! Marcador no definido.
Figura 24: Cementerio	¡Error! Marcador no definido.
Figura 25: Castillo de Armagrog.....	¡Error! Marcador no definido.
Figura 26: Detección de colisiones y elementos estáticos de la escena.....	39
Figura 27: Habilidad del doble salto.....	39
Figura 28: Elementos dañinos de la escena	39
Figura 29: IA de los enemigos.....	39
Figura 30: Origen del bestiario.....	39
Figura 31: Toques de RPG al juego.....	39
Figura 32: Salvado de estadísticas	39
Figura 33: Estructura de la carpeta Assets	46
Figura 34: Concept art de la portada.....	47
Figura 35: Concept art del logo de EldritchMonn Studios	47
Figura 36: logo final de EldritchMonn Studios.....	47
Figura 37: Tilemap del pueblo seccionado por componentes	48
Figura 38: Imagen de background	48

Figura 39: Sprites de mecánicas usadas por nuestro caballero.....	48
Figura 40: Enemigo B (Hombre lobo).....	¡Error! Marcador no definido.
Figura 41: Pociones	¡Error! Marcador no definido.
Figura 42: sistema monetario y su valor	¡Error! Marcador no definido.
Figura 43: diagrama de estados de Inferno's Court.....	57
Figura 44: Arquitectura de componentes del juego.....	57
Figura 45: Movimiento entre dos puntos fijos.....	¡Error! Marcador no definido.
Figura 46: Inspector del vigía y collider de detección del jugador	59
Figura 47: Enemigo a la espera del jugador dentro de su rango de movimientos ..	¡Error! Marcador no definido.
Figura 48: IA de los enemigos con animación de ataque.....	60
Figura 49: Calculo de la trayectoria en tiempo real hacia el jugador.....	¡Error! Marcador no definido.

Índice de tablas

Tabla 1: Comparativa Motores analizados	29
Tabla 2: Ranking de ventas de la saga Castlevania	31
Tabla 3: Especificaciones del dispositivo	44
Tabla 4: Requisitos máximos y recomendados del motor Unity.....	44
Tabla 5: Listado de recursos utilizados	55

1.Introducción

1.1. Contexto y justificación del trabajo

1.1.1. Motivación personal para el desarrollo del proyecto

La elección del área de videojuegos frente a otras supuso una elección no tan difícil dado que había otras áreas que tenían más relación con el ámbito laboral en el que me desarrollo. Trabajo como investigador de I+D, licenciado en ciencias biológicas, en una empresa de biotecnología y el grado de ingeniería informática me supuso adquirir conocimientos multidisciplinarios para poder en un futuro desarrollarme personalmente en el área de la bioinformática.

Visto mi trasfondo uno pensaría que por ejemplo el área de inteligencia artificial podría suponer un tema más que plausible para realizar un trabajo final de grado pero al haber realizado el master de bioinformática previo al grado ya realicé un trabajo final de master centrado en el uso de algoritmos de aprendizaje automático sobre una base de datos genéticos (HapMap). Por este motivo sentí que seguramente estaría “reproduciendo” el mismo trabajo pero con otros data set razón por la cual seleccione otra área que pudiera darme un desafío a la vez de poder realizarme haciendo algo que me gusta.

El desarrollo de este proyecto no viene de un afán puramente académico o laboral sino que proviene de una afición que ha sido cultivada a lo largo de los años desde que me regalaron en los albores de la década de los 90 una Nintendo NES (Nintendo 1983) con el juego super Mario Bros (Nintendo 1985).

Las aventuras de un pequeño fontanero que intenta rescatar a una princesa de una especie de tortuga con pinchos, visto desde mi infante perspectiva, no hizo más que fomentar una pasión que creció debido a dos principales factores.

El primero fue con las visitas a los salones arcades, lugar de reunión donde se te ponía a prueba factores tan importantes como la paciencia al tener a varias personas alrededor esperando que perdieras para poder jugar ellos y también a la economía pues las partidas requerían de dinero y con esa tierna edad no es que manejáramos mucho salvo la paga semanal de modo que había que economizar a que juego querías jugar, ya fuera el Street fighter 2 (Capcom 1991), Pacman (Nanco 1980), Bubble Bobble (Taito 1986)...

El otro factor fue la popularización que tuvieron los dispositivos domésticos con los lanzamientos de las consecuentes generaciones asociadas a los saltos tecnológicos que supusieron las máquinas de 16 bits, 32 bits... Durante estas épocas se habla de la famosa guerra de consolas que dividieron en la época de 16 bits entre los usuarios de SNES (Nintendo 1990) y Mega drive (Sega 1988) donde no se apreciaba los niveles de la toxicidad actual [1], donde la gente a pesar de catalogarse entre “nintenderos” o “segeros” era bastante habitual entre estos prestarse las consolas para jugar al catálogo de la competencia.



Figura 1: Salón recreativo

Al igual que con muchas aficiones que se cultivan suele surgir el interés por conocer más en profundidad aquello que te apasiona, de modo que empiezas por consumir otra serie de productos asociados al tema de interés.

A comienzos de la industria era muy difícil que los desarrolladores hicieran declaraciones del proceso de creación de videojuegos pues la única fuente de información que se tenía eran las revistas mensuales como Hobby Consolas o Nintendo Acción (editorial Axel Springer SE). Con la llegada de internet, el conocimiento se convirtió en algo accesible y los desarrolladores son más abiertos a la hora de hacer documentales sobre el desarrollo del juego o a creadores de contenido que hacen la tarea de recopilar y transmitir dicha información [2].

Y es en este punto cuando, al ver el final del grado en el horizonte y con los conocimientos adquiridos durante el proceso que uno se preguntaría ¿Podría ser capaz de hacer un juego yo

Considero que una vez desarrollado toda una trayectoria a un lado de la pantalla quemando las pilas/baterías o incluso desgastando los mandos, aún recuerdo esas partidas al Mario Party (Nintendo, 1998)[3], sea un paso de evolución natural el querer ponerse al otro lado y pasar de consumidor a editor.

Una analogía la podemos encontrar en los juegos de rol. Los que vimos el auge de juegos como D&D (Wizard of the Coast, 5^o edición 2014) [4] o de Vampiro la mascarada (NOSOLOROL, 5^o edición 2018) en los 90 donde lo más normal era que el papel de Dungeon master fuera rotando entre los miembros del grupo al terminar las sesiones de juego, de modo que pasabas de tirar los dados e intentar sobrevivir a una emboscada de orcos a poner un dragón para hacer la puñeta a tus amigos y reiros un rato.

1.1.2. Justificación del interés y la relevancia de la propuesta

La justificación de la realización de este juego se fundamenta en poner los conocimientos adquiridos durante el grado para desarrollar un ciclo completo de desarrollo de software, aplicado en este caso al ámbito de los videojuegos.

Para el desarrollo se seguirá el marco MDA, que significa Mechanics-mecánicas, Dinamycs-dinámicas y Aesthetics-estéticas) y que tiene su objetivo en que desean los usuarios tras jugar.

Dentro de las mecánicas encontraremos las reglas que queremos que se apliquen y que podrá hacer o no los elementos del juego, como tendrá toques de RPG se espera que nuestro personaje se fortalezca según va jugando.

En este apartado también tendría que considerarse la narrativa pues influye bastante dentro del aspecto final que tendrá el juego, si hablamos de animales del bosque saldrá un juego del tipo Donkey Kong Country pero si hablamos de un señor oscuro con ínfulas de dominación tendremos lo opuesto, Splatterhouse (Namco, 1988).

En el apartado de dinámicas tenemos una descripción más abstracta pues habrá que desarrollar mecánicas para fomentar la satisfacción del usuario como por ejemplo el *backtracking* con nuevas habilidades, todo teniendo en cuenta la pirámide de Maslow para el diseño del juego pero que nos distorsione las propuestas de aprendizaje y objetivos propuestos en este trabajo.

Por último, con la estética se espera que el jugador obtenga alguno de las siguientes sensaciones tras jugar al juego. Principalmente se espera que al terminar el juego el usuario obtenga una respuesta emocional de descubrimiento que le llevara a tener curiosidad por el género y conectar con otros jugadores que disfrutaran de este género.

Siguiendo estas premisas se espera desarrollar un software de calidad que permita al jugador pasar un buen rato y que disfrute de la historia que este humilde desarrollador quiere ofrecer. Esta es la premisa fundamental del modelo MDA, la interacción entre el jugador y el desarrollador [38].

Durante el desarrollo del juego debe tenerse en cuenta este modelo para garantizar su éxito, dado que el diseñador quiere crear su mundo pero deben poner una serie de reglas para que la experiencia del jugador sea debida a dichas reglas [39].

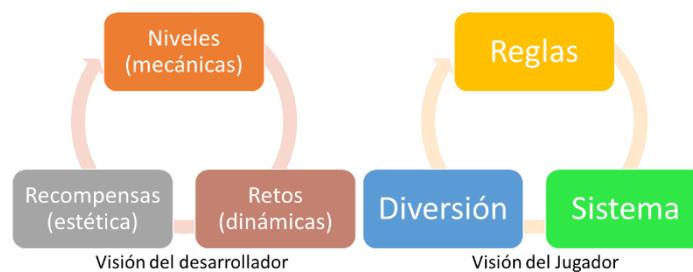


Figura 2: Diferentes perspectivas del modelo MDA según el rol

Debido a esto el desarrollo de videojuegos es referido un problema de diseño de segundo orden, donde se implementa mecánicas que llevan a la solución de un problema. Esto dificulta el desarrollo de videojuego pues no se trata de crear el juego perfecto sino que todo es un puzle que debe armarse con paciencia y tesón para que todo fluya pues el desarrollador solo tiene control sobre la parte de las mecánicas del modelo dejando las otras dos en manos del jugador.

Como se ha indicado en la introducción es importante que se inviertan los roles para poder comprender mejor cuanto de trabajo lleva el desarrollo de un juego para así poder disfrutar más de las posibilidades que el diseñador ha incluido en el juego, por medio de las mecánicas desarrolladas.

Sin embargo esto podría ser un arma de doble filo para los desarrolladores amateurs pues les da la impresión al desarrollar su propio juego de querer poner todas aquellas emociones que ha recibido de un juego.

Esto supondría que empezarían el desarrollo del juego desde las estéticas y progresarían hasta las mecánicas, lo cual es un error pues como se ha indicado en el modelo MDA las reglas definidas por las mecánicas serán las que guíen las otras dos partes y no al revés.

Una vez establecida la orientación que se seguirá durante todo el desarrollo del juego pasaremos a esgrimir las cuestiones que nos han llevado a elegir la estética, genero... que han dado forma a Inferno's Court.

El interés de este subgénero, metroidvania, en detrimento de otros como podrían ser las plataformas o los role playing game (RPG en adelante), se fundamenta en que los metroidvania parten de un concepto donde la mecánica de la acción y la ambientación tiene más peso que otros factores como la narrativa.

Por ejemplo, en el caso principal de los máximos exponentes del género como son Metroid (Nintendo, 1986) o Castlevania - symphony of the night (Konami, 1997), en adelante SOTN, donde el personaje parte del punto A y tiene que llegar al punto B librándose de muchos enemigos en el camino.

Mientras que en juegos de la saga Final Fantasy (Square-Enix) o la saga Tales of (Bandai-Namco) el desarrollo de la historia supone una trama más elaborada para dar sentido a la creación del personaje. Poner el foco sobre el personaje se debe a que en parte se realizaran mecánicas de misiones (principales o secundarias) que darán un mayor trasfondo a la sique del personaje haciendo que el jugador empatice más o menos con estos.

Por ejemplo recuerdo que en Final Fantasy IX, personajes como Quina o Steiner no me aportaban nada pues eran los típicos arquetipos (maga azul y soldado respectivamente) pero Vivi era diferente, empezaba como el típico mago negro pero a medida que pasaba la aventura adquiría, sin entrar en *spoilers*, una evolución que los otros no llegaban a tener, a excepción de Yitan y Garnet.



Figura 3: Final Fantasy IX (Square-Enix, 2000). Vivi buscaba dar sentido a su vida.

Aunque es cierto que últimamente los metroidvania están empezando a tener una carga de narrativa mayor, donde cada objeto (clave o no), enemigos, magias... tienen una descripción que amplía el lore del juego base y proporcionan una mayor inmersión al jugador. Adquiriendo más mecánicas más próximas a los juegos tipo de aventuras por medio de la adición de cinemáticas.

Un caso podría ser el último juego de la franquicia Metroid, Metroid Dread (Nintendo, 2021) o Blasphemous (The Game Kitchen, 2019), este último recoge mecánicas de narrativa de los juegos de la saga Souls, Mauricio García productor de Blasphemous señala que además de las mecánicas para matar enemigos hay otras mecánicas narrativas opcionales para pararse a admirar el escenario o PNJ (personajes no jugadores) que darán información referente al mundo donde transcurre la acción [5].

Eliminando mecánicas opcionales te permite centrarte en un proceso de desarrollo de software más corto al tener un número limitado de recursos ya sea a nivel personal, temporal... Sin embargo, te permitiría pasar por cada una de las fases de desarrollo de un videojuego más rápidamente [6].

Aunque la elección del subgénero a desarrollar podría considerarse como parte de la motivación personal, se incluye en este apartado pues enlaza con lo expuesto dado que este

tipo de géneros en un nivel de profundidad mayor supone que el desarrollo de niveles implique una gran carga de *backtracking* dado que se piensan para que el jugador tenga que volver a un punto dado una vez haya conseguido la habilidad necesaria.

Esto supone un añadido que los fans del género agradecen pues les obligan a tener que recorrerse cada palmo del mapa descubriendo todos sus secretos si quieren desbloquear el 100%.

Otro motivo de interés por querer hacer un metroidvania viene de la demanda de juegos de este género que se está revitalizando actualmente ya que desde la salida de Castlevania order of Ecclesia (Konami, 2008) para Nintendo DS hasta que llegó Hollow Knight (Team Cherry, 2017) para pc originalmente. Desde entonces y gracias desarrolladores indies se está recuperando el género. A modo de ejemplo tenemos la vuelta de Koji Igarashi, padre del SOTN, que recuperó la esencia gótica con Bloodstained (ArtPlay, 2018)[7] o que la propia Nintendo lanzase Metroid 19 años después del último juego de la saga publicado.



Figura 4: Metroid dread. Samus escapando de un EMMI.

Como queda demostrado la elección de tema es relevante dado los tiempos que corren y la dirección que las grandes compañías están tomando en sus desarrollos. También, por lo descrito es más que factible que sea más entretenido jugar a este género frente a otros más pausados como los RGP o un simple plataformero.

La relevancia de la propuesta consiste en aplicar los conocimientos en la programación orientada a objetos para el desarrollo de un juego de estas características teniendo en consideración el paradigma de la programación orientada a objetos (en adelante POO) para el desarrollo de las diferentes clases que van a intervenir en nuestro juego como podrán ser el jugador, los enemigos, los niveles...las cuales tendrán que realizarse las correspondientes relaciones para que por ejemplo el suelo permita que nuestros personajes puedan moverse sobre estas o gestionar a los PNJ (enemigos principalmente) por medio de una inteligencia artificial básica y que respondan en tiempo real a acciones del jugador, por ejemplo perseguirle por la pantalla o atacarle en cuanto pueda.

Todas las situaciones que dependan de la reacción del jugador, el cual como humano podrá ser impredecible, obliga al programador a desarrollar métodos que permitan adaptarse a las situaciones que pueda encontrarse.

Es por ello que el tema seleccionado puede ser desafiante por todo el trabajo inherente pero que también a la vez de entretenido y cuyo producto final podría ser una lección importante en áreas desarrolladas durante la carrera así como las competencias adquiridas en POO o Proyecto de desarrollo de software por citar algunas.

Además, permitirá adquirir conocimientos que podrían considerarse de postgrado al tener que utilizar un motor de juego y las diferentes particularidades que este proporciona para que el juego pueda desarrollarse correctamente [8].

1.2. Descripción/Definición

El propósito de este trabajo consiste en la elaboración de un videojuego 2D que se englobaría dentro del subgénero de los search-actions o “metroidvania” como se conocen comúnmente. El juego seguirá las convenciones más puristas del género que fueron introducidas en SOTN.

Dentro del juego se desarrollarán mecánicas que serán descritas a continuación junto con lo que se espera que el jugador perciba para que la impresión de este sea positiva y que por ende lleve al éxito del juego. La metodología usada para el diseño del ha sido determinada por el modelo MDA.

Estas convenciones consistirían en el uso de las plataformas combinado con dosis de acción, ambientado en un mundo con secciones de juego claramente definidas donde habrá enemigos acordes en su mayor medida a la sección por donde el jugador pasara.

Las secciones del juego son parte de la mecánica donde se situará la acción cada área en este tipo de género conlleva la introducción de contenido diferente como podrían ser enemigos o habilidades.

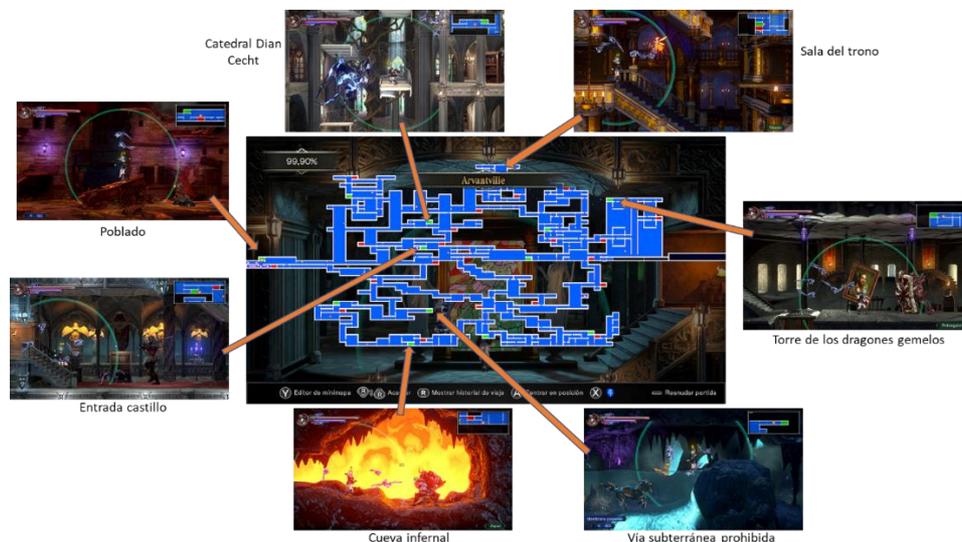


Figura 5: BloodStained y sus diversas áreas. Capturas de Nintendo Switch, de mi partida personal.

Por ejemplo se espera implementar un área que transcurra en un cementerio de modo que lo enemigos que lo pueblen serán más del tipo no muertos (zombies, esqueletos, fantasmas...). Esto supone que se incluye la mecánica de imponer un reto al jugador para poner a prueba sus habilidades y donde podríamos encontrar el uso de misiones principales/secundarias las cuales ofrecen recompensas. Todo esto conllevaría a que el jugador decida implicarse más o menos dentro de la aventura pues da una sensación de progreso al “sentirse útil”.

En este proyecto para abreviar el uso del bestiario que compondrán los diversos enemigos se planteara una aproximación de reciclaje de modelados para dar la sensación de una mayor diversidad de enemigos. En juegos de este calibre las diferencias entre las diferentes

subespecies del mismo tipo se basan en características cosméticas y en un cambio de parámetros mientras que las animaciones y patrones de ataque son las mismas.



Figura 6: Subclases de dragones en Bloodstained

Tampoco podremos olvidarnos de otros aspectos de la saga, como podrá ser la adquisición de experiencia para fortalecer a nuestro jugador, el uso de diferentes sets de armas y armaduras que dotaran al juego de un perfil más típico de los RPG y la lucha frente a enemigos finales que salvaguardaran el pase a otra área o que guardan habilidades necesarias para el acceso a áreas imposibles de acceder hasta ese punto de la historia.

Usando un motor de juego predefinido, en lugar de la elaboración de uno propio al igual que el uso de unos assets de código libre, se pretende acelerar la producción del mismo para centrarse en tareas de diseño de niveles y la interacción de los diferentes personajes entre sí y respecto al entorno en el que se encuentran.

Es por tanto que desde mi punto de vista debe cubrirse un nicho de mercado que en los últimos años pocos juegos han realizado, dado que muchos están siendo muy influenciados por las mecánicas jugables vistas en la saga souls, actualmente de más moda por la salida de Elden Ring (FromSoftware, 2022).

No es del ámbito de este trabajo determinar que la implementación de mecánicas de otros géneros pueda desvirtuar al propio género sino que más bien lo contrario, aunar dos géneros en uno y que el producto final sea de calidad y que sea del agrado del público es difícil y gratificante a la vez.

Es por ello que en ocasiones habría que dar un paso atrás y volver a las raíces del género para poder reformularlo y que siga siendo atractivo para el público actual. Si nos remontamos en la industria los últimos juegos metroidvania puristas han sido Bloodstained, Ori and the wild of the Wisps (Moon Studios, 2020) y Metroid dread. Esto supone tres juegos en casi 5 años pues títulos como Blasphemous y Hollow Knight tienen mecánicas souls que penalizan la muerte del avatar de alguna manera.

El tema del desarrollo de un juego de este género es relevante pues la comunidad no ha perdido el apetito por este género, sino que muchos de ellos se lanzan a desarrollar sus propios juegos para cubrir este nicho. La propia Konami no está trabajando actualmente en la saga Castlevania, ni siquiera por su 35 aniversario, sino lo que es peor y ha generado mucha polémica por el uso de esta licencia para la venta de NTF [9][10].



Figura 7: Castlevania Mirror of Fate. Trevor Belmont en el cementerio del castillo

Con los conocimientos adquiridos durante el desarrollo del juego se pretende aplicar el uso de las mecánicas que definen el género además de generar alguna propia que pueda surgir durante la fase de desarrollo en el tiempo impuesto. Por ejemplo se espera desarrollar una mecánica que suponga un tipo de dificultad añadido, en la forma de un monstruo, y que sea para el jugador una molestia de la que tenga que estar pendiente constantemente y que favorezca así una mayor inmersión del jugador.

Al final del proyecto se espera tener un juego implementado donde el personaje vaya desplazándose por el entorno interactuando con este y con los enemigos que lo pueblan.

1.3. Objetivos generales

Los objetivos marcados en este proyecto estarán relacionados con la creación de un videojuego desde cero. Los conocimientos adquiridos durante la carrera en las diversas asignaturas, ya sean del ámbito de la computación o del desarrollo de software, para alcanzar la fase Gold y liberar el juego para el disfrute de la comunidad.

A continuación se detalla un listado de los objetivos ordenados por relevancia.

1.3.1. Objetivos principales

- Definición del videojuego: temática, estética...
 - Crear un diseño propio y original.
- Elección de la plataforma y del lenguaje.
 - Aprender el uso de Unity Engine para el desarrollo de videojuegos.
- Desarrollo de una versión jugable, codificación y testing.
- Establecer la viabilidad del proyecto al inicio y tras prueba de testing.
- Desarrollar plan de actualización o de desarrollo futuro.
- Medir el grado de entretenimiento en el target objetivo del juego a través de una encuesta.
- Gestionar todos los aspectos del desarrollo de un software.

1.3.2. Objetivos secundarios

- Diseño de escenarios y personajes.
 - 4 niveles interconectados.
 - Enemigos con diferentes mecánicas por medio de una IA básica.
- Implementación de mecánicas y coleccionables.
 - Mecánicas de RPG.
 - Sistema de inventariado
 - Sistema loot...
- Divertirse en el proceso de creación.
- Ampliar competencias, y experiencia en el ámbito de la computación, en este caso aplicada a videojuegos.
- Conocer los diferentes motores actuales del mercado.
- Obtener una nueva perspectiva del proceso de creación de videojuegos.

- Versión WebGL para poder ser jugado en navegadores.
- Elaboración de material adicional a la memoria, para acompañar la salida de juego: Manual de usuario, videos de desarrollo...

1.4. Metodología y proceso de trabajo

Como se ha indicado durante la introducción de este trabajo, la importancia de ponerse en la piel de un desarrollador en lugar del habitual papel de crítico es uno de los objetivos planteados y realizar los mismos procesos que serán desarrollados de manera incremental. Los procesos por los que pasaremos a lo largo de este trabajo serán:

- Análisis: planificación, análisis y especificación de los requisitos.
- Diseño: diseño y especificación del sistema.
- Implementación: programación y pruebas unitarias.
- Verificación: integración de sistemas, pruebas de sistema y de integración.
- Mantenimiento: entrega, mantenimiento y mejora.

Esta metodología de trabajo sigue el modelo lineal de desarrollo en cascada [11] donde cada fase se hará de manera secuencial y ordenada de la primera a la última, donde no podremos pasar a la siguiente etapa sin cerrar previamente la que actualmente está activa tras comprobar los hitos y entregas asociados a la misma. Todo esto nos permitirá determinar qué elementos deben quitarse/añadirse/modificarse en la siguiente etapa.

La idea principal de este trabajo es desarrollar un producto nuevo a partir de un producto existente mediante el uso de assets de distribución libre, los cuales permitirán agilizar la etapa de diseño y centrarnos directamente a la implementación del juego estableciendo las mecánicas de los diferentes componentes (avatar del usuario, la escena...) que compondrán el juego.

Se ha decidido que esta es la aproximación correcta visto el tiempo limitado para el desarrollo a la par de la limitación de personal y equipamiento para ello. Dado que la fase de diseño en términos de preproducción que incluiría el dibujo de bocetos, renders preliminares, animaciones... requiere de mucho tiempo y unas habilidades que no se poseen.

El uso de un motor gráfico nos permitirá la aplicación de los assets para dar vida a cada uno de los componentes del juego. Unity, motor elegido, permitirá por el uso de scripts gestionar toda la lógica del juego centrando la parte gráfica al propio motor.

Esta aproximación mixta entre la programación de novo vista durante el grado o el uso de un sistema cerrado que realiza todos los procesos del juego por medio de un point-n-click donde en este último únicamente hay arrastrar a la escena los diferentes elementos y que ya incluyen la parte lógica. Esto último se simulará por medio de los prefabs de Unity, lo cual permitirá una vez codificada toda una clase poder gestionar la animación y la lógica sin tener que volver a relacionar scripts con sprites.

Una analogía de los sistemas point-n-click sería lo visto en las aventuras gráficas como la mítica Monkey Island (LucasArts, 1990) o Age of Empire (Ensemble Studios, 1999) donde en este



Figura 8: Soy Guybrush Threepwood y quiero ser pirata.

último las construcciones se basaban en seleccionar el elemento y arrastrarlo a la posición deseada y el motor del juego ya se encargaría del resto.

Así, con el sistema mixto se nos permitirá tener un mayor control del flujo de trabajo y además poder tener más libertad para añadir nuevas mecánicas a lo largo del tiempo establecido.

Para el desarrollo del mismo se ha determinado que la mejor herramienta para conseguir los objetivos propuestos dentro del marco temporal establecido es Unity Engine. Previa elección del motor se realizó una evaluación de los diferentes motores actuales en el mercado (ver capítulo 2.2).

Al querer desarrollar un juego en 2D de estética retro, se tiene que tener en consideración que no será necesario un motor gráfico que requiera de muchos recursos a la hora de implementar el proyecto.

Unity al permitir el desarrollo de juegos en 2D y/o 3D podemos determinar que se ajusta a lo establecido y que además presenta una tienda donde poder adquirir recursos de forma gratuita para ser importados en el proyecto, lo cual ahorra tiempo de búsqueda y da al usuario la confianza de la procedencia de los mismos.

La curva de aprendizaje es de vital importancia para aquellos que no estén familiarizados con el desarrollo de videojuegos, como en mi caso, pasar de *gamer* a desarrollador supone que tengo que tener en consideración cada uno de los elementos que conforman la escena donde el jugador pasara el tiempo.

Unity presenta una curva de aprendizaje intuitiva que se basa en el desarrollo de scripts, los cuales van asociándose a los componentes (activos o pasivos) del juego, de modo que una vez establecido pueden ajustarse los parámetros en tiempo de compilación sin tener que estar recurriendo al lenguaje de programación.

También, decir que parte de la tarea de compilación recae en el motor por medio de carpetas autogeneradas que quitan trabajo de programación al usuario. Entre estos archivos se incluyen por ejemplo las mecánicas de físicas que son de vital importancia para dar una sensación de inmersión.

El proyecto desarrollado puede ser implementado en una plataforma, en este caso un ordenador y que será jugado por medio del teclado/ratón y que posteriormente podrá ser lanzado en diferentes plataformas como Android, iOS/Mac o de consola.

Esto debe tenerse en cuenta pues habrá dispositivos que según los gráficos no puedan ser ejecutados o que la tarea de implementación en un sistema sea más costosa que el propio desarrollo. Esto último es así pues ha habido muchos ejemplos que se justifican por ejemplo en que no pueden lanzarse juegos en Nintendo Switch por carecer esta de potencia gráfica pero que posteriormente se pueda disfrutar de *The Witcher 3* (CD projekt) en modo portátil. Este logro, entre otros similares, se consiguió sacrificando únicamente la distancia de dibujo y a 30 fps estables es digno de elogio y un llamamiento a aquellos que optan por no querer currárselo.

La portabilidad en último punto será un tema a desarrollar en las últimas fases del proyecto y Unity permitirá que sea posible pues nuestro desarrollo podrá ser implementado en estos dispositivos al haber otros referentes de la misma temática.

Unity nos permitirá un reto a la hora de crear los diferentes comportamientos para los enemigos que se interpondrán frente a nuestro jugador. Cada enemigo no tendrá que comportarse de la misma manera y por tanto el código que tendrá que usar variará.

Por ejemplo, un enemigo estático no tendrá que tener funciones que determinen su movimiento, lo que podría llevar a fallos de compilación, pero si heredaran características comunes de la clase padre.

Del mismo modo habrá que desarrollar diferentes elementos y abordar el proyecto bajo el prisma del paradigma de la programación orientada a objetos para tener un código lo más limpio posible. Esto se conseguirá con una estructura de carpetas.

Puede decirse que Unity 3D es una herramienta sólida para principiantes y también para usuarios que no tienen muchos recursos en términos de arquitectura para llevar a cabo el proyecto de sus sueños.

Por último se prevé que con la versión Alpha del juego se desarrolle un proceso de entrevistas para mejorar la versión gold del mismo, de manera análoga a lo realizado con los juegos como Octopath traveler (Square-Enix, 2018) [12] o recientemente con Triangle Strategy (Square-Enix, 2022) [13] para que sean los propios usuarios los que valoren el producto y permitan al equipo mejorar los puntos fuertes de cara al lanzamiento.

1.5. Planificación

Para la planificación del proyecto se ha tenido en consideración las entregas orientadas dentro del aula virtual del TFG establecidas por los hitos alcanzables al final de cada entrega.

- Prototipo (PEC 2)
- Versión Alpha (PEC 3)
- Versión gold (PEC 4)

Cada una de las fases que aquí se enumeran constituyen un sprint dentro del proyecto con las tareas más importantes asociadas a esa iteración. Los últimos días de cada iteración quedan reservados para la redacción y revisión de los materiales a entregar.

Inferno's Court								
Color	Número de EDT	Nombre de tarea / Título	Fecha de inicio planificada	Fecha de finalización planificada	Duración (horas)	Tipo	Nivel	
	1	Pec 1 - Plan de proyecto	16/02/2022	27/02/2022	96	proyecto	1	
	1.1	Idea del juego	16/02/2022	17/02/2022	16	tarea	2	
	1.2	Conceptualización	18/02/2022	18/02/2022	8	tarea	2	
	1.3	Selección del engine	19/02/2022	19/02/2022	8	tarea	2	
	1.4	Roadmap	20/02/2022	22/02/2022	24	tarea	2	
	1.5	Preparación documentación y material	23/02/2022	26/02/2022	24,08	tarea	2	
	1.6	Entrega	26/02/2022	27/02/2022	16	tarea	2	
	2	Pec 2 - Estado del arte y primera versión del proyecto	28/02/2022	03/04/2022	280	proyecto	1	
	2.1	Crear cuenta GitLab	28/02/2022	01/03/2022	16	tarea	2	
	2.2	Descarga de Unity3D y primeros pasos	28/02/2022	02/03/2022	24	tarea	2	
	2.3	Busqueda de assets	02/03/2022	03/04/2022	264	tarea	2	
	2.4	Prototipo	03/03/2022	13/03/2022	80,02	grupo de tareas	2	
	2.4.1	Creación de escenario	03/03/2022	06/03/2022	32	tarea	3	
	2.4.2	Creación de personaje	05/03/2022	08/03/2022	32	grupo de tareas	3	
	2.4.2.1	Movimientos	05/03/2022	06/03/2022	16	tarea	4	
	2.4.2.2	Ataque	07/03/2022	08/03/2022	16	tarea	4	
	2.4.3	Creación de NPCs	07/03/2022	11/03/2022	40	grupo de tareas	3	
	2.4.3.1	Movimientos	07/03/2022	10/03/2022	24,02	tarea	4	
	2.4.3.2	Ataque	09/03/2022	11/03/2022	24	tarea	4	
	2.4.4	Entrega Prototipo	12/03/2022	13/03/2022	8,02	tarea	3	
	2.5	Feedback y Marketing	21/03/2022	25/03/2022	40	grupo de tareas	2	
	2.5.1	Abrir cuenta	21/03/2022	24/03/2022	24,08	tarea	3	
	2.5.2	Subir prototipo	24/03/2022	25/03/2022	16	tarea	3	
	2.6	Revisión y mejoras	13/03/2022	29/03/2022	136	grupo de tareas	2	
	2.6.1	Desarrollo UI	13/03/2022	22/03/2022	80	tarea	3	
	2.6.2	Desarrollo de objetos	20/03/2022	29/03/2022	80	tarea	3	
	2.7	Preparación documentación y materiales	30/03/2022	02/04/2022	32	tarea	2	
	2.8	Entrega	02/04/2022	03/04/2022	16	tarea	2	
	3	Pec 3 - Implementación versión jugable	04/04/2022	08/05/2022	280	proyecto	1	
	3.1	Desarrollo escenarios y enemigos	04/04/2022	01/05/2022	224	tarea	2	
	3.2	Control de cámara	10/04/2022	12/04/2022	24	tarea	2	
	3.3	Menus	13/04/2022	15/04/2022	24	tarea	2	
	3.4	Salvado de partida	15/04/2022	17/04/2022	24	tarea	2	
	3.5	Sistema de experiencia	18/04/2022	21/04/2022	32	tarea	2	
	3.6	Música y ambientación	21/04/2022	24/04/2022	32	tarea	2	
	3.7	Revisión y mejoras	25/04/2022	01/05/2022	56	tarea	2	
	3.8	Actualización repositorio	02/05/2022	03/05/2022	16	tarea	2	
	3.9	Preparación documentación y materiales	04/05/2022	07/05/2022	32	tarea	2	
	3.10	Entrega	07/05/2022	08/05/2022	16	tarea	2	
	4	Pec 4 - Memoria y productos finales	09/05/2022	05/06/2022	216,02	proyecto	1	
	4.1	Ultimas mejoras	09/05/2022	15/05/2022	56	tarea	2	
	4.2	Trailer	12/05/2022	14/05/2022	24	tarea	2	
	4.3	Generación versión Gold	13/05/2022	15/05/2022	24	tarea	2	
	4.4	Preparación memoria TFG	15/05/2022	22/05/2022	64	tarea	2	
	4.5	Preparación defensa	23/05/2022	31/05/2022	72	tarea	2	
	4.6	Actualización repositorio	31/05/2022	01/06/2022	16	tarea	2	
	4.7	Informe autoevaluación	02/06/2022	03/06/2022	16	tarea	2	
	4.8	Preparación documentación y materiales	02/06/2022	04/06/2022	24	tarea	2	
	4.9	Entrega	04/06/2022	05/06/2022	8,02	tarea	2	
	5	Pec 5 - Defensa virtual	15/06/2022	22/06/2022	64	proyecto	1	
	5.1	Defensa	15/06/2022	22/06/2022	64	tarea	2	

Figura 9: Desglose de las tareas del proyecto

1.6. Breve resumen de los productos obtenidos

A lo largo de las diferentes entregas que se irán liberando durante el transcurso del proyecto sin incluir la memoria que aquí se suscribe en orden cronológico.

- Prototipo Inferno's Court (PEC 2)
- Video making of de Inferno's Court (PEC 2 y PEC 3)
- Juego Inferno's Court (PEC 4)
- Tráiler de Inferno's Court (PEC 4)
- Defensa Inferno's Court (PEC 4)
- Repositorio del proyecto (PEC 4)
- Informe de autoevaluación (PEC 4)

1.7. Estructura del resto del documento

Estado del arte y Análisis de mercado

En este capítulo visualizaremos el género Metroidvania a lo largo de la historia y su lugar dentro del mercado de los videojuegos ya sea por medio del target al que van dirigidos y un análisis de mercado donde compararemos nuestro producto con otros de la misma índole. También se expondrán las diferentes tecnologías de creación de videojuegos realizando un desglose de los principales motores del mercado.

Propuesta del juego

En este capítulo definiremos el juego, comenzando por la conceptualización y siguiendo por la influencia de otros títulos sobre el mismo. También describiremos los elementos principales que darán valor al proyecto por medio del guion del mismo

Diseño

En este capítulo se explicarán en mayor detalle todos los componentes que componen el proyecto. Se detallará todos los productos (propios o ajenos) que han sido utilizados y que dan forma al producto final. Nos centraremos en el lenguaje y en el motor grafico usado para crear los diferentes scripts para la implementación de las animaciones, prefabs, etc....

Diseño de niveles

En este capítulo se explicarán en mayor detalle todos lo relacionado con la escena donde transcurrirá el juego así como las decisiones tomadas para combinar el sistema de combate en el entorno del juego.

Implementación y Demostración

En este capítulo se describirán los pasos necesarios para poder ejecutar la versión gold del proyecto, indicando los requerimientos mínimos para poder ejecutarlo.

También, en este capítulo se describirán los diferentes prototipos liberados durante todo el proceso. También, se describirán los bocetos llevados a cabo para dar vida al juego y la prueba de usuarios realizada cuyos resultados se incluirán y se analizarán en los anexos. Por último, se detallará el manual de juego, el cual tendrá los comandos para poder jugar.

Conclusión y líneas de futuro

En este capítulo se valorará la trayectoria del roadmap propuesto y el grado de cumplimiento del mismo, critica objetiva del resultado obtenido y trataremos las mejoras a futuro después de terminar el proyecto para que el producto final sea más atractivo además

de revisar y mejorar los puntos que hayan sido detectados durante la fase de test con los usuarios.

Bibliografía

En este capítulo se incluyen los recursos bibliográficos utilizados para el desarrollo de este trabajo.

Anexos

En este capítulo se incluyen los elementos de ayuda para el desarrollo de este trabajo ya bien sean los glosarios, manual de usuario, resultados de la prueba con los stakeholders, enlaces de interés en relación al proyecto...

Se incluirán un archivo complementario a los anexos aquí presentados.

2.Estado del arte y Análisis de mercado

2.1. Género del juego

En punto 1 de la memoria se ha introducido el modelo MDA de como hay que entender el desarrollo de los videojuegos para alcanzar una coherencia sistemática cuando se alcanza un equilibrio entre todas las partes del juego.

Pensar en los videojuegos como artefactos, encuadrados dentro de un género, ayuda a que se construya el comportamiento de los elementos que interaccionaran, permitiendo así un diseño más claro.

Está claro que el fin último (estética) de un juego es la diversión, aunque admite múltiples metas, pero queremos huir de una palabra tan genérica como diversión y por ello en este juego a desarrollar como fin último usaremos la palabra juego como fantasía pues ofrecerá un ambiente épico donde los caballeros se enfrentaran al mal para salvar el reino.

En este apartado describiremos las mecánicas que se han ido creando y asentando para dar forma a lo que actualmente conforman el género que actualmente está en la cresta de la ola otra vez, por ejemplo Death's Gambit: Afterlife ofrece la mecánica nueva de los roles tipo visto también en Elden Ring y que permiten un juego diferente, en términos de mecánicas, con cada clase [40].

Inferno's Court se basa en el género de plataformas 2D con toques de acción y rol que comúnmente se denomina Metroidvania por las mecánicas que ofrece. Este género nació a mediados de los 80 [14] con tibios inicios hasta la salida de Metroid (Nintendo, 1986) la cual sentó las bases sobre las que gira el género.

Hasta este momento la mayoría de juegos de plataformas en consola domestica de esta época consistían en adaptaciones de las maquinas arcade como podría ser Pacman o aquellos en que el personaje fuera del punto A al punto B evitando obstáculos por medio de plataformas cuyo máximo exponente fue Super Mario Bros (Nintendo, 1985). Esto demuestra que Nintendo ha sido una de las grandes pioneras dentro del sector, tanto en términos de software y hardware, y modelo de inspiración para las desarrolladoras actuales [15].

Gracias a la popularización que tuvo la Nintendo NES en el mercado doméstico permitió el desarrollo de la saga Metroid a la par de la saga Castlevania, del que se crearía el ya famoso subgénero search-action que nos ocupa en este proyecto.

En estas sagas a diferencia de la del fontanero, aun compartiendo el desplazamiento en dos direcciones (en adelante scroll 2D) en estos se permitía volver sobre los pasos creando un desarrollo no tan lineal , invitando más al jugador a explorar las pantallas según se adquirían habilidades. Otro de las principales diferencias respecto al género de plataformero es que las tramas solían estar más elaboradas.

Sin embargo las fórmulas de antaño deben redefinirse para que el género pueda seguir siendo atractivo a la nueva generación de jugadores que surgían por la década de los 90, amén de seguir manteniendo a los iniciales. El salto a los 16 bits no supuso un gran cambio



Figura 11: El origen de todo. Samus explorando Zebes.

respecto a las mecánicas ofrecidas en la generación de 8 bits salvo por el aumento gráfico que supuso contar los dispositivos SNES y Mega Drive.



Figura 12: Comparativa Castlevania III en SNES (Izq.) vs Mega Drive (Dcha.)

Sin embargo, el género no terminó de consolidarse hasta la salida de Castlevania SOTN [16], donde su productor Koji Igarashi realizó una deconstrucción de los cánones de la saga para ofrecer una nueva perspectiva.

Sirviéndose de inspiración en el propio Metroid y en la saga de The Legend of Zelda, el desarrollo de niveles dejó de ser lineal para convertirse en mosaico de pequeñas pantallas conectadas entre sí de modo que el jugador puede perderse intentando encontrar el camino establecido para progresar en la historia. La exploración se ve fomentada por la adquisición de ítems o habilidades que permitan acceder a otras áreas del castillo permitiendo la reexploración.



Figura 13: SOTN, la mezcla perfecta de dos sagas

Otro punto introducido en la saga es la experiencia que nuestro jugador adquiere consiguiendo que se modifiquen las estadísticas del personaje, parámetros que incluso se pueden modificar por habilidades pasivas a lo largo del juego. También, se incluyó la personalización del equipamiento permitiendo equipar diferentes tipos de armas (lanzas, espadas, hachas...) desligándose del clásico látigo *vampire Killer* [17]. El arma equipada influirá en el tipo de ataque pues armas pesadas otorgan un gran poder de ataque pero son lentas y podría dejarte expuesto a un ataque mientras que armas ligeras no hacen tanto daño.

Irónicamente en los juegos del género que optan por parecerse a la saga souls dejan de lado la personalización RPG, personalización incluida en la saga original, teniendo nuestro héroe un solo arma que tendrá que mejorar por medio de artefactos o pagando un tributo, el cual también te servirá para la mejora de las habilidades. Esto fuerza al jugador a perfeccionar su estilo de juego en lugar de atacar a lo bruto con el arma de mejores estadísticas que poseyeras.

Por tanto según lo descrito anteriormente podemos describir una serie de mecánicas que todo metroidvania debe seguir para ser considerado como tal [41-44].

- Género de acción-aventura con secciones de plataformas.
- Gran mapa dividido en muchas habitaciones diferentes, pasillos y espacios abiertos

- Enemigos que reaparecen en la mayoría de las áreas al volver sobre tus pasos.
- El progreso en el juego se basa en el descubrimiento de habilidades que permiten sortear obstáculos y "desbloquear" nuevas zonas.
- Backtracking para descubrir nuevos secretos, áreas, objetos por el juego al obtener cierta habilidad.
- Progresión lineal del juego.
- Contiene elementos de RPG, como equipamiento para aumentar las estadísticas o un sistema de niveles.

Estas mecánicas han sido aceptadas como estándares dentro del género de modo que la evolución natural ha sido la introducción de penalizaciones al jugador en caso de que pierda toda la vida. Es habitual en este tipo de juegos donde los puntos de guardado están tan dispersos que el jugador se confía en exceso y siga explorando de modo que un descuido pueda acabar en una muerte segura con todas las consecuencias que ello conlleva.

En este punto es cuando se produjo otro cisma dentro del género de modo que hubo juegos que siguen la estela de los metroidvania clásicos, donde al morir vuelves al último punto de guardado y tienes que repetir todos tus pasos. La otra alternativa fue la introducción de mecánicas de los géneros souls donde al morir vuelves a un punto de guardado pero el jugador pierde una cantidad de ítems (objetos, dinero...) que se incita a recuperar pues en algunos casos recuperas los objetos perdidos o recuperas una sección de las estadísticas, por ejemplo en Blasphemous cada vez que se muere se pierde parte de la vida de magia del personaje impidiéndole así acceder a sus habilidades.



Figura 14: El penitente debe expiar la culpa para alcanzar la madre de madres.

Para la realización de este proyecto se ha tomado la saga Castlevania en su línea más purista para la implementación de las mecánicas, animaciones, movimientos además de la escenografía gótica.

La elección se debe principalmente a que nos encontraremos patrones que suelen repetirse en este género de modo que así reducimos la curva de aprendizaje al encontrarse ya familiarizado con las mecánicas de otros juegos similares.

2.2. Estado del arte

En esta sección se centrará en las tecnologías existentes para el desarrollo de videojuegos y que incluyen los motores de juego que se encuentran en el mercado y que son accesibles para este tipo de proyecto pues es bien sabido que las grandes compañías como CD Projekt Red o Square-Enix tienen sus propios motores (REDengine 4 y Luminous respectivamente) pero otras desarrolladoras suelen usar motores más comerciales como los que serán descritos.

La elección del motor gráfico supondrá el acceso a las herramientas que al final permitirán hacer viable el proyecto, teniendo en cuenta que muchos de ellos tienen diferentes opciones de licencias según se quiera un acceso gratuito o de pago.

Para la realización del proyecto se evaluaron una serie de motores gráficos para la vida al proyecto, siendo Unity el seleccionado de entre ellos por las características que ofrece y que serán detalladas a continuación.

	Unity	Unreal engine	RPG maker	GameMaker Studio 2
Lenguaje	C#	C# y Blueprints	(Arrastrar y soltar)	GML (Arrastrar y soltar)
Precio	De pago y gratuito	De pago y gratuito (con royalties)	De pago (juego base + DLCs)	De pago
Curva de aprendizaje	Baja-media	Alta	Baja	Baja
Tipo de juegos	2D y 3D (todos los géneros)	2D y 3D (todos los géneros)	2D (RPG únicamente)	2D

Tabla 1: Comparativa Motores analizados

2.2.1. Unity

Motor para el desarrollo de juegos en 2D o 3D. Presenta una gran comunidad y tutoriales asociados que permite que se reduzca la curva de aprendizaje para el desarrollo de videojuegos de corte indie. Presenta varias modalidades de pago, aunque hay una versión gratuita. Su motor gráfico permite generar físicas de manera sencilla por medio de su app, también permite un modelo de creación de animaciones y viene vinculado con Visual Studio para el desarrollo de scripts en lenguaje C#.

Dentro de su uso destaca su interface sencilla la cual permite el desarrollo sea fácil de seguir sin tener que marearte con multitud de ventanas para cada una de las partes que conforman el proyecto (script, animaciones, paleta...).



Figura 15: Interface Unity

Por último destacar que tiene una página de assets de la que puedes adquirir recursos para tus juegos (gratuitos o de pago) para multitud de proyectos y además permite la exportación de los proyectos a diferentes plataformas ya sea móviles, consolas...

Por todas estas características Unity ha sido el motor de juego seleccionado para el desarrollo de Inferno's Court.

Dentro de los juegos actuales más famosos desarrollados en unity encontraremos Cuphead, la saga Ori, Hollow knight [19].

2.3. Público objetivo (i.e. *target audience*) y perfiles de usuario

El público objetivo obtenido a través de una encuesta [20] sobre el género son:

- Personas con una edad comprendida entre los 26-30 años en adelante. Esta franja de edad cubre a aquellos que vivieron el paso del género a los 32 bits con SOTN y que siguieron con las entregas portátiles.
- Prácticamente en su totalidad los jugadores de este género son hombres.

- Actualmente, y debido a la política de ofertas de las diferentes plataformas digitales no se puede generar un perfil socio-económico del target al que está destinado el juego. Sin embargo, los precios a los que se pueden encontrar los juegos más clásicos (de PlayStation One hacia tras) pueden alcanzar grandes sumas de dinero si se encuentra en un estado aceptable con embalaje y manual correspondiente [21]. Pero debido a la emulación o su inclusión en diversos recopilatorios hace que el perfil del coleccionador sea más estricto.
- Respecto al perfil geográfico, los jugadores predominantes se encuentran en el llamado mundo siendo América el país donde más se consume este género.
- Dentro del perfil cultural, no puede establecerse una relación respecto al nivel de formación pero puede extrapolarse que el target son individuos con un alto nivel de formación en el uso de ordenadores/consolas y que disfrutaran del ocio digital al haber crecido durante la generalización de las consolas domésticas.
- Dentro de las convenciones del juego los usuarios disfrutaban más de la exploración y de la adquisición de habilidades para completar el mapa.
- Por último, destacar que dentro de los juegos más valorados por el género se encuentra en primera posición Hollow Knight, seguido de Dead Cells y de Blasphemous. En este punto podemos ver que los más valorados son curiosamente a los juegos enmarcados dentro del auge del género de los últimos años, lo cual puede significar que estos jugadores que se reincorporaron debido a la ausencia de la saga principal.

2.4. Análisis de mercado

Castlevania y los Metroidvania desde sus inicios no ha sido un juego que fuera candidato a vender muchos ejemplares a diferencia de otros juegos como podría ser Elden Ring que en solo una semana ha superado los 10 millones de copias en todo el mundo [22].

Aunque es bien cierto que tiene muchos fans dentro de la comunidad y que se llenan los foros pidiendo el regreso de la serie los números no mienten y posiblemente se deba a la plataforma donde es lanzado.

Paradójicamente, una de las entregas más laureadas es la trilogía compuesta por Dawn of Sorrow (2005), portrait of Ruin (2006) y Order of Ecclesia (2008) lanzadas en Nintendo DS y que se encuentran entre las posiciones 10 y 20 del ranking [23] de juegos de la saga más vendidos en una plataforma con un parque de más de 150 millones de consolas vendidas [33].

Como se comentó en la introducción de este trabajo, el género ha tenido una nueva revitalización desde el abandono de Konami por la saga [25-28], gracias al panorama indie. Otro factor que fomentó el abandono de la saga han sido los fracasos comerciales que supusieron Castlevania 64 (Nintendo 64, 1999) que provocó que no se realizaran juegos en 3D hasta que salió Lord of Shadows (2010).

Lord of Shadows, tanto por las críticas como las ventas propició que se quisiera seguir con la saga en este spin off con una vuelta de guion sobre la lucha entre los Belmont y Drácula. Para el desarrollo de la segunda parte se confió en un equipo español para ello, Mercury Steam.

Sin embargo el juego no estuvo a la altura de lo esperado por motivos que escapan a la índole y temática de este trabajo, provocando que se abandonara definitivamente la serie en el olvido por parte de Konami [29]. Aunque posteriormente el estudio se resarcíó con un remake de Metroid II en Nintendo 3DS que acabo haciendo doblete en el premiado Metroid Dread en 2021, el cual ha sido el juego más vendido de toda la saga Metroid [30].

Juegos como los citados a lo largo de este trabajo a los que podemos sumarles Dead Cells (Motion Twin, 2017), Owl boy (D-Pad Studio, 2016) o Celeste (Matt Makes Games, Extremely OK Games, Ltd., 2018) han contribuido a que el interés por el género haya resucitado, de hecho cualquiera de estos juegos ha vendido más que los propios juegos en los que se basan.

Aun así, estos juegos suelen ser creados por estudios pequeños, y a pesar de su calidad y aceptación por el público no serán considerados como AAA dado que su target objetivo no es tan extenso como la comunidad de Fortnite u otro juego de tipo MOBA.

Posición Ventas	Juego	Consola	Ventas	Fecha de lanzamiento
1	Symphony of the Night	PS1	1.27 m	Oct 1997
2	Castlevania	NES	1.23 m	May 1987
3	Lord of Shadows	PS3	1.11 m	Oct 2010
4	Lament of Innocence	PS2	0.94 m	Oct 2003
5	Castlevania II: Simon's quest	NES	0.93	Nov 1988
6	Castlevania III: Dracula's curse	NES	0.85	Oct 1990
7	Lord of Shadows	360	0.66 m	Oct 2010
8	Castlevania	N64	0.64 m	Ene 1999
9	Super Castlevania IV	SNES	0.63 m	Dic 1991
10	Circle of the Moon	GBA	0.89	Jun 2001
11	Dawn of Sorrow	DS	0.43	Oct 2005

Tabla 2: Ranking de ventas de la saga Castlevania

Teniendo en consideración todo lo anterior, podemos afirmar que actualmente la saga este en la cresta de la ola otra vez y que las ventas que se generan por los lanzamientos suponen que la inversión en el proyecto es retornable.

De todas maneras, juegos como el que aquí compete son de carácter *fan made* y se pueden encontrar muchos en plataformas como itch.io para ser disfrutados gratuitamente por los usuarios [31].

2.5. Competencia/Antecedentes (o marco teórico/estado del arte)

El análisis DAFO es una herramienta de estudio de Marketing necesaria para conocer la situación del proyecto ya sea internamente como externamente y que nos pueda ayudar para elaborar las acciones necesarias para alcanzar los objetivos presupuestos en el plan de

negocio. Este análisis presenta unas características que ayudan a esa toma de decisiones de manera rápida por:

- Sencillo y fácil de aplicar.
- Muy concreto y detallado.
- Proporciona una visión global del proyecto.
- Favorece la detección de problemas en el ámbito de la organización y responder a estos.



Figura 16: Cuadrícula DAFO. Obtenido de <https://www.mastermarketing-valencia.com/ventas-y-gestion-comercial/blog/que-es-dafo/>

Análisis interno

- **Debilidades**

- **Mercado muy competitivo:** Actualmente con el renacimiento del género la cantidad de juegos al año que son publicados es enorme y llamar la atención sobre el público es más complicado [32-33]. Es más que probable que el usuario se encuentre abrumado y “empachado” tras haber jugado a alguno. y que muchos de ellos no alcancen los objetivos previstos debido a esta saturación.
- **Falta de experiencia con las herramientas de desarrollo:** No tener un dominio de las herramientas a usar puede acarrear problemas y posibles retrasos en los proyectos.
En este caso no influiría debido a que el objetivo de este trabajo es el aprendizaje en términos generales pero en el ámbito de la industria puede suponer que un proceso deba ser cancelado o retrasado indefinidamente [34].
- **Tiempo de desarrollo elevado:** Un proyecto de este calibre, realizado por una única persona teniendo que desarrollar todos los aspectos incluidos (assets, scripts, motor de juego...) en un plazo de tiempo muy corto supone que deba invertirse una gran cantidad de tiempo para abarcar todo el proyecto.
- **Marketing:** Es necesario llevar a cabo una gran campaña de marketing que ponga en el candelero al producto y pueda llegar así tanto al target objetivo como a los posibles potenciales que se sientan atraídos.

Un proyecto de este calibre no tiene el musculo necesario para conseguir llegar al target, por ello se tendera a buscar vías alternativas, por ejemplo publicar en itch.io.

- **Exceso de trabajo:** Es necesario con un equipo mínimo organizar el cronograma correctamente para evitar sobrecargas de trabajo y que se alcancen los hitos dentro de los tiempos establecidos. Con el equipo de una persona cualquier incidencia supone una modificación sustancial del cronograma.
- **Fortalezas**
 - **Pasión por el tema a elaborar:** El realizar un trabajo que además es parte de tus hobbies permite al desarrollador tener una motivación alta y ver realizados sus esfuerzos durante las fases de pruebas, aunque tenga que lidiar con unos ritmos de trabajo altos, sobre todo si el desarrollo del proyecto no forma parte de sus tareas laborables.
 - **Género en auge:** El género metroidvania ha tenido un relanzamiento con título potentes en los últimos años gracias al panorama indie y a la apuesta de grandes estudios para recuperar un género que hasta el momento de autos los máximos exponentes habían abandonado.
Que se sigan desarrollando y que tengan en pensamiento lanzar numerosas propuestas hace pensar que los gustos de los usuarios están variando más allá de Fornite o el Fifa de cada año [62-64].
 - **Coste de desarrollo casi nulo:** Aunque el uso del motor Unity en su versión más profesional es de pago, las herramientas a disposición en la licencia gratuita permitirán desarrollar el proyecto en su totalidad.
Esto sumado también al uso de assets gratuitos permitirá obtener un mejor acabado centrando la atención del desarrollador en combinar todos los elementos a su disposición.
 - **Plataforma objetivo:** Que el título pueda disfrutarse en cualquier ordenador que tenga un sistema operativo Windows es una apuesta segura al querer desarrollar un producto dado que te aseguras llegar a la cuota de mercado más predominante (73%) en relación a los dispositivos no móviles [35].

Análisis externo

- **Amenazas**
 - **Juegos del mismo estilo:** La vuelta de este género al candelero ha ocasionado que el número de lanzamientos de este tipo sea muy numeroso, del mismo modo los grandes estudios, como MercurySteam, también lanzan juegos y de una gran calidad técnica lo cual hace que el listón este muy alto. Por otra parte si nos centramos en estudios de menor calibre o de corte indie vemos que el precio de estos juegos es bajo de modo que la competencia es aún mayor para acceder al target.
 - **Mercado muy exigente:** Las propuestas más clasistas del género palidece en cuanto a lo que los usuarios demandan pues en la actualidad lo que impera es que cada juego, independiente del género, tenga un aroma a la saga Souls de FromSoftware además de tener que reinventar la rueda para diferenciarse de otros competidores.

- Cambios en la tendencia del mercado: Aunque estemos diciendo que actualmente el género se encuentra viviendo una segunda luna de miel hay que tener cuidado pues podría darse el caso, como lo sucedido con Fornite, que mañana un streamer mundial diga que el juego del género “xxx” es lo más para que todo el núcleo de jugadores se mude a ese nuevo género. Muchos de estos casos lo vivimos con el Fornite, ya que el público estaba hastiado de los FPS pues sagas como Call Of Duty (Activision) sacaban juegos anuales sin novedades y reciclando elementos, de modo que la entrada de los Battle royale ocasiono que estos juegos fueran reimaginados y que a posteriori la propia Call of duty lo incorporase para poder competir en igualdad.
- Cambios legislativos para la publicación del software: Si se produjeran cambios por ejemplo dentro la legislación que otorga la calificación por edades (Pan European Game Information (PEGI)) [36] podría suponer que todo el proyecto deba ser revisado para poder ser publicado bajo la etiqueta que se desea en caso contrario se publicaría por una etiqueta con la edad correspondiente por el organismo y de este modo perder a parte de tu target.
- **Oportunidades**
 - **Aprendizaje:** Para el desarrollo del título se ha tenido que adquirir unas competencias de las que antes se carecían, aunque en parte adquiridas durante el grado. Esto permite que el que suscribe salga de su ámbito de confort y explore nuevas áreas que podrían servirle para abrirle puertas en un futuro no muy lejano.
 - **Situación mundial:** Aunque actualmente estamos en un proceso donde la pandemia provocada por la COVID-19 está desapareciendo, el consumo de ocio digital y en especial el de los videojuegos creció por culpa del confinamiento.
 - **Mercado en auge:** Los videojuegos en un sector en constante crecimiento debido en principal lugar a la aparición de dispositivos móviles de gama media alta que poseen unas características que le permiten la portabilidad de juegos que en otra época habría sido imposible ejecutarlos en uno. El ejemplo más claro lo tenemos con el juego Snake que aparecieron en los móviles Nokia a mediados de la década de los 90. En aquella época en consolas domesticas nos encontrábamos con la Nintendo 64, PsOne y Sega Saturn donde teníamos ya juegos de 32 o 64 bits con entornos poligonales en 3D mientras que los teléfonos ejecutaban juegos de aspecto GameBoy. Ahora se pueden jugar a juegos como Genshin Impact que no tienen diferencias significativas con su versión de OC [65-67].
 - **Distribución digital:** El mercado indie y los juegos de bajo desarrollo han encontrado en las plataformas de la Play Store o App Store el lugar ideal para poder distribuir el juego y así alcanzar un mayor número de targets. Así, los juegos estarán disponibles para millones de personas en todo el mundo. A parte de las stores de los dispositivos móviles o de consolas existen páginas que permiten a desarrolladores amateurs la distribución del juego al depositarlos en su plataforma, por ejemplo Itch.io.

3. Propuesta del Juego

3.1. Definición del producto

3.1.1. Descripción del juego

Inferno's Court es un videojuego de aventuras y acción en 2D. Nuestro personaje deberá superar todos los desafíos para alcanzar el castillo donde se encuentra el señor oscuro Armagrog y salvar así a Idraldor de la pesadilla en la que viven.

A lo largo del recorrido se deberá hacer uso de todas las habilidades para sobrepasar todos los obstáculos y salir victorioso en el proceso.

3.1.2. Historia, ambientación y/o trama

El reino de Idraldor era como cualquier otro reino, su población vivía en armonía con las otras razas que pueblan, elfos, enanos, draconidos... evitando así la discriminación que otros países promueven contra ellas.

Este factor promueve que la capital sea un crisol de culturas lo cual le otorgo la bendición de la diosa Selûne, guardiana del orden, confiriéndoles un poder sagrado que guardan en la capilla.

Idraldor tiene además recursos naturales lo cual provoca envidia por otros países, pero debido a la relación con el actual emperador del reino de Ethermoor propicia que no sean el objetivo de acciones hostiles contra ellos.

Pero más allá de las conspiraciones humanas, en este mundo hay un mal que se mueve entre las sombras y es aquí cuando sucede la caída de Idraldor a manos del señor oscuro Armagrog.

Al principio, Armagrog que era un general respetado entre el pueblo y sus soldados no presentaba un alineamiento caótico malvado sino que fue corrompido por el dios Myrkul lo que hizo que las raíces del mal arraigaran en lo más profundo de su ser.

Armagrog corrompido por el nuevo poder que poseía, irrumpió en la capilla donde se guardaba el tesoro sagrado y absorbió su poder convirtiéndose en el señor oscuro que ha sometido hasta ahora el reino. Con su poder atrajo como una polilla a la llama a criaturas que podrían describirse como surgidas de las más oscuras pesadillas.

Con este ejército Armagrog siguió conquistando los reinos aledaños, despojando a sus habitantes de la bendición de los dioses como Selûne y sumergiendo a todos en un infierno del que no pueden despertar.

Ni siquiera los gremios de aventureros fueron capaces de frenar la horda de oscuridad que crecía sino que cualquier intento de ataque a la fortaleza de Armagrog era repelido por los ejércitos de este, de modo que con el tiempo nadie osó levantarse en armas para recuperar el reino y su libertad.

Pero con el tiempo una nueva esperanza emergerá pues las ruedas del destino son caprichosas y en un intento de desafiar a la tiranía del caos la diosa Selûne altero el destino para que hubiera un atisbo de luz en el mundo dentro de un ser capaz de derrotarle pues esa fue la profecía que la diosa creo para salvar a sus fieles.

Armagrog al enterarse de esto reforzó su campaña de represión buscando personas que pudieran tener una chispa y que denotara el origen divino de sus habilidades, pero se equivocó. La chispa no vendría del nacimiento de un elegido sino que por ironía del destino cualquiera con el coraje de enfrentarse al mal sería capaz de derrotarle de modo que toda la humanidad tenía la solución a su problema si se unieran pero el miedo les hacía incapaz de darse cuenta de este hecho.

Sin embargo, en una pequeña aldea un caballero se alza contra las hordas, pero ¿Qué motiva a un caballero a enfrentarse a los infiernos de los que el resto del mundo huiría, exponiéndolo a una muerte segura?

La respuesta...es el amor. La pérdida de su prometida fue razón más que suficiente para lograr lo que ningún otro mortal hubiera soñando, pero esa historia todavía está por contar. Ahora es aquí donde, al principio de nuestra historia donde nuestro caballero llega al poblado habitado más cercano al antiguo castillo de Idraldor...

3.1.3. Definición de los personajes/elementos

3.1.3.1 El caballero

El caballero es el protagonista de nuestra aventura y el jugador deberá de guiarle por las oscuras tierras de Idraldor para que logre el objetivo de salvar el mundo. El caballero puede interactuar y colisionar con el resto de elementos del juego.



Figura 17: Personaje principal del juego, el caballero

3.1.3.2 Enemigos y trampas

En esta sección se describirán todos los elementos que suponen un riesgo para nuestro jugador. Por el escenario habrá enemigos que se dediquen a patrullar un área establecida, otros perseguirán o atacarán a nuestro personaje nada más detectarle.

Los enemigos podremos diferenciarlos dentro de dos grupos, aquellos que no disponen de un patrón de ataque y cuya función es ser unos kamikazes que colisionaran contra el jugador frente a los que tienen un ataque activo ya sea por medio de las armas que dispongan o hagan uso de la magia.

Tendremos una gran variedad de enemigos y como es natural en juegos de esta saga se reciclarán los sprites para generar diferentes tipos de enemigos de la misma clase pero que tendrán estadísticas diferentes (ver anexos bestiario).



Figura 18: Enemigos

Al igual habrá objetos en el escenario cuyo contacto con el jugador supondrá que este pierda vida.

3.1.3.3 Elementos con los que puede interactuar

El caballero podrá interactuar con diversos elementos en pantalla como podrán ser las diferentes pociones que repercutirán en sus estadísticas ya hablemos de la recuperación de vida, el mana o la adquisición de experiencia. También, habrá coleccionables como podrán

ser monedas que podrían ser usadas para comprar objetos como podrán ser pociones o potenciadores.



Figura 19: Ítems y coleccionables

3.1.3.4 Elementos que forman parte de la escena

Los elementos de la escena condicionan los muros que nuestro caballero no puede atravesar. Estos serán los muros y el suelo, los cuales soportarán el peso de los diferentes elementos que tendrán un componente activo en la escena.

De igual manera, se desarrollarán *checkpoints* los cuales serán usados por el jugador para guardar el progreso de su partida.

Por último, habrá elementos como elementos decorativos o NPCs, en segundo plano que darán una sensación de un juego con más “vida” en él.



Figura 20: La estatua de la diosa funciona como Check Point y Save Point

3.1.4. Interacción entre los actores del juego

En Inferno's Court hay dos partes bien diferenciadas, una será la parte donde el jugador explorara los diferentes niveles que conforman el mundo de Idraldor y la otra parte será el combate contra la horda infernal que quiere barrer el mundo.

Además de nuestro protagonista tendremos al escenario y a los enemigos como actores con los que interactuar. Tanto ciertos elementos del escenario como los enemigos serán hostiles para nuestro caballero de modo que si el caballero colisiona con estos elementos perderá vida. En caso de perder toda la vida morirá y se mostrará la pantalla de *Game Over*.

Tanto el combate como la exploración se hará por medio del teclado. Donde podremos encontrar las siguientes acciones los controles son descritos en la sección 6.3.2 de esta memoria.

3.1.5. Objetivos planteados

El objetivo que se plantea al jugador es salvar a la tierra y sus gentes de Idraldor de las garras de Armagrog para ello el personaje recorrerá una serie de escenarios adquiriendo experiencia para derrotar al jefe final que le espera en la sala del trono de su castillo. A medida que el jugador progresa se hará más fuerte y por ende también los enemigos que le salgan al paso.

3.1.6. Concepts Arts

En esta sección se incluyen bocetos e ideas que serán implementados dentro del producto final.

3.1.6.1 Niveles

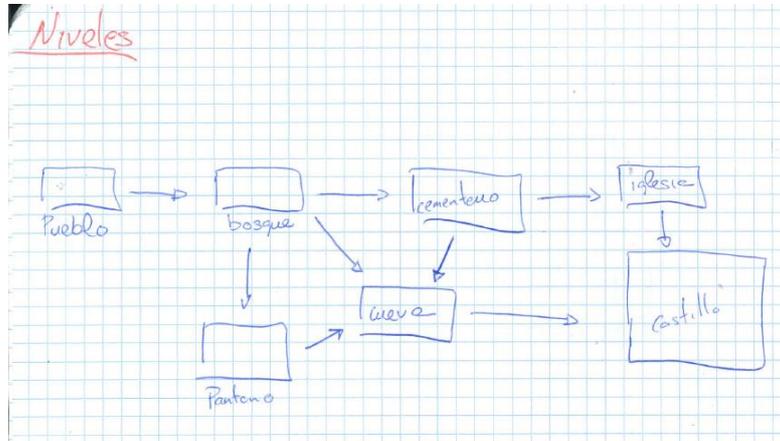


Figura 21: Estructura del reino de Idraldor

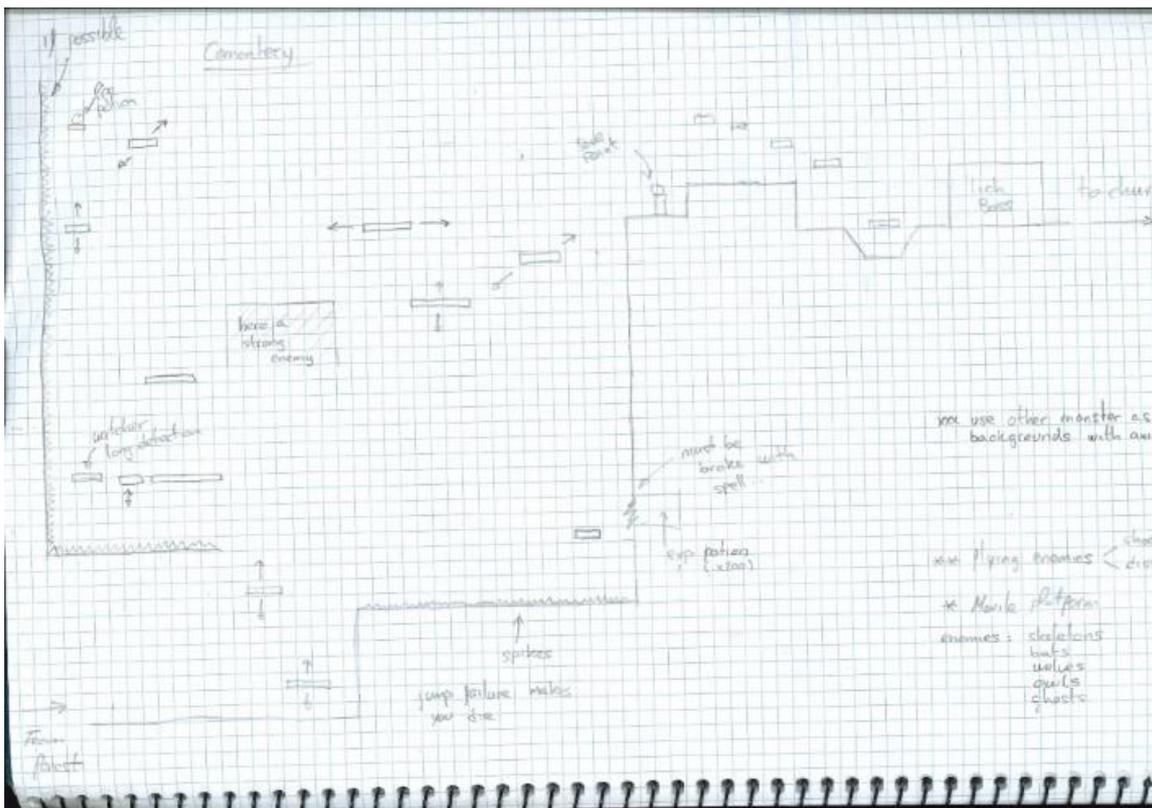


Figura 22: Diseño preliminar de la fase del cementerio con anotaciones sobre los elementos a incluir.

3.1.6.2 Funcionalidades/Mecánicas

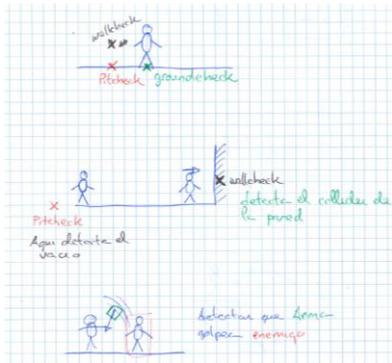


Figura 23: Detección de colisiones y elementos estáticos de la escena

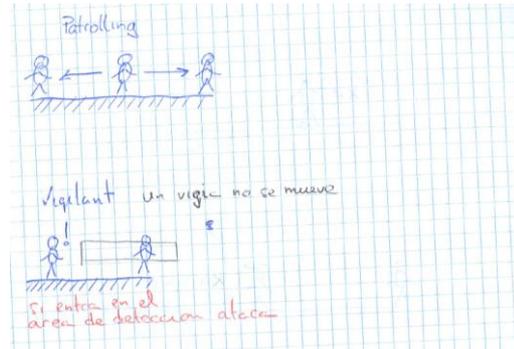


Figura 26: IA de los enemigos

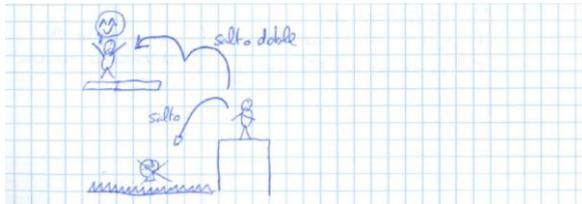


Figura 24: Habilidad del doble salto



Figura 27: Origen de la idea del bestiario

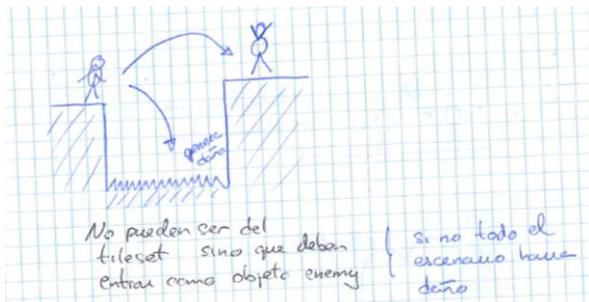


Figura 25: Elementos dañinos de la escena

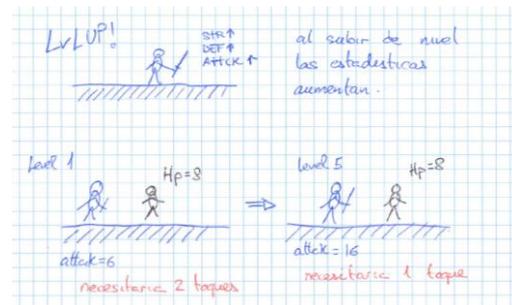


Figura 28: Toques de RPG al juego

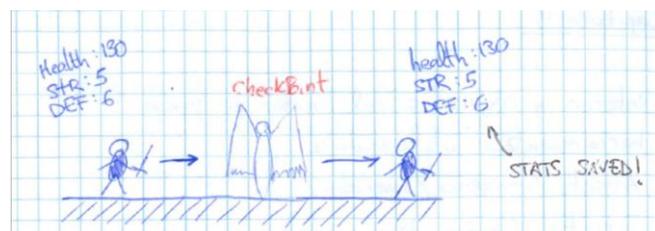


Figura 29: Salvado de estadísticas

3.1.7. Plataforma de destino

La plataforma de destino es el escritorio, el cual debe ser ejecutado bajo el sistema operativo de Windows. También se implementará una versión jugable online a través de la plataforma itch.io.

Por último comentar que no se descarta realizar una versión portátil que corra en sistemas operativos Android o una versión ejecutable desde cualquier navegador web, ya que este tipo de juegos al requerir de pocos comandos son fácilmente adaptables para generar botones táctiles.

También esta plataforma es un objetivo debido a la alta concurrencia de usuarios casuales se concentran en estas plataformas y supondrá un incremento del *target* objetivo del proyecto.

3.2. Definición de objetivos/especificaciones del producto

El objetivo principal del producto es entretener al jugador con una jugabilidad pulida. Dicho producto.

Especificaciones:

- **Explora un mundo no lineal:** Derrota a enemigos terroríficos y supera trampas mortales mientras exploras una gran cantidad de zonas, y busca salvar Idraldor, un mundo oscuro.
- **Combate brutal:** Masacra a los enemigos con tu espada, que nada se interponga en tu camino.
- **Personaliza tu personaje:** mejora las características de tu personaje por medio de equipamiento. Prueba con distintas combinaciones para encontrar una que encaje con tu estilo de juego.
- **Batallas épicas contra jefes:** Para llegar al castillo tendrás que derrotar a hordas de criaturas gigantes y malignas. Descubre sus patrones de movimientos y sobrevive hasta conseguir la victoria.

3.3. Modelo de negocio

No se han desarrollado una valoración económica de los mismo pues se han usado los medios disponibles por el firmante y se ha usado software de distribución gratuita para el desarrollo del mismo.

El juego de acuerdo a lo que se incluye en los créditos/Copyrights es de distribución libre y no se espera tener beneficios por su distribución. Este proyecto es puramente educativo y de carácter freemium [37].

3.4. Estrategia de marketing

La estrategia de marketing de Inferno's Court consistirá en la publicación de material referente al juego en las diversas plataformas (ver anexos, enlaces) hasta la publicación final del juego.

Los comentarios obtenidos tras la realización de las pruebas serán analizados y publicados en esta memoria, aunque podrán ser consultados en la carpeta del drive del proyecto. Los comentarios referentes tanto a los fallos vistos por los testers como sus sugerencias serán analizados en el apartado 6.5 de esta memoria.

De igual manera, en dicho apartado, se comentarán las medidas oportunas realizadas para incluir/corregir/eliminar los componentes que más han llamado la atención durante la prueba de usuarios. De esta manera se recibe un feedback que será usado para mejorar el juego y que tenga una mayor repercusión en la comunidad.

Se publicarán al finalizar cada fase de trabajo tanto la demo jugable en dicha fase como los diarios de desarrollo que nos narrarán como el proyecto avanza y cobra forma.

Junto a las versiones parciales que se vayan generando y sus publicaciones tanto por el foro interno como en games jams se recogerán los comentarios obtenidos por los beta-testers para analizar la variabilidad de las propuestas y determinar la inclusión de los mismos dentro del motor de juego.

De igual manera cuando el juego se encuentre en fase gold (ver apartado 6.4.2) y se encuentre cerca de la fecha de lanzamiento se realizará un tráiler de lanzamiento para recordar al target objetivo cuando podrá disponer del título en las plataformas seleccionadas.

4. Diseño

En el diseño de videojuegos es donde uno podría encontrarse el mayor obstáculo pues es necesario poner en obra todas las ideas que se han realizado en las fases anteriores. Por ello hay que tener en consideración la tecnología y los recursos a disposición para ello.

Durante el desarrollo del juego se ha tenido en consideración que el jugador entre en un estado mental establecido por Samuel Coleridge y denominado "suspensión de la incredulidad" [45] el cual se sucede en otros contenidos multimedia como podrían ser las películas o la música amen de los videojuegos. Este estado mental se caracteriza por dejar atrás el mundo real a favor del universo ficticio según la trama del mismo avanza.

Como diseñador se ha intentado que el jugador este en este estado el mayor tiempo posible, habrá que intentar evitar que la frustración del jugador alcance un máximo deje el juego perdiendo la suspensión de incredulidad. Si se alcanza este punto podríamos decir que habría fracasado como desarrollador pues dicho jugador podría no recomendar el juego a sus amigos perdiendo así targets.

Ron Hilbert, padre del mítico The secret of Monkey Island que actualmente va a terminar su trilogía [46], propuso en 1989 en su publicación "Why adventure games sucks" [47] propuso un decálogo de reglas (simples y con bastante sentido común) que deberían seguirse para desarrollar buenas historias y utilizadas por grandes estudios de nuestros días como Naughty Dog (Uncharted, The Last of Us), Telltale Games (The Walking Dead) o Double Fine (psychonauts 2) [48].

Aquí mostraremos y detallaremos las reglas aplicadas a nuestro desarrollo y como han sido implementadas en el juego:

- **Regla 1: El objetivo debe ser claro**

Desde el principio el jugador debe tener claro su objetivo que le permitirá llegar al final del juego. Para aplicar esta regla fue creado un NPC, el cual se encontrará nada más arrancar el juego con nuestro caballero, explicándole lo sucedido (a falta de una cutscene) y diciéndole que debe hacer para mostrar en este caso la pantalla de juego completo.

- **Regla 2: Los puzles (o la escena) al revés**

Para que un puzle o mapa sea placentero para el jugador, este debe ser construido desde la solución hasta su inicio. La pantalla debe ser construida al revés por este motivo las plataformas se han construido de modo que desde la salida el jugador pueda seguir una ruta sin que haya ningún impedimento hasta alcanzar el punto de partida. De este modo nos aseguramos que el jugador no tendrá ningún tipo de

problemas para superar la fase y así de paso evitamos fallos de diseño que podrían ser detectados en fases más tardías del proyecto.

- **Regla 3: Los puzles (o la escena) debe alimentar la historia**

Hay que huir de los puzles o escenas que no tengan ningún impacto en la historia. En este punto podríamos haber realizado más fases, por ejemplo, una cueva, las cuales nos obligarían a ser encajadas dentro del marco argumentativo del juego, pero si no fuera así sería una fase cuya entrada y salida sería lo mismo y por tanto no tendría más sentido que distraer al jugador de su propósito.

- **Regla 4: Recompensa progresiva**

El placer de un juego es descubrir nuevas escenas, NPC, situaciones... de modo que se han implementado que los enemigos tengan diferente temática según la escena y cuyo comportamiento varíe también de modo que la introducción de estas mecánicas se realiza moderadamente, pero de manera constante (por medio de la subida de nivel) y así satisfaciendo los instintos más básicos del jugador.

En los siguientes apartados se abordarán los diferentes aspectos técnicos usados para dar vida a la idea de Inferno's Court.

4.1. Entorno y requisitos

4.1.1. Entorno

Tras haber detallado nuestro proyecto y haber dado pinceladas del estilo y las mecánicas a desarrollar podemos comentar las principales características que tendrá el juego y que determinaran la selección del motor.

- Juego en 2D con desplazamiento lateral.
- Recursos gráficos sencillos de aspecto pixel art.
- No es necesario una carga muy grande en pantalla en relación a polígonos ni fps.
- Juego principalmente desarrollado para jugar en ordenadores.
- Tiempo y recursos limitados para la elaboración del proyecto.
- Fuerte carga lógica en la implementación de las mecánicas del juego.

El motor elegido para el desarrollo del juego Inferno's Court es Unity. El principal motivo para su elección fue el uso de la una licencia gratuita que permitiera con las herramientas disponible acceder a los objetivos propuestos. Otra razón es que es una buena elección para el desarrollo de juegos 2D usando el sistema de sprites y tiles y que el propio motor requiere de unos requisitos muy bajos para poder ejecutarse y comenzar a desarrollar un videojuego.

También, existe una gran comunidad de usuarios, tanto amateurs como profesionales, que usan este motor de modo que habrá recursos a disposición de los usuarios además de prestar asistencia para poder distribuir los proyectos de cada uno. Unido a esto destacar la facilidad de Unity para poder exportar el proyecto a diferentes plataformas ya sean de tipo web, móvil o de ordenador.

Unity además posee una tienda que permite acceder a estos recursos como pueden ser escenarios para enriquecer nuestra escena, personajes jugables tanto como elementos relacionados con la interfaz de usuario, elementos fácilmente implementables utilizando el lenguaje C#.

Para finalizar decir, que Inferno's Court es mi primera aproximación al desarrollo amateur de videojuegos y que Unity presenta un entorno fácil, intuitivo y muy directo, a través del inspector y de las otras pantallas para acceder rápidamente tanto a la paleta de escenario, como al de animadores pasando por una previsualización del juego a medida que se va construyendo.

4.1.2. Requisitos técnicos

El equipo sobre el que se ha desarrollado tiene las siguientes características:

Procesador	AMD Ryzen 5 3500x
RAM	8 GB DDR4 3200 MHz
Tipo de sistema	Sistema operativo de 64 bits
Sistema operativo	Windows 11 Pro
Tarjeta Grafica	Gigabyte GeForce GTX 2070 8GB GDDR6

Tabla 3: Especificaciones del dispositivo

La versión usada para el desarrollo del juego es la 2021.2.9f1. Los requerimientos para usar Unity, en un entorno Windows, se describen en la siguiente tabla [49-50]:

	Requisitos mínimos	Requisitos recomendados
Sistema operativo	Windows 7 (SP1+), Windows 10	Windows 11 64 bit
CPU	Arquitectura x86, x64 con soporte del conjunto de instrucciones SSE2.	Arquitectura x64 con soporte del conjunto de instrucciones SSE2.
API grafica	DX10, DX11	DX12
Requerimientos adicionales	Controladores con soporte oficial del proveedor de hardware. Para el desarrollo: El backend de scripting IL2CPP requiere Visual Studio 2015 con el componente C++ Tools o posterior y el SDK de Windows 10.	Controladores con soporte oficial del proveedor de hardware. requiere Visual Studio 2019 con el componente C++ Tools o posterior y el SDK de Windows 10
Gráficos	Tarjeta de video de 512 Mb (GTX 650 en adelante)	RTX 3070(para proyectos 3D)
RAM	Desde 4 GB	16 GB
Espacio en disco	10 GB	10 GB

Tabla 4: Requisitos máximos y recomendados del motor Unity

4.2. Herramientas utilizadas

- **Unity:** Motor grafico sobre el que se implementa la construcción del juego.
- **Visual studio 2019:** Visual Studio es el Integrated Development Environment (IDE) que se ha utilizado para crear los scripts, en lenguaje C#, que implementan las diferentes funcionalidades del juego.
- **Google Drive:** Herramienta para albergar copia y la encuesta realizada sobre la versión pre Alpha del juego.

- **Paquete Office:** Herramientas ofimáticas de Microsoft empleado para la elaboración de la memoria y de los anexos (hojas de cálculo, presentaciones...).
- **Paint3D:** Editor de imágenes utilizado para la manipulación de los assets.
- **GanttPro:** Herramienta para el diseño de la planificación del proyecto.
- **BitBucket:** Repositorio del código fuente del proyecto.
- **SourceTree:** Aplicación de escritorio que permite interactuar entre el repositorio de la nube con el sistema de fichero de nuestro proyecto de manera fácil e intuitiva.
- **Obs studio:** Herramienta para la grabación de los diferentes videos comentando el desarrollo del juego para las diferentes entregas del proyecto. Herramienta que permite capturar la pantalla y la cámara del ordenador para la elaboración de streamings.
- **Xilisoft Video Convertidor Ultimate:** Herramienta para la conversión del audio usado en el proyecto para convertir al formato adecuado y aceptado por el motor gráfico.
- **DaVinci Resolve:** Herramienta para la edición del tráiler y de la defensa del proyecto. La versión utilizada es la 17.4.6 versión gratuita.
- **Draw io (web versión):** Herramienta de software gratuito y de código abierto de dibujo gráfico multiplataforma desarrollado en HTML5 y JavaScript. Se utiliza para crear diagramas tales como diagramas de flujo, wireframes, diagramas UML...

4.3. Recursos del videojuego

Los recursos utilizados para el desarrollo de Inferno's Court son totalmente gratuitos, se trata de elementos, gráficos o sonoros, con una licencia libre. Los recursos se han ido recopilando de diversas plataformas como podrían ser al Asset Store de Unity [51] o páginas de internet como Itch.io [61] o The Spriters Resource [53].

Estos recursos han sido desarrollados por terceras personas que o bien se tratan de recursos propios o son recursos de juegos licenciados por las grandes compañías y que han sido desglosados del juego donde fueron desarrollados.

El uso de estos recursos de licencia libre ha permitido al proyecto a desarrollarse más rápidamente para centrarse en las ideas principales que has sido desarrolladas siguiendo el modelo MDA y poder ofrecer una mayor variedad de mecánicas de haber tenido que desarrollar recursos propios no habrían sido posibles debido a la planificación impuesta.

Estas mecánicas y dinámicas propuestas esperan que una vez completadas y ajustadas a la temática/idea del juego pueda proporcionar una grata satisfacción al jugador. Como desarrollador este es el principal objetivo que espera, transmitir una idea al público y que esta cale y tenga repercusión.

4.3.1. Assets

A continuación podemos apreciar la estructura de la carpeta Assets del proyecto.

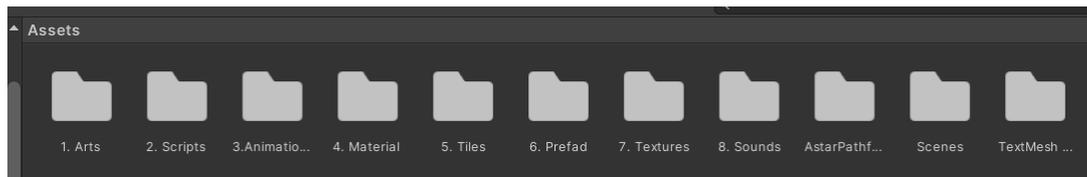


Figura 30: Estructura de la carpeta Assets

Las carpetas están estructuradas para albergar los diferentes elementos que serán usados en cada uno de los componentes para desarrollar ya sean o bien mecánicas o las dinámicas. Algunos elementos serán descritos en mayor profundidad en su apartado correspondiente.

- **Arts:** En esta carpeta se albergarán todos los recursos gráficos usados en el juego.
- **Scripts:** En esta carpeta encontraremos los archivos de programación en lenguaje C# para donde están configuradas las mecánicas implementadas desde el inicio del juego.
- **Animation:** Todas las animaciones que se han desarrollado para los personajes, escenario, menús...
- **Materials:** en esta carpeta guardaremos los elementos que permitirán mejorar las mecánicas físicas del juego.
- **Tiles:** La escena del juego ha sido desarrollada mediante el tile palette de Unity a partir de los recursos gráficos. Esto favorece que la escena pueda construirse con mayor rapidez al no depender de código.
- **Prefabs:** objetos reutilizables, y creados a partir de unos elementos dentro del proyecto los cuales serán instanciados cada vez que se estime oportuno y tantas veces como sea necesario.
- **Texturas:** Encontraremos los elementos que conforman parte de la interfaz de usuario (UI abreviatura procedente del término *sajón*).
- **Sonidos:** En esta carpeta guardaremos los archivos de audio utilizados en el proyecto.
- **Scene:** Estas escenas son las que componen el juego.

4.3.2. Imágenes

Debido a la gran cantidad de recursos gráficos usados en el proceso, sobre todo en la parte de los escenarios como de los enemigos la explicación se hará con una parte representativa de los mismos dejando relegados el resto a una tabla posterior enumerando todos los recursos usados.

Logos e Iconos

Se ha desarrollado un logo del juego que será usado como *concept art* para la portada del juego físico como para la del manual de usuario (ver anexos adicionales). También, se ha diseñado un logo de empresa para darle al proyecto un aspecto de profesionalidad.

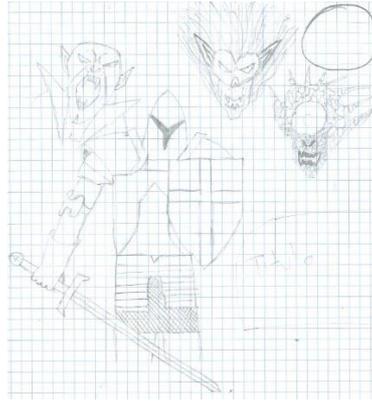


Figura 31: Concept art de la portada

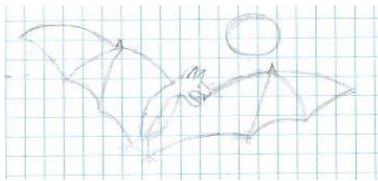


Figura 31: Concept art del logo de EldritchMoon Studios



Figura 32: logo final de EldritchMoon Studios

Los iconos del juego han sido desarrollados usando la aplicación online de game-icons [61].

Ambientación

Gran parte de la riqueza de este tipo de género radica en los escenarios por donde transcurre la acción. Dentro de los tilesets podremos desarrollar diferentes capas para alcanzar una ambientación satisfactoria.

La evolución de las pantallas en 2D supuso de pasar desde una pantalla con 1 capa (PacMan), pasando por 2 (Super Mario Bros) a un número mayor donde las capas de fondo cobraban más importancia para darle una "mayor" profundidad.

Habiendo determinado que la ambientación es un factor importante en el desarrollo de un juego podemos establecer que las diferentes secciones que conformaran las zonas por las que nuestro avatar pase deben seguir una construcción que otorgue una mayor expresividad y huir de unos simples escenarios 2D planos.

Por ellos los tilesets usados han sido seleccionados entre un gran conjunto de materiales y que se adaptan a la idea del que suscribe para la realización del proyecto. Algunos de ellos podrían no tener un estilo parecido entre ellos debido a las ideas creativas de los autores, la técnica de creación (más pixel art o más acuarela por citar alguno) y la idea de

juego que estos tuvieran al desarrollar los escenarios prrenderizados pero se ha trabajados con ellos para generar una integración lo más orgánica posible en el videojuego.



Figura 32: Tilemap del pueblo seccionado por componentes



Figura 33: Imagen de background

Personaje principal

Para el personaje principal, nuestro caballero, se seleccionó entre una multitud de recursos que se encontraban en las páginas de recursos anteriormente citadas. Dentro de la gran variedad, de personajes principales de este tipo, se seleccionó uno que siguiera la estructura del juego planteada y la idea que se tenía desde el principio respecto a la historia.

Por ejemplo, este personaje ha permitido desarrollar las mecánicas de saltar, atacar, correr... y con estas mecánicas poder desarrollar las dinámicas que permiten al Player acceder a nuevas áreas/progresar por el escenario, derrotar enemigos y poder desplazarse para explorar el escenario respectivamente.

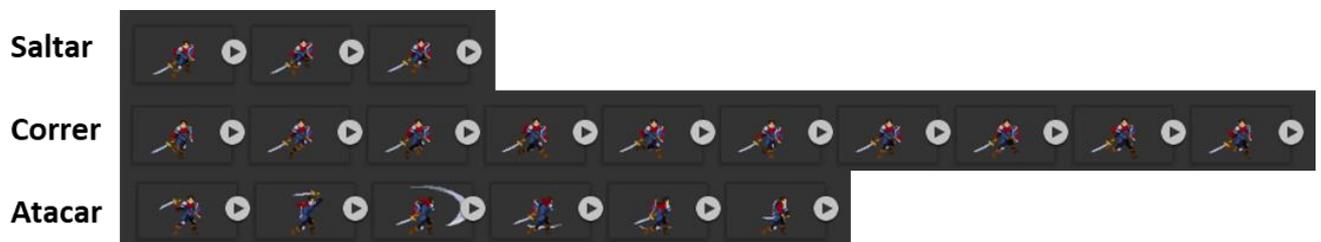


Figura 34: Sprites de mecánicas usadas por nuestro caballero

Así disponemos de 5 tipos de mecánicas a disposición del jugador:

- Mover: El jugador se mueve a los lados.

- Saltar: permite al jugador saltar obstáculos o enemigos (algunos).
- Atacar: usando la espada del personaje se puede atacar a los enemigos.
- Magia: Permite al jugador liberar un ataque mágico que puede ser usado como ataque de media-corta distancia o defensivo.
- Hablar: Permite al jugador hablar con el NPC cuando se active el indicador pertinente.

Para las físicas del jugador, y para el resto de elementos de los enemigos, se han asociado a cada elemento los componentes de Unity que otorgan un componente sólido dotando al elemento en cuestión de masa sobre la que la gravedad ejerce su fuerza por medio de los *rigidbody 2D*.

Para las colisiones se usaron *box colliders* de este modo nos permitirá tener un control de los elementos que interaccionan entre sí pudiendo luego, por medio del script, de saber si se pisa el suelo, si se hace daño al enemigo (o viceversa) ...

El control de cambio de estados, y por ende que determina que una mecánica sea la ejecutada en ese momento, se hará por medio de diferentes booleans que reacciona al input que realiza el jugador a través de los controles y que son determinados por las reglas implementadas en el script de movimiento.

Para la mecánica de ataque del jugador (y de los enemigos) se ha desarrollado un *collider* que se activara en un momento fotograma específico de la animación. Este *collider* interacciona con otro elemento para desencadenar la mecánica de pérdida de salud implementada.

A parte de los movimientos se han desarrollado mecánicas de RPG de modo que al obtener experiencia nuestro personaje se hace más fuerte y las estadísticas como el ataque (afecta al daño causado) o la defensa (relacionado con el daño recibido).

Con esto se espera que el jugador note una satisfacción de mejora al entrenar a su personaje, igual que sucede cuando uno entrena a un Pokémon para el juego competitivo.

Otras mecánicas implementadas suponen un sistema de inventario para guardar los coleccionables que suelten los enemigos o que se encuentren por la escena. El uso del inventario permite al jugador sus estrategias a la hora de encarar el título.

Enemigos

Los enemigos componen el mayor número de assets usados ya que cada zona que compone Idraldor está representado por gran variedad de enemigos diferentes cuyos niveles de dificultad varían según el jugador se aproxima a la sección final del juego. Los enemigos se encuentran localizados según la temática de la escena

Al igual que sucede con el jugador principal podemos encontrar que los assets están configurados para desarrollar diferentes mecánicas según la inteligencia artificial desarrollada para ellos.

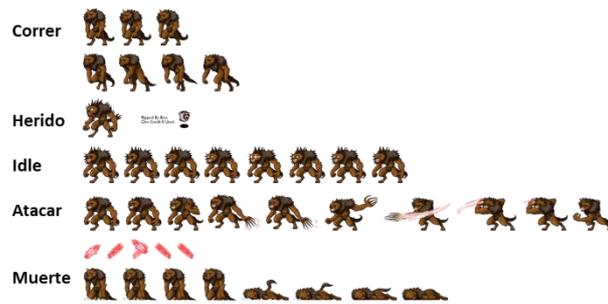


Figura 35: Enemigo B (Hombre lobo)

Al igual que sucede en juegos de este tipo los enemigos pueden clasificarse por categorías/especies y para desarrollarlos se han reciclado assets para crear variantes que difieren en sus estadísticas. Los enemigos podrán ser vistos en la sección del bestiario de los anexos.

La lista de assets usados será desarrollada en la tabla 5. Las mecánicas que controlan su inteligencia artificial serán comentadas en el apartado correspondiente de esta memoria.

La lista de assets usados será desarrollada en la tabla 5. Las mecánicas que controlan su inteligencia artificial serán comentadas en el apartado correspondiente de esta memoria.

Ítems

Elementos que se utilizan para mejorar la experiencia de juego al jugador al considerarse una recompensa al jugador por explorar. Se desarrollarán ítems para que el jugador pueda recuperar alguno de los parámetros que componen parte de la UI como será la vida/mana/experiencia.



Figura 36: Pociones

La recolección de estos objetos como se ha indicado en el apartado del jugador principal tiene implementadas dos mecánicas claramente diferenciadas para que el jugador pueda adaptar su estilo de juego.

Otro de los ítems a recoger es el dinero, se ha implementado una serie de divisa de diferente valor. ¿Y por qué el juego requiere diseñar dinero? ¿Hay algún tipo de compra in game? La respuesta viene heredada de los clásicos, pues que no sería un Castlevania si no hubiera una tienda de ítems/armas...

El dinero está pensado para desarrollar una mecánica de compra-venta de ítems por medio de un menú donde el jugador podrá compra-vender ítems para hacerse con otros.

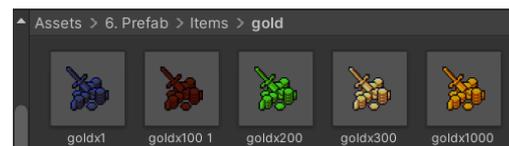


Figura 37: sistema monetario y su valor

De este modo tendrá mayor control sobre el juego para desarrollar sus estrategias. Con la mecánica del dinero se espera que el jugador decida si quiere por ejemplo comprar 2 pociones de vida superiores porque está en un enfrentamiento o comprar un arma que duplique sus parámetros y así hacer más daño.

De igual manera estos objetos podrán obtenerse de los enemigos al ser derrotados, cada enemigo tendrá asociado un sistema de loot en el que al morir dejará caer un objeto de una lista asociada a estos. Así podremos catalogar los objetos que sueltan como comunes, raros y raros míticos dependiendo de la probabilidad de aparición.

4.3.3. Sonidos

El juego dispone pistas de audio entre BSO (Banda Sonora Original) y efectos sonoros en formato mp3. Tanto los efectos sonoros como la BSO presentan una epicidad que anima al jugador a seguir más allá aumentando la inmersión de este en el juego.

Para incorporar los sonidos se ha implementado una clase de AudioManager que será la que gestione los sonidos por medio del script implementado para ello, y que habrá un componente por el UI que permitirá calibrar el volumen de la BSO y de los efectos durante la ejecución del juego.

4.3.4. Fuentes

Se han añadido las fuentes de texto "cretino" [54] y "AvQuest" [55], que la fuente utilizada para los títulos, la cual es una fuente con una forma más específica para la ambientación medieval gótica que se pretende con la escenografía del juego.

Para los textos de los diálogos se ha utilizado una fuente estándar, para una mayor legibilidad por parte del jugador, además de permitir técnicas de manipulación de texto que el componente estándar no es capaz de hacer.

Para la mayoría de los textos se ha usado del componente *TextMeshPro* [56], dado que al ser una herramienta más óptima para el desarrollo de UI (user interface) y para los diálogos de los NPC. Este componente debe ser "instalado" dentro del proyecto para poder ser usado en cada uno de los elementos que compongan el juego.

4.3.5. Listado de Assets

Como se ha ido comentando a lo largo de toda la memoria, por motivos de tiempo y falta de recursos para la elaboración de recursos propios ha sido necesario utilizar varios assets

de terceras partes. Estos recursos han sido usados para completar tanto la parte visual como la sonora del proyecto.

A continuación se detallará todos los elementos que componen el juego y donde han sido obtenidos.

Área	Nombre	Autor	Licencia	Enlace
Tilesets	Final Taiga Forest	Anokolisa	CC0 1.0	https://anokolisa.itch.io/mountain-forest-asset-pack
	Gothicvania Swamp files	Ansimuz	CC0 1.0	https://ansimuz.itch.io/gothicvania-swamp
	Gothicvania Rocky Pass Environment	Ansimuz	CC0 1.0	https://ansimuz.itch.io/gothicvania-rocky-pass-environment
	GothicVania Church	Ansimuz	CC0 1.0	https://ansimuz.itch.io/gothicvania-church-pack
	GothicVania Cemetery	Ansimuz	CC0 1.0	https://ansimuz.itch.io/gothicvania-cemetery
	GothicVania Town	Ansimuz	CC0 1.0	https://ansimuz.itch.io/gothicvania-town
	Gothicvania Junk Wasteland	Ansimuz	CC0 1.0	https://ansimuz.itch.io/gothicvania-junk-wasteland-environment
	Dancing Girl Sprites	Ansimuz	CC0 1.0	https://ansimuz.itch.io/dancing-girl-sprites
	GothicVania Patreon Collection	Ansimuz	CC0 1.0	https://ansimuz.itch.io/gothicvania-patreon-collection
	Legacy-Fantasy-VL.1 - Castle Prison	Anokolisa	CC0 1.0	https://anokolisa.itch.io/castle-prison
	Mushroom-grove	Anokolisa	CC0 1.0	https://anokolisa.itch.io/mushroom-grove
	Mobile - Minecraft Earth - Dialog Boxes	SuperPupperDogg	CC0 1.0	https://www.sprites-resource.com/mobile/minecraftearth/sheet/147775/
	gameOverScreen	ND	CC0 1.0	https://www.reddit.com/r/dayz/comments/11oo45/please_keep_the_you_are_dead_screen_in_the/
	ending screen	ND	CC0 1.0	https://www.reddit.com/r/FinalFantasy/comments/bgx9v1/over_100_hours_its_always_sad_seeing_these_games/
	dead-death-game-over-grave-gravestone-marker-passed-peace-801770	ND	CC0 1.0	https://flyclipart.com/dead-death-game-over-grave-gravestone-marker-passed-peace-game-over-clipart-801770
	86-865581_dialogue-assets-speech-bubble-icon-flat-clipart-removebg-preview	ND	CC0 1.0	https://www.pincliptart.com/pindetail/hmbhJh_dialogue-assets-speech-bubble-icon-flat-clipart/
Player	Hero Knight	Sven	CC0 1.0	https://sventhole.itch.io/hero-knight
Enemies	bandits	Sven	CC0 1.0	https://sventhole.itch.io/bandits
	Huntress	luizmelo	CC0 1.0	https://luizmelo.itch.io/huntress
	Medieval-king-pack-2	luizmelo	CC0 1.0	https://luizmelo.itch.io/medieval-king-pack-2

	Monsters Creatures Fantasy	luizmelo	CC0 1.0	https://luizmelo.itch.io/monsters-creatures-fantasy
	Wizard Pack	luizmelo	CC0 1.0	https://luizmelo.itch.io/wizard-pack
	Bringer-of-death-free	Clembod	CC0 1.0	https://clembod.itch.io/bringer-of-death-free
	Fish Head	SmithyGCN	CC0 1.0	https://www.sprisers-resource.com/playstation/cvsotn/sheet/3684/
	Red Skeleton Knight	Sosfiro	CC0 1.0	https://www.sprisers-resource.com/pc_computer/koumajoudensets uiistrangersrequiem/sheet/88790/
	Hammer	SmithyGCN	CC0 1.0	https://www.sprisers-resource.com/playstation/cvsotn/sheet/162710/
	Bat	Sosfiro	CC0 1.0	https://www.sprisers-resource.com/pc_computer/koumajoudensets uiistrangersrequiem/sheet/88715/
	Black Dragon	Maxim	CC0 1.0	https://www.sprisers-resource.com/pc_computer/heroesofmightand magic2/sheet/29291/
	Owl	Sosfiro	CC0 1.0	https://www.sprisers-resource.com/pc_computer/koumajoudensets uiistrangersrequiem/sheet/88743/
	Werewolf	Boo	CC0 1.0	https://www.sprisers-resource.com/pc_computer/maplestory/sheet/24259/
	Plant (Green #2)	kendotlibero	CC0 1.0	https://www.sprisers-resource.com/pc_computer/koumajoudensets uscarletsymphony/sheet/89073/
	Goblin Shaman	Duker	CC0 1.0	https://www.sprisers-resource.com/pc_computer/lostruins/sheet/165696/
	Dark Cloud Fox	Mageker	CC0 1.0	https://www.sprisers-resource.com/pc_computer/maplestory/sheet/21769/
	Bone Halberd	Sosfiro	CC0 1.0	https://www.sprisers-resource.com/pc_computer/koumajoudensets uiistrangersrequiem/sheet/88759/
	Sorcerer	kendotlibero	CC0 1.0	https://www.sprisers-resource.com/pc_computer/koumajoudensets uscarletsymphony/sheet/88963/
	Minotaur	elpegaso7	CC0 1.0	https://www.sprisers-resource.com/playstation/cvsotn/sheet/133089/
	Green Demon	Sosfiro	CC0 1.0	https://www.sprisers-resource.com/pc_computer/koumajoudensets uiistrangersrequiem/sheet/90725/

	Malachi	elpegaso7	CC0 1.0	https://www.spriter-resource.com/playstation/cvsotn/sheet/146762/
	Sword Lord	elpegaso7	CC0 1.0	https://www.spriter-resource.com/playstation/cvsotn/sheet/133951/
Items	fantasy-icons-pack	cheekyinkling	CC0 1.0	https://cheekyinkling.itch.io/shikashis-fantasy-icons-pack
	gold_treasure_icons_16x16	Bonsaiheldin	CC0 1.0	https://opengameart.org/content/gold-treasure-icons-16x16
Magic	Firebomb Magic	arecbarwin	CC0 1.0	https://www.spriter-resource.com/arcade/ghoulsnghosts/sheet/65829/
	Bolt Magic	Magma MK-II	CC0 1.0	https://www.spriter-resource.com/game_boy_advance/cvhod/sheet/52078/
	Fire Magic	Magma MK-II	CC0 1.0	https://www.spriter-resource.com/game_boy_advance/cvhod/sheet/52079/
	Ice Magic	Magma MK-II	CC0 1.0	https://www.spriter-resource.com/game_boy_advance/cvhod/sheet/52080/
	Light 1	Davias	CC0 1.0	https://www.spriter-resource.com/pc_computer/rpgmakervx/sheet/100077/
	Save Point	rasgar	CC0 1.0	https://www.spriter-resource.com/mobile/demonhuntlgnd3/sheet/44060/
Audio	Gold	ND	CC0 1.0	https://mixkit.co/free-sound-effects/discover/gold/
	Potion	ND	CC0 1.0	https://mixkit.co/free-sound-effects/discover/potion/
	Level up	ND	CC0 1.0	https://elements.envato.com/es/level-up-86JQGU2?_ga=2.107207362.1913851292.1647768097-1894744702.1647768097&utm_campaign=elements_mixkit_cs_sfx_search_no_results&utm_medium=referral&utm_source=mixkit
	Enemy Death	ND	CC0 1.0	https://mixkit.co/free-sound-effects/monster/
	Enemy Hitted	ND	CC0 1.0	https://mixkit.co/free-sound-effects/monster/
	Player Death	ND	CC0 1.0	https://opengameart.org/content/hurt-death-sound-effect-for-character
	Player Hitted	ND	CC0 1.0	https://mixkit.co/free-sound-effects/hurt/
	Player Sword	ND	CC0 1.0	https://mixkit.co/free-sound-effects/sword/
	Jump	ND	CC0 1.0	https://pixabay.com/sound-effects/search/jump/

	Castlevania_ Order of Ecclesia OST - Sorrow's Distortion	Michiru Yamane	CC0 1.0	https://www.youtube.com/watch?v=2-lrP9mX4QQ
	Castlevania_ Symphony of the Night - Dracula's Castle	Michiru Yamane	CC0 1.0	https://www.youtube.com/watch?v=dh1qqLbuBww

Tabla 5: Listado de recursos utilizados

4.4. Cámara del juego

La cámara del juego tiene tres características importantes.

- La escena está dividida en pequeños sectores que dejan una sensación de cámara fija.
- Es una cámara que persigue al jugador por el mapa en secciones donde el tamaño de la sección es mayor que la de la cámara.
- La cámara tiene un límite establecido por un *collider*. Con esto se pretende que la cámara se quede acotada de modo que al inicio del juego no se vea lo que hay a la izquierda y lo mismo ocurrirá en los otros tres lados del *collider* que componen la sección.
- Los elementos que se encuentra contrapeados al plano o capa (*layer*) donde se realiza la acción principal generaran un efecto estereoscópico [57] que generara un aspecto más próximo a los juegos 2.5D sin llegar a tener un efecto 3D.

El código de la cámara se describirá en el apartado de scripts.



Figura 38: Mapa del juego visto en secciones de cámara

4.5. Escenas

Para este proyecto se han utilizado 2 escenas principales.

La primera escena corresponde al menú inicial del juego y en ella el jugador podrá iniciar la partida, consultar los controles y ajustar los efectos de sonido.

La otra escena recoge todos los elementos que constituyen el reino de Idraldor y donde transcurrirá la aventura.

4.6. Scripts

La programación del juego se ha llevado a cabo mediante la elaboración de scripts desarrollados en lenguaje C# para el juego. Cada script controla una parte de toda la lógica del proyecto dejando otras al motor de Unity.

En este apartado desglosaremos de manera somera los diferentes scripts que tiene el proyecto.

- **Dialog:** Dialog recoge los diferentes scripts que son utilizados para el desarrollo de las conversaciones entre en jugador y los diferentes NPCs.
- **Effects:** En effects tenemos los scripts desarrollados para generar físicas que no son ofrecidas por el paquete de Unity. Aquí encontraremos el script de Blink el cual ocasiona que al golpear a un enemigo o ser golpeado el sprite del jugador/enemigo cambie a uno antes de volver al sprite por defecto simulando una herida.
- **Enemies:** Dentro de los códigos relacionados con los enemigos podremos encontrar los scripts agrupados dentro del tipo de enemigo al que representan ya sean terrestres, aéreos o los jefes que tienen su propia IA. En el apartado de inteligencia artificial (4.8) se amplía la descripción de las mecánicas codificadas en estos scripts.
- **Items:** En ítems tendremos los scripts que recogen el comportamiento de las pociones, el dinero y el uso de los ítems para el inventario.
- **Player:** La función de los scripts es controlar las mecánicas del personaje principal. Las mecánicas del personaje se harán a varios niveles y regulados por diferentes scripts según sea el objetivo del mismo. Aquí encontraremos:
 - Mecánicas que afectan a las estadísticas y del movimiento del jugador.
 - Mecánicas para la salud y su relación con los demás objetos que puedan repercutir en dichos parámetros.
 - Mecánicas para el uso de magia por del personaje.
 - Mecánicas del progreso de experiencia.
- **Scenes:** En Scenes tenemos los scripts que guardan la lógica de control del flujo de juego y que no dependen en cierta manera del input del jugador durante la partida. Dentro encontraremos scripts para:
 - Control de los menús principales.
 - Control de la cámara del juego.
 - Sistema de pausa y de guardado.
 - Inventario y tienda.
 - Control de los efectos sonoros y de la banda sonora.

4.7. Arquitectura del juego

Mediante el siguiente diagrama de estados, se puede observar el comportamiento que tiene el juego en función de las interacciones que realice el jugador.

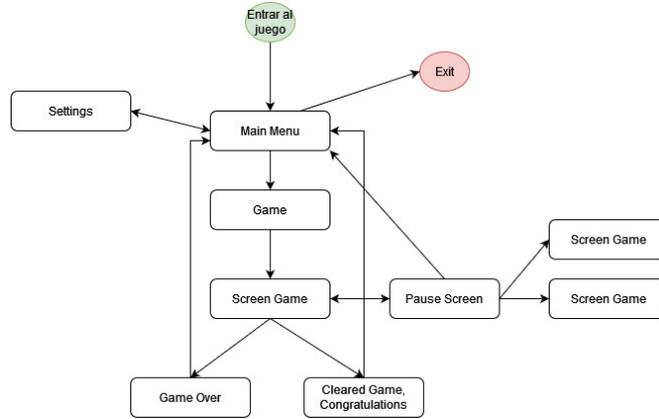


Figura 39: diagrama de estados de Inferno's Court

En la siguiente imagen se puede observar el diagrama que representa a los diferentes assets del proyecto y que por ende conforman el videojuego y la manera en la que estos componentes interaccionan entre sí.

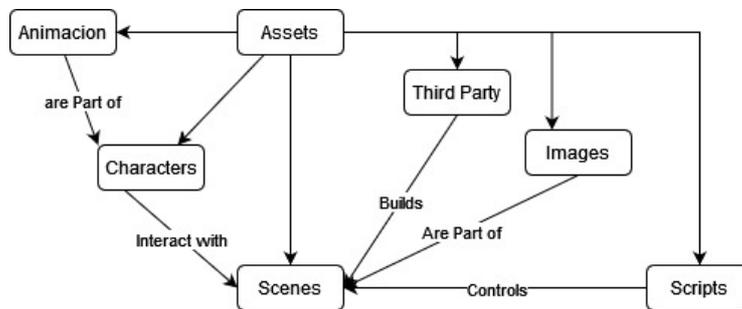


Figura 40: Arquitectura de componentes del juego

4.8. Inteligencia artificial

En Inferno's Court se han implementado varios tipos diferentes de enemigos los cuales presentan un comportamiento basado en una inteligencia artificial (en adelante IA) básico. El que el mecanismo de inteligencia de estos sea básico no supone un detrimento en la jugabilidad sino que esta se encuentra acorde a otros títulos de este género y visto el resultado estamos bastante satisfecho con las opciones que el código implementado ofrece y que puede ofrecer en futuras actualizaciones.

Los enemigos se pueden agrupar entre enemigos terrestres y enemigos aéreos. Dentro a su vez podemos encontrar diferentes variaciones de comportamientos según sea su objetivo.

- Enemigos aéreos
 - Enemigos de patrulla o de desplazamiento fijo (patrulleros)
 - Enemigos vigías (conjuradores)
 - Enemigos con búsqueda activa (acechadores)
- Enemigos terrestres
 - Enemigos fijos
 - Enemigos de patrulla o de desplazamiento fijo (patrulleros)
 - Enemigos vigías (conjuradores)
 - Enemigos con búsqueda activa y ataque (perseguidores)
- Enemigo con mecánica "Némesis"

Todas las variantes de comportamientos que a continuación se describen se agrupan en ocasiones en un mismo script pudiendo al desarrollador poder cambiar de comportamiento y no tener así una masificación de scripts, una para cada tipo de enemigo.

Enemigos de patrulla o de desplazamiento fijo

Este tipo de enemigos se fundamenta en una inteligencia artificial por *WayPoints*, en la que se marcan dos puntos en el mapa sobre los que enemigos se moverán de un punto a otro y vuelta a repetir el ciclo.

Esta mecánica es muy útil para delimitar una zona por donde queremos que se sitúe nuestro enemigo, de modo que podríamos evitar fallos como el de caerse fuera de la plataforma.

Dentro de este tipo se han desarrollado 3 aproximaciones. La primera se basa en el uso de dos *gameObject* que harán de delimitadores en ambos lados de la zona de patrulla. La segunda aproximación se basa en un control por medio de unos *gameObject* que tendrán un registro de los *colliders* del escenario, de activarse la IA cambia la mecánica del enemigo para mantenerse sobre esa plataforma.

Este comportamiento se ha desarrollado tanto para los enemigos terrestres como para los aéreos, la única diferencia entre ellos es el uso de detectores de suelo y de vacío en el terreno, pues estos no están limitados por la gravedad aplicada al entorno.

Enemigos fijos

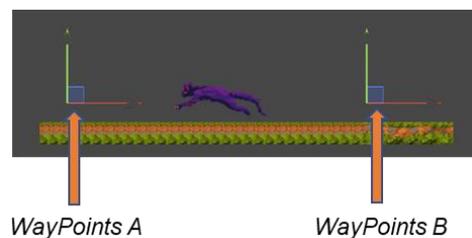


Figura 41: Movimiento entre dos puntos fijos

Se implemento una mecánica para que los enemigos se pudieran quedar parados en el sitio y que tuvieran la animación de Idle (parado) activa en ese momento.

Esta mecánica permitirá establecer enemigos en zonas estratégicamente de modo que pueda suponer un estorbo para el jugador.

Enemigos vigías

Los vigías están controlados por una IA que se caracteriza por asociar un *collider* al enemigo y que al hacer contacto con el jugador ocasionara que se active una mecánica de disparo en la dirección del *collider*.

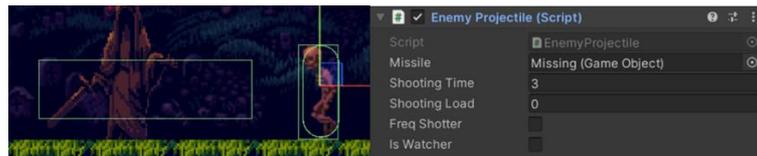


Figura 42: Inspector del vigía y collider de detección del jugador

Dentro de esta modalidad podemos encontrar 2 mecánicas, la primera es de disparo continuo y la segunda es la anteriormente descrita. Con esta mecánica se consigue se sea más difícil al jugador poder moverse por la escena forzarle a que varíe su estrategia. Las mecánicas de IA pueden variarse desde script diseñado específicamente para la mecánica de disparo.

Enemigos con búsqueda activa

Los enemigos adheridos a esta clase presentan una mecánica de movimiento/persecución del jugador cuando este entra dentro de un área delimitada que actúa como detonante o *trigger*.

Mientras el jugador no se encuentre dentro de esta área, el enemigo se encontrará en una posición estática a la espera (animación Idle) que será considerada como el punto de partida (starting point), cuando se active el trigger, este enemigo cambiara de estado (walk) y se dirigirá hacia el jugador.

Este comportamiento será diseñado para enemigos voladores principalmente como podrían ser murciélagos, fantasmas... de modo que puedan volver a una posición de espera en una zona donde el jugador de por si no pudiera acceder.

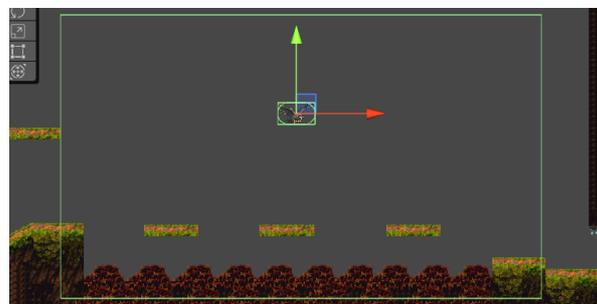


Figura 43: Enemigo a la espera del jugador dentro de su rango de movimientos

Enemigos con búsqueda activa y ataque

Estos enemigos presentan una mecánica de ataque al estar próximo al jugador. A diferencia del enemigo anteriormente descrito, la detección del jugador no se basa en un área delimitada sino en una variable que calcula la diferencia de espacio respecto al jugador (detection range).

Por este motivo, mientras que la distancia de detección sea mayor, el enemigo estará en una posición de espera, pero al detectar al jugador cambiara de estado para iniciar la mecánica de persecución.

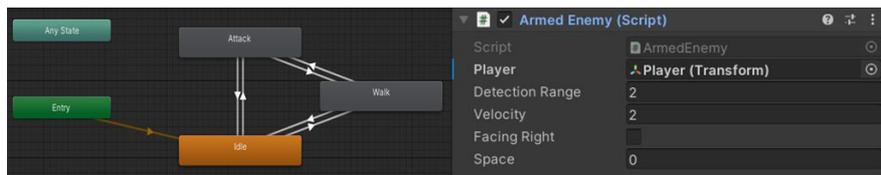


Figura 44: IA de los enemigos con animación de ataque

Esta IA se ha diseñado especialmente para enemigos terrestres y armados. Es por ello que estos enemigos detectaran al jugador cuando este más próximo a ellos, simulando que son detectados por un contacto visual entre ambos.

Enemigo con mecánica Némesis

Este enemigo presenta una IA que va un paso más a los dos descritos anteriormente en términos de persecución. En las anteriores el jugador podía eludir la persecución alejándose de los perseguidores pero con el sistema Némesis esta posibilidad queda descartada.

Esta mecánica no debe ser confundida con la desarrollada para TLOTR: Shadow of Mordor (Monolith Productions, 2014) aunque tengan el mismo nombre [58].

Este enemigo en si debe suscitar una amenaza constante al jugador pues se ha implementado con la idea de que ocasione un *instant kill* (muerte instantánea) al colisionar con el jugador dado que sus estadísticas son demasiado elevadas.

De este modo actuara de modo parecido al que tenía Némesis en el juego de Resident Evil 3 [59] donde a lo largo del juego ponía al jugador en la tesitura de tener que elegir entre huir y/o luchar (con premio tras superarlo).

Esta mecánica ha sido implementada de novo para que el jugador tenga que estar consciente en todo momento de la escena pues "Némesis" puede aparecer en cualquier momento y va directo hacia el jugador.

La mecánica ha sido desarrollada en primera instancia usando como modelo el algoritmo para Unity A* pathfinding [60] para

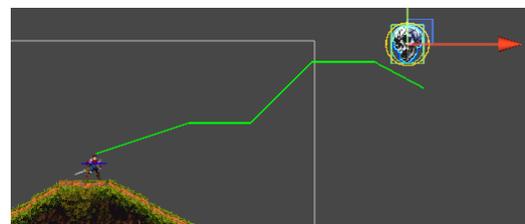


Figura 45: Calculo de la trayectoria en tiempo real hacia el jugador

comprobar la viabilidad de la misma y su posible aplicación, aunque posteriormente fue desarrollado un script que reproduce dicho comportamiento.

El objeto que tenga la mecánica implementada determinara la ruta más corta por el escenario, representada por la línea verde que los une y la variara dependiendo de la posición del jugador.

5. Diseño de niveles

Actualmente Inferno's Court se componen de las siguientes pantallas: Menú inicial, juego, GameOver y Final.

Cada pantalla permite al usuario navegar entre las opciones que ofrece, aunque como resulta obvio el mayor énfasis se sitúa en la pantalla de juego donde tendrá lugar una acción frenética y re jugable.

5.1. Diseño del mundo

5.1.1. El último poblado

El nivel introductorio tiene como objetivo servir como un tutorial donde se explicarán los controles y las mecánicas principales del personaje para que el jugador se familiarice y pueda presentar batalla en los siguientes escenarios. El mapa está separado en dos secciones, la superior es donde los NPC presentaran las mecánicas y una parte baja donde aguardan la primera prueba de fuego al jugador.

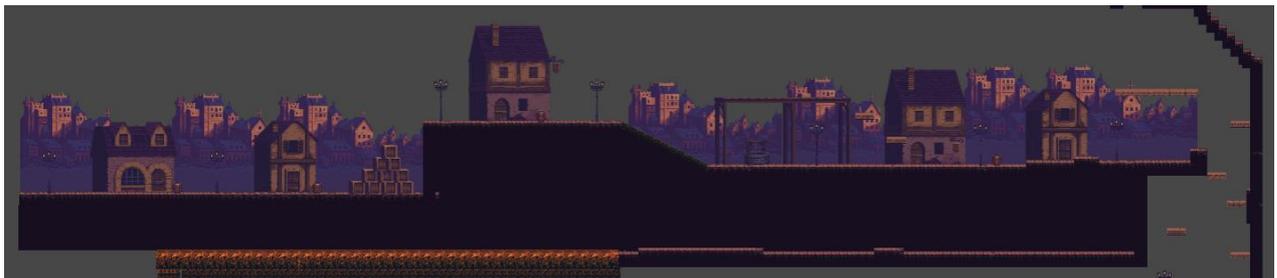


Figura 46: Zona del pueblo donde se realiza el tutorial de las mecánicas.

5.1.2. Bosque

La primera sección fuera de la seguridad del pueblo es un bosque lúgubre y un nivel más extenso donde al jugador se le pondrá a prueba de las habilidades y mecánicas introducidas en el poblado. El jugador tendrá que superar varias zonas como la guarida de los bandidos, el nido de goblins y territorio lycan, pero el peligro no vendrá únicamente por medio de los enemigos, el bosque estará también en contra del jugador (ver anexos adicionales).

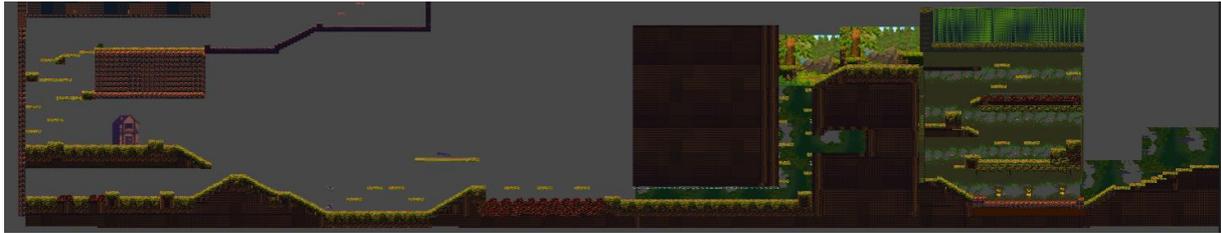


Figura 47: Fase del bosque

5.1.3. Cementerio

El cementerio es una extensión más vertical que el bosque y que además recoge partes de un escenario descartado, las cuevas, al cual se le ha dado una pequeña ambientación. En este escenario el nivel de dificultad no decae presentando al primer jefe de nivel del juego. En este escenario debido a la verticalidad el jugador tendrá que medir sus pasos para poder salir del cementerio y no salir lastimado (ver anexos adicionales).

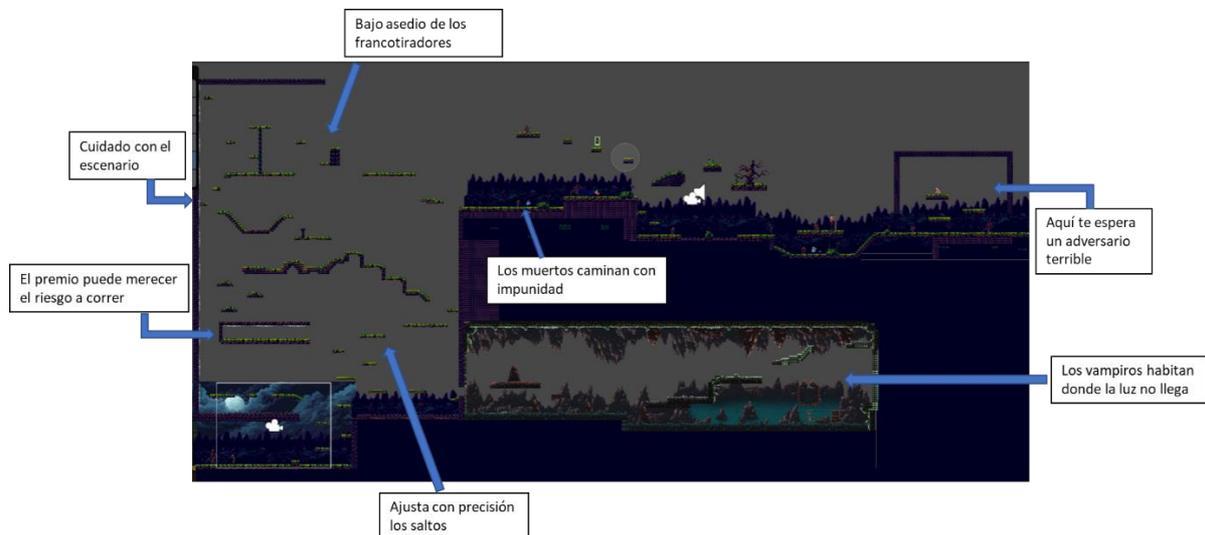


Figura 48: Fase del cementerio con pistas al jugador

5.1.4. Castillo

En el último nivel, pese a no contener ningún elemento ni peligro nuevo la dificultad de los enemigos aumenta al encontrarte cerca del jefe final del juego, Armagrog. El jugador deberá poner en práctica todo lo aprendido durante el transcurso del juego ya sea sortear ríos de lava, furiosos centinelas, un mal que ansía el oro con rabia... hasta encontrar la luz sagrada en lo más alto del castillo (ver anexos adicionales).



Figura 49: Fase del castillo, detrás del trono te espera tu destino.

5.2. Diseño de combate

El diseño de combate del juego es sencillo pues únicamente se fundamenta en el uso de dos botones de modo que no hay necesidad de hacer combos ni combinaciones de botones.

Se ha mantenido así de sencillo para que el jugador pudiera centrarse en poder atacar de manera agresiva recomendado para aquellos que les gustan los hack and slash.

Derrotar enemigos, junto con la obtención de ciertas pociones, propicia que nuestro personaje suba de nivel siendo recompensado con un aumento de sus parámetros (vida, fuerza, defensa...).

El cálculo de subida de nivel y de parámetros se encuentra recogido en el archivo de estadísticas.

5.3. Instrucciones de instalación

Para jugar a Inferno's Court no es necesario realizar ninguna instalación, ni requiere de software adicional para su ejecución.

6. Implementación y Demostración

6.1. Requisitos mínimos

Este juego no requiere de una instalación permanente en el equipo. Los requerimientos básicos para jugar al juego son los siguientes:

- Versión Windows
- SO: Windows 7
- Versión: Sistema operativo de 64 bits.
- CPU: Intel® core™ duo T5450 1.66 GHz
- Gráfica: NVIDIA GeForce 8600 GT (2007)
- Ram: 2 GB

La resolución de pantalla debe ser 1920x1080. Resoluciones superiores provocan errores en el canvas del menú de pausa al descuadrar los elementos de pantalla.

6.2. Formas de iniciar el juego

La manera de ejecutar el juego en su versión Windows es descomprimir la carpeta del proyecto en nuestro escritorio o en la carpeta que el jugador desee. Una vez realizado este paso y dentro de la carpeta se deberá ejecutar el archivo Inferno's Court.exe.

La segunda forma de iniciar el juego será para aquellos usuarios que deseen disfrutarlo a través de un navegador web por medio de la versión WebGL [70]. Por este motivo la versión final del juego fue publicada en una game jams [71] y debido a los paquetes instalados para la realización de los textos y de la IA no se pudo publicar vía explorador en la página de itch.io sin embargo se pudo subir el proyecto para ser jugado vía web en la plataforma simmer.io.

6.3. Instrucciones de uso

6.3.1. Menús

Nada más iniciar el juego encontramos el menú inicial del cual podremos acceder a los menús de configuración de audio y a un menú explicando los controles. Dentro del juego se podrá acceder al menú pausa pulsando la tecla de "Esc" desde la cual podremos navegar hacia el menú de audio y de vuelta al menú principal.



Figura 50: Navegabilidad entre los diferentes menus del juego (pantalla de pausa no definitiva)

6.3.2. Controles

- Movimiento: podremos usar tanto las teclas WASD como las teclas direccionales.
- Ataque: podremos usar tanto Ctrl izquierda como el botón izquierdo del ratón.
- Magia: podremos usar tanto Alt izquierda como botón derecho del ratón. Para la versión WebGL se sustituyó por la tecla "x" dado que alt se utiliza para los atajos de teclado y podría romper una ruptura con la dinámica de juego.
- Saltar: podremos usar la tecla de espacio.
- Interaccionar: podremos interactuar con NPC con la tecla shift izquierda.

6.4. Prototipos

6.4.1. Prototipos Lo-Fi

Los sketches de las mecánicas a implementar durante el desarrollo del proyecto se pueden encontrar en la sección 3.1.6 Concept arts de este documento y en los anexos adicionales.

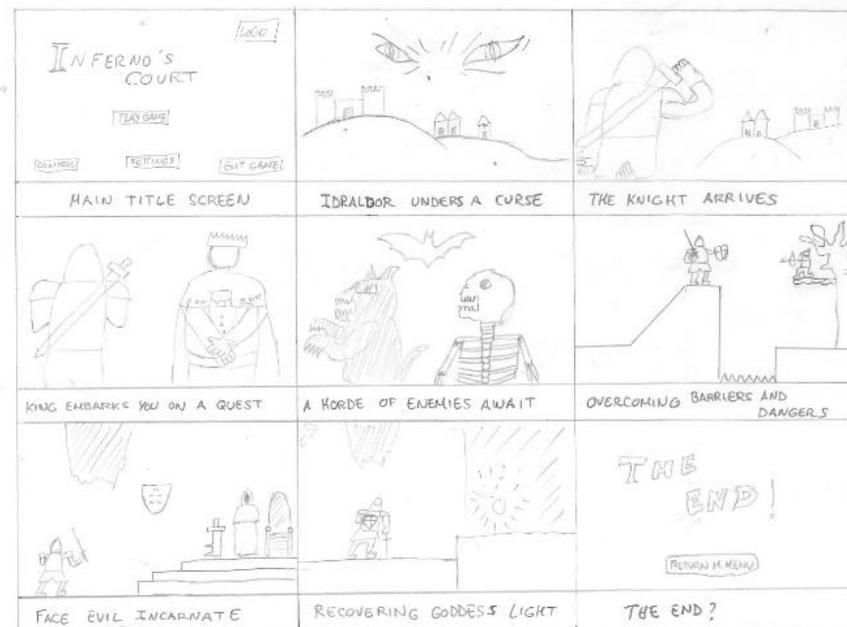


Figura 51: StoryBoard del argumento del juego

6.4.2. Prototipos Hi-Fi

Se han realizado unas series de prototipos funcionales a lo largo del proyecto y que han coincidido con las diferentes fases del proyecto coincidentes con los entregables del mismo. Igualmente algunos de estos prototipos han sido publicados en páginas especializadas como itch.io. Las imágenes del progreso del juego serán entregado en un documento aparte (ver documento adicional anexos).

Versión Alpha (PEC 2)

Primera versión funcional del mismo. Aunque inestable, a menudo es la primera versión a la que los testers o probadores del juego tiene acceso para reportar errores. Los errores reportados se recogen en la sección de test.

Versión Beta (PEC 3)

Primera versión prácticamente completa de un videojuego, y generalmente la última versión de prueba antes de la final que será comercializada. En este caso se considerará una versión prácticamente completa en lugar de una completa dado el tiempo de desarrollo, así mismo y debido a la temporización del trabajo y a la disponibilidad de los testers los resultados de la versión Alpha son recogidos y analizados en esta fase del trabajo.

ChangeLog from PEC 2:

- Desarrollo del resto de partes que componen el juego: Pueblo, bosque, y el castillo.
- Desarrollo de enemigos de búsqueda activa, estos perseguirán al Player cuando entre en su campo de detección.
- Nuevos tipos de enemigos tematizados con las áreas a las que pertenecen.
- Los enemigos tienen efecto de daño indicando la vida perdida al recibir un golpe.
- El pueblo sirve como tutorial de los controles. Los lugareños darán información al Player por medio de cuadros de teto interactivos.
- Desarrollo de nuevos menús tanto iniciales como de pausa.
- Se desarrolla un prototipo de inventario.
- Se añade props para dar ambiente en las escenas, por ejemplo en la guarida de los goblins o del dragón.
- Se elimina los corazones de la interfaz del juego. Los corazones servían para tener un control del uso de armas secundarias por parte de nuestro personaje. Se sustituye por un sistema de magia.

Versión Candidata Gold (PEC 4)

Versión final de un videojuego a la que se le han aplicado todos los cambios y modificaciones resultantes de la versión beta, y que por tanto es candidata a ser la definitiva que será comercializada (gold). Se considera que es la versión Gold cuando el desarrollo ya ha alcanzado la versión definitiva del mismo, lista para ser producida en masa.

Respecto a la planificación del proyecto en esta fase será cuando el feedback de los testers será implementado.

ChangeLog from PEC 3:

- Reajuste de las estadísticas de personajes, enemigos y objetos.
- Pulido de bugs en el escenario.
- Mejora del sistema de cámara.
- Sistema de loot de objetos al derrotar enemigos.
- Sistema de inventario con tienda.
- Versión WebGL implementada en simmer.io.

6.5. Test

Al finalizar la PEC2 se realizó una beta del juego donde se enviaron invitaciones para probar el juego junto a un cuestionario (ver anexos) donde se recopilaban los datos de los usuarios que participaron y cuyos resultados se han publicado en este informe de manera

anónima por medio de la ley de protección de datos publicada en el BOE-A-2018-16673 Ley Orgánica 3/2018 [68].

La prueba consistió en probar el juego sin limitación de tiempo durante el desarrollo de este trabajo dando acceso a los encuestados a la carpeta del repositorio del juego de modo que pudieran descargárselo y jugar.

A continuación, se analizarán los resultados correspondientes al juego que aquí compete (sección 3 del cuestionario). La nota media del juego fue de 7.8 siendo la nota predominante el 8. Los resultados muestran que los encuestados estuvieron contentos con la prueba del juego, a pesar visto en los datos de demografía y afición que son usuario más “casuals” al no jugar con frecuencia. Esto refuerza la idea presentada en el análisis DAFO que los juegos de este género están teniendo una segunda juventud y que su estilo de juego todavía es demandado en el mercado.

En el análisis se solicitó a los usuarios/stakeholders que señalaran tanto los puntos fuertes como los débiles además de preguntarles que mecánica/opción echan en falta para que así el juego fuera más atractivo para sus intereses.

En referencia a lo que más les gusta en referencia al juego encontramos algunas opiniones destacadas. Varios opinan que el estilo artístico elegido es lo que más les ha gustado pues la ambientación recuerda a los juegos de los años 90 cuando muchos de los encuestados (incluido un servidor) éramos niños y jugábamos en las consolas de 8 o 16 bits.

Otros encuestados destacan que la dificultad está bien ajustada pues supone un reto al jugador según van apareciendo nuevos tipos de enemigos según se progresa en la historia.

En colación al desarrollo de la historia del punto anterior a muchos les gusta que el personaje tuviera un sistema de progreso por medio del sistema de niveles implementado para hacer al héroe más fuerte según avanza la historia dando la sensación de que tanto el jugador como su avatar “maduran” a la vez, reforzando así la inmersión del jugador.

Por último destacan que el juego presenta una accesibilidad para cualquier perfil de jugadores independientemente de su edad o de la experiencia con videojuegos que haya tenido al permitir el control del personaje únicamente con dos botones (análogo al control de NES).

En cuanto a lo que no les gusta del juego, destacaron que el nivel de dificultad era un poco abusivo al haber incluido un sistema de “instant kill” en el escenario dado que si el personaje colisionaba con ciertos elementos de la escena (no enemigos) este moría automáticamente. Esta mecánica se sustituyó primero ajustando el daño que producía y luego por el sistema némesis dando así al jugador más tranquilidad respecto al plataformeo pero sin perder de vista la tensión por un final abrupto.

También hubo quejas respecto a la velocidad del avatar lo que hacía que fuera difícil de controlar y por tanto provocaba que el jugador muriese con mayor frecuencia. Se reajustó la velocidad base del personaje para que tuviera un movimiento más pausado y controlable.

Por último, los encuestados sintieron la falta de una historia y unos objetivos claros dentro del juego. Esta cuestión era asumible pues la demo se centraba en una sección intermedia del mapa sin ningún tipo de contexto, mientras que en la versión final nada más comenzar una serie de NPCs te darán información sobre el trasfondo del juego y de los objetivos a cumplir.

Dentro de las recomendaciones sugeridas se encuentran las siguientes:

- Sistema Loot: el sistema loot permite que los enemigos suelten objetos de manera aleatoria los cuales podrán ser usados en el inventario o en la tienda.
- Mas opciones de control para el personaje: Se sugería que el personaje pudiera realizar mecánicas típicas de juegos como Dark souls, por medio de uso de magias, bloques, esquivas... Todo esto podría suponer que la vertiente del juego se alejaría de la perspectiva más clásica del género para adaptarse a lo que el público actualmente está solicitando [69].

En este sentido alguna de las solicitudes se ha implementado como el uso de otro botón para la realización de una magia que permite ser usada como ataque o defensa.

- Mas personajes jugables: La inclusión de otros personajes jugables ha sido estudiada durante todo el desarrollo del juego. Para ello se valoró la posibilidad de iniciar el juego desde el inicio con cualquiera de los personajes como ocurría en cualquiera de los beat 'em up de los 90 como en Cadillacs and dinosaurs.

Esta idea se descartó pues habría que realizar diferentes mecánicas para cada uno de ellos y que además se produjeran cambios en todo el juego ya fuera en la escena, en los menus de inicio...

En su lugar se pensó la posibilidad de intercambiar de personaje en cualquier momento del juego de modo que el jugador pudiera elegir entre uno u otro. La idea era que hubiera un caballero con ataque de tipo melee y un mago con ataque de rango, cada uno tendría sus estadísticas diferenciadas.

Al igual que en la opción anterior supondría un lavado completo de todo el proyecto de modo que se descartó su implementación. Sin embargo, la segunda opción es más plausible de realizarse y se estipula como objetivo a futuro (ver sección 7.2).

6.6. Ejemplos de uso del producto (o manual de usuario)

El manual de usuario será entregado en un documento aparte junto con el resto de entregables.

7. Conclusiones y líneas de futuro

7.1. Conclusiones

En este punto de la memoria es momento de reflejar las conclusiones alcanzadas durante el desarrollo de *Inferno's Court* para este trabajo de final de grado.

Como se indicó en la introducción, una de mis pasiones desde que era pequeño ha sido el mundo de los videojuegos. Durante todo el grado tenía la idea de realizar un videojuego como trabajo final, al igual que otros proyectos de otras áreas y que con los conocimientos adquiridos durante el grado espero que se vean realizados con el tiempo.

Cuando piensa uno en videojuegos se imagina realizar el próximo juego de la saga Zelda o GTA, pero hay que remarcar que ese trabajo tiene detrás cientos de trabajadores que son capaces de llevarlos a cabo y que el que suscribe es un equipo de una persona, lo cual es más parecido a lo que vivió Ron Gilbert cuando desarrollo *The secret of Monkey Island* allá por 1990.

La única diferencia entre Ron y un servidor era que él ya había participado en títulos de LucasArts y tenía experiencia con las herramientas de desarrollo (incluso creo el motor SCUMM) yo por mi parte tuve que adentrarme en un área desconocida pero que con el tiempo me siento satisfecho con el aprendizaje adquirido.

En primer lugar, la satisfacción que produce tener una versión final de una idea y que se haya podido ver materializada pues aunque siempre es mejorable (ya hablaremos de ello en el siguiente apartado) el resultado ha quedado prácticamente igual al establecido durante los *concepts arts*. Crear un video juego desde cero es una tarea difícil, hay que pensar en el destinatario y desarrollar un método MDA que permita un grado de satisfacción en el jugador y que se ajuste a la propia visión del programador. De modo que el reto propuesto ha sido grande ya bien por los conocimientos de partida como de la programación establecida.

En segundo lugar, y esto se compagina con mi línea de trabajo habitual como investigador en biotecnología, se trata de tener en cuenta una planificación de proyecto inicial donde se recogían los objetivos e hitos del proyecto que siguieran el modelo SMART (MARTE en español) para poder ofrecer un producto de calidad al consumidor final dentro del marco teórico establecido en el cronograma.

Tener en mente un cronograma facilita la elaboración de los objetivos para realizar una correcta estimación de los tiempos para poder ofrecer, bien es sabido que en la situación actual de la industria muchos juegos son lanzados "antes de tiempo" y ya desde el día 1 requieren de un parche para solucionar problemas o que por la apretada agenda tienen que sacar el contenido previsto desde el inicio del proyecto en forma de DLCs o actualizaciones dando al usuario la sensación de que el juego lo sacaron a medias.

Por eso es importante estipular dentro los objetivos y delimitar que ofrecer al público, si prometerles la luna o darles una experiencia innovadora. Para ello hay que considerar desde las primeras fases

que el modelo MDA debe orientar el diseño del juego para ofrecer una experiencia única centrándose en el desarrollo en las mecánicas y dinámicas para alcanzar la estética deseada.

En tercer lugar, el uso de Unity en un principio, que era una herramienta desconocida hasta ese momento, además de que conocía de su existencia por la entradas de los juegos al indicar bajo que motor habían sido realizadas y pese a tener una curva de aprendizaje baja, supuso el desconocimiento por mi parte de las diferentes herramientas que podrían ser necesarias para la elaboración del proyecto, si bien es cierto que habrá mecánicas que deban ser pulidas en posteriores versiones/actualizaciones del juego he quedado muy satisfecho con lo aprendido con el motor y deseoso de poder abarcar en nuevos proyectos.

Junto al uso de Unity decir que la elaboración de los scripts en C# me ha servido para profundizar más en este lenguaje y ver que las funciones y métodos desarrollados en estos tenían un reflejo en Unity y que podía manipularlo para una mayor depuración del juego me sirvió como guía y para comprobar que el juego estaba yendo por el camino deseado.

Por último, decir que el proceso de creación de un videojuego es mucho más complicado y laborioso de lo que en una primera aproximación puede ser, incluso hay tutoriales en YouTube que te ofrecen aprender a crear videojuegos desde cero en un par de horas, pero tras haber finalizado el proyecto puedo determinar que solo rasgan la superficie de lo que realmente un programador amateur puede hacer con su equipo y mucha imaginación.

En este proceso hay que tener en consideración muchos factores como el guion, las mecánicas, diseño de los diferentes componentes gráficos y sonoros... y hay que hacer que todo se coordine y que no haya ningún tipo de *crasheo* que entorpezca la experiencia. Es preferible que se tenga una escena de juego más pequeña y que todos los elementos tengan sentido que intentar recrear un mapa del tamaño de *Zelda Breath of the wild* y pecar de ambicioso. En definitiva, un gran reto para una sola persona y para el poco tiempo del que se ha tenido para elaborarlo.

Inferno's Court como producto final, y siendo un servidor un programador novel podría ser el punto de partida para la elaboración de nuevos proyectos en el horizonte. Estoy bastante satisfecho del progreso entrega a entrega del juego dando vida a una idea que a principios de semestre pensaba que sería una epopeya digna de una trilogía de películas.

En el plan personal, desarrollar el juego y rendir homenaje a la saga Castlevania ha sido una delicia y aprender a ver el negocio de los videojuegos desde otra perspectiva que me servirá para valorar los juegos pasados y venideros.

Reflexión crítica de los objetivos conseguidos vs planteados

En el capítulo 1 de esta memoria se establecieron los objetivos principales y secundarios del proyecto. En cuanto a los objetivos principales considero que se han cumplido satisfactoriamente. El producto final es un juego de acción con inicio y final claramente delimitados.

Dentro de los objetivos era el desarrollo de un juego con clara inspiración a los clásicos Castlevania para ello se tenía que elaborar una estética en conjunto que homenajeara a estos juegos y consideramos que esto se ha conseguido con la versión final del juego.

La gestión de proyecto se ha llevado a cabo de la manera adecuada cumpliendo los objetivos y los hitos propuestos en el cronograma, en ciertos puntos del proyecto se realizaron pequeños periodos de crunch para poder proporcionar un mayor contenido al juego.

En referencia a los objetivos secundarios también podemos considerar que han sido cumplidos en su mayoría. Aunque fue necesario estar inscrito en dos plataformas (itch y simmer) para poder desarrollar la versión WebGL, dado que nos hubiera gustado que el proyecto estuviera todo agrupado dentro del mismo sistema.

Se partía de una historia que podría ser genérica pero que en la mayoría de los juegos cumple con la premisa, además de este modo se cumple la 1ª regla del decálogo de buenas prácticas del diseño de juegos de aventura al indicar al jugador cual es la misión que tiene que cumplir.

Quizá dentro de los objetivos propuestos se pecó de ambicioso al querer construir un mundo con secciones claramente definidas por una temática donde hubiera enemigos típicos de esos hábitats pero se prefirió por dejar pinceladas de esas secciones dentro de otras más grandes, por ejemplo en el cementerio hay una cueva que hace referencia a toda una sección. Esto se decidió cambiarlo a este formato al ver que los escenarios (otro personaje más de la historia) requería una planificación mucho más elaborada.

También se podría haber realizado el juego con un número menor de enemigos en lugar de todo el plantel presentado en los anexos pero el desarrollo de las diferentes IA de estos propiciaba, junto con el tiempo, que pudieran crearse nuevos prefabs de enemigos para aplicarles dichos algoritmos.

En cuanto a los personajes, nos hubiera gustado que tuvieran un control más fluido y que en ocasiones su relación con el escenario fuera más orgánica. También, me habría podido introducir nuevas animaciones para los enemigos y el jugador para que pudieran tener un abanico más amplio de movimientos y que los primeros reaccionaran con diferentes estrategias al jugador.

Heos aquí, igual que en las grandes historias, haciendo un balance final del proyecto con cierta retrospectiva crítica cuando se puede decir que se ha conseguido cumplir todos los objetivos propuestos, además de alguno que surgió en el proceso ya fueran académicos o personales.

Seguimiento de la planificación y metodología

Si nos atenemos a la planificación propuesta en el cronograma, decir que se ha intentado seguir los plazos ahí descritos con la mayor rigurosidad.

En líneas generales los plazos e hitos impuestos han sido completados, aunque en algunas fases del mismo como se ha comentado anteriormente se adelantaron tareas programadas correspondientes a otras PECs o incluso se varió ligeramente el desarrollo de las tareas según comodidad del marco temporal donde se realizaron.

Por ejemplo, en una primera instancia el desarrollo de enemigos habría tenido lugar en dos fases pero al desarrollarlos se decidió realizar un crunch para poder seguir creando nuevos enemigos en lugar de pasar a otra tarea.

Alguna tarea ha podido llevar más tiempo del establecido, por ejemplo, los reajustes de los escenarios, y modificadas desde la idea original, para que tuvieran consistencia con las mecánicas introducidas tanto en el personaje principal como en los enemigos.

Por estos motivos, podemos asegurar que la planificación no se ha seguido al 100% con la prevista en el cronograma adjunto en el capítulo 1 de esta memoria. Sin embargo, de lo que sí se puede garantizar es de haberse cumplido los plazos establecidos con el material necesario requerido para cada evaluación.

7.2. Líneas de futuro

Como se ha comentado anteriormente, los plazos condicionan mucho la visión del desarrollador sobre lo que quiere introducir y lo que realmente va a poder introducir en la versión gold. Por este motivo, en esta sección se enumerarán los diferentes componentes y/o mecánicas que habría hecho de Inferno's Court un juego más redondo y satisfactorio tanto para el desarrollador como para el usuario entre las que podríamos citar.

- Versión Android.
- Sistema de habilidades.
- Sistema de inventariado/equipamiento mejorado.
- Sistema de misiones secundarias (SideQuest) ofrecidas por los NPCs del pueblo favoreciendo así el backtracking.
- Mejora de los escenarios: Crear nuevas secciones y dotarlas de más elementos artísticos.
- Ampliación del bestiario: nuevos niveles (ver punto anterior) requiere de nuevos enemigos, ya sean variantes más fuertes de unos establecidos o enemigos desarrollados de novo. En este apartado se incluyen nuevas mecánicas para dotar de mayor variedad a los enemigos.
- Nuevos personajes jugables: Se espera añadir nuevos personajes, desde la pantalla de selección o intercambiable, que ayuden al caballero a libra las tierras de Idraldor de las manos de Armagrog. Por ejemplo, dado que el caballero es de tipo melee se añadirá un hechicero con ataques de media y larga distancia (tipo shooter) para así añadir un componente estratégico pudiendo intercambiar entre ellos según convenga al jugador dependiendo de la situación.
- Cinemáticas: desarrollarlas para darle al juego un tono más de película, las cuales irían como mínimo al inicio/final del juego o durante la pantalla de GameOver. Las animaciones in game se usarían para ampliar el lore del juego y darle más profundidad a la historia.
- Para finalizar, hay que comentar que durante el desarrollo de este trabajo se ha disfrutado mucho programándolo y que se seguirá desarrollando videojuegos de manera amateur

variando de género y estética (3D) y el motor de juego, probando tal vez la potencia de Unreal Engine. Seguramente el siguiente será un RPG por turnos elaborando una historia propia y que sea un reto para el jugador.

- Sistema de guardado de inventario: sistema para guardar los objetos del inventario de modo que al morir los mantenga y no se pierdan.
- Compatibilidad con otras resoluciones de pantalla.

Bibliografía

- [1] Kwak H, Blackburn J. Linguistic analysis of toxic behavior in an online video game [Internet]. arXiv [cs.SI]. 2014. Disponible en: <http://arxiv.org/abs/1410.5185>
- [2] Nintendatos. Curiosidades de Zelda: Ocarina of Time | Nintendatos [Internet]. Youtube; 2022 [citado 19 de febrero de 2022]. Disponible en: <https://www.youtube.com/watch?v=32ZRvE2uTmk>
- [3] Frankie MB. El día que el mando de N64 y el primer Mario Party casi le costaron a Nintendo 80 millones de dólares en guantes [Internet]. Vidaextra.com. Vida Extra; 2021 [citado 19 de febrero de 2022]. Disponible en: <https://www.vidaextra.com/casual-games/dia-que-mando-n64-primer-mario-party-casi-le-costaron-a-nintendo-80-millones-dolares-quantes>
- [4] The (not-so) secret origin of D&D [Internet]. Wizards.com. [citado 19 de febrero de 2022]. Disponible en: <https://dnd.wizards.com/articles/features/not-so-secret-origin-dd>
- [5] Bishop y David Caballero S. The Game Kitchen y la narrativa ambigua de Blasphemous [Internet]. Gamereactor. 2018 [citado 19 de febrero de 2022]. Disponible en: <https://www.gamereactor.es/the-game-kitchen-y-la-narrativa-ambigua-de-blasphemous/>
- [6] Guzman HC. Las 7 fases más importantes en el desarrollo de juegos [Internet]. Hektorprofe.net. [citado 19 de febrero de 2022]. Disponible en: <https://docs.hektorprofe.net/escueladevideojuegos/articulos/fases-del-desarrollo-de-videojuegos/>
- [7] Kickstarter.com. [citado 19 de febrero de 2022]. Disponible en: <https://www.kickstarter.com/projects/iga/bloodstained-ritual-of-the-night>
- [8] Cortés J. Top 9 mejores Motores para crear Videojuegos 2D & 3D (Gratis y de Pago) [Internet]. Notodoanimacion.es | noticias, recursos, tutoriales y empleo para Artistas Digitales. 2021 [citado 19 de febrero de 2022]. Disponible en: <https://www.notodoanimacion.es/motores-para-crear-videojuegos/>
- [9] Laguna D. Fans de Castlevania arremeten contra Konami por su proyecto NFT [Internet]. Yahoo Finanzas. 2022 [citado 19 de febrero de 2022]. Disponible en: https://es.finance.yahoo.com/noticias/fans-castlevania-arremeten-konami-proyecto-153400182.html?quce_referrer=ahr0chm6ly93d3cuz29vz2xllmvzlw&quce_referrer_sig=agaaakexks-ixiik5w5fst545kzz6vtk2xdiugbfncuuykb9ril_j8e2jhvmc_cuudyp4vnjpcq2gk42q9fw_3zux2-vu7e-qbooahecle7w9nx9edptdu4brh_sv_ippjadrrgun4bsc8hhgmkilcjmnomas8otwxwlphe8e7clqzk
- [10] Ferreiro D. ¿Qué es un NTF, cuál es su valor y por qué se están poniendo de moda? [Internet]. COPE.es. 2021 [citado 19 de febrero de 2022]. Disponible en: https://accesible.cope.es/actualidad/tecnologia/noticias/que-ntf-por-que-estan-poniendo-moda-20210415_1239258
- [11] Pressman R. Ingeniería del *software*: Un enfoque práctico, 3.^a Edición, Pag. 26-30
- [12] España N. project OCTOPATH TRAVELER - Resultados de la encuesta sobre la demo (Nintendo Switch) [Internet]. Youtube; 2018 [citado 21 de febrero de 2022]. Disponible en: <https://www.youtube.com/watch?v=jSFHvuYJc3k>
- [13] ¡Descubrid los cambios realizados en TRIANGLE STRATEGY™ a raíz de la encuesta de la versión de prueba! [Internet]. Nintendo of Europe GmbH. [citado 21 de febrero de 2022]. Disponible en: <https://www.nintendo.es/Noticias/2021/septiembre/-Descubrid-los-cambios->

[realizados-en-TRIANGLE-STRATEGY-a-raiz-de-la-encuesta-de-la-version-de-prueba--2047085.html](#)

[14] Torrejón R. La evolución del subgénero Metroidvania: explora, vuelve atrás y redescubre [Internet]. Meristation. 2021 [citado 25 de marzo de 2022]. Disponible en: https://as.com/meristation/2021/05/29/reportajes/1622277939_617548.html

[15] Brodersen J. Nintendo: una historia de innovación que comenzó a fines del siglo XIX [Internet]. Clarín. 2017 [citado 25 de marzo de 2022]. Disponible en: https://www.clarin.com/tecnologia/nintendo-historia-innovacion-comenzo-fines-siglo-xix_0_BkdGLtMIg.html

[16] Davis B. 'castlevania: Symphony of the night': When A franchise switches to "very easy" and creates a masterpiece [Internet]. Collider. 2021 [citado 25 de marzo de 2022]. Disponible en: <https://collider.com/castlevania-sotn-easy-why-is-it-good/>

[17] King W. Matavampiros. Timun Mas; 2004. 2021 [citado 25 de marzo de 2022]. Disponible en: <https://castlevania.fandom.com/es/wiki/Matavampiros>

[18] Masterclass.com. [citado 25 de marzo de 2022]. Disponible en: <https://www.masterclass.com/articles/metroidvania-definition#what-is-a-metroidvania-video-game>

[19] Wikipedia contributors. List of unity games [Internet]. Wikipedia, The Free Encyclopedia. 2022. Disponible en: https://en.wikipedia.org/w/index.php?title=List_of_Unity_games&oldid=1078664395

[20] Short survey about Metroidvania games [Internet]. Google Docs. [citado 25 de marzo de 2022]. Disponible en: <https://docs.google.com/forms/d/1ygHENk5yw997T9t2h8GVBdhufngSz4e0EN7sYr9jFyc/vie/wanalytics>

[21] castlevania en venta - [Internet]. Ebay.es. [citado 25 de marzo de 2022]. Disponible en: https://www.ebay.es/sch/i.html?from=R40&nkw=castlevania&sacat=0&LH_BIN=1&sop=16

[22] Gómez M. Que Elden Ring haya vendido 10 millones de copias en PC parece ser falso: en realidad, el RPG de FromSoftware apunta aún más alto [Internet]. 3DJuegospc.com. 3DJuegos PC; 2022 [citado 25 de marzo de 2022]. Disponible en: <https://www.3djuegospc.com/rpg/que-elden-ring-haya-vendido-10-millones-copias-pc-parece-ser-falso-realidad-rpg-fromsoftware-apunta-alto>

[23] Video game charts, game sales, top sellers, game data [Internet]. VGChartz. [citado 25 de marzo de 2022]. Disponible en: <https://www.vgchartz.com/games/games.php?name=castlevania&keyword=&console=®i>

[on=All&developer=&publisher=&goty_year=&genre=&boxart=Both&banner=Both&ownership=Both&showmultiplat=No&results=50&order=Sales&showtotalsales=0&showtotalsales=1&howpublisher=0&showpublisher=1&showvgchartzscore=0&shownasales=0&showdeveloper=0&showcriticscore=0&showpalsales=0&showreleasedate=0&showreleasedate=1&showuserscore=0&showjapansales=0&showlastupdate=0&showlastupdate=1&showothersales=0&showshipped=0](#)

[24] Wikipedia contributors. Nintendo DS [Internet]. Wikipedia, The Free Encyclopedia.

Disponible en: https://es.wikipedia.org/w/index.php?title=Nintendo_DS&oldid=142379973

[25] Synbioso. Konami deja el futuro de Castlevania en manos de los fans [Internet]. NextN:

¡Actualidad Nintendo! ¡Todas las noticias Nintendo Switch! 2021 [citado 25 de marzo de

2022]. Disponible en: <https://www.nextn.es/2021/09/konami-deja-el-futuro-de-castlevania-en-manos-de-los-fans-o-eso-dice-yota-tsutsumizaki/>

[26] Hernández DA. Dave Cox, productor de Castlevania, abandona Konami [Internet].

Hobbyconsolas. 2014 [citado 25 de marzo de 2022]. Disponible en:

<https://www.hobbyconsolas.com/noticias/dave-cox-productor-castlevania-abandona-konami-83786>

[27] Sandoval C. Productor de Castlevania, Koji Igarashi, abandona Konami [Internet].

Com.mx. Xataka México; 2014 [citado 25 de marzo de 2022]. Disponible en:

<https://www.xataka.com.mx/videojuegos/productor-de-castlevania-abandona-konami>

[28] Konami habla sobre la posibilidad de trabajar en un nuevo Castlevania [Internet].

LevelUp. 2021 [citado 25 de marzo de 2022]. Disponible en:

<https://www.levelup.com/noticias/643888/Konami-habla-sobre-la-posibilidad-de-trabajar-en-un-nuevo-Castlevania>

[29] Díez P. Los creadores de Castlevania: Lords of Shadow cargan contra Konami

[Internet]. SomosXbox. 2016 [citado 25 de marzo de 2022]. Disponible en:

<https://www.somosxbox.com/los-creadores-de-castlevania-lords-of-shadow-cargan-contra-konami/88495>

[30] Álvarez C. Metroid Dread Ventas - Casi 3 Millones de Unidades [Internet].

NextGame.es | Tu web de videojuegos, cine y series. 2022 [citado 25 de marzo de 2022].

Disponible en: <https://www.nextgame.es/nintendo/metroid-dread-ventas/>

[31] Search results for «castlevania» [Internet]. itch.io. [citado 25 de marzo de 2022].

Disponible en: <https://itch.io/search?q=castlevania>

[32] Bonthuys D. 15 best metroidvania games to play in 2022 [Internet]. Gamespot. 2022

[citado 25 de marzo de 2022]. Disponible en:

<https://www.gamespot.com/articles/metroidvania-best-games-like-metroid/1100-6499170/>

- [33] Yuri. Indie-gems [Internet]. <https://indie-gems.com>. 2021 [citado 25 de marzo de 2022]. Disponible en: https://indie-gems.com/upcoming-metroidvania-games-for-2022-beyond-to-look-out-for/?utm_source=rss&utm_medium=rss&utm_campaign=upcoming-metroidvania-games-for-2022-beyond-to-look-out-for
- [34] Malo E. Metroid Prime 4 no sólo se retrasa, se rehará desde cero [Internet]. MuyComputer. 2019 [citado 25 de marzo de 2022]. Disponible en: <https://www.muycomputer.com/2019/01/27/metroid-prime-4-retrasado/>
- [34] Wikipedia contributors. Usage share of operating systems [Internet]. Wikipedia, The Free Encyclopedia. 2022. Disponible en: https://en.wikipedia.org/w/index.php?title=Usage_share_of_operating_systems&oldid=1078806797
- [36] Wikipedia contributors. Pan European Game Information [Internet]. Wikipedia, The Free Encyclopedia. Disponible en: https://es.wikipedia.org/w/index.php?title=Pan_European_Game_Information&oldid=140035426
- [37] Camargo G. Freemium: conoce cómo funciona, sus ventajas y 3 ejemplos prácticos [Internet]. Rock Content - ES. 2021 [citado 25 de marzo de 2022]. Disponible en: <https://rockcontent.com/es/blog/freemium/>
- [38] Jimenez S. La web como un videojuego: el MDA para la Analítica Web - Sergio Jiménez [Internet]. Sergio Jiménez. 2015 [citado 7 de mayo de 2022]. Disponible en: <https://sergiojimenez.net/la-web-como-un-videojuego-el-mda-para-la-analitica-web/>
- [39] gamedesignconcepts / Nivel 5: Mecánicas y Dinámicas [Internet]. Pbworks.com. [citado 7 de mayo de 2022]. Disponible en: <http://gamedesignconcepts.pbworks.com/w/page/11829118/Nivel%252>
- [40] Wiki. Classes [Internet]. Deaths Gambit Wiki. [citado 7 de mayo de 2022]. Disponible en: <https://deathsgambit.wiki.fextralife.com/Classes>
- [41] Metroidvania [Internet]. TV Tropes. [citado 7 de mayo de 2022]. Disponible en: <https://tvtropes.org/pmwiki/pmwiki.php/Main/Metroidvania>
- [42] Hollow knight and metroidvania genre conventions [Internet]. Hexagram.ca. [citado 7 de mayo de 2022]. Disponible en: <https://tag.hexagram.ca/hollow-knight-and-metroidvania-genre-conventions/>
- [43] r/metroidvania - Defining the «Metroidvania» Genre [Internet]. reddit. [citado 7 de mayo de 2022]. Disponible en: https://www.reddit.com/r/metroidvania/comments/95yfrf/defining_the_metroidvania_genre/
- [44] What the difference between Metroid and Metroidvania?.Gamespot.com. [citado 7 de mayo de 2022]. Disponible en: <https://gamefaqs.gamespot.com/boards/323656-metroid-dread/79713208?page=7>
- [45] Izquierdo A. La suspensión de la incredulidad [Internet]. Espinof.com. Espinof; 2012 [citado 7 de mayo de 2022]. Disponible en: <https://www.espinof.com/diccionario-cine-television/la-suspension-de-la-incredulidad>

- [46] Return to Monkey Island [Internet]. Return to Monkey Island | Coming 2022. [citado 7 de mayo de 2022]. Disponible en: <https://returntomonkeyisland.com/>
- [47] Grumpy gamer [Internet]. Grumpy Gamer. [citado 7 de mayo de 2022]. Disponible en: https://grumpygamer.com/why_adventure_games_suck
- [48] Deneschau, N., 2022. Los misterios de Monkey Island. 1st ed. Ciudad Real: GamePress Editorial, pp.56-60.
- [49] Unity Technologies. System requirements for unity 2021 LTS [Internet]. Unity3d.com. [citado 7 de mayo de 2022]. Disponible en: <https://docs.unity3d.com/Manual/system-requirements.html>
- [50] Rawi. Requisitos recomendados de unity: pros y contras 2022 [Internet]. La Fortaleza Gamer. 2021 [citado 7 de mayo de 2022]. Disponible en: <https://www.lafortalezagamer.online/unity-que-es-requisitos-minimos-peso-en-disco-pros-y-contras/>
- [51] Unity asset store - the best assets for game making [Internet]. Unity.com. [citado 7 de mayo de 2022]. Disponible en: <https://assetstore.unity.com/>
- [52] Top free game assets [Internet]. itch.io. [citado 7 de mayo de 2022]. Disponible en: <https://itch.io/game-assets/free>
- [53] The spriters resource [Internet]. Sriters-resource.com. [citado 7 de mayo de 2022]. Disponible en: <https://www.sriters-resource.com/>
- [54] Cretino [Internet]. Dafont.com. [citado 7 de mayo de 2022]. Disponible en: <https://www.dafont.com/es/cretino.font>
- [55] AvQest Regular true type fuente [Internet]. Ttfonets.net. [citado 7 de mayo de 2022]. Disponible en: https://ttfonets.net/es/font/3221_AvQest.htm
- [56] Unity Technologies. TextMeshPro [Internet]. Unity3d.com. [citado 7 de mayo de 2022]. Disponible en: <https://docs.unity3d.com/Manual/com.unity.textmeshpro.html>
- [57] Wikipedia contributors. Estereoscopia [Internet]. Wikipedia, The Free Encyclopedia. Disponible en: <https://es.wikipedia.org/w/index.php?title=Estereoscopia&oldid=143309740>
- [58] Redacción. Qué es Nemesis, la mecánica del videojuego 'Shadow of Mordor' que ha sido patentada en medio de una polémica [Internet]. RPP. 2021 [citado 7 de mayo de 2022]. Disponible en: <https://rpp.pe/videojuegos/juegos/juego-del-senor-de-los-anillos-que-es-nemesis-la-mecanica-del-videojuego-shadow-of-mordor-que-ha-sido-patentada-en-medio-de-una-polemica-noticia-1319953>
- [59] Nemesis-T Type [Internet]. Resident Evil Wiki. [citado 7 de mayo de 2022]. Disponible en: https://residentevil.fandom.com/es/wiki/Nemesis-T_Type
- [60] A* Pathfinding Project [Internet]. Arongranberg.com. [citado 7 de mayo de 2022]. Disponible en: <https://arongranberg.com/astar/features>
- [61] 4122 free SVG and PNG icons for your games or apps [Internet]. Game-icons.net. [citado 7 de mayo de 2022]. Disponible en: <http://game-icons.net>
- [62] Escapistmagazine.com. [citado 27 de mayo de 2022]. Disponible en: <https://onlysp.escapistmagazine.com/the-rise-and-rise-of-metroidvania/>
- [63] r/metroidvania - Is Metroidvania becoming the next over-crowded genre much like rogue-likes? So many games are coming in 2020 and many of them are really good quality [Internet]. reddit. [citado 27 de mayo de 2022]. Disponible en:

https://www.reddit.com/r/metroidvania/comments/fcz6ld/is_metroidvania_becoming_the_next_overcrowded/

[64] Nutt C. The undying allure of the Metroidvania [Internet]. Game Developer. 2015 [citado 27 de mayo de 2022]. Disponible en: <https://www.gamedeveloper.com/design/the-undying-allure-of-the-metroidvania>

[65] Video games industry market size in the U.S. 2010-2022 [Internet]. Statista. [citado 27 de mayo de 2022]. Disponible en: <https://www.statista.com/statistics/246892/value-of-the-video-game-market-in-the-us/>

[66] Dec TW. The games market and beyond in 2021: The year in numbers [Internet]. Newzoo. 2021 [citado 27 de mayo de 2022]. Disponible en: <https://newzoo.com/insights/articles/the-games-market-in-2021-the-year-in-numbers-esports-cloud-gaming>

[67] Witkowski W. Videogames are a bigger industry than movies and North American sports combined, thanks to the pandemic [Internet]. MarketWatch. 2020 [citado 27 de mayo de 2022]. Disponible en: <https://www.marketwatch.com/story/videogames-are-a-bigger-industry-than-sports-and-movies-combined-thanks-to-the-pandemic-11608654990>

[68] BOE.es - BOE-A-2018-16673 Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales [Internet]. Boe.es. [citado 27 de mayo de 2022]. Disponible en: <https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673>

[69] Alex CD. Por qué lo comparamos todo con Dark Souls [Internet]. Vidaextra.com. Vida Extra; 2016 [citado 27 de mayo de 2022]. Disponible en: <https://www.vidaextra.com/cultura/por-que-lo-comparamos-todo-con-dark-souls>

[70] Play Inferno's Court in your browser [Internet]. SIMMER.io. [citado 28 de mayo de 2022]. Disponible en: <https://simmer.io/@jpulidol/inferno-s-court>

[71] Metroidvania month 16 [Internet]. itch.io. [citado 28 de mayo de 2022]. Disponible en: <https://itch.io/jam/metroidvania-month-16>

Anexos

Anexo A: Entregables del proyecto

- PEC 2
 - Diario de desarrollo PEC2: descripción de los aspectos más destacados del juego al alcanzar el hito de entrega de la PEC2.
 - Demo técnica PEC2: prototipo del juego tras implementar las funciones descritas en el diario de desarrollo PEC2.

- Archivo de cálculo de los diferentes recursos del juego para equilibrar las estadísticas de los mismo.
- Encuesta realizada al finalizar la PEC2 de cara a la implementación y mejora del proyecto.
- PEC 3
 - Diario de desarrollo PEC3: descripción de los aspectos más destacados del juego al alcanzar el hito de entrega de la PEC3.
 - Versión PEC3: versión alpha del juego tras implementar las funciones descritas en el diario de desarrollo PEC3.
 - Actualización del archivo de cálculo de los diferentes recursos del juego para equilibrar las estadísticas de los mismo.
- PEC 4
 - Versión PEC4: versión Gold del juego tras completarse el desarrollo del juego.
 - Actualización del archivo de cálculo de los diferentes recursos del juego para equilibrar las estadísticas de los mismo.
 - Manual de usuario (Instruction booklet).
 - Tráiler del juego.
 - Video de defensa del proyecto.
 - Documentos de cesión de derechos y autoevaluación.
 - Memoria final de proyecto.
 - Anexos de la memoria.

Anexo B: Enlaces

- Enlace al repositorio Bitbucket: <https://bitbucket.org/jorgepul/infernos-court/src/master/>
- Enlace al canal de Youtube: <https://www.youtube.com/channel/UCa3DKujRvlnZuSKa1WSYoZA>
- Enlace al canal de Itch.io: <https://jpulidol.itch.io/infernos-court>
- Enlace al canal de simmer.io: <https://simmer.io/@jpulidol/inferno-s-court>
- Enlace al drive del proyecto: <https://drive.google.com/drive/u/1/folders/1dbfWMcd7mtauVDmQAj6HiqwihlQ79aDf>