UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Videojuego PAWN: Multiplataforma, desarrollo de mecánicas y efectos de sonido

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Canuto Soler, Jennifer

Tutor/a: Mollá Vayá, Ramón Pascual

CURSO ACADÉMICO: 2021/2022

Resumen

Debido a que “Pawn” nació de la unión de los alumnos de la ETSINF y BBAA para la realización de un proyecto de final de asignatura, este videojuego ya había sido desarrollado anteriormente, específicamente hasta su nivel 1. Es por ello por lo que, junto con dos compañeros más, Sergio Muñoz Alejandre y Diego Ramos Nebot, nos encargaremos de la realización del siguiente nivel, además de un nuevo modo que nos permitirá jugar con otros jugadores conectándonos a la red.

Por lo tanto, el trabajo que se realizará en este TFG será el de desarrollar la jugabilidad de los niveles, es decir, el modo de juego que se incorporará en el nivel 2 y en el modo multijugador; las distintas mecánicas del personaje, tales como esquivar, cambiar de arma, entre otras más; los efectos de sonido, usando herramientas de edición de audio; y, por último, el desarrollo del juego en al menos plataforma PC, Android y otras plataformas de Realidad Virtual.

Todo esto se realizará mediante la herramienta principal *Unity 3D*.

Palabras clave: videojuego, multiplataforma, sonido, Unity, mecánicas, jugabilidad, realidad virtual.

Resum

Pel fet que “Pawn” va nàixer de la unió dels alumnes de l'ETSINF i BBAA per a la realització d'un projecte de final d'assignatura, aquest videojoc ja havia sigut desenvolupat anteriorment, específicament fins al seu nivell 1. És per això que, juntament amb dos companys més, Sergio Muñoz Alejandre i Diego Ramos Nebot, ens encarregarem de la realització del següent nivell, a més d'un nou mode que ens permetrà jugar amb altres jugadors connectant-nos a la xarxa.

Per tant, el treball que es realitzarà en aquest TFG serà el de desenvolupar la jugabilitat dels nivells, és a dir, la manera de joc que s'incorporarà en el nivell 2 i en el mode multijugador; les diferents mecàniques del personatge, com ara esquivar, canviar d'arma, entre altres més; els efectes de so, usant eines d'edició d'àudio; i, finalment, el desenvolupament del jugue en almenys plataforma PC, Android i altres plataformes de Realitat Virtual.

Tot això es realitzarà mitjançant l'eina principal Unity 3D.

Paraules clau: videojoc, multiplataforma, so, Unity, mecàniques, jugabilitat, realitat virtual.

Abstract

As "Pawn" was born from the union of the students of the ETSINF and BBAA to carry out a final project, this video game had already been developed previously, specifically up to level 1. So, together with two more partners, Sergio Muñoz Alejandre and Diego Ramos Nebot, we will be developing the next level, besides of a new mode that will allow us to play with other players by connecting to the network.

Therefore, the work that will be done in this TFG will be to develop the gameplay of the level, that is, the game mode that will be incorporated in level 2 and in multiplayer; the different mechanics of the character, such as dodging, changing weapons, among others; sound effects, using audio editing tools; and, finally, the development of the game on at least the PC, Android, and other Virtual Reality platforms.

All this will be done using the main Unity 3D.

Keywords: video game, multiplatform, sound effects, Unity, mechanics, gameplay, virtual reality.

Tabla de contenidos

Índice de imágenes

Índice de tablas

1. Introducción

La historia del entretenimiento es larga y con una evolución constante y en continuo auge. Al principio, los juegos eran simples, solo se necesitaba del uso de la imaginación para poder entretenerse. Sin embargo, cuando se usa la palabra "videojuegos" no se hace referencia a las actividades que se realizaban con objetos caseros ni nada al respecto, sino más bien se refiere a un juego digital interactivo, donde se pueda conectar dispositivos de entrada (mandos o controles) o de salida (pantallas, ordenadores, etc.). Es por ello por lo que no se pudo dar a conocer este tipo de juego hasta finales de las décadas del siglo XX.

Al principio se trataban de videojuegos que se podía encontrar en máquinas de arcade situadas en centros públicos, por las cuales había que pagar cada vez que se deseaba jugar, pero con el paso del tiempo, y la incorporación de las TIC (Tecnologías de la Información y Comunicación), fueron apareciendo distintos dispositivos electrónicos con los que el usuario objetivo podía jugar. Los más conocidos hoy en día son los ordenadores, la *PlayStation* o las distintas *Nintendo*, pero existen más consolas que se han ido inventando durante estos años de evolución y también dispositivos portátiles, cómo podrían ser los teléfonos móviles.

Ahora bien, es cierto que existen todos estos elementos electrónicos con los que se puede jugar a varios tipos de juegos, pero también existen distintos dispositivos de entrada que los hacen más inmersivos. Un buen ejemplo serían los cascos o gafas de realidad virtual.

Así pues, a lo largo de este proyecto se irá mostrando la elaboración de un videojuego, haciendo hincapié en su jugabilidad y su migración a distintas plataformas, en las que se incluye la realidad virtual para dispositivos Android, haciendo uso de unas gafas de realidad virtual para teléfonos móviles y un *gamepad* [1]que podremos conectar por *bluetooth*, además de la creación de los efectos de sonido. Todo esto junto con la colaboración de dos compañeros, que se encargarán de la elaboración de otros aspectos del videojuego:

- Sergio Muñoz Alejandre - Videojuego PAWN: Desarrollo del modo multijugador online.
- Diego Ramos Nebot - Videojuego PAWN: Implementación de la Inteligencia Artificial y las animaciones.

1.1 Motivación

Hoy en día podemos ser testigos de la gran variedad de juegos que hay en el mercado, abarcando la mayoría de los géneros. Debido a ello, han ido ganándose el interés de las personas, logrando que poco a poco vayan evolucionando con ayuda de las TIC.

Así pues, esta fue una de las principales razones por la que esta idea de proyecto nació. Desde que me sumergí en este mundo comencé a tener interés y curiosidad sobre los videojuegos y su desarrollo: comenzando desde lo más básico, como su historia y narración, hasta llegar al producto final, terminando con la creación de un juego interesante y atrayente para el usuario objetivo.

De esta manera, surgió la motivación del desarrollo del videojuego presentado en esta memoria, pero migrando a otras plataformas para facilitar su portabilidad en caso de querer jugar fuera de casa, además de hacerlo más inmersivo para que el jugador disfrute de una realidad

[1] Descrito en el glosario de términos [1]

aumentada, envolviéndose en un mundo bélico ambientado en uno de los mayores juegos estratégicos: el ajedrez.

1.2 Objetivos

Tal y como se ha indicado en puntos anteriores, el objetivo principal de este proyecto es el desarrollo de un videojuego 3D usando como herramienta base Unity y su migración a otras plataformas. Otro objetivo importante que se tratará será el uso de unas gafas de realidad virtual, el cual será uno de los núcleos de este TFG.

De esta manera, podemos dividirlo en distintos aspectos que se tratará más adelante:

- Desarrollar la jugabilidad del juego, junto con las mecánicas del jugador y cada uno de los sonidos que nos harán sentirnos más inmersos en la trama.

- Experimentar el ambiente del videojuego en realidad virtual, pudiendo jugar desde el propio teléfono móvil.

- Aplicar los conocimientos adquiridos durante estos años cursando el grado de Ingeniería Informática, junto con los aprendidos en la asignatura de Desarrollo de videojuegos 3D.

- Todo ello se hará usando varias herramientas, pero la más importante y principal será Unity.

1.3 Estructura

Como todo videojuego, tiene que haberse realizado a priori un documento en el que se explique de manera detalla todos los aspectos de este, tanto técnicos como artísticos, al cual denominamos GDD[2]. Debido a que este proyecto se comenzó como un trabajo de asignatura, dicho documento se puede leer en el Anexo I.

Sin embargo, a pesar de que se trata de una continuación de un proyecto, también se ha realizado un estudio del estado del arte de la industria de los videojuegos y las múltiples plataformas con las que se pueden jugar. Con ello, se ha realizado un análisis del mercado con el que se han podido analizar mejoras para que este proyecto resulte mejor opción que otros parecidos.

Una vez hecho esto, faltaría desarrollar el videojuego y los distintos aspectos que se han ido desarrollando en este proyecto, pero antes de ello, se especifica las tecnologías empleadas, junto al diseño de la interfaz necesaria.

A parte, al final de la memoria, antes del anexo del GDD, se ha dejado un glosario de términos donde se explican vocablos que pueden llegar a ser desconocidos si no se tiene conocimiento previo del uso de la herramienta de Unity o del desarrollo de un videojuego.

Gracias a esta previa explicación, ya podemos empezar a mostrar el desarrollo del videojuego y su migración a otras plataformas, comenzando con una explicación de la jugabilidad del nivel

[2] Game Design Document o Documento de Diseño, es el documento escrito en el que se recogen todos los aspectos del diseño y/o la producción de un videojuego (más información en el Anexo I).

2, es decir, se detallará cómo es su funcionamiento y la manera en la que se ha hecho; pasando a comentar el proceso de desarrollo de *Pawn* en PC, Android y realidad virtual, agregando de paso las mecánicas del jugador, ya que según la plataforma usada tienen unos controles u otros y resulta más fácil de comprender uniendo ambos en un solo apartado. Tras esto, se terminaría la explicación comentando los efectos de sonidos que se han utilizado y su incorporación en el videojuego.

Otros aspectos que existan en el juego se tratarán en las memorias de mis compañeros, introducidos en el apartado de Introducción, o se hará de manera conjunta, tal y como es el diseño del escenario o armas.

2. Estado del arte

A pesar del tiempo que ha transcurrido, no se puede establecer cuál fue el primer videojuego de la historia, pero hay uno que puede considerarse el primero de todos, pero sin afirmarlo por completo: *Nought and crosses*, desarrollado por Alexander D. Douglas en el año 1952. Se trata del tres en raya que todo el mundo conoce, pero computerizado y sin poder jugar con una persona real, solamente contra la máquina en concreto.

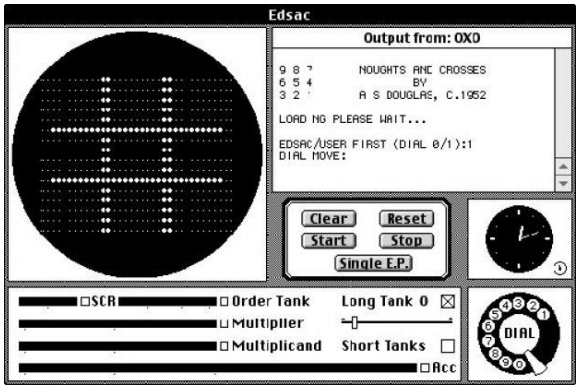


Ilustración 1. Interfaz del juego "Nought and crosses"

Los demás juegos que fueron desarrollándose eran prácticamente lo mismo. Juegos arcade básicos que consistían en mecánicas simples y no muy complejas. Un buen ejemplo sería *Spacewar!*, creado en 1962 por Steve Russel, un estudiante del Instituto de Tecnología de Massachussets. A pesar de que fue un éxito, no se dio a conocer por ser un proyecto universitario. Sin embargo, Nolan Bushell, el fundador de una de las empresas más importantes en la historia de los videojuegos, Atari, fue el causante de que en 1971 *Spacewar!* se diese a conocer y de que, además, los videojuegos comenzasen a tomar más importancia, y todo ello con la creación de la máquina recreativa *Computer Space*, basada en el juego anteriormente nombrado. No obstante, estas máquinas no eran de uso doméstico y, en el caso de *Computer Space*, había que pagar por cada uso.

Lo que hoy en día conocemos como consolas surgió en 1972, con la invención de *Magnavox Odyssey*, considerada la primera consola de uso doméstico, creada por Ralph Baer. Esta idea tan innovadora que consistía en conectarla a la pantalla para jugar a varios juegos que tenía instalados, generó un impacto en la industria, logrando que muchas empresas comenzaran a interesarse en el desarrollo de estos dispositivos, pero no fue hasta principios de los años 90 que las videoconsolas comenzaron a impactar de gran manera en la industria de los videojuegos.

Ilustración 2. Primera videoconsola, Magnavox Odyssey

Diversas compañías, tales como Capcom o Nintendo, comenzaron a desarrollar diferentes consolas con las que se podía jugar a juegos con mejores gráficos que los que tenían años anteriores. Sin embargo, el mayor golpe a esta industria fue con la aparición de videojuegos 3D, gracias a la creación de videoconsolas que hoy en día conocen la mayoría de las personas: Sony PlayStation, la primera de todas; Sega Saturn; y Nintendo 64.

Ilustración 3. Diseño de las consolas de PlayStation, Sega Saturn y Nintendo 64

Ahora bien, a pesar de que esa fuese la época de oro de los videojuegos, por así decirlo, debido al impacto que generó en esos años y en generaciones futuras, la tecnología que tenían no era la suficiente para poder jugar en un dispositivo móvil, después de todo, los que hoy en día conocemos como *smartphones* fueron comercializados en 1992 y el procesador que contenían no era lo suficiente potente como para poder cargar un juego parecido a los de la PlayStation o la Nintendo. Se podía jugar, por ejemplo, al *Snake*, un juego 2D que, a pesar de carecer de color y de mucha jugabilidad, causó una revolución en el mercado. Sin embargo, no fue hasta el siglo XIX donde se comenzó a comercializar poco a poco juegos con color o en 3D para dispositivos móviles, videojuegos que comenzaron a aumentar la industria. A su vez, la multiplataforma en teléfonos móviles y otros dispositivos también comenzó a hacerse notar.

Ilustración 4. Genshin Impact, videojuego para PC, PS4, IOS y Android

En cuanto a los ordenadores personales, no se quedan atrás. Empresas como Epic Games[3]o Ubisoft[4] han creado múltiples videojuegos para estos dispositivos y, además, muchos de ellos tienen la posibilidad de jugarlos en otras plataformas, siendo la *PlayStation* la que domina en este ámbito.

Hasta ahora se ha hablado un poco de la historia de los videojuegos, principalmente para ver la evolución de estos y la importancia que han ido tomando a lo largo de los años, sobre todo la multiplataforma. El poder desarrollar un videojuego que se pueda jugar en distintos dispositivos ha ayudado a las empresas a generar bastantes beneficios, después de todo, si una persona no cuenta con una *PlayStation*, por ejemplo, pero sí es propietario de un ordenador, la posibilidad de que juegue a ese juego es más alta que si se crea para un dispositivo en concreto. Es un aspecto que puede ayudar a que el videojuego sea más atractivo para el cliente.

Sin embargo, uno de los inventos que más impacto ha generado en el público, es el desarrollo de la realidad virtual. Esta tecnología ha sido usada para muchos ámbitos, tales como la medicina o la ingeniería, pero también se ha acabado usando en los videojuegos. Poder jugar y adentrarte dentro de un mundo virtual y diferente al real, puede llegar a ser cautivador para algunas personas, sobre todo si puedes interactuar con el escenario. Con ayuda de unas gafas, como podrían ser las *Oculus* [5]o las *Cardboard*[6], el usuario puede meterse en un mundo distinto, donde poder interactuar al mover la cabeza 360º, como si estuviese ahí mismo. Además, si lo unimos con dispositivos de entradas que ayuden a interactuar con el juego, como podría ser un mando o incluso un guante, la inmersión se hace más completa.

Ilustración 5. Ejemplos de gafas de realidad virtual

Ilustración 6. Mando bluetooth para jugar en el móvil

[3] Página oficial de la empresa: Epic Games
[4] Página oficial: Ubisoft
[5] Gafas de realidad virtual con una tecnología y unos gráficos avanzados. Se puede obtener más información en la página oficial de los desarrolladores de este dispositivo: Meta
[6] Gafas de cartón con cristales incrustados. Han sido diseñados por Google: Cardboard.

2.1 Crítica al estado del arte

Difícilmente podemos encontrar juegos de dispositivos móviles para realidad virtual. La mayoría son *demos*[7] lanzadas en videoconsolas, esencialmente para PS4[8] o juegos multijugador que al final acaban siendo repetitivos. Ahora bien, la calidad de una videoconsola u ordenador es mucho mejor que la de un teléfono móvil, a pesar de que tengan pantallas AMOLED o un procesador potente como sería el Snapdragon 8 Gen 1, pero un inconveniente a favor de los dispositivos móviles es el coste.

Las *Oculus Quest* 2 cuentan con un buen rendimiento gracias a su procesador y una gran calidad de gráficos con resolución 2K, pero debido a todo esto su precio es algo elevado y posiblemente muchas personas no se lo puedan permitir. Sin embargo, unas *Cardboard* o unas gafas de realidad virtual para dispositivos móviles, pueden llegar a costar alrededor de 30 euros, mientras que las *Oculus* cuestan alrededor de 400. Se puede ver la diferencia de precio a simple vista. Una de las razones es, obviamente, su calidad gráfica, tal y como se ha dicho, pero esto no es un inconveniente para que un juego de realidad virtual en móvil pueda ganar algo de interés del su público objetivo.

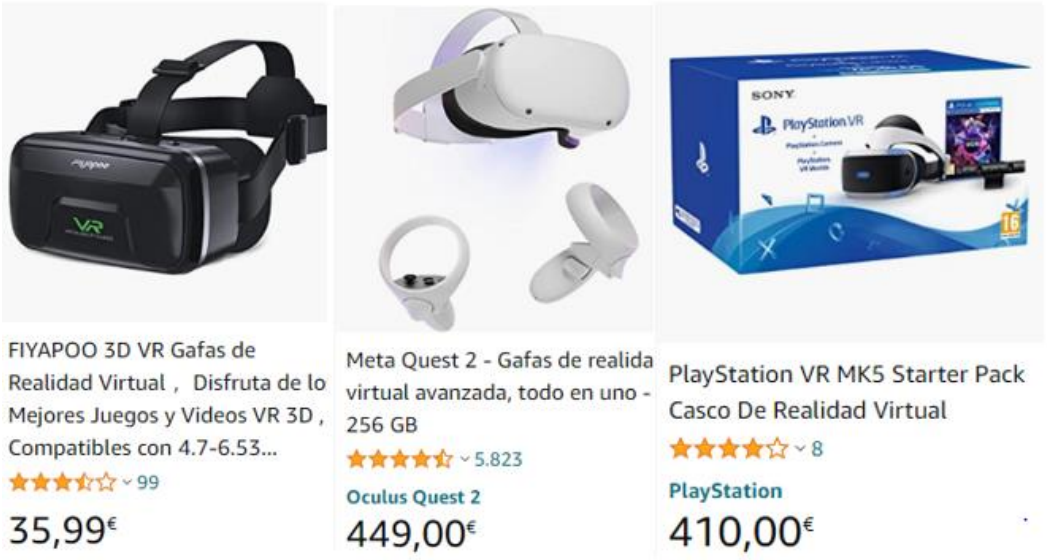


Ilustración 7. Comparación de los distintos precios que hay en Amazon

No obstante, la cantidad de juegos de calidad de este tipo es escasa y sobre todo si hablamos de videojuegos del género de plataforma. Cuando se piensa en juegos de realidad virtual, lo primero que se puede venir a la mente son los que hay en la *PlayStation*, compatibles con las gafas que la misma marca han creado, o los que se pueden jugar en el ordenador gracias a las *Oculus* o similares, por lo que no es muy común pensar en un juego de Android o de IOS para este ámbito.

Muchos de estos juegos son con una jugabilidad simple. Puedes mover la cabeza hacia todos los lados, pero la interacción suele ser escasa, aunque algunas veces se puede conectar un dispositivo de entrada para hacerlo más inmersivo. Un ejemplo sería el juego de móvil VR *Crazy Swing*[9], un juego en el que consiste estar montado en un columpio a una gran altura. Cómo se

[7] Pequeña versión del videojuego usado para hacer una demostración de cómo sería jugarlo.
[8] PlayStation 4
[9] Desarrollado por Fibrum y se puede encontrar en la tienda de Play Store

puede ver, la jugabilidad es escasa y al principio puede llegar a ser interesante probarlo, pero luego acaba siendo repetitivo al no poner hacer más cosas.

Ilustración 8. Imagen promocional del videojuego "Crazy Swing"

De este modo, un juego de plataformas, con la opción de poderlo jugar en ordenador, Android o en este mismo dispositivo, pero en realidad virtual, puede llegar a ser llamativo para el usuario objetivo, sobre todo si se trata sobre fichas de ajedrez.

3. Propuesta de trabajo

En este apartado se procederá a explicar el análisis del mercado de algunos de los videojuegos que pueden considerarse una competencia para el proyecto. Hay que tener en cuenta que el género de Pawn es de plataformas y que en la actualidad existen muchos juegos de este estilo, así que simplemente se analizará los que han servido de referentes y los que más parecidos son a nivel de jugabilidad. Además, también se estudiará juegos multiplataforma y juegos para dispositivos Android con la opción de poder jugar en realidad virtual.

3.1 Análisis del mercado

Hoy en día es casi imposible idear un proyecto completamente original, después de todo, el mercado está repleto de videojuegos de todo tipo y para todas las edades, sobre todo los de plataformas. No es extraño que cada año salga al mercado un juego de ese estilo, después de todo, se trata de un género que resulta entretenido y atrayente para el cliente, sea de la edad que sea. Unos claros ejemplos podrían ser los que se van a describir a continuación y que, por tanto, se consideran la competencia y referentes del videojuego que se desarrollará en esta memoria, *Pawn*.

Crash Bandicoot[10]

Serie de videojuegos creada en el año 1996 por *Naughty Dog*[11] para múltiples plataformas, tales como PlayStation, Xbox, Nintendo, Android, IOS y PC, pero la principal siempre ha sido PlayStation.

En un principio esta saga fue creada con la idea de pertenecer al género de plataformas, desarrollando de esta manera sus tres primeros juegos para PlayStation: *Crash Bandicoot* (1996), *Crash Bandicoot 2: Cortex Strikes Back* (1997), *Crash Bandicoot 3: Warped* (1998). Sin embargo, con el paso de los años el contenido de sus videojuegos fue cambiando. En el año 1999 distribuyeron su primer juego de carreras, *Crash Team Racing* (1999), y al año siguiente uno de minijuegos, *Crash Bashl* (2000).

Ilustración 10. Imagen promocional de Crash Bandicoot

Ilustración 9. Imagen promocional de Crash Team Racing

Hasta ese año todos y cada uno de sus juegos fueron desarrollados para PlayStation, pero al año siguiente pusieron en el mercado un videojuego de plataformas para esta consola, Xbox y Nintendo Gamecube, *Crash Bandicoot: La venganza de Cortex* (2001). Fue la primera vez que crearon uno para múltiples plataformas y donde inició una larga saga de videojuegos para distintos

[10] Se adjunta la página oficial de la franquicia: Crash Bandicoot
[11] Para más información: Naughty Dog

dispositivos, incluido dispositivos móviles, tales como se hizo con el juego titulado *Crash Twinsanity*.

Spyro[12]

Franquicia de videojuegos de género de plataformas, desarrollado en 1998 por *Insomniac Games*[13]. Al igual que ocurría con *Crash Bandicoot*, *Spyro* fue lanzado principalmente para PlayStation, sus primeros juegos eran solo para esta consola, pero más adelante fueron desarrollando videojuegos para otras plataformas, como, por ejemplo, *Spyro: Enter the Dragonfly* (2001), lanzado para PlayStation 2 y GameCube, o *Spyro: A Hero's Tail* (2004), para PlayStation 2, Xbox y GameCube.

Ilustración 11. Captura de la adaptación de la trilogía de videojuegos de esta franquicia, Spyro Reignited Triology

Ratchet & Clank[14]

Saga de videojuegos del género de plataformas, creada en el año 2000 por la compañía *Insomniac Games*. Al igual que los dos ejemplos anteriores, esta serie también fue desarrollada principalmente para consolas de PlayStation, con juegos tales como *Ratchet & Clank* (2002), *Ratchet & Clank 2: Totalmente a tope* (2003) o *Ratchet & Clank 3* (2004), todos para PS2[15] o PS3[16]. Sin embargo, esta saga también distribuyó jugos para otras plataformas, específicamente para teléfonos móviles: *Ratchet & Clank: Going Mobile* (2005) y *Ratchet & Clank: Before the Nexus* (2013).

Su modo de juego es muy parecido a los dos anteriores, pero añadiendo una mecánica de disparos que le dan un toque diferente, además de añadir conducción de vehículos, puzles y pilotaje de naves.

[12] Información de la franquicia de videojuegos: Spyro
[13] Empresa de videojuegos. Su página oficial es: Insomniac Games
[14] La información sobre los videojuegos de esta serie se puede encontrar en la página oficial de PlayStation
[15] PlayStation 2
[16] PlayStation 3

Ilustración 12. Captura del juego de PlayStation 5, Ratchet & Clank: Una dimensión aparte (2021)

Ilustración 13. Captura del juego de móvil, Ratchet & Clank: Before the Nexus (2013)

Super Mario[17]

Se trata de una saga de videojuegos bastante conocida, desarrollada en 1985 por la compañía *Nintendo*, siendo también su propio distribuidor. La mayoría de sus juegos son de plataformas, a excepción de algunos cómo, por ejemplo, *Mario Kart* (1992), siendo un juego de carreras lanzada para la consola Super Nintendo; o *Mario Party* (1998), primer juego de la saga con el mismo nombre, siendo todos y cada uno de ellos del género de minijuegos.

Ilustración 14. Imagen promocional de Super Mario Run

Todos los juegos de *Super Mario* han sido creados principalmente para consolas de *Nintendo*, sin embargo, existen otros videojuegos a los que se pueden jugar en dispositivos móviles, cómo es el caso de *Super Mario Run* (2016 para IOS, 2017 para Android), siendo este el que más se parece a una parte del nivel 2 que se va a desarrollar en el videojuego de este proyecto, debido a que se trata de un juego *endless runner*[18] y nuestro objetivo es aplicar este modo de juego a mitad del nivel, solo que en nuestro caso sería en 3D; además, se trata de un videojuego creado por la consola de Unity.

[17] Página oficial de *Nintendo* donde muestran la franquicia de Super Mario
[18] Descrito en el glosario de términos [2]

En cuanto a diseño, es importante destacar que sus primeros juegos, tales como *Super Mario Bros* (1985), fueron en 2D, sin embargo, con la evolución de la tecnología y el incremento de consolas con mejores gráficos, se empezaron a desarrollar juegos en 3D, como es el ejemplo de *Super Mario Galaxy* (2007), el cual, en cuanto a jugabilidad, es bastante semejante a Pawn.

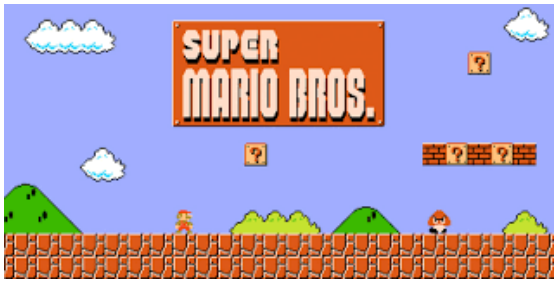


Ilustración 15. Interfaz de Super Mario Bros

Ilustración 16. Captura del juego de Super Mario Galaxy

Subway Surfers[19]

Videojuego *Endless Runner* desarrollado por Kiloo[20]. Originalmente se creó para dispositivos con sistema operativo Android, pero con el pasar del tiempo, y gracias a la gran fama que obtuvo, y con ello beneficios, los creadores decidieron en ponerlo también en otras plataformas, tales como Windows e IOS.

Lo que lo asemeja con este proyecto es su jugabilidad. *Subway Surfers* podría considerarse el referente número uno a la hora de crear la parte del *Endless Runner*, debido a que su movilidad es similar y la posición de la cámara también.

Ilustración 17. Imánes del juego "Subway Surfers"

[19] Página oficial del videojuego Subway Surfers
[20] Empresa de videojuegos. Se pueden encontrar muchas de sus creaciones en su página web

Sonic Generations[21]

Se trata también de un videojuego de plataformas 3D perteneciente a la serie de *Sonic the Hedgehog*, producido por la empresa Sega[22] y lanzado al mercado en 2011. Está disponible para múltiples plataformas, tales como Microsoft Windows, Nintendo 3DS, Xbox 360 y PlayStation 3.

La mayoría de los videojuegos de esta serie son de plataformas también, pero el motivo por el cual se ha decidido centrarse más en este juego en específico es por ser en 3D[23], cosa que no ocurría con algunos de sus antecesores. Aparte, es un buen ejemplo de cómo se vería el nivel que se va a desarrollar en esta memoria.

Figure 19. Captura del juego de Sonic Generations

Como podemos ver, todos estos juegos son del mismo género que *Pawn*, además de que la mayoría, exceptuando juegos antiguos como los primeros de Super Mario, tienen un diseño 3D y algunos de ellos son para múltiples plataformas. Todos estos detalles son los que rivalizan con este proyecto y, al mismo tiempo, sirven de referentes. Sin embargo, en ninguno de estos casos hemos visto que se juntara un ambiente bélico con fichas de ajedrez. Este aspecto es lo que lo diferencia de la competencia, aparte de que se podrá jugar en realidad virtual desde un dispositivo móvil.

Por eso mismo, a continuación, se mostrarán una serie de ejemplos que puedan rivalizar en ese aspecto, solo que el modo de juego o de diseño no será parecido al que se realizará en esta memoria, es por esa razón por la que se ha dividido estos ejemplos en dos partes: los que son del género de plataformas o se pueden jugar en varios dispositivos, que serían los videojuegos anteriores; y los que son para juegos de móvil y en realidad aumentada.

[21] Página oficial de Sega donde se muestra información sobre este videojuego: Sonic Generations
[22] Página oficial de la empresa: Sega
[23] Video en el que se muestra la jugabilidad del juego <https://www.youtube.com/watch?v=sCZKfC70xGU>

Need For Speed: No limits VR

Juego de carreras en realidad virtual[24] para dispositivos Android, creado en 2016 por *Electronic Arts*[25]. El jugador se encuentra dentro de un coche, del cual no se puede salir, solamente se puede conducir el vehículo y mirar hacia los lados girando la cabeza. Este juego pertenece a una saga llamada *Need for Speed*[26], pero es el único con el que se tiene la opción de jugar de manera aumentada.

Ilustración 18. Captura del juego de Need For Speed: No limits VR

Mekorama VR

Juego 3D de rompecabezas en realidad virtual[27], desarrollado por Martin Magni[28]. El objetivo es mover diferentes elementos para que el personaje principal, el cual es un robot, llegue a la meta y de esta manera pasarse el nivel. Cuenta con una versión gratuita para jugar desde el móvil, sin la opción de realidad virtual.

Ilustración 19. Captura del juego de Mekorama VR

[24] Video demostrativo <https://www.youtube.com/watch?v=1fPnj0jZHaI>
[25] Empresa de videojuegos bastante conocida: Electronic Arts
[26] Más información: Need for Speed
[27] Vídeo demostrativo <https://www.youtube.com/watch?v=LF1UqwvwaS0>
[28] El juego se encuentra en la tienda de Google y tiene un coste 4,29€

Hardcore VR

Juego 3D para dispositivos móviles al que puedes jugar en realidad aumentada[29]. Se centra más en el modo de juego online, donde se podrá enfrentar a otros jugadores mediante disparos y puzles. Fue desarrollador por Mike Farlenjov, el cual solo cuenta con este y otro juego, *Need For Jump VR*[30], ambos para realidad virtual.

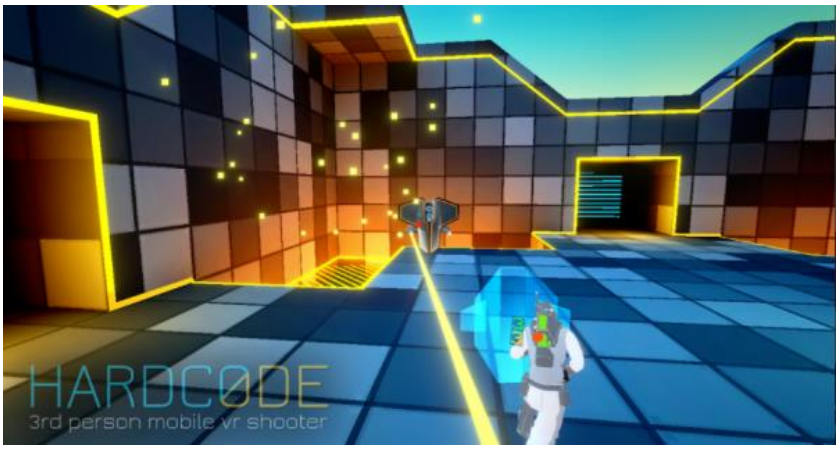


Ilustración 20. Imagen promocional del juego de Hardcore VR

Voyage Wrong VR

Mezcla bastantes estilos de juego, como los rompecabezas, los juegos de disparos o los de plataformas, todo esto en realidad virtual[31]. Ha sido desarrollado por Fguillotine y, al igual que los ejemplos anteriores, es para dispositivos Android.

Ilustración 21. Captura del juego de Voyage Wrong VR

[29] Vídeo demostrativo <https://www.youtube.com/watch?v=8aslwyzWtWQ>
[30] Se puede encontrar gratis en Google Play
[31] Vídeo demostrativo <https://www.youtube.com/watch?v=LKawfPm6LQg>

Análisis Comparativo

Una vez se ha presentado y explicado cada uno de los videojuegos que se pueden considerar competencia, se va a mostrar una tabla con un análisis comparativo que una las características que más destacan, comparándolas con el uso de los asteriscos, siendo uno solo el más bajo, y cinco el más alto. En el caso de que haya un hueco sin nada significa que dicho videojuego carece de esa característica.

	Historia narrativa	Posibilidad de jugarlo en distintas plataformas	Mecánicas y sistemas de recompensa	Realidad virtual	Jugabilidad	Multijugador
Crash Bandicoot	*****	****	***		*****	
Spyro	*****	****	***		*****	
Ratchet & Clank	****	****	***		*****	
Super Mario	***	*	***		****	****
Subway Surfers	**	**	****		***	
Sonic Generations	**	*****	****		***	
Need for Speed: No limits VR	*		*	****	****	
Nekorama VR	*		**	****	*	
Hardcore VR			**	****	**	****
Voyage Wrong VR			**	****	**	

Tabla 1. Análisis comparativo de los distintos juegos presentados

Tras este análisis, se ha llegado a una conclusión de cuáles serían los requisitos que debería tener el juego:

- Una historia que explique el por qué el protagonista hace lo que hace y que, a su vez, sea original y atractiva.
- Multiplataforma, aunque sea para PC y dispositivos móviles, pero con la posibilidad de jugar con un mando.
- Mecánicas que pasen de lo básico, pero que no llegue a un extremo de dificultad que pueda llegar a ser tedioso para un juego de ese género.
- Buena jugabilidad. Desarrollar enemigos que puedan ponerle las cosas difíciles al jugador y que no resulte aburrido.
- Poder jugar con otros jugadores.

Estos serían los requisitos que debería tener *Pawn* en un principio, pero no se van a mostrar todos en ese TFG. La parte de la inteligencia artificial y el modo multijugador se desarrollará en

las memorias de mis compañeros de trabajo, pero sí que se comentará los demás aspectos, salvo la narrativa, ya que está incluido en el GDD.

Resumen

Con todo esto podemos observar que, a pesar de que existen una multitud de juegos del género de plataformas, muy pocos con la opción de realidad virtual para Android tienen este estilo de jugabilidad, un aspecto que favorece a *Pawn* respecto a la competencia.

Hemos podido ver que, a pesar de que el modo de juego sea semejante, cada uno de ellos se diferencian en algunas cosas, como por ejemplo la ambientación. Al igual que *Spyro* está ambientando en un mundo de fantasía en el que existen los dragones, donde el jugador maneja a uno de ellos, *Pawn* se engloba en un mundo bélico con guiños al ajedrez y fichas de este juego de mesa, solo que con toques de diseño diferentes.

También hay que destacar que no todos los juegos están hechos para diversas plataformas, sobre todo si hablamos de realidad virtual, y más si es para dispositivos móviles. Es algo complicado ver un juego de este estilo en un dispositivo móvil y con la opción de jugar en realidad virtual, además de poder jugar con otros jugadores por red.

3.2 Análisis DAFO

Una vez hecho el análisis del mercado, es hora de realizar el análisis DAFO, del proyecto, donde podremos observar sus puntos débiles y fuertes respecto a su competencia. De esta manera, a la hora de desarrollar el videojuego, se podrá pulir mejor algunos detalles que ayudarán a destacar entre los demás y llamar la atención del público objetivo.

Fortalezas	Oportunidades
• Múltiples formas de juego (Realidad virtual, online, PC, teléfonos móviles). • Ambiente y personajes originales.	• Poca variedad en el mercado de juegos de plataformas para dispositivos móviles en realidad virtual. • Juego inspirado en el ajedrez.
Debilidades	**Amenazas**
• Equipo pequeño e inexperto. • Escasez de fondos para invertir en el proyecto. • Juego de poca duración. • Poca popularidad en el mercado. • Pocos recursos para realizar campañas de marketing.	• Mucha variedad de juegos del mismo género. • Juegos con mejor diseño 3D. • Tiempo escaso para exprimir al máximo el potencial del juego.

Tabla 2. Matriz DAFO para la elaboración de Pawn

Al ser un proyecto de final de carrera, no se puede contar con el tiempo y los recursos suficientes para poder realizar el juego con mejor calidad. Actualmente el grupo se compone de tres programadores y ningún diseñador, por tanto, en el estilo visual se verá más ineficiente que otros juegos del mercado. Sin embargo, a pesar de que el tiempo sea limitado y no se cuente con

una gran financiación, lo que se busca es crear una base en el que apoyarse y, más adelante, pulirlo del todo. Es cierto que hay muchos juegos similares, pero por ello se busca crear un proyecto que pueda llamar la atención a pesar de tener mejores opciones en el mercado.

3.3 Especificaciones

Pawn se trata de un videojuego de plataformas con una ambientación bélica medieval. Sus personajes son fichas de ajedrez, siendo el protagonista, de nombre igual al del juego, un peón normal, con una espada simple y sin ningún rasgo a destacar. Los enemigos también son fichas de ajedrez, incluso existe el rey y la reina en el juego, solamente que no aparecen hasta el nivel final o, en el caso de la reina, en el modo multijugador, siendo ésta el jefe final con el que todos los jugadores tendrán que pelear, tal y como en el modo historia.

Se divide en cinco niveles con cinco enemigos finales. El primer nivel representa al peón, por lo que sus enemigos son este tipo de fichas, pero con tres rasgos diferentes: el delgado, siendo más rápido que el resto; el alto, con la ventaja de que tiene más vida; y el más corpulento, con la característica de que es más fuerte y, por tanto, sus golpes hacen más daño. Manejando a Pawn, se tendrá que pelear con cada uno de estos enemigos hasta llegar al final del nivel, donde aparecerá el Peón jefe, siendo este más grande y fuerte que todos los demás. A este punto de la partida, se tendrá que derrotar para poder avanzar al siguiente nivel y poder obtener como recompensa por derrotarle una lanza que servirá de arma durante el resto del juego.

El nivel dos representa a los caballos y será el que se desarrollará en este proyecto y en el de mis compañeros. Esta vez no solo habrá peones como enemigos, sino también jinetes montados a caballos. Estos personajes son más rápidos que los demás y cuentan con una lanza y un escudo. El enemigo final de este nivel es como los nuevos enemigos, pero más grande y con más habilidades, y en este caso su recompensa será un escudo.

Los siguientes niveles no se van a poder desarrollar, pero a modo de resumen cada nivel representa una ficha de ajedrez: en el tres aparecerá el alfil como enemigo y le dará a Pawn la habilidad de doble salto una vez lo derrotes; el siguiente es la torre, el cual le otorgará un cañón que servirá de arma a distancia; y en el último nivel se podrá pelear con los jefes que representan al caballo, al alfil y a la torre, dando a entender que hay dos por cada categoría, al igual que en el ajedrez, para luego pelear contra la reina. Una vez derrotada, el rey huye, abandonando la corona y dando fin al juego.

Esta sería la historia principal y el modo de juego de un solo jugador, pero *Pawn* también cuenta con una opción de multijugador donde se podrá realizar batallas con la ayuda de otros jugadores que se conectarán por red.

3.4 Plan de trabajo

Para poder organizar las distintas tareas a la hora de realizar la memoria y el trabajo, se ha hecho un diagrama de Gantt inicial, con el objetivo de cumplir con los plazos establecidos.

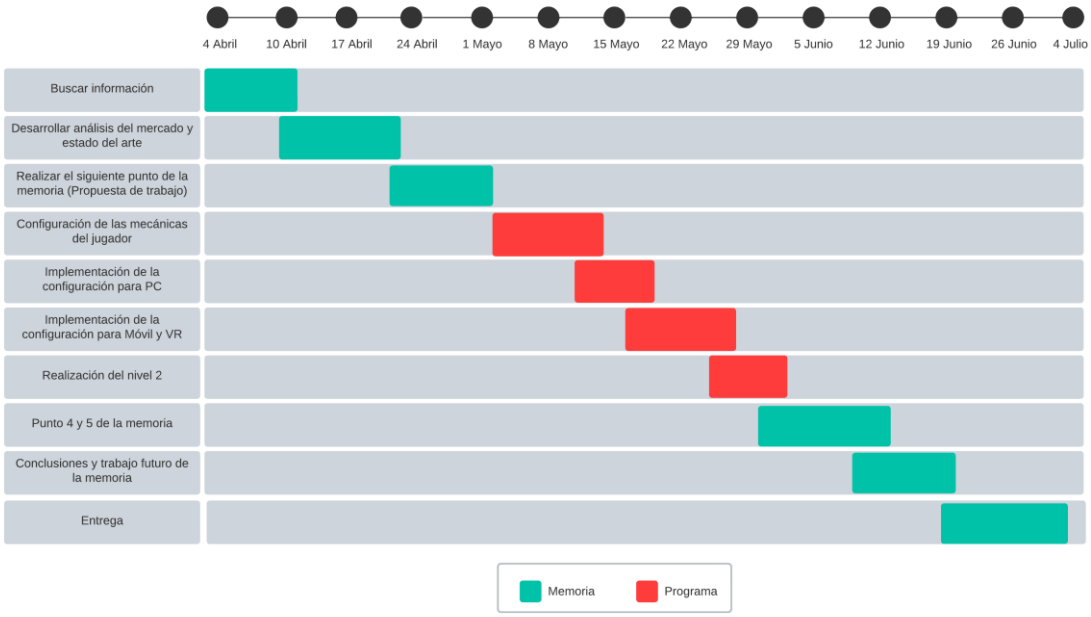


Ilustración 22. Diagrama de Gantt inicial

Sin embargo, no se ha cumplir con exactitud dicha planificación. El diagrama final es diferente y, además, se ha tenido en cuenta la organización en grupo con los distintos participantes del videojuego. Lo que se ha complicado a la perfección son las tres primeras tareas, correspondiente a la realización de los tres primeros puntos de la memoria, además de buscar la información que proporciona secretaría para saber la estructura y la recomendación a la hora de realizar la memoria.

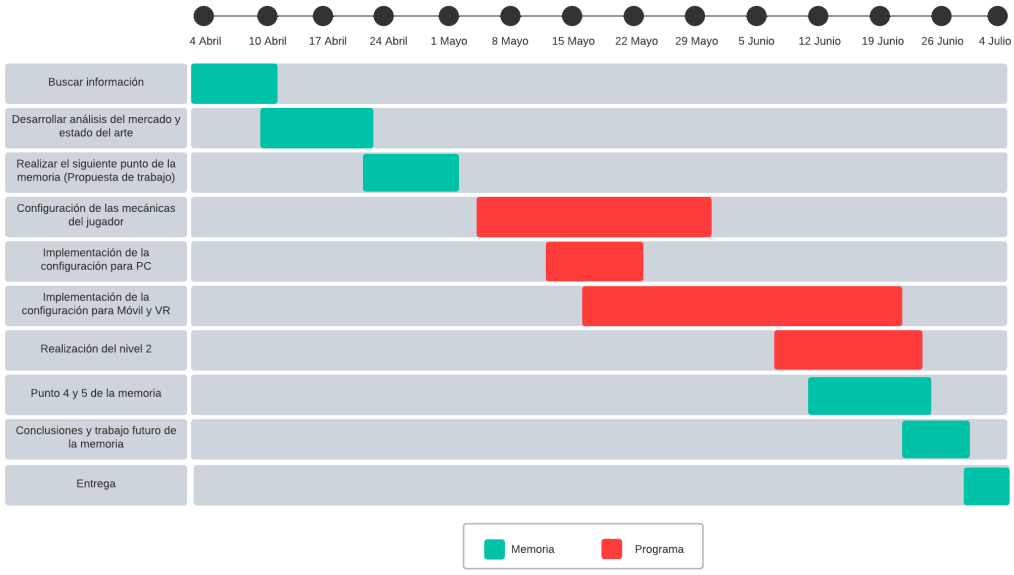


Ilustración 23. Diagrama de Gantt final

Como se puede ver, el desarrollo de las plataformas de Android y realidad virtual ha tomado más tiempo de lo previsto, además las mecánicas. Esto es debido a los problemas que generaban algunas configuraciones. Esto se explica más adelante en la memoria, así que no profundizaré

en este apartado, pero si se ha de avisar que el retraso respecto a esas tareas es debido a dichos problemas y que, por tanto, las tareas que venían por detrás se han visto afectadas.

4. Diseño y especificación

En este apartado se hablará sobre las herramientas utilizadas para el desarrollo de los distintos elementos del juego, tanto a nivel de programación como a nivel de diseño. Además, aun sabiendo que el núcleo de este proyecto se basa por completo en la programación, también se ha tenido que diseñar algunos aspectos para darle sentido al juego. Por eso mismo, en los siguientes apartados se mostrará con mayor detalle cada uno de estos aspectos.

4.1 Tecnología a emplear

Durante la realización del videojuego se han ido usando diferentes herramientas aparte de Unity 3D, algunas para el desarrollo del proyecto y otras para planificar correctamente las tareas del equipo o individuales, de esta manera conseguimos una mayor organización y comunicación entre todos. No obstante, antes de explicar cada una de ellas, se va a hacer un análisis de las diferentes propuestas que hay y justificar la razón del por qué finalmente se han elegido dichas herramientas.

Motor de videojuegos

Existen varios motores en el mercado, tanto gratis como de pago. También hay algunos que son gratuitos, pero tienen versiones de pago con más opciones que la principal. A veces esto suele ser beneficioso, pero la mayoría de las veces la versión gratuita es suficiente para realizar los objetivos establecidos. Por eso mismo, los motores que se mostrarán a continuación son gratuitos o se han elegido la versión gratuita para hacer la comparación.

Una de las características que debería tener esta herramienta es la posibilidad de realizar videojuegos con gráficos 3D, después de todo, el que se va a realizar en este trabajo va a ser con dichos gráficos, por lo tanto, se han buscado opciones que contengan esta opción y se han encontrado programas como Unreal Engine, Unity, Source 2, Cry Engine y Godot, entre otros, pero las mejores opciones que se ha visto son estas.

Ahora bien, a pesar de que cada uno de ellos desarrollan videojuegos en 3D, hay otros aspectos que ver para poder reducir las elecciones. Uno de ellos es en qué plataformas se puede crear los videojuegos. En este caso, se necesitaría que tengan la opción de generar el ejecutable del juego en ordenador y en Android, como mínimo, y que además tenga soporte para realidad virtual. Por ello, los que cumplen este requisito son todos a excepción de Source 2.

Estos serían los requisitos principales e importantes que debería tener el motor de videojuegos que se vaya a utilizar, pero obviamente no se van a usar los cuatro que han quedado, por eso se ha mirado entre ellos cual podría ser la mejor opción. Una opción viable podría ser Unreal Engine, ya que es el más completo. Cuenta con diferentes herramientas de renderizado, animaciones fluidas y para el manejo de físicas que puedan llegar a ser complejas. Sin embargo, un motor que estaría por encima suya, en este caso, sería Unity, por dos razones: el equipo está familiarizado con él, ya que es el que se ha estado utilizando en la asignatura del primer cuatrimestre, y cuenta con una comunidad mayor que la de Unreal Engine. Por tanto, cuando de información se trata, es más sencillo encontrarla en foros de Unity que en los del otro motor de videojuegos. Estas serían las razones por las que se ha elegido este motor por encima de los otros.

Ilustración 24. Logo de Unity

Entorno de desarrollo integrado (IDE)

Hay distintos entornos que se podrían usar para el desarrollo de *scripts*[32] en Unity, pero al final de un análisis comparativo, y unas ventajas bastante claras, se ha optado por uno en concreto: Visual Studio. La razón de esto es simple, es el programa que se ha estado usando durante la asignatura y con el que más familiarizado se está, pues a parte, también se ha estado utilizando en otros cursos.

Esta es la principal razón por la que se ha elegido este programa en vez de otros, aun así, hay que destacar que no es la mejor opción para Unity, a pesar de que tener una amplia variedad de liberias. Tras una investigación se ha encontrado un entorno bastante compatible con ese motor de videojuegos, y ese es Rider. Su usabilidad es similar a la Visual Studio, es sencillo de manejar y contiene distintos API's que son útiles en Unity. Sin embargo, no se ha optado por esta opción por lo dicho anteriormente, la comodidad que generaba Visual Studio al conocer bastante bien su uso fue la razón decisiva para quedarse con este programa.

Ilustración 25. Logo de Visual Studio

Herramienta de efectos de sonido

Durante el transcurso de la asignatura y el desarrollo de este proyecto, se ha estado buscando opciones en internet que nos proporcionase los audios necesarios para usarlos en el videojuego, pero a pesar de haber varias opciones, se ha decidido por las que no contienen derechos de *Copyright.* Por lo tanto, con esto en cuenta, la herramienta web que más se ha utilizado es Zapsplat, pero también se han usado otras herramientas que hay por internet en los casos en los que no se encontraba el audio que se deseaba.

Dicho esto, no se puede quedar con uno en concreto, pero sí se destacaría la que se ha dicho por ser la más utilizada.

[32] Descrito en el glosario de términos [3]

Ilustración 26. Logo de Zapsplat

Gestión de proyectos

Has diferentes herramientas para ello, incluso Visual Studio tiene una opción de poder gestionar los proyectos entre diversas personas, pero a pesar de ello no es la que se ha utilizado. Azure también era una opción viable, además es compatible con el entorno de desarrollo utilizado, pero había una herramienta que resultaba mejor opción que todas estas, y esa es GitHub Desktop. Las razones son varias, pero la más importante es la facilidad de crear diferentes ramas y fusionarla con la original. Esto facilita bastante que, en caso de pérdida de datos por conflictos que pueden haberse generado, se puede recuperar la información desde la rama con la se había fusionado. También es posible revertir cambios de manera sencilla en caso de que, por algún motivo que no se puede localizar, el proyecto deje de funcionar como se desea.

Con todas estas características, se vio a GitHub como la mejor opción para gestionar el proyecto entre varias personas.

Ilustración 27. Logo de Github

Junto a ello, también se han usado otras herramientas para gestionar mejor las tareas del proyecto. A la hora de que el equipo se ponga de acuerdo, las reuniones son la mejor opción, para ello, cada martes de la semana el equipo se reunía en Discord para establecer las tareas que se debían realizar. La razón del por qué se ha elegido esta opción en vez de otras, como por ejemplo Microsoft Teams o Skype, es porqué se puede generar un servidor que se divida en distintas categorías, pudiendo así ordenarlo por tutoriales, recursos y referencias. De esta manera, cada uno de ellos contienen distintos canales para distintos aspectos del juego: diseño, audio, inteligencia artificial, etc.

Resultaba más fácil ordenarlo de esta manera que creando un grupo en Teams. Por lo tanto, era más sencillo reunirse por Discord y subir en el canal apropiado los aspectos de los que se iban hablando en la reunión semanal.

Ilustración 28. Logo de Discord

Con ello, se ha usado de manera paralela el uso de la herramienta Google Docs para apuntar todo de lo que se ha hablado en la reunión y de lo que se necesitaba tener para la siguiente, pudiendo así organizarse mejor. Es cierto que hay herramientas mejores, como podría ser Trello, pero resultó más cómodo usar la herramienta de Google debido a que había libertad de escribir todo lo que se quisiera de manera simultánea.

Ilustración 29. Logo de Google Docs

Sin embargo, a nivel personal, se ha usado la herramienta Microsoft To Do para las tareas individuales. De esta forma se podía llevar una planificación de cuáles eran las tareas que se debía realizar para una fecha determinada o qué faltaba por completar. El motivo por el que se ha decidido el uso de esta herramienta es la facilidad de usarlo en distintos dispositivos con la misma cuenta. Al ser una herramienta de Microsoft, se podía tener la aplicación en el móvil y ver o añadir las tareas desde el dispositivo, solamente sincronizándolo con la cuenta que estaba en el ordenador.

Ilustración 30. Logo de Microsoft To Do

Herramienta de diseño

Existen diferentes programas de modelado 3D, unos gratis y otros de pago. En este caso se ha comparado las opciones gratuitas, que no son muchas, y se ha optado por Blender. Las razones siguen siendo las mismas que con los otros programas: el conocimiento tras su uso en la asignatura. Aun así, otros motivos por lo que se ha elegido esta herramienta entre otras más es por la facilidad de encontrar tutoriales para este programa y por ser completamente gratuito. No

se puede realizar una comparación extensiva, ya que las herramientas que están por encima de Blender acaban siendo de pago, como por ejemplo 3ds Max.

Ilustración 31. Logo de Blender

4.2 Diseño

Como se ha estado hablando durante la realización de esta memoria, se tratarán más los aspectos técnicos que los de diseño. No obstante, esto no quita el hecho de que se ha tenido que diseñar algunos detalles, tales como el menú o escenario. Antes que nada, los modelados de algunos objetos y de los enemigos se mostrará en la memoria de otro de los participantes del equipo, ya que en este trabajo se ha diseñado solamente la parte del *Endless Runner*, junto a sus *prefabs*[33], y el menú de controles, el cual se ha añadido debido a que se ha introducido la opción de jugar con mando.

Menú principal

A pesar de que el menú se ha desarrollado en la asignatura de videojuegos, se va a mostrar un esquema de la navegabilidad entre las opciones del menú principal, para entender mejor el funcionamiento de éste y tener un contexto antes de mostrar el diseño de la opción "controles". Lo mismo se hará para el menú de pausa.

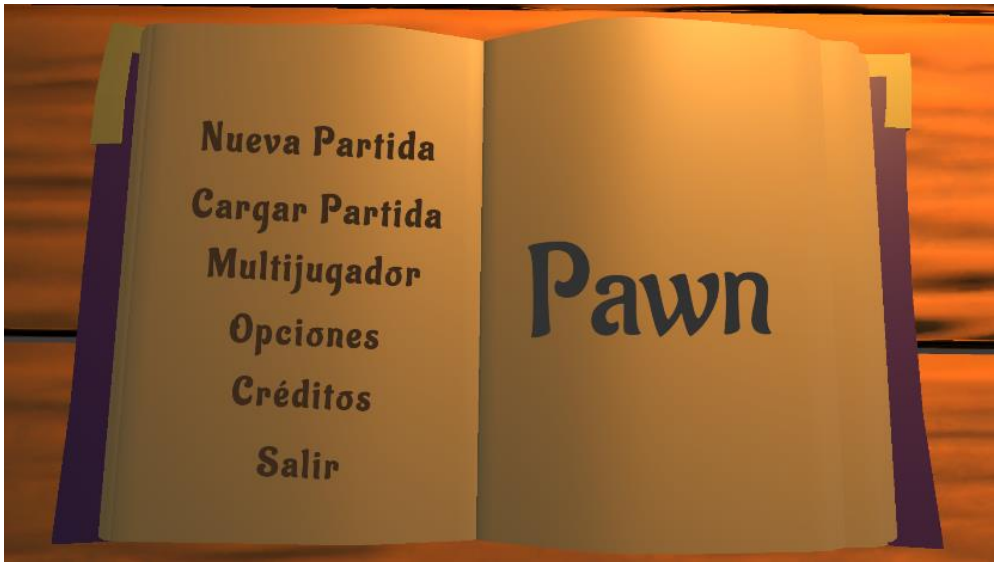


Ilustración 32. Captura del menú principal

En el menú principal tenemos varias opciones, como podemos ver en la Ilustración 32. Captura del menú principal. Si se selecciona la primera de ellas, saldrá una pequeña ventana

[33] Descrito en el glosario de términos [4]

preguntando si se desea o no crear una nueva partida. Si la respuesta es no, se cerrará dicha ventana y no pasará nada; en cambio, si se aceptar comenzar una partida nueva, se comenzará el juego desde el principio del todo.

La segunda opción solamente saldrá una ventana de aviso, diciendo que no hay partida guardada, ya que no se ha guardado ninguna.

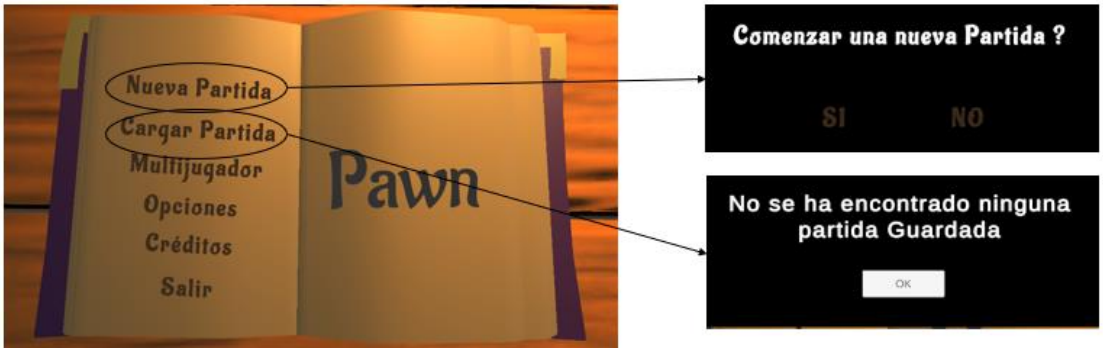


Ilustración 33. Navegabilidad para los botones "Nueva partida" y "Cargar partida"

Si se decide seleccionar la línea de "Multijugador", se abrirá una escena de carga y cuando termine, aparecerá una interfaz en el que se puede crear salas y acceder a otras. El diseño de esta interfaz lo ha diseñado mi compañero Sergio.

La siguiente va a tener un diseño parecido a la opción que hay en el menú de pausa, pero más adelante se mostrará. En este caso, si se decide abrir dicha pestaña, se abrirán distintas opciones: Gráficos, Sonido, Controles y el botón de volver hacia atrás.

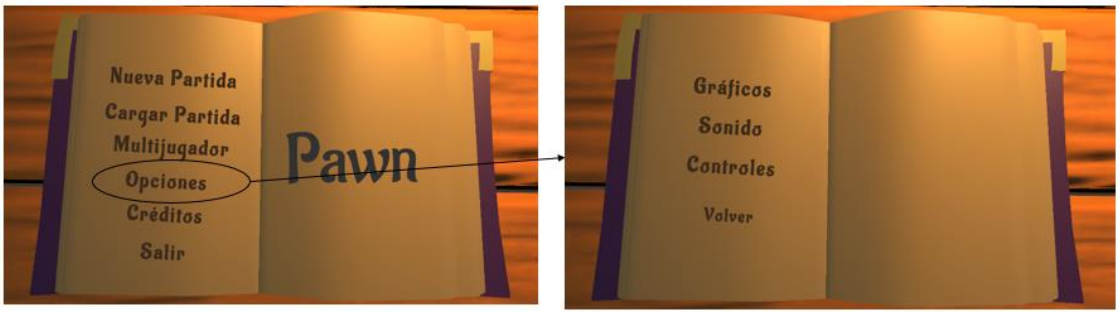


Ilustración 34. Captura de los botones que aparecen al seleccionar "Opciones"

Algo que se va a notar es que la opción de controles tiene un diseño diferente a las otras dos. Esto se debe a que era complicado meter los controles de teclado y mando en la hoja derecha del libro, por lo que se ha hecho es reutilizar el diseño del menú y meterlo en el menú principal. Lo importante, y lo que en realidad se buscaba al añadir esta interfaz, era que el jugador conociese cuales eran los controles, que tuviese un esquema de cómo se juega.

Una vez se ha comentado este detalle, se muestra a continuación la pestaña que aparece al seleccionar la primera opción. En ella, hay un *slider*[34] para ajustar el brillo, aunque actualmente no funciona correctamente; un *dropdown*[35] con las distintas resoluciones que puede tener el juego,

[34] Descrito en el glosario de términos [5]
[35] Descrito en el glosario de términos [6]

siendo 320 x 200 la más baja, y 1360 x 768 la más alta; por último, se muestra la opción de poner la aplicación pantalla completa o no.

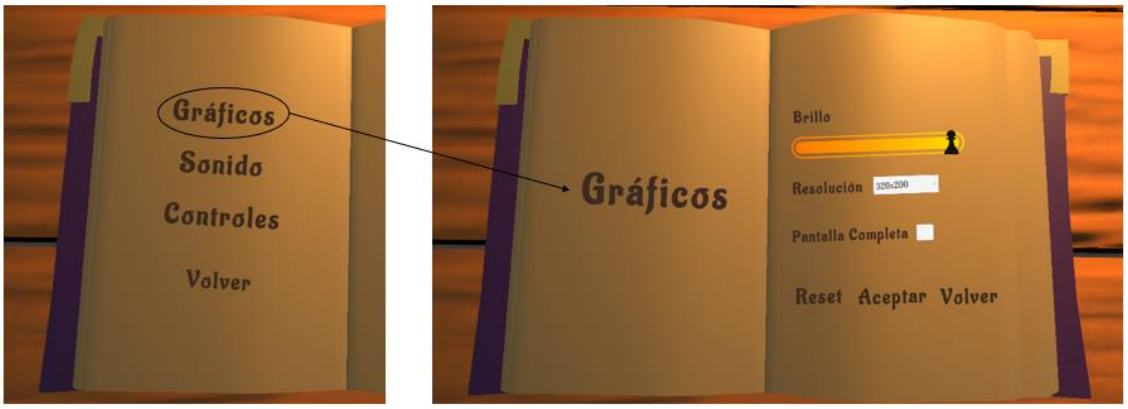


Ilustración 35. Captura de la interfaz de los gráficos

La pestaña de sonido es simplemente para ajustar el volumen Global, ajustando todos al mismo tiempo; la música del juego; los efectos de sonido (SFX); y los sonidos de ambiente, como el viento, el agua o el fuego.

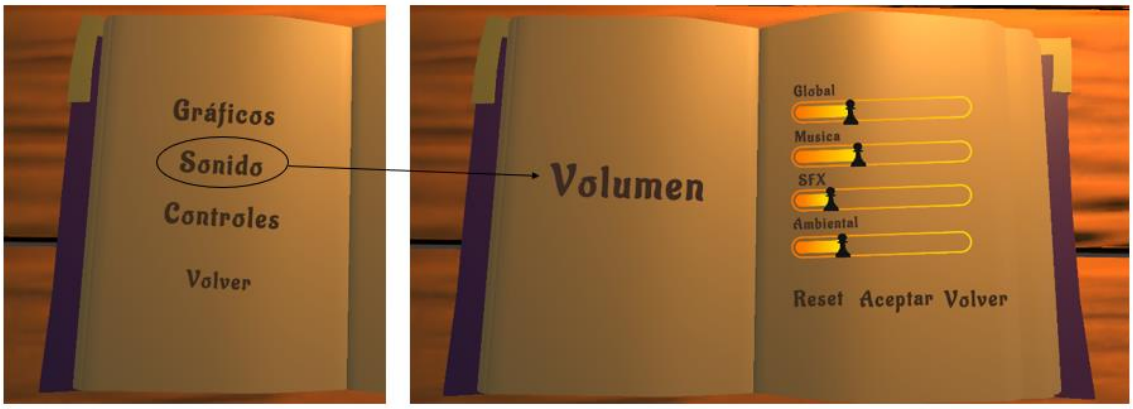


Ilustración 36. Captura de la interfaz del volumen

Por último, la opción controles, que es la que se ha diseñado para este proyecto. Al igual que con la de gráficos, se muestra un *slider* con el nivel de sensibilidad del ratón y el mando, pero por desgracia, por falta de tiempo no se ha podido configurar bien, así que se deja como trabajo futuro terminar su funcionamiento. Debajo de él, hay un *dropdown* donde se puede elegir entre "teclado" o "mando", mostrando la imagen con los controles para ambas opciones, según la que se elija. Estas imágenes se verán mejor en el apartado 5.1.

Ilustración 37. Captura de la interfaz de los controles para teclado

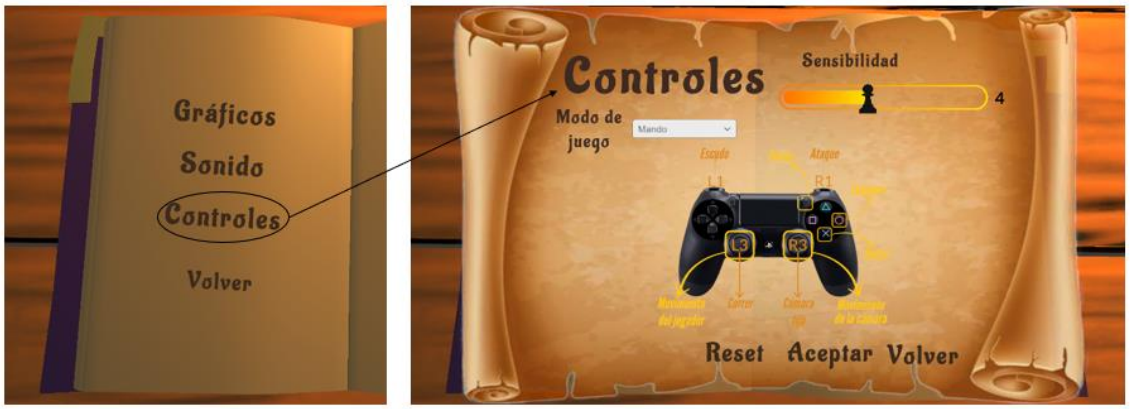


Ilustración 38. Captura de la interfaz de los controles para mando

Los botones "Reset", "Aceptar" y "Volver" de las tres opciones funcionan igual. El primero borra lo que se haya cambiado y se deja por defecto; el segundo guarda los cambios y vuelve a la ventana anterior; y el último botón borra los cambios, dejándolo como estaba antes de tocar nada, y regresa también a la ventana anterior.

Una vez explicado la pestaña de "opciones", la siguiente que hay, "créditos", muestra las personas que han trabajo en el videojuego. Se han añadido tanto los miembros del equipo que había en la asignatura, como personas externas que han ayudado con algún detalle.

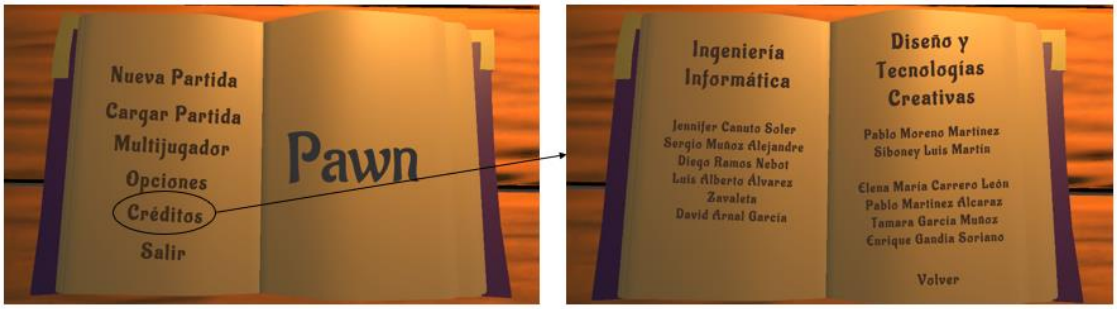


Ilustración 39. Captura de la interfaz de créditos

Por último, el botón de "volver" del menú principal solamente cierra el juego, cerrando la ventana por completo.

Menú de pausa

Para acceder al menú de pausa se tiene que hacer presionando una vez el botón “esc” del teclado, el de “pausa” del mando (en caso de que sea de PS4), “R3” cuando se juegue en realidad virtual (PS4 de nuevo) y la opción de “pausa” para el móvil, situado en la parte superior de la pantalla.

Una vez se abre la pestaña de pausa, se muestran cuatro opciones. La primera de ellas es simplemente para continuar jugando; la segunda para lo que dice su propio nombre, mostrando la misma ventana que la opción de “cargar partida” del menú principal; la siguiente es la misma que el anterior, pero cambiando el diseño; y, por último, la opción que nos llevaría al menú del principio.

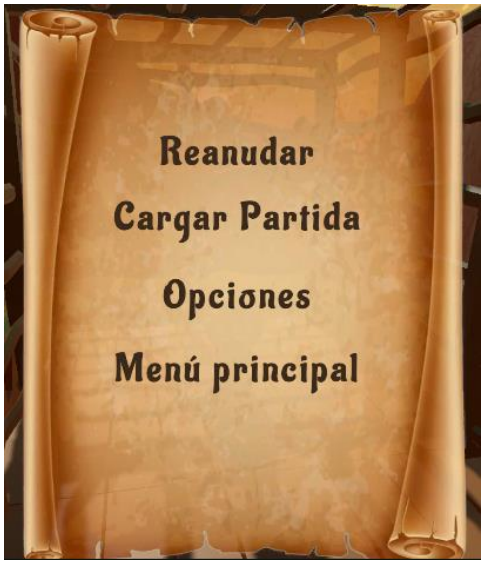


Ilustración 40. Captura de la interfaz del menú de pausa

Ilustración 41. Captura de la interfaz de controles

Icono de la aplicación

El diseño es el mismo para todas las aplicaciones en todas las plataformas. Se basa en un peón, con una corona y ojos de enfado. Esto hace un guiño a la historia principal del juego. Hay que recordar que el objetivo de Pawn es obtener la corona debido a que estaba harto de sentir que los peones eran herramientas de usar y tirar, posicionándolos en una categoría baja. Es por eso por lo que se han usado estos rasgos.

Ilustración 42. Icono que se ha diseñado para la aplicación de Pawn

Se ha realizado en la herramienta web Adobe Express[36], usando la versión gratuita. La razón del porqué se ha usado esta y no otra es por qué resultaba bastante cómodo. Adobe tiene varias opciones y entre ellas estaba la de diseñar logos. Además, al igual que otras herramientas de diseño, cuenta con la distribución de capas, haciendo las modificaciones más sencillas.

Botones del móvil

Se ha usado imágenes de Google para el diseño de los iconos que hay en cada botón y la letra Aladin para las palabras, que es la misma fuente para los menús. El color que se le ha puesto es el que tiene los *sliders* que se ha mostrado anteriormente, con el objetivo de lograr una mejor comodidad visual. En cuanto a los botones en sí, estos han sido diseñados desde el mismo Unity, usando los elementos *button* que encontramos en la pestaña "Component", concretamente en "UI".

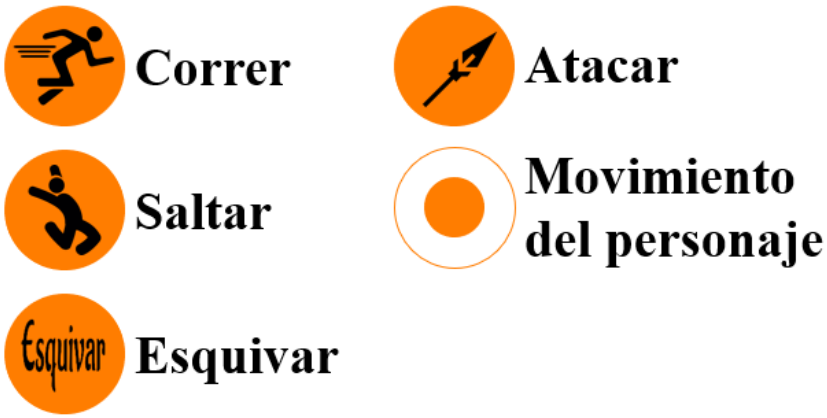


Ilustración 43. Diseño de los distintos botones para el móvil

[36] Adobe Express

Prefabs Endless Runner

A la hora de desarrollar esta parte, se ha tenido que diseñar un total de once *prefabs*, los cuales contienen terreno base, creado en el *prefab* "Terrain 2", y distintos elementos del escenario u enemigos que se han ido diseñando a lo largo de la asignatura o a manos de uno de los miembros del equipo.

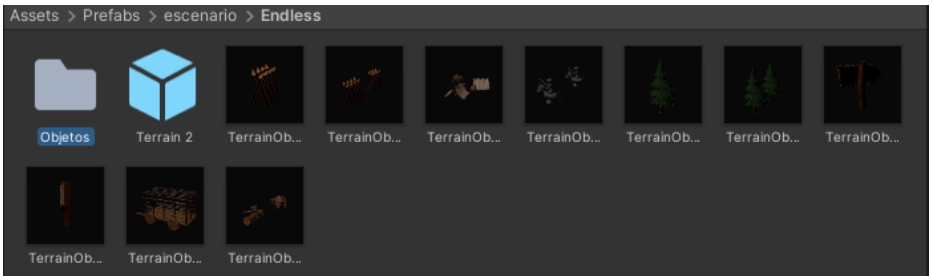


Ilustración 44. Captura que muestra los distintos prefabs creados para la parte Endless runner

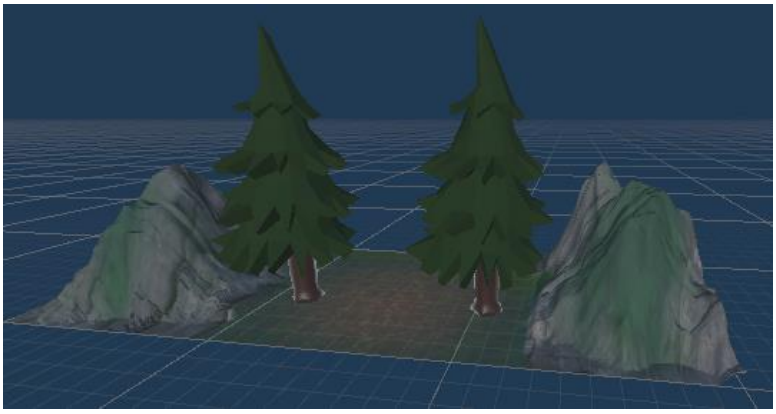


Ilustración 45. Captura de un ejemplo de los prefabs

La idea es crear diferentes obstáculos, pero con la misma dimensión, por eso mismo todos y cada uno de ellos tienen la misma estructura: los objetos que haya dentro de cada uno dependen del terreno, que es el mismo que se ha dicho antes, y su distribución se ha pensado de forma que, a la hora de generarlos en el escenario de manera automática, no salgan 5 seguidos con la misma posición de los objetos, es decir, dejando un camino en medio o los lados libres de obstáculos.

Hasta aquí serían los distintos diseños para tener en cuenta, el resto se verá en los otros trabajos de los miembros del equipo, según lo que haya hecho cada uno de ellos.

5. Desarrollo del videojuego

Para este proyecto, se ha decidido realizar el nivel 2 del videojuego para PC, móvil y realidad virtual en dispositivos Android. Sin embargo, tras varios intentos, las versiones de Unity que se han utilizado para ordenador y Android son distintas. En este primero, se ha usado la 2020.3.14f, que es la que se ha estado utilizando en la asignatura de Desarrollo y videojuegos 3D, pero para el caso de dispositivos Android, se tuvo que optar por crear un nuevo proyecto con una versión inferior, la 2019.4.40f. Esto se debe a la incompatibilidad del Android SDK con Unity. Se intentó con varios SDK's, pero no era posible solucionar el error.

Por lo tanto, teniendo esto en cuenta, el diseño del nivel 2 en ambos dispositivos no son similares, pues al migrar las escenas de un proyecto a otro no se podía copiar el terreno que se había diseñado en el proyecto original, y lo mismo ocurre con el *NavNesh*[37] para la inteligencia artificial. Así pues, las mecánicas funcionan para todas las plataformas que se han nombrado, pero los enemigos no tendrán el mismo funcionamiento que en ordenador, debido a que mi compañero Diego se ha encargado de programarlos en este dispositivo solamente.

Una vez explicado esto, se pasará a desarrollar cada uno aspectos que se han ido desarrollando en este trabajo, pasando primero por la explicación de la jugabilidad del nivel, después por el desarrollo de las múltiples plataformas y terminando con los efectos de sonido.

5.1 Jugabilidad

Lo que llama la atención de un videojuego, a parte de su atractivo visual, es lo entretenido que puede llegar a ser. Hay muchas maneras de hacer que esto sea posible, pero al final lo que acaba llamando la atención del usuario es su jugabilidad. Se puede jugar a uno cuyo diseño sea lo mejor que se ha visto últimamente, pero después a nivel de juego puede llegar a resultar poco llamativo e incluso aburrido. Por eso mismo es muy importante tener en cuenta los demás aspectos de un videojuego, tales como las mecánicas del jugador o su jugabilidad. Así pues, estos serán los detalles en los que nos centraremos a continuación, pero en este apartado específicamente se explicará el desarrollo del nivel 2.

El comienzo de éste es similar al anterior nivel, es decir, su modo de juego es semejante, el jugador tiene que avanzar, eliminando a distintos enemigos dispersos por el escenario. Entre ellos, están los peones, divididos entre los tres tipos que se ha hablado en el apartado 1.3, y los caballos. Éstos no son iguales a los del ajedrez, se les ha dado un estilo propio y se les ha añadido un jinete arriba.

Cada uno de ellos tendrán un patrón de ataque distinto y el jugador deberá evitarlos o derrotarlos para poder llegar al final del escenario, donde hay un caballo, el cual, si se acerca a él, se activará un evento y mostrará la siguiente parte del nivel y la que se ha programado en este proyecto. enemigos hasta llegar a un caballo.

En esta parte los enemigos se convierten en obstáculos y no atacarán al personaje al verlo, por tanto, tampoco podrá él atacar. Lo único que debe hacer es moverse hacia los lados y saltar para evitar chocar con los objetos. Si colisiona con ellos, le quitará un total de 40 entre 100 puntos de vida, de modo que, si los toca 3 veces, estará muerto y se le mostrará el mensaje de muerte.

[37] Descrito en el glosario de términos [7]

Se ha programado de manera que sea el propio escenario el que se mueve hacia el jugador y no al revés. Esto quiere decir que los objetos que componen la escena van a estar en continuo movimiento, a excepción de cuando muere o se pausa el juego. La forma en la que se ha configurado esto es creando un *GameObject*[38] vacío (Camino), con un objeto dentro de su jerarquía (Object).

Ilustración 46. Estructura del objeto que genera el camino

Estos dos elementos se encargan de generar un camino con los *prefabs*, que se han nombrado en el apartado 4.2, de manera completamente aleatoria, de esta forma ningún camino será exactamente igual, cada vez que se reinicie esa parte, habrá un terreno distinto al anterior.

Ilustración 47. Cápura muestra cómo se ha referenciado los prefabs

Esto se logra con el *script* que contiene cada *GameObject* que se ha nombrado. El objeto padre contiene el fichero que se encarga de obtener una cantidad determinada de elementos de tipo Pool, nombre del *script* del objeto Object, y generarlos de manera aleatoria en la escena, posicionándolos en una cola (última línea del siguiente código). De esta manera, el último elemento de la cola, que es el que se posiciona detrás del personaje, se elimina (desaparece) y se posiciona otro al principio de todo.

```
for (int i = 0; i < cantidad; i++)
{
    var elemenTransf = ObjectItems.GetRandom();
    elemenTransf.position = offSet - direccion * separacion * i;
    elemenTransf.gameObject.SetActive(true);
    elemetos.Enqueue(elemenTransf);
```

[38] Descrito en el glosario de términos [8]

```
    }
```

A parte, los elementos que se vayan incorporando a la escena tendrán una velocidad de desplazamiento que hará que se muevan hacia donde está el personaje, dando la impresión de es el mismo jugador el que va hacia delante.

Ahora bien, tal y como se había dicho antes, los objetos con los que se colisione quitarán 40 puntos de vida sobre los 100 que tiene el personaje. Esto se ha hecho a través de *triggers*[39] en los *collider*[40] de cada uno de los elementos del escenario, a excepción del terreno, obviamente. Por tanto, todos ellos tendrán un *script* que determine si el objeto que colisiona con ellos es el jugador o no. En caso de que así sea, se encargará de eliminar la vida correspondiente al personaje, activando la animación de daño y bajando la barra de vida.

```
[SerializeField] private float damage = 40f;
private void OnTriggerEnter(Collider other)
{
    if (other.tag == "Player")
    {
          other.gameObject.GetComponentInParent<PlayerControllerEndless>().Ta
          keDamage(damage);
    }
 }
```

Una vez muerto, no comenzará la partida desde el principio del nivel 2, solamente se reiniciará esa escena hasta que pase un minuto y se cargue la siguiente, comenzando la batalla con el jefe de ese nivel.

5.1.1 Estructura del personaje

Unity se basa en la creación de escenas, estructuradas en diferentes *GameObjects* que, a su vez, están compuestos por diferentes componentes, siendo el más importante el denominado *Transform*[41], encargado de darle al objeto o componente una posición, escala y rotación.

Ya se había dado a entender, pero para tener una mejor idea de cómo funciona la jerarquía de Unity, este tiene una forma de árbol, es decir, hay un padre que contiene distintos hijos, quienes heredan sus variables. Si éste primero se mueve, los que dependan de él también se moverán. Por eso es muy importante que, a la hora de insertar componentes en un *GameObject*, se tenga claro qué componente depende de otro.

Con esto en cuenta, se han creado dos *prefabs* para el desarrollo de las mecánicas del jugador, que se explicarán en los subapartados del punto 5.2. Uno de ellos tiene como arma una lanza y el otro tiene, además, un escudo para defenderse. El uso de estos serán para el nivel 2 y el modo multijugador, respectivamente.

.

[39] Descrito en el glosario de términos [9]
[40] Descrito en el glosario de términos [10]
[41] Descrito en el glosario de términos [11]

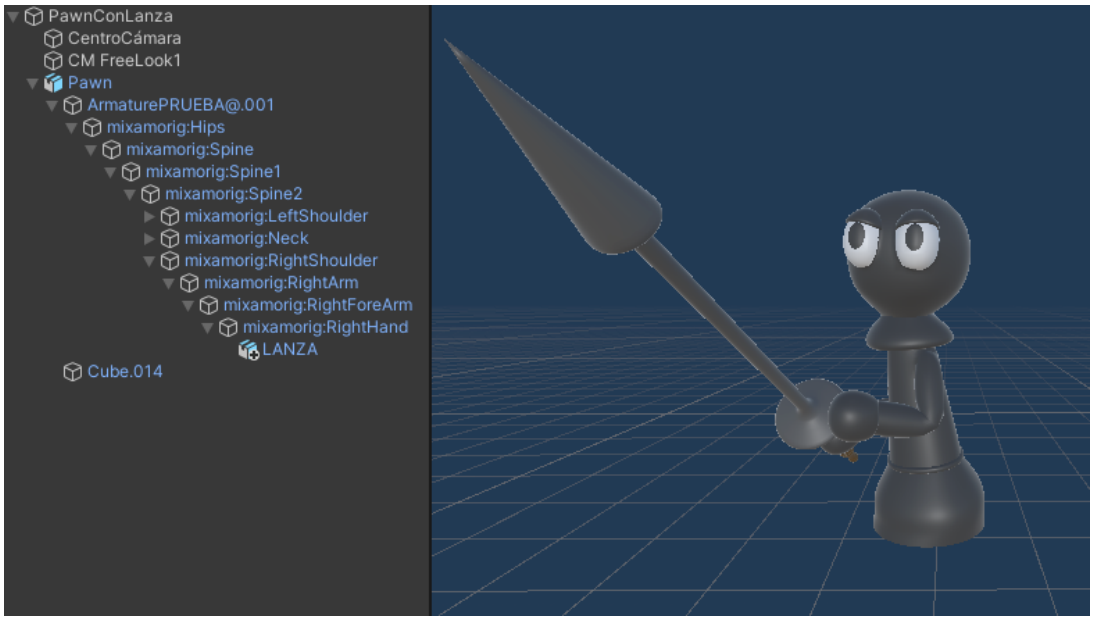


Ilustración 48. Captura de la estructura del prefab "PawnConLanza"

Cómo se puede ver en la Ilustración 48. Captura de la estructura del prefab "PawnConLanza", el *prefab* está compuesto por varios componentes, siendo los dos primeros elementos útiles para el movimiento de la cámara y los otros dos el esqueleto del personaje, modelado en Blender por una compañera externa a este proyecto de fin de carrera.

El script que configura el movimiento del personaje sea cual sea el dispositivo con el que se juegue, está contenido en el objeto “PawnConLanza”, mostrado en la imagen y el de las animaciones en el componente “Pawn”, junto con sus *audio source*[42] y el *script* que configura los sonidos que se efectúan al realizar una animación determinada.

En cuanto a la lanza, este objeto también contiene un *script* que configura el daño que otorga al enemigo cuando colisiona con su *collider*.

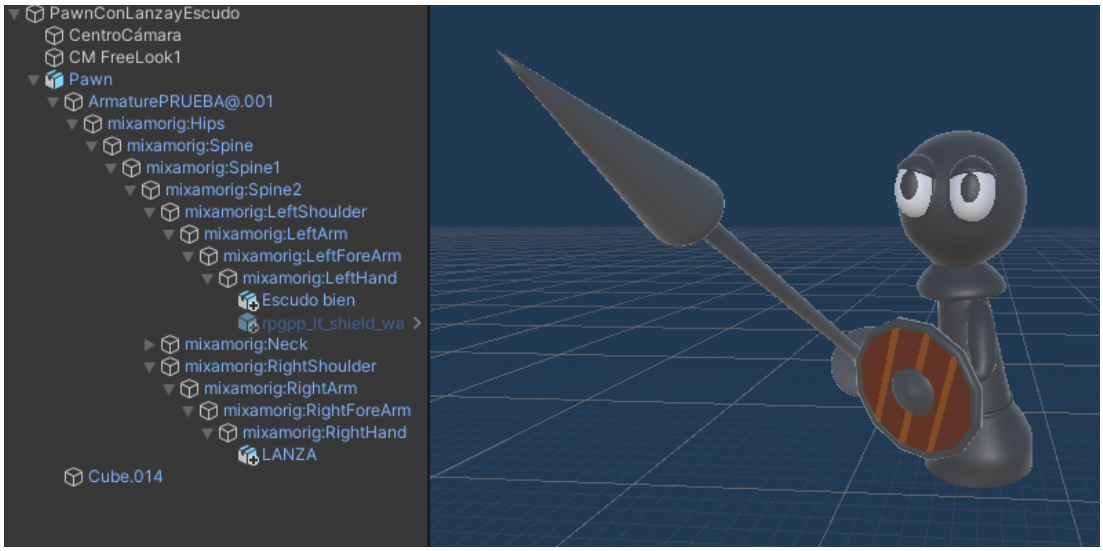


Ilustración 49. Captura de la estructura del prefab "PawnConLanzayEscudo"

[42] Descrito en el glosario de términos [12]

La estructura del *prefab* “PawnConLanzayEscudo” que se puede ver en la Ilustración 49. Captura de la estructura del prefab "PawnConLanzayEscudo" es exactamente igual a la de “PawnConLanza”, pero con tres diferencias; los ficheros de movimiento y de las animaciones son diferentes y, además, se ha añadido al brazo izquierdo un escudo con su *collider*, de tal manera que sí el arma del enemigo colisiona con éste, no efectuará ningún daño en el jugador, pero solamente cuando se esté usando el escudo y el enemigo lo golpeé por delante, si lo hace por la parte de atrás, sí que le quitará vida.

5.2 Multiplataforma

Una vez explicado la jugabilidad del nivel y la estructura del personaje, podemos pasar a comentar cómo se ha desarrollados las distintas plataformas y cuáles son las mecánicas en cada una de ellas.

5.2.1 PC

Lo bueno que tiene Unity es que puedes cambiar de plataformas desde el programa, pudiendo elegir la que desees para compilar tu aplicación. En este caso, para poder jugar a *Pawn* en un ordenador, sea cual sea su sistema operativo, solamente hay que seleccionar la opción “PC, Mac & Linux Standalone” en el apartado “Build Settings”.

De esta manera, el programa creará un ejecutable con las escenas que se hayan añadido a esta misma pestaña. En el caso de este proyecto, para la aplicación de ordenador se han añadido un total de ocho escenas, comenzando con un vídeo donde se puede ver el nombre que se le puso al equipo del proyecto desde el principio, es decir, desde la asignatura de videojuegos; y finalizando con la escena del jefe del nivel 2, donde, una vez derrotado, saldrá por pantalla un mensaje que da a entender que se ha llegado al final de la demo, terminando así el juego. Se puede ver mejor en la Ilustración 50. Esquema de la navegabilidad entre escenas.

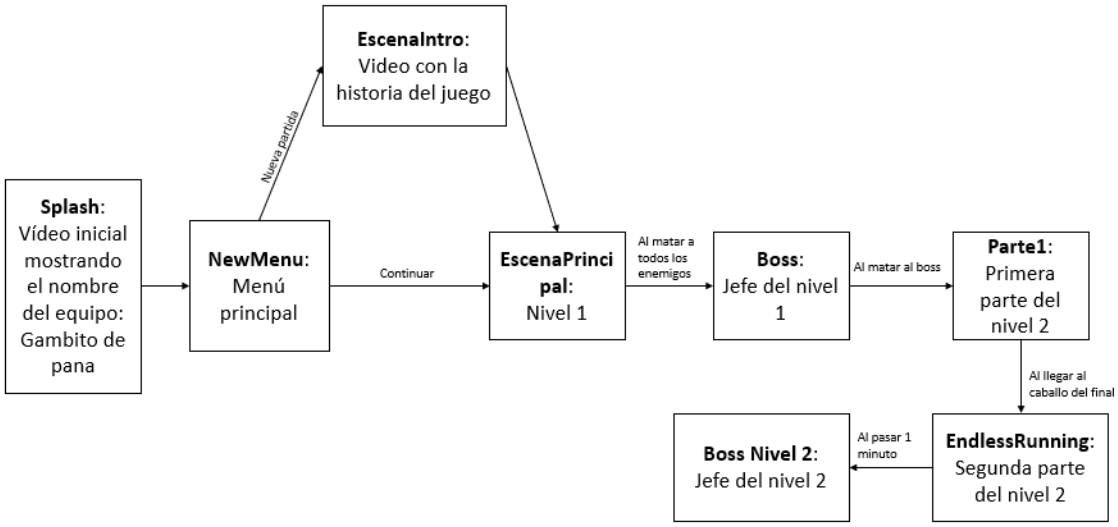


Ilustración 50. Esquema de la navegabilidad entre escenas

Mecánicas

Una vez explicado la navegación entre escenas, se va a pasar a comentar cómo se ha realizado las distintas mecánicas del jugador, tanto para teclado y ratón, como para mando, usando de referencia el de la PS4.

Para ello, se ha usado una unión entre el *Input Manager*, que viene por defecto en Unity, y el *Input System*, paquete que se puede descargar desde la pestaña de "Package Manager". Ambos se encargan de lo mismo, desarrollar los diferentes controles del juego, tanto como movimiento del personaje, como de la cámara. La mejor opción, según se ha investigado, sería *Input System*, debido a que es más actual y mejora bastantes cosas respecto al otro *Input*, sin embargo, no es la que se ha elegido para este proyecto. La razón es debido a los múltiples problemas que podría conllevar a la hora de usarlo. Hay que tener en cuenta que este videojuego se inició como un trabajo de asignatura, por lo que en ese momento no se pensaba usar ningún dispositivo de entrada para poder jugar sin el teclado, así que bastaba con usar el *Input Manager* para configurarlo; es por eso por lo que al intentar cambiarlo por el *Input System*, había que cambiar muchas cosas de los *scripts* y eso podía generar pérdida de tiempo innecesaria.

Al final, se acabó configurando los controles del mando con el *Input* de Unity y se ha usado el otro para poder reconocer si el mando está conectado o no, de esta manera, si se juega a *Pawn*, se podrá jugar con el teclado y ratón todo lo que uno quiera, pero en el momento que se desee conectar el mando, el programa cambiará la configuración de manera automática y establecerá los controles correspondientes a este dispositivo.

```
var gamepad = Gamepad.current;

if (gamepad == null)
{
    ...
}
else
{
    ...
 }
```

Para ello, en el código de los distintos "PlayerController" de cada escena se ha importado el paquete de *Input System* con la instrucción "`using UnityEngine.InputSystem;`" y se ha insertado el código anterior, añadiendo entre los corchetes las instrucciones que capturan el movimiento horizontal y vertical, correspondiendo el dispositivo de entrada que se use, además de llamar al método que ejecuta el movimiento de la cámara, el cual es el mismo para ambos casos.

Algo que tener en cuenta, es que en este proyecto se ha usado como referencia los controles del mando de la PlayStation 4, de tal forma que, si se juega con un mando de la Xbox, por ejemplo, posiblemente el funcionamiento de los botones cambie.

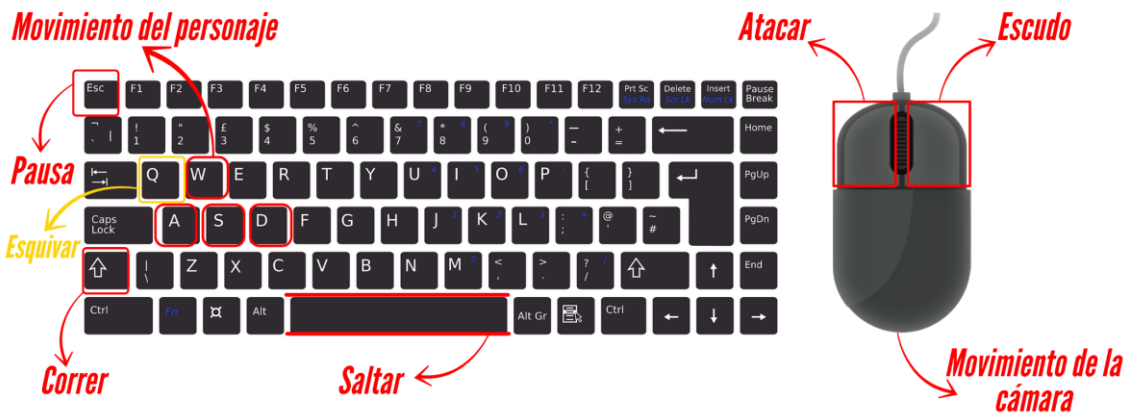


Ilustración 51. Controles para el teclado y ratón

El movimiento del personaje del que se ha hablado se realiza con las teclas W, A, S y D del teclado. De esta manera, cuando se presionan dichas teclas, las instrucciones comentadas en el párrafo anterior se activan y capturan el movimiento. Lo mismo ocurriría con las animaciones, solo que las instrucciones cambian un poco.

En este caso, se crea una condición que confirme si algunas de las teclas anteriores se han activado y, en caso de que así sea, se guarda en un *booleano*[43] la variable *true* o *false* en caso contrario.

Con esto, se puede activar la animación de la máquina de estados del personaje comprobando que dicha tecla está activada o no y que, además, la animación de andar no está en uso. Lo mismo para detenerla, pero comprobando lo contrario a lo anterior.

Respecto al movimiento en el *endless*, hay que recordar que solo se puede mover hacia los lados, de tal manera que las teclas W y S (delante y atrás) no se están habilitadas. Por eso, el contenido de dentro de los corchetes del código anterior cambia un poco. Se captura solamente el movimiento horizontal, para mando y teclado, y se le multiplica la velocidad que se le asigna y el tiempo real entre cada transición (*Time.deltaTime*). Una vez hecho esto, se pone un límite en los movimientos hacia la derecha y la izquierda, para evitar que se salga del mapa generado y no se pueda mover por encima de dichos valores.

Pasando a hablar de mecánicas, éstas tienen una programación muy parecida. Cuando se quiere correr, lo que hay que hacer es mantener presionado la tecla *shift* situado a la izquierda del teclado, de tal manera que cuando eso ocurre, el programa lo capta y llama al método "RunOn()", aumentando por dos su velocidad normal. Al mismo tiempo, el *script* de las animaciones se encarga de activar la animación de correr al detectar la pulsación de la tecla y comprobar que ésta no estaba ejecutándose. Una vez se suelta, llama al método "RunOff()" y deja la velocidad por defecto, desactivando al mismo tiempo la animación de correr.

Si lo que se pretende es saltar, se tendrá que pulsar la tecla "espacio" del teclado, pero solamente se activará el salto al momento de pulsarlo, si se mantiene la pulsación el salto no volverá a realizarse a no ser que se levante el dedo y se vuelva a presionar la tecla.

Esto se ha hecho mediante variables, sin usar ninguna animación. El programa detecta que, si el jugador está tocando el suelo y se presiona dicha tecla, mediante el uso de *Input Manager*, se

[43] Descrito en el glosario de términos [13]

llama al método “Jump()” y se le pasa un valor, correspondiente a la fuerza de salto, a la posición del personaje en el eje y.

En cambio, en la parte del *endless*, el desarrollo del salto se ha hecho con el uso de un *AnimationCurve*[44] con el fin de diseñar la curva que realizará el personaje al saltar, pasándole la duración y fuerza del salto. Para activarlo se hace mediante una corrutina.

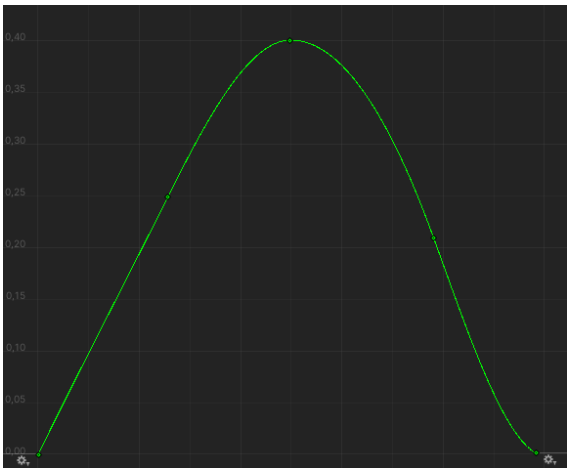


Ilustración 52. Curva diseñada para el salto

Para atacar, hay dos movimientos. El primero es una estocada, presionando una sola vez el botón izquierdo del ratón; el otro ataque es un barrido, pulsando dos veces seguidas dicha tecla. La forma en la que se ha programado esto está en el *script* de las animaciones, de tal manera que se ha creado un entero que guarde las veces que se pulsa la tecla del ratón.

```
if (attackPressed)
{
    if(!inCoroutine) StartCoroutine("ResetPressTime");
    pressTime++;
}

if (cur_health > 0 && esperaClic && !estoyAtacando)
{
    esperaClic = false;
    if (pressTime >= 2)
    {
        animator.SetTrigger("isAttacking");
    }
    else
    {
        animator.SetTrigger("isLanzando");
    }

}
```

Como se puede ver, el código lo que hace es detectar que la tecla de ataque se ha activado, entrando en la corrutina que reinicia el valor de la variable que detecta las pulsaciones, evitando que el numero suba y no se reduzca en ningún momento, esto en caso de no estar ya dentro. Una vez hecho esto, el entero se incrementa en uno y se pasa a comprobar que la vida no es nula y no se encuentra el jugador atacando. Una vez ahí, si el conteo de pulsaciones es igual o superior a

[44] Descrito en el glosario de términos [14]

dos, se activará la animación del barrido, que es la misma que la de la espada del nivel 1; en caso contrario, se activará la de la estocada.

Si se quiere esquivar a los enemigos en vez de atacarlo, se puede hacer presionando la tecla Q. Lo que pasará es que el jugador se moverá un poco hacia atrás, activando la animación de esquivar. Para ello se le pasa al eje z el valor de la fuerza de esquivo, la cual es negativa para que así se mueva hacia atrás.

La última mecánica del jugador que queda por explicar, a parte del movimiento de la cámara, es la del escudo, pero ésta solo se puede usar en el multijugador. Según lo que se ha establecido en el GDD que se puede ver en el Anexo 1, este objeto lo obtiene al final del nivel 2, por lo tanto, solamente se ha añadido para el modo de juego de múltiples jugadores y no en el nivel que se está desarrollando.

De esta manera, cuando se mantenga pulsado el botón derecho del ratón, se realizará una animación donde el brazo izquierdo levanta el escudo, activando el *collider* de este objeto y evitando que cualquier ataque frontal del enemigo haga daño al personaje. Esto se ha hecho activando los *trigger* de los *collider* escudo en el momento que se activa la animación, mediante un *script* que contiene el mismo objeto. De igual forma, cuando se deja presionar dicha tecla, la animación se desactiva, al igual que los componentes que detectan las colisiones.

Por último, quedaría la cámara. Se puede controlar moviendo el ratón hacia los lados, gracias al uso del paquete *Cinemachine*, el cual se ha descargado para configurar mejor la cámara del juego. Éste lo que hace es capturar el movimiento del ratón con el *Input Manager* y moverla según los ejes del dispositivo.

Ilustración 53. Controles del mando para PC

Una vez se ha explicado las mecánicas con el teclado, poco queda explicar para el mando, ya que prácticamente casi todo se ha hecho de la misma forma, pero cambiando parámetros. De igual forma, se hará una breve explicación para cada una.

Para el movimiento funciona exactamente igual que con el teclado, si el mando está conectado, el programa capturará el movimiento horizontal y vertical haciendo uso del *Input Manager*. Lo mismo ocurre con la cámara, ya que esta usa el mismo método y e *input* que el ratón.

En cuanto al salto, al ataque y al movimiento de esquivar, es exactamente igual, pero cambiando las llamadas al *Input Manager* y los parámetros que detectan las pulsaciones de los botones, pero lo demás tiene la misma estructura que se ha comentado antes, incluso la tecla correr y el escudo.

5.2.2 Android

Para poder hacer una aplicación para dispositivos Android, lo primero que hay que hacer es instalar los módulos que dan soporte para la compilación en Android.

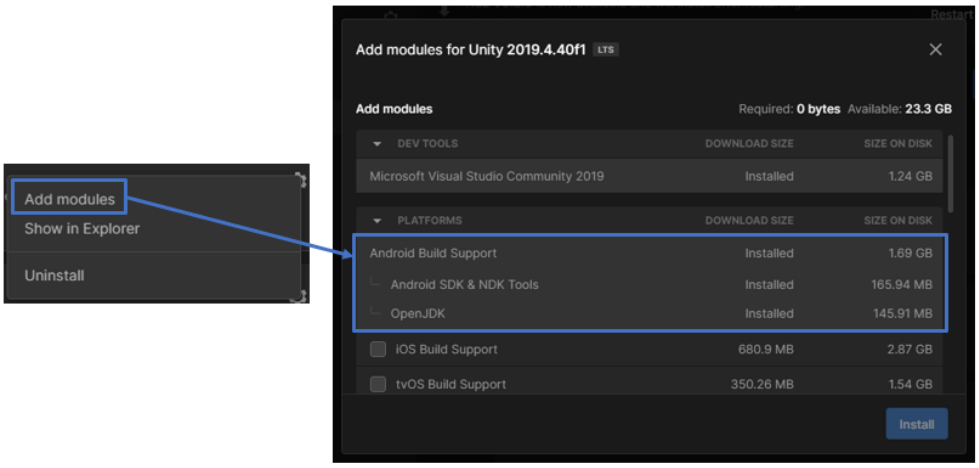


Ilustración 54. Demostración de cuáles son los módulos de Android

Sin esto, prácticamente es imposible crear la aplicación, ya que se tendrán que poner las distintas herramientas de Android manualmente y esto al parecer puede generar conflictos y errores que a lo largo de estos días no ha sido posible solucionar, debido a que volvía a salir otro error. Además de esto, también se ha tenido que especificar en la pestaña "Project Settings" cuál es la versión mínima de Android que puede soportar el videojuego. En este caso ha sido la API 24, correspondiendo a la versión 7 de Android.

A la hora de crear la aplicación del móvil, se han seleccionado un total de cinco escenas, comenzando por el vídeo que muestra el nombre del equipo y terminando con la escena del jefe del nivel 2. En este caso no se ha añadido el nivel 1, debido a lo que se ha comentado antes respecto al terreno, y en cuanto al a la opción de "Nueva partida" del menú principal, ésta abrirá la escena del nivel 2 en vez del anterior.

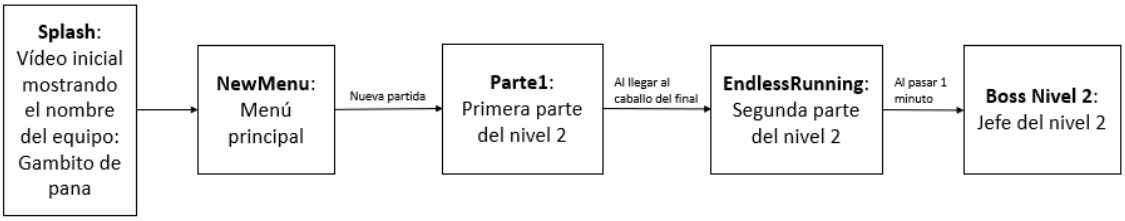


Ilustración 55. Esquema de la navegabilidad entre escenas para la aplicación de móvil

Con esto, se puede realizar ya la compilación e instalación en un dispositivo móvil, conectándolo mediante USB. Al principio, no reconocerá el dispositivo, por ello hay que activar las opciones de desarrollador. Los pasos que seguir para activarlas son similares en todas las marcas, pero al mismo tiempo puede llegar a ser distinto. Como se ha usado un Xiaomi para la

realización de las pruebas en este proyecto, se explicará cómo se ha hecho en este dispositivo, teniendo en cuenta que para los demás son de manera similar.

Lo primero que hay que hacer es ir a la aplicación de Ajustes y entrar en el apartado donde se especifique la información del dispositivo. Una vez hecho se pulsa varias veces donde ponga "Versión de MIUI" hasta que diga que ya se han habilitado las opciones de desarrollador. En ese momento ya se puede activar la depuración USB para que Unity reconozca el dispositivo, además de activar la instalación vía USB.

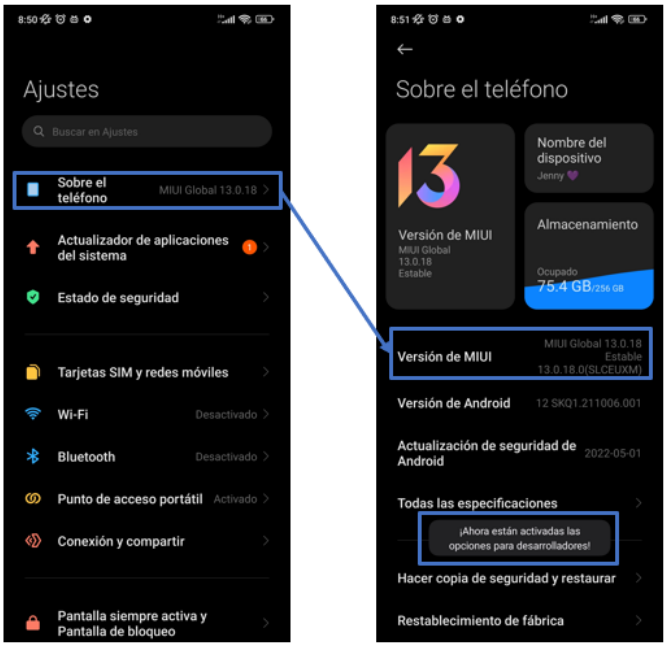


Ilustración 56. Pasos para activar la depuración USB (1)

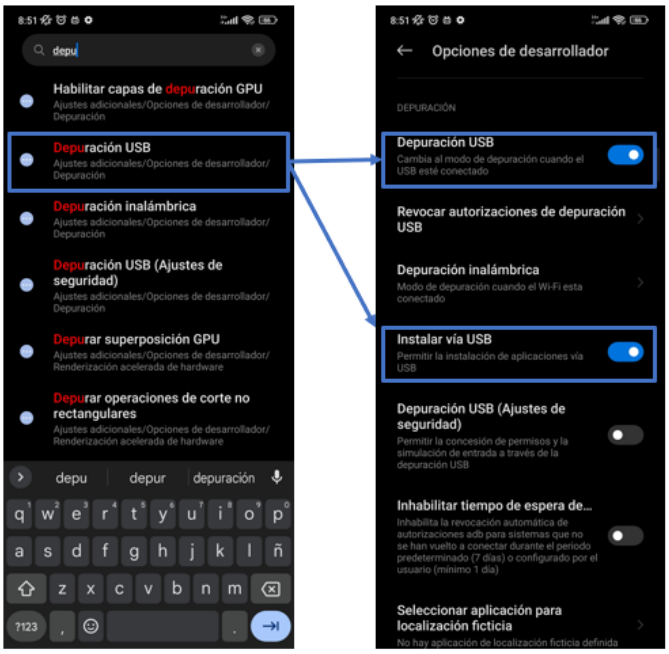


Ilustración 57. Pasos para activar la depuración USB (2)

Una vez hecho esto, y se haya terminado la instalación, aparecerá en la aplicación en la interfaz del móvil.

Mecánicas

En cuanto al desarrollo de las mecánicas en un dispositivo Android se ha hecho uso de los botones de Unity y de un *joystick*[45] que se ha descargado de la tienda online. El paquete es completamente gratuito y se llama "Joystick Pack" [5].

Ilustración 58. Captura del videojuego para mostrar la interfaz del móvil

Debido a que se ha reutilizado el mismo proyecto para la aplicación de realidad virtual, se ha tenido que crear un objeto vacío que sirva para establecer la plataforma que se va a usar. Por tanto, dentro de ese objeto hay que asignarle el *script* "GameManager", heredando el mismo nombre que se le asigna a dicho objeto.

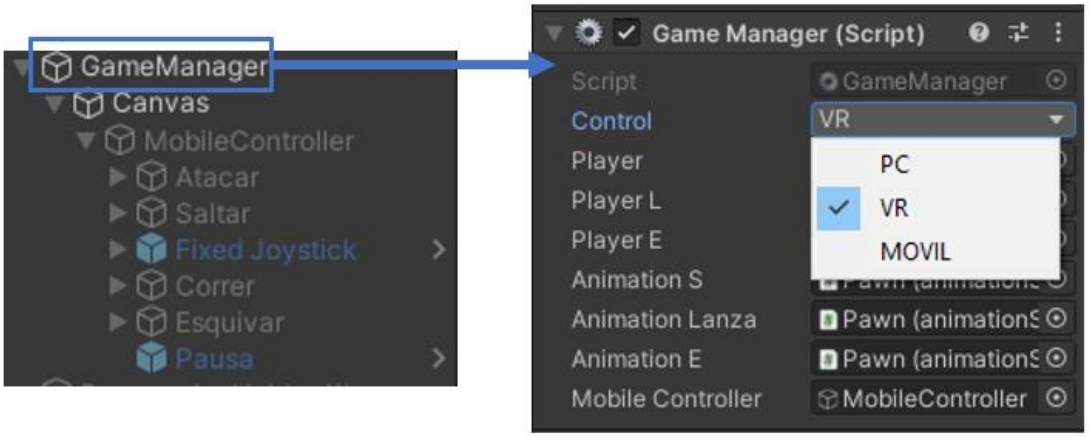


Ilustración 59. Componente GameManager

[45] Palanca de control que permite desplazar algún objeto en concreto.

Si se selecciona la opción VR, se desactivará la interfaz del móvil y se activarán los controles del mando en caso de que esté conectado por bluetooth al dispositivo, manteniendo pulsado al mismo tiempo los botones "share" y el que tiene el icono de la PlayStation (esto en caso de mando de PS4). La opción MOVIL simplemente activa los botones y sus funciones.

Con ayuda de dicha interfaz, podremos mover al personaje moviendo simplemente el *joystick* de la izquierda. Al ser perteneciente de un paquete, éste viene configurado con un *script* que logra que el círculo del medio se mueva cuando se arrastre. Para que el jugador se pueda mover junto a él, se ha capturado el movimiento horizontal y vertical de dicho *joystick* y se le ha pasado al *CharacterController*[46] del personaje. Esto es lo mismo para el nivel 2 entero.

En cuanto a las habilidades, si se mantiene pulsado el botón de correr, se lanzará un evento que active al mismo tiempo el método "RunOn()", multiplicando por dos la velocidad normal, y el método "CorrerMovilOn()", el cual activa la animación de correr. Una vez se suelta, se activan los métodos que se encargan de devolverle el valor de la velocidad por defecto y el que desactiva la animación.

Los botones de atacar y esquivar también funcionan mediante eventos, de tal manera que, si se pulsan, se activan sus respectivos métodos. Para el ataque solamente se activará la animación, mientras que, al esquivar, se le asignará al eje z del jugador un valor que lo mueva hacia atrás y se activará la animación de manera simultánea.

Por último, el botón de saltar. Éste funciona exactamente igual que con los controles del PC, pero haciendo uso de un *script* que determina cuándo se pulsa y cuándo no.

```
[HideInInspector]
public bool Pressed;

public void OnPointerDown(PointerEventData eventData)
{
    Pressed = true;
}

public void OnPointerUp(PointerEventData eventData)
{
    Pressed = false;
}
```

Con esta variable, se puede llamar al método "Jump()", haciendo una referencia al *script* y llamando al booleano "Pressed". Si detecta una pulsación, la variable se iguala a *true* y el salto se activa; una vez deje de pulsarlo, se cambiará el valor a *false*, para evitar que el personaje siga saltando a pesar de no estar pulsando nada.

5.2.3 Realidad virtual

Como se trata de una aplicación de móvil, los módulos del apartado anterior han de estar instalados junto con el resto de los aspectos que se han nombrado. Una vez hecho esto, hay que activar la opción "Virtual Reality Supported" en la pestaña "Preferences", además de las *Cardboard*.

[46] Descrito en el glosario de términos [15]

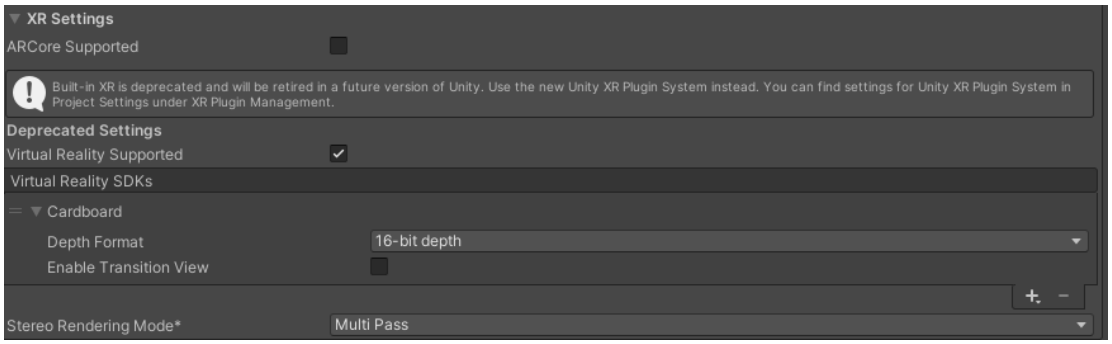


Ilustración 60. Captura donde muestra la opción que hay que activar para la realidad virtual

De manera adicional, se ha de importar la última versión SDK de Google VR para Unity [6] mediante las pestañas "Assets > Import Package > Custom Package". Una vez instalado, aparecerá una nueva opción "Google VR" en el menú superior de Unity. Con todo esto, ya podemos comenzar a configurar el juego para realidad virtual.

Antes que nada, para esta aplicación habrá que compilar un total de cuatro escenas, concretamente el video inicial con el nombre y las tres del nivel 2. El menú principal no se ha puesto debido a que no se ajustaba del todo bien, ya que se ha diseñado principalmente para ordenador.

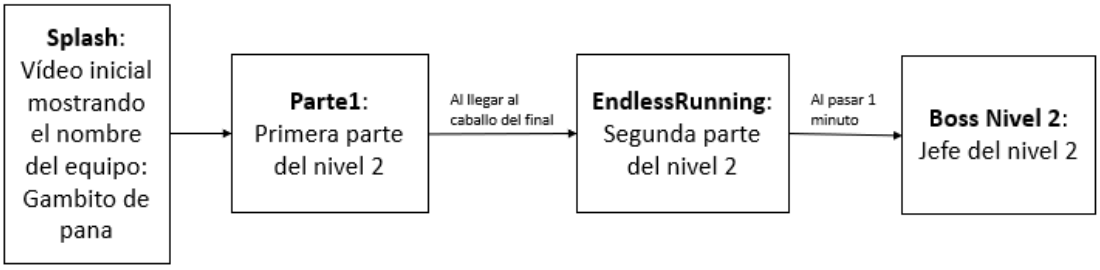


Ilustración 61. Esquema de la navegación entre escenas para la aplicación de realidad virtual

Tal y como se ha puesto la opción MOVIL en el *GameManager* para el apartado anterior, en este caso habrá que seleccionar la que pone VR. Una vez hecho la interfaz se desactiva y se capturan los controles del mando.

Mecánicas

En este caso, los controles cambian un poco respecto a la versión de PC, seguramente se deba a la versión de Unity, pero se ha dejado tal y como está.

Ilustración 62. Controles para la realidad virtual

Igualmente, la configuración mediante código de estos controles sigue siendo exactamente la misma que en PC, no se ha hecho de manera diferente ya que se sigue usando el *Input Manager*. Por tanto, lo que se ha cambiado para esta versión es la cámara. Se ha puesto en primera persona para que parezca que el jugador es el propio Pawn y sea, de esa manera, más inmersivo. La cámara se ha vuelto fija, ya no hay movimiento si se mueve el joystick del mando, pero sí que se mueve por todos los ángulos al mover la cabeza. Esto es gracias al paquete de Google VR SDK.

Una vez desarrollada las mecánicas, y comprobado que todo funciona correctamente, se instala la aplicación en el móvil tal y como se ha dicho en el apartado anterior. Al abrirla se verá el juego duplicado, con bordes negros alrededor. Esto es para que, a la hora de ponerse las gafas y ajustarlas de manera correcta, el juego se vea como si estuviese el propio jugador dentro de él.

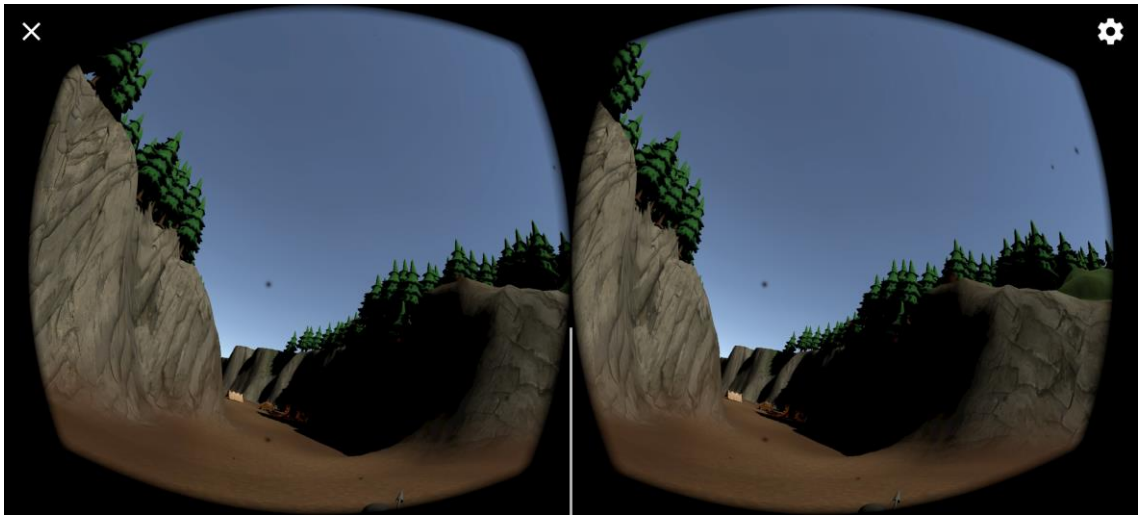


Ilustración 63. Captura del videojuego en VR

5.3 Efectos de sonido

Un videojuego no es lo mismo si no cuenta con efectos de sonido. Sin ellos puede llegar a ser aburrido o extraño, así que al fin y al cabo termina siendo un aspecto importante del cual no se puede olvidar.

En Unity, para que el audio se reproduzca, se tiene que hacer mediante el uso de *audio source* y *audio listener*[47]. Éste último debe ser un componente de la cámara, para que así se pueda escuchar todo sonido que haya en la escena. No se puede poner más de uno, en cambio, los *audio source* pueden haber más de uno, es más, es lo que se debe hacer si un mismo objeto contiene más de un sonido.

En este apartado se dividirán los efectos de sonido en tres categorías: Ambientación, Enemigos y Jugador. Hay un *audio mixer*[48] creado para la ambientación, otro para la música y el último para los efectos de sonido, metiendo en este a los enemigos y al jugador. Con esto, se logra que las barras de volumen del menú funcionen de tal manera que, si se desliza hacia un lado u otro, el audio baje o aumente para cada uno de ellos.

5.3.1 Ambientación

Para darle una mejor ambientación al juego, se ha creado una serie de componentes que reproducen la música de fondo y los distintos sonidos del ambiente. En primer lugar, la canción que se escucha de fondo, tanto para el menú principal como para los niveles, se ha sacado de un paquete gratuito que había en la tienda de Unity [7]. Se ha elegido la temática medieval por ser la época en la que está inspirado el juego.

En cuanto a los sonidos del ambiente, como los árboles y la hoguera, se han sacado de herramientas de internet, pero la más utilizado ha sido *Zapsplat*, de la que se ha hablado en el aparto 4.1.

A la hora de incorporarlos en el juego, para el caso de la música de fondo y los sonidos del bosque, se ha creado un objeto vacío que contenga dos *audio source*, uno para cada uno, asignándoles el *audio mixer* Música y Ambiental, respectivamente.

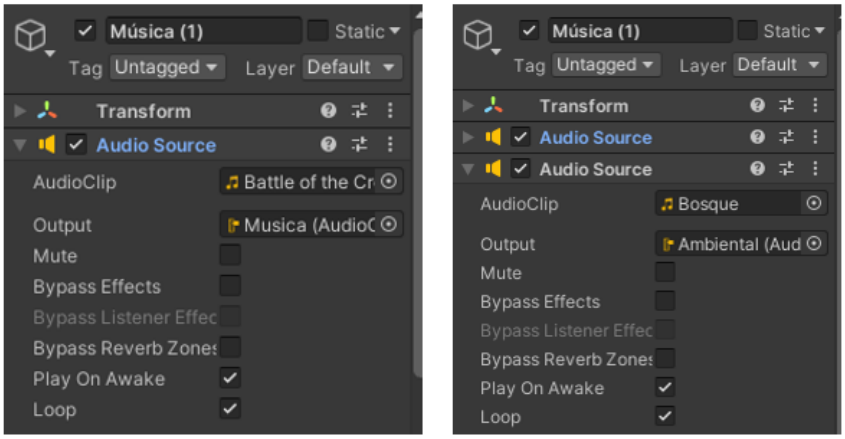


Ilustración 64. Componentes audio source del objeto Música

[47] Descrito en el glosario de términos [16]
[48] Descrito en el glosario de términos [17]

Se han activado las opciones “Play on awake” y “Loop” para que una vez se inicie la escena, el audio que se le ha asignado a “AudioClip” comience a escucharse en bucle.

Respeto a los otros sonidos del ambiente, las hogueras, por ejemplo, contienen un *script* que cogen el audio que se ha descargado para dicho objeto y se lo pasa al *audio source* que tiene como componente. En este caso, la opción “AudioClip” está vacío hasta que se inicia el juego, donde el código se encarga de asignarle el audio correspondiente. El *audio mixer* asignado es el mismo que a la música de ambiente.

5.3.2 Enemigos

Los efectos de sonido de los enemigos están dentro del *audio mixer* “Sfx”. Cada tipo de enemigo tiene su propio *script* y la estructura es igual en todos, pero cambiando los audios y algunas instrucciones.

```
public enum Sonidos
{
    ORBE,
    CAMINAR,
    MUERTEENEMIGO,
    OUCH,
    TOTAL_SONIDOS
}

public AudioSource[] audios;
private float cur_health;
private bool alreadyDead = false;
public Animator animator;

string[] nombreSonidos = { "Orbe", "Caminar", "MuerteEnemigo", "Ouch" };
// Start is called before the first frame update
void Start()
{
    audios = GetComponents<AudioSource>();
    animator = GetComponentInParent<Animator>();

    for (int i = (int)Sonidos.ORBE; i < (int)Sonidos.TOTAL_SONIDOS; i++)
    {
        audios[i].clip = Resources.Load<AudioClip>(nombreSonidos[i]);
    }
}
```

Se crea una enumeración que corresponde al audio que se le ha de asignar al objeto y se guarda cada uno de ellos dentro de un array de *audio source*. Para activarlos se hace mediante las animaciones, es decir, en el momento que el programa detecta que el enemigo que contiene dicho *script* ha realizado una determinada animación, se inicia el audio correspondiente, y una vez deja de realizar la animación, se pausa.

```
if (animator.GetBool("isWalking") || animator.GetBool("isRunning"))
{
        if (!audios[(int)Sonidos.CAMINAR].isPlaying) {
            audios[(int)Sonidos.CAMINAR].Play();
        }
}
else {
        if (audios[(int)Sonidos.CAMINAR].isPlaying) {
            audios[(int)Sonidos.CAMINAR].Pause();
        }
```

```
}
```

5.3.3 Jugador

Para el caso del jugador es exactamente igual que con los enemigos. El personaje contiene un *script* que, según la animación que se active, se ejecuta un audio u otro. Los *audio source* de éste están dentro del *audio mixer* "Sfx" y no contienen ningún "AudioClip" hasta que el propio programa se lo pasa al iniciar la escena.

La estructura también es similar a la de los enemigos. Se crea una enumeración y se asigna cada audio a una lista. Con esto, cuando el personaje comienza a andar, se le asigna al componente de reproducción de audio el clip que corresponde a este movimiento; lo mismo ocurre para cuando se ejecuta un ataque, un salto o cuando el protagonista se muere.

Para este caso se ha usado solamente un *script* para todos los modelos del personaje, usando las instrucciones de "*try catch"* para detectar qué diseño se está usando para capturar los puntos de vida, de esta manera se puede ejecutar el audio de muerte en caso de que esta variable llegue a cero.

```
try
{
      cur_health =
      transform.parent.gameObject.GetComponent<PlayerController>().cur_he
      alth;
}
catch (NullReferenceException) { }

try
{
      cur_health =
      transform.parent.gameObject.GetComponent<PlayerControllerMulti>().c
      ur_health;
}
catch (NullReferenceException) { }

try
{
      cur_health =
      transform.parent.gameObject.GetComponent<PlayerControllerLanza>().c
      ur_health;
}
catch (NullReferenceException) { }

try
{
      cur_health =
      transform.parent.gameObject.GetComponent<PlayerControllerEndless>()
      .cur_health;

      jumping =
      transform.parent.gameObject.GetComponent<PlayerControllerEndless>()
      .Jumping;
}
   catch (NullReferenceException) { }
```

6. Conclusiones

Tras la realización de este proyecto se han podido realizar los distintos objetivos que se han planteado al principio de la memoria. Ha habido bastantes obstáculos de por medio, problemas que parecían difíciles de solucionar, pero que, por suerte, se ha dado con la solución correcta. Se ha podido crear una aplicación para las plataformas que se han planteado, incluso se ha configurado los controles para el mando en caso de que el usuario desee jugar con este dispositivo en vez del teclado. Es cierto que hay cosas que estarían bien de pulir, pero eso se hablará en el siguiente punto.

El desarrollo del nivel 2 ha dejado un buen sabor de boca, se ha cumplido con las expectativas que se tenía desde el principio. La funcionalidad y jugabilidad de la segunda parte de este nivel ha resultado divertida y funciona como se deseaba desde el principio.

Se ha podido explicar, de manera que se pueda entender, los pasos que se han realizado para la configuración en realidad virtual y Android, incluso se ha dicho cómo instalar la aplicación mediante Unity para realizar las distintas pruebas.

En cuanto a los efectos de sonido, el resultado también ha cumplido con las expectativas iniciales del trabajo. Se ha logrado que los sonidos sean envolventes, que el usuario pueda escuchar lo que ve en pantalla.

Lo que más tedioso ha podido resultar ha sido la compilación en dispositivos móviles. Durante estos años de carrera no se han establecido los conocimientos necesarios para crear una aplicación en el móvil, mucho menos con Unity, ni siquiera en la asignatura de videojuegos, por tanto, ha resultado ser un problema. No obstante, gracias a los foros de esta herramienta y la amplia información que hay sobre ella, se ha podido terminar el trabajo, cumpliendo con los objetivos establecidos y obteniendo un buen resultado respecto al proyecto desarrollado.

7. Trabajo futuro

El objetivo del grupo de desarrollo es acabar el. Se ha hablado entre los miembros del equipo y todos están de acuerdo en que la idea es atractiva. Por lo tanto, se deja como pendiente terminar los niveles que falta, sus enemigos, sus mecánicas, sus afectos de sonido, mejorar la eficiencia del código, depurar mejor el proyecto entero, entre otras cosas más. El objetivo es mostrar el videojuego al público, con lo que se debe tener una versión limpia y mejor desarrollada que la actual.

Habrá que mejorar problemas que no se han solucionado, conflictos respecto a versiones de Unity, intentar usar la misma para todas las plataformas y añadir otras más, como la incorporación de las *Oculus* para la realidad virtual. Ahora que el tiempo ya no se echa encima, se puede tomar con más calma y poder matizar todos estos aspectos.

A parte, también se planea desarrollar otros videojuegos, sean de distinto género o no, pero la idea es seguir por este camino.

8. Referencias bibliográficas

[1] Simone Belli y Cristian López Raventós (2018). Breve historia de los videojuegos. Artículo de la revista *Athenea Digital*.

[2] Jose Luis Eguia Gómez, Ruth S. Contreras-Espinosa y Lluís Solano-Albajes (2013). Videojuegos: Conceptos, historia y su potencial como herramientas para la educación. Publicado en la revista de investigación *[3]Ciencias.*

[3] Francisco Pérez Martínez (2011). Presente y futuro de la Tecnología de la Realidad Virtual. Publicación de la revista *Creatividad y Sociedad.*

[4] Alba Mora (21 febrero 2022). Los mejores cascos de realidad virtual del momento. Artículo de *PCWorld.*

[5] Paquete Joystick Pack de la tienda de Unity. https://assetstore.unity.com/packages/tools/input-management/joystick-pack-107631

[6] Google VR SDK para Unity. Se ha instalado la versión v1.200.1. https://github.com/googlevr/gvr-unity-sdk/releases

[7]Paquete RPG Medieval Themes de la tienda Unity. https://assetstore.unity.com/packages/audio/music/orchestral/rpg-medieval-themes-196607#reviews

[8] Documentación de Unity. https://docs.unity3d.com/Manual/index.html

Glosario de términos

[1] **Gamepad**.

Dispositivo de entrada utilizado para jugar a videojuegos en diferentes dispositivos. Se pueden conectar por cable o de manera inalámbrica.

[2] **Endless Runner**.

Se trata de un modo de juego en el que el jugador solamente podrá moverse hacia los lados (en caso de que se trate de un juego 3D) o saltar. No tendrá que moverse hacia delante, ya que lo hace de manera automática.

[3] **Script.**

Fichero que contiene el código de programación necesario.

[4] **Prefab**.

Permite almacenar diferentes componentes y propiedades en un solo objeto padre, de tal manera que se pueda usar en distintas escenas.

[5] **Slider**.

Barra deslizadora que permite al usuario establecer un valor determinado y pasárselo al programa.

[6] **Dropdown**.

Desplegable con diferentes opciones a seleccionar.

[7] **NavNesh**.

Área en el que se puede mover un objeto. Usado para la inteligencia artificial.

[8] **GameObject**.

Hace referencia a todo objeto que se encuentre en la escena.

[9] **Trigger**

Eventos usados para llamar métodos o activar algún elemento del programa.

[10] **Collider**

Componente que sirve para detectar colisiones entre objetos.

[11] **Transform**

Componente importante v encargado de darle al objeto una posición, escala y rotación.

[12] **Audio Source**

Reproduce el audio que le asigne.

[13] **Booleano**

Tipo de variable con dos valores: true y false.

[14] **AnimationCurve**

Lo que hace es establecer una línea que el jugador debe ejecutar en el momento que se le ordena. Se puede hacer rectas, curvas, parábolas, etc.

[15] **CharacterController**

Componente que controla la aceleración y movimiento del personaje en el escenario, sin necesidad de usar físicas.

[16] **Audio Listener**

Actúa como un micrófono. Recibe los clips de los *audio source* y reproduce el sonido para que se escuche en el juego.

[17] **Audio Mixer**

Mezclador de audio. Se puede ajustar el volumen y los efectos de audio a todo clip que se catalogue con el mismo *audio mixer*.

Anexo I: GDD

1. Datos principales

Título: “Pawn”.

Concepto: Eres Pawn, un peón al servicio de sus reyes. Cansado de las injusticias y la guerra constante a la que eres sometido, decides empezar tu propia guerra, esta vez, contra tu ejército, y así derrocar al rey con tus propias manos.

Tema: Ajedrez, mundo distópico medieval.

Género: Plataforma 3D, arcade, aventura.

Plataforma: PC, Android, Realidad virtual para dispositivos Android.

Mercado: Todos los públicos.

PEGI: Edad mínima recomendada: 7 años. Esto es debido a que, pese a no haber violencia explícita, sangre o palabras malsonantes, sigue siendo un juego cuyo objetivo es matar, aunque sean figuras de ajedrez.

2. Pitch doc

2.1 Resumen

En el siglo VI d.C. estalló una guerra entre dos bandos. Las piezas blancas intentaban conquistar el territorio de las piezas negras. Pawn, un peón negro, está harto de tanta violencia y de ser utilizado por el Rey a su conveniencia. Un día empieza a cuestionarse por qué pelea y llega a la conclusión de que la disputa ya no tiene sentido. Tanta es su determinación que incluso decide rebelarse y acabar con la guerra cueste lo que cueste. Incluso si ha de matar a su Rey con sus propias manos.

2.2 Gameplay

El juego empieza manejando a Pawn, concretamente dentro de una jaula de la que habrá que escapar. Tras las puertas de esta se encuentra un camino lleno de enemigos y obstáculos que habrá que superar para poder avanzar hacia el objetivo inicial, el castillo del Rey.

Será en tercera persona y para moverse se usarán las teclas de dirección (WASD), además, se podrá saltar (espacio), atacar (botón izquierdo del ratón), esquivar (Q) y bloquear (botón derecho del ratón).

Cada nivel tiene al final un jefe representativo del nivel, siendo correspondiente a la jerarquía del ajedrez: Peón, Caballo, Alfil, Torre, y Dama.

De igual forma, cada nivel tendrá un escenario diferente, donde se encontrará a enemigos de niveles inferiores, además de los del propio nivel.

Al acabar con el jefe de cada nivel se recibirá un objeto o habilidad como recompensa.

2.3 Estética

El juego está ambientado en los campos de batalla medievales, donde el objetivo de cada ejército era conquistar el castillo enemigo.

El estilo escogido es *cartoon low poly*, con colores variados y brillantes, basados en una paleta de colores concreta. Es importante recalcar que Pawn es un peón negro, al igual que todos los enemigos a los que se enfrentará, por ello es importante tener en cuenta los colores que podrían contrastar en la ambientación

Los personajes son figuras representativas del ajedrez caricaturizadas.

3. Análisis competitivo

3.1 Referentes

A nivel jugable: *Crash Bandicoot, Spyro, Ratchet & Clank*.

Ilustración Anexo 1. Captura del videojuego Crash Bandicoot

A nivel estético: *Fall guys, Human: Fall Flat.*

Ilustración Anexo 2. Imagen promocional de Fall Guys

3.2 Competencia

Además de los juegos nombrados con anterioridad, a nivel de jugabilidad podemos destacar también los siguientes títulos: *Super Mario (64, Galaxy, 3D World), Sonic Generations, Jak and Daxter*. Esto sin tener en cuenta los juegos de *plataforma 2D*, de los cuales hay una gran variedad actual.

Ilustración Anexo 3. Captura del juego de Jack and Daxter

Ilustración Anexo 4. Captura del juego de Super Mario Odyssey

3.3 Características diferenciadoras:

Estilo artístico *low poly*, figuras caricaturescas.

Los personajes están basados en las figuras del ajedrez. Protagonizado por un peón que, por una serie de sucesos, se rebela contra el sistema soberano. Una referencia clara es la novela de George Orwell, *1984*.

Además, no se ha podido encontrar ningún juego en el que los protagonistas sean figuras de ajedrez, sin contar el ajedrez en sí.

4. Estructura Narrativa

4.1 Deseo principal:

El deseo principal del protagonista, Pawn, es avanzar entre las líneas enemigas (antes aliadas) con el objetivo de derrocar al Rey y vencer a todo su ejército.

4.2 Actos narrativos:

Inicio: Pawn comienza con un cuento ilustrado en las páginas de un libro que dará contexto de la historia principal del videojuego.

Primer nivel: Pawn se encuentra en medio de un campo de batalla con el objetivo de derrotar a todos los enemigos del nivel para así avanzar hasta la caballería.

Cuando se derroten a todos, en la villa del final aparecerá un peón gigante aparecerá con el que tendremos que pelear. Este enemigo corresponde al jefe del primer nivel.

Segundo nivel: Tras su derrota, se pasará al siguiente nivel, con una lanza como arma.

Este nivel está dividido en dos partes. La primera será similar al nivel 1, pero añadiendo nuevos enemigos (los caballos); la segunda corresponde a un modo de juego que se denomina *endless runner*, basado en correr hacia delante de manera automática, sin parar, pudiendo solo moverse hacia los lados y saltar. Si se colisiona con un obstáculo, quitará vida al personaje. Con tres golpes está muerto.

Al pasar un minuto este nivel terminará y aparecerá un peón montado a caballo. Este es el jefe de este nivel, el cual correrá hacia el personaje e impactará contra él con su lanza. Tras derrotarlo, se conseguirá un escudo como recompensa.

Tercer nivel: Comienza estando fuera de las murallas que custodian el castillo real. Aquí habrá que vencer a los enemigos que aparezcan en oleadas para detenerlos. Entre los enemigos se podrá encontrar peones, caballos y alfiles. Estos últimos se dividen en dos tipos: alfil verde y alfil rojo. El verde se encarga de curar a otros enemigos o protegerse con un escudo mágico para que no le ataquen; el rojo lanza orbes de energía que harán daño a Pawn mientras retrocede cada vez que se acerca el protagonista.

Tras superar varias oleadas de enemigos mientras se avanza hacia la muralla del castillo, aparecerá el jefe alfil, al derrotarlo se verá una cinemática donde lo lanzaremos con nuestra lanza contra la pared de una de las torres, rompiéndola y creando un agujero que permitirá avanzar al siguiente nivel. El ítem que dejará este *boss* es una mitra obispal que permitirá al jugador realizar un doble salto.

Cuarto nivel: Al adentrarse en la torre a través del hueco surgido en la anterior batalla, deberá subir las escaleras en espiral que hay en la torre, derrotando a los enemigos y saltando los espacios vacíos entre escalones y obstáculos en el camino. Los enemigos que hay son todos los anteriores, además de unos peones subidos en pequeñas torres que disparan cañones que empujan hacia atrás al personaje, pero si se usa el escudo se podrá amortiguar el impacto.

Al llegar a la cima de la torre se encuentra con otra torre que será el jefe de este nivel. Tras acabar con ella se obtendrá un pequeño cañón como arma adicional a distancia.

Quinto nivel: Una vez en la cima de la torre, se tendrá que bajar hasta el patio de armas. Para ello habrá que buscar la manera óptima de hacerlo mediante saltos.

Al llegar al patio de armas habrá que enfrentar a oleadas de enemigos y, entre ellas, aparecerán 3 jefes que son: un caballo, un alfil y una torre. Cada uno dejará como recompensa por derrotarlo una parte de una llave. Al tener las 3 partes, se unirán para poder acceder al castillo real.

Sexto y último nivel: Tras derrotar a los enemigos anteriores, se podrá acceder a las puertas del castillo. Una vez dentro se verá una cinemática donde se encuentran el Rey y la Dama, al percatarse de la presencia del jugador, la Dama apartará al Rey para enfrentarse a él y, de este modo, comenzar la pelea final contra el último jefe del juego.

Desenlace o final del juego: Cuando la Dama es derrotada, se verá una cinemática donde el Rey demuestra ser un cobarde al huir. En su huida deja caer la corona real y se podrá ver cómo Pawn se acerca y la recoge, acabando el juego con la imagen del protagonista alzando la corona en sus manos, dudando entre acabar con la figura del monarca definitivamente, o simplemente ocupar el trono que acaba de quedar vacío.

5. Guion de la historia principal

Aparece un libro con ilustraciones y un texto en la parte inferior que narrará la historia.

Primera página del libro:

- Se ve una imagen del campo de batalla, con las piezas blancas y negras peleando.

- Narrativa en formato novela visual (el texto se vería en la parte inferior de la pantalla):

Texto 1 → Corría el siglo VI d.C. Las piezas blancas ansiaban más tierras de las que tenían y decidieron invadir el territorio de las piezas negras.

Texto 2 → El bando atacado, como es normal, no aceptó y estalló la guerra.

Texto 3 → La guerra duraría años, e incluso milenios...

Segunda página:

- Se ve una imagen de Pawn.

- Narrativa:

Texto 1 → Esta historia cuenta las aventuras de Pawn, un peón negro sin ninguna habilidad especial que le diferencie de los demás peones.

Texto 2 → Pawn, al contrario que sus compañeros, tenía unos ideales diferentes. No le bastaba con ser una herramienta que pudiera morir con el fin de proteger al Rey.

Texto 3 → Él quería ser más valioso, ser algo más en la vida, que la sociedad le reconociera y le recordaran como una gran figura de ajedrez.

Tercera página:

- Imagen de Pawn dando una charla a sus compañeros.

- Narrativa:

Texto 1 → Harto de ver morir a tantos de los suyos, compañeros y compañeras que habían dado su vida a una causa injusta. Pawn había tomado una decisión…

Texto 2 → Una revolución junto a sus compañeros sería el golpe perfecto para derrocar a su vil rey, pero…

Cuarta página:

- Imagen de Pawn encerrado.

- Narrativa:

Texto 1 → Pero Pawn no contaba con que sus propios aliados lo delataran como traidor y lo encarcelaran por su conspiración contra el Rey.

Quinta página:

- Imagen de Pawn mirando al castillo desde su jaula.

Texto 1 → Decepcionado y aislado de todos, su visión del mundo cambió.

Texto 2 → Él acabaría con la guerra a cualquier precio.

Sexta página:

- Imagen de Pawn mirando una espada apoyada contra la rueda de la jaula en la que se encuentra encerrado.

Texto 1 → Destruiría a todos aquellos que se pusieran en su camino hasta su propio Rey y así arrebatarle la corona para finalizar la guerra de una vez por todas.

Texto 2 → Esa sería su meta en la vida. La partida acababa de comenzar...

Finaliza la narrativa y comienza el juego → Se cierra el libro y se va oscureciendo la pantalla para luego mostrar el primer nivel.

6. Flowchart de la narrativa:

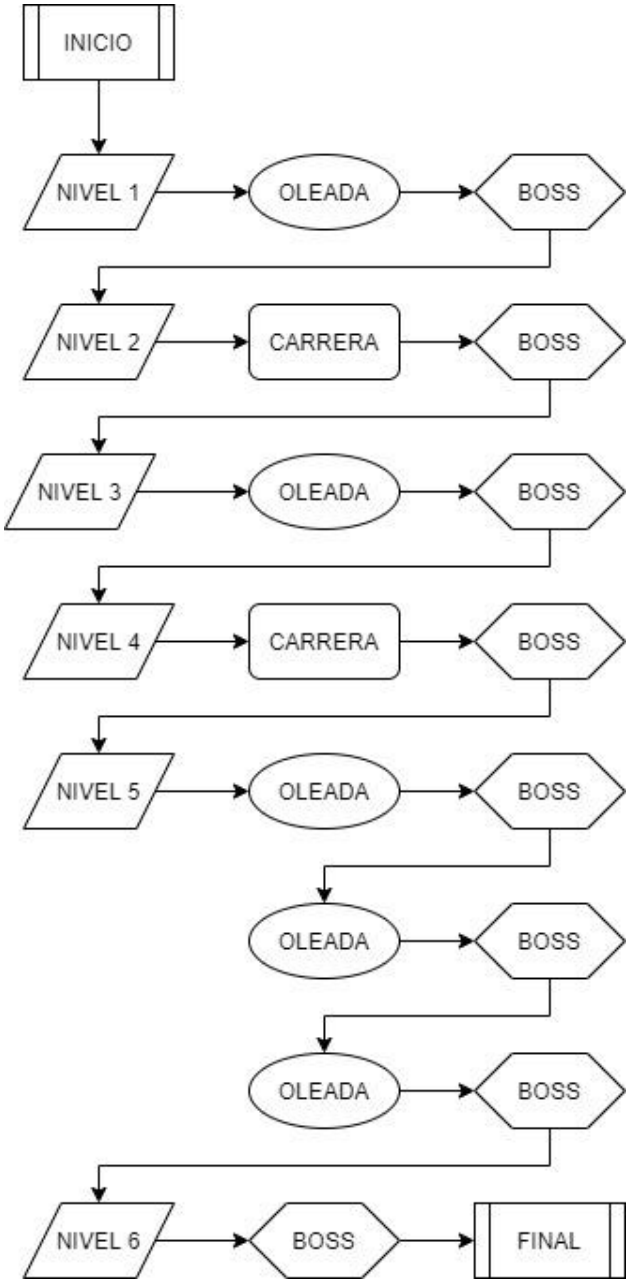


Ilustración Anexo 5. Flowchart de la narrativa

7. Curvas de interés:

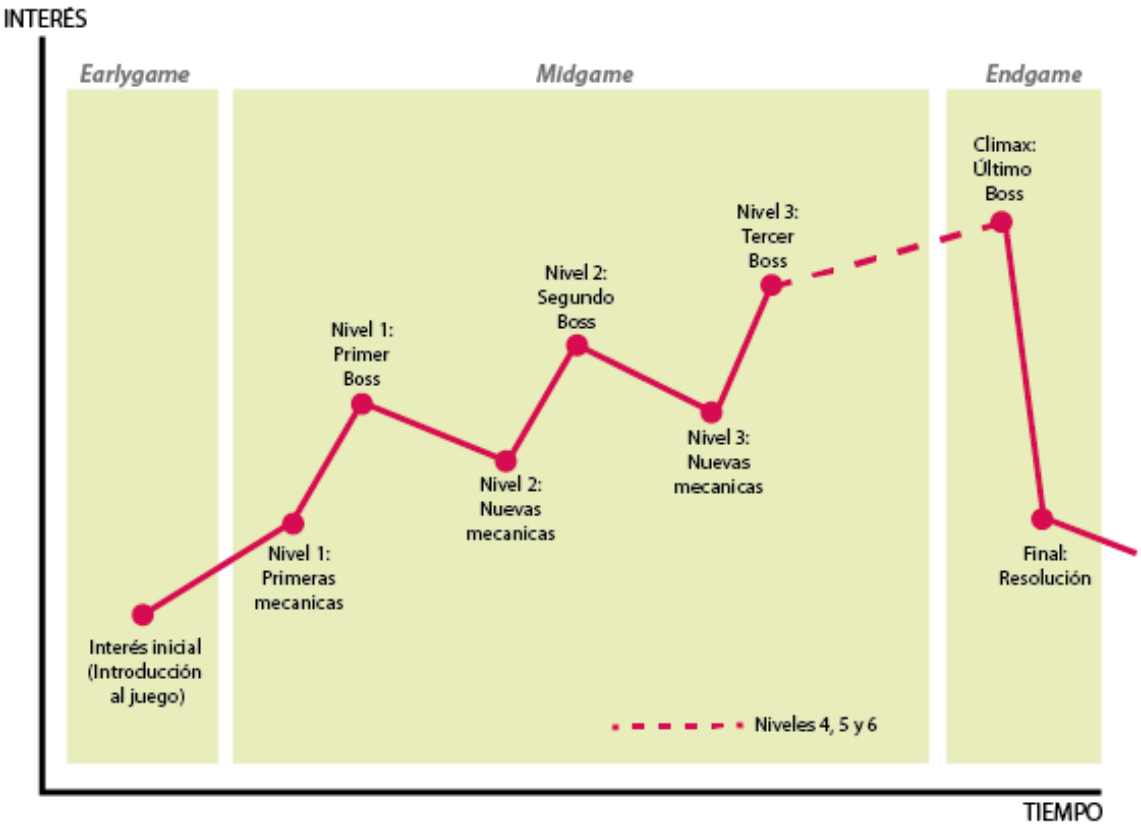


Ilustración Anexo 6. Curvas de interés

Earlygame:

Interés inicial, este inicio estaría vinculado al prólogo. Es muy importante crear un marco de referencia y, generar así, un lugar idóneo para que los jugadores puedan comenzar su aventura.

Midgame:

Primera toma de contacto real con el juego, las primeras mecánicas de este, primeras interacciones con el mundo y, por supuesto, los primeros enemigos. Se nos presentan los retos, misiones y recompensas. Una vez que hemos pasado este primer "subidón", pasamos un periodo de "relax transitorio", que nos prepara para las siguientes experiencias venideras.

Endgame:

Y en la última fase, en el fin del juego, es donde se decide todo. Vivimos un gran apogeo donde nos enfrentamos al reto final para conseguir la recompensa definitiva. Clímax, solución final y epílogo. El cierre de nuestro arco argumental, aquí tendremos el mayor punto de interés y donde se verán auténticamente reflejadas las intenciones de nuestro protagonista, así como la verdadera personalidad de nuestro Rey.

8. Personajes

8.1 Protagonista:

Pawn

Es un peón negro que quiere derrocar a su propio rey. Físicamente, es un peón clásico del tablero de ajedrez, pero caricaturizado y dotado de expresión facial gracias a unos grandes ojos, también tiene dos brazos, pero no tiene piernas. Puede moverse hacia delante y los lados dando pequeños saltitos, además puede saltar más alto para superar obstáculos.

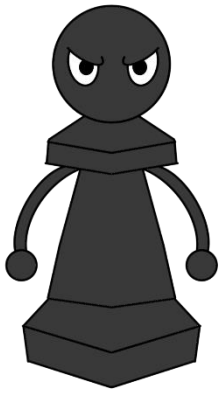


Ilustración Anexo 7. Diseño de Pawn

Sus ataques varían a lo largo del juego y son los siguientes:

Nivel 1: Pawn cuenta con una espada y al pulsar el clic izquierdo del ratón podrá atacar con ella. El ataque es un barrido horizontal.

Nivel 2: En este nivel se cambia de arma a una lanza y se añade un nuevo ataque: un clic del ratón será una estocada y dos un barrido. La segunda parte del nivel irá montado en un caballo, donde podrá moverse solamente horizontalmente y saltar.

Nivel 3: Pawn ahora tiene un escudo (clic derecho). Al usarlo, el personaje quedará completamente inmóvil.

Nivel 4: Sigue teniendo los ataques nombrados anteriormente, así como su escudo, la novedad en este nivel es que al saltar (espacio) se podrá volver a saltar en el aire, realizando un "doble salto".

Nivel 5: En este nivel se puede usar un pequeño cañón que Pawn llevará bajo el brazo. Existe un *delay* entre disparos, este ataque es el que llega más lejos, pero también tiene más daño que los anteriores, aunque es más lento. Para apuntar se usa el clic derecho y para disparar el clic izquierdo. Cuando tenga equipado el cañón no podrá usar el escudo.

8.2 Antagonistas:

Peones:

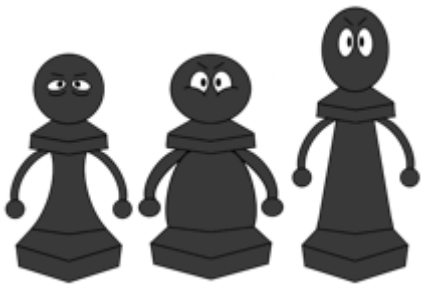


Ilustración Anexo 8. Diseño de los peones

Hay cuatro tipos de peones diferentes. Tres de ellos son peones básicos, la diferencia entre ellos y Pawn es que su físico: más delgado, más alto o rechoncho.

El ataque es igual que el ataque inicial de Pawn, pero atacarán de uno en uno dejando un margen de reacción al jugador. Cada uno de estos peones tendrá una característica diferente según su físico, siendo el delgado más rápido, el alto más fuerte y el rechoncho tiene más vida.

El cuarto tipo de peón es el jefe, que es mucho mayor en tamaño, fuerza y vida que los anteriores. Tiene pinchos que cubren su cabeza, cuello y base, además de unos guantes. Las características de sus ataques son los siguientes:

- **1ª fase (100% HP)**: Entra con una maza y sus ataques son poderosos y desde arriba.
- **2ª fase (50% HP)**: Se deshace de su espada y empieza a saltar y dar cabezazos contra el suelo para intentar aplastarte. Entre saltos habrá un tiempo en el que estará cansado y no podrá atacar de nuevo.

Ilustración Anexo 9. Diseño del peón jefe

Caballos:

Existen dos tipos de caballos diferentes en el juego: los estándares y el jefe. Lo componen el propio caballo y un peón jinete cuya arma es una lanza.

El caballo básico perseguirá a Pawn y saltará encima de él.

En cuanto al jefe, es mucho mayor en tamaño, fuerza y vida que los anteriores. Las características de sus ataques son los siguientes:

- **1ª fase (100% HP)**: Cuenta con una lanza, sus ataques son estocadas y saltos que irá intercambiando. Correrá hacia el personaje y se deberá conseguir que se golpee contra los pinchos de la pared para inmovilizarlo y atacarle.
- **2ª fase (50% HP)**: Lanza su arma, la cual comenzará a dar vueltas y perseguirá el jugador. Hay que seguir con lo mismo, hacer que se haga daño con los pinchos.

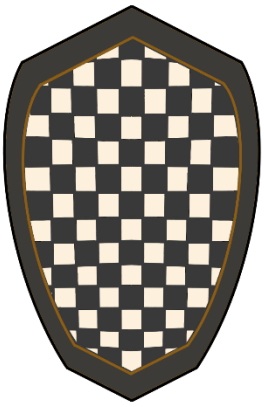


Ilustración Anexo 10. D

Alfiles:

Existen 3 tipos de alfiles diferentes en el juego. Los alfiles son peones que han dedicado su vida al servicio de Dios, lo que comúnmente conocemos como Obispos. Estos tienen una mitra en la cabeza y su arma es un báculo.

Habrá un alfil verde y uno rojo. El primero curará a los enemigos mientras se protege del jugador; el rojo escapará de él al mismo tiempo que le lanza orbes de energía.

En cuanto al tercer alfil, este corresponde al jefe, el cual es mucho mayor en tamaño, fuerza y vida que los anteriores, las características de sus ataques son los siguientes:

Ilustración Anexo 11. Diseño del alfil

- **Pre-Fase**: Aparecen alfiles y peones (devotos), y no sabes cuál es el alfil jefe. Una vez derrotados todos, se abren las puertas de la muralla y aparece realmente el Alfil Jefe.
- **1ª fase (100% HP)**: Su arma es un báculo, es rápido y además salta más alto que nosotros.

- **2ª fase (50% HP)**: Aparecen 4 alfiles que empiezan a curar poco a poco al alfil jefe y tendremos que acabar con ellos primero mientras evitamos los ataques del jefe (máximo 75% HP).

- **3ª fase (25% HP)**: Comienza también a hacer *dashes* para atacar, como los del Caballo Jefe, pero mucho más cortos y rápidos.

Torres:

Existen 2 tipos de torres diferentes en el juego. Están compuestas por una torre con un peón en su interior que dispara tres cañones. Son personajes inmóviles que se puede encontrar en el nivel y cuyos ataques hay que bloquear con el escudo. Habrá que subir las plataformas de alrededor suyo para llegar al botón de arriba y golpearle.

El último enemigo de este tipo es el jefe del nivel. Es mucho mayor en tamaño, fuerza y vida que los anteriores, y además puede moverse, las características de sus ataques son las siguientes:

- **1ª fase (100% HP)**: Su arma es un cañón, y tendremos que aprovechar el doble salto junto a los escombros que vayan cayendo para poder acercarnos a la cabeza y golpearla.

- **2ª fase (50% HP)**: Podrá realizar ráfagas de cañonazos que tendremos que evitar posicionándonos detrás de escombros.

- **3ª fase (25% HP)**: Balas explosivas que tienes que esquivarlas, en caso de cubrirte te harían mitad de daño y destruirá los escombros.

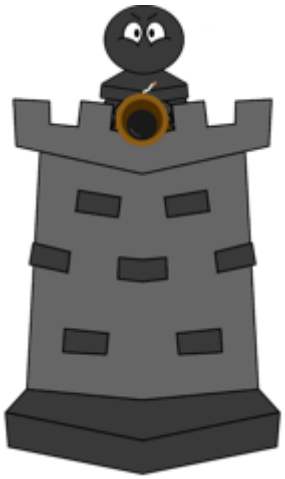


Ilustración Anexo 12. Diseño de la torre

Dama:

La dama es la figura más poderosa, solo hay una. Tiene una apariencia parecida a la de un peón, pero más esbelta y con una corona y un cetro mágico. Además, tiene mucha más vida que los demás.

- **1ª fase (100% HP)**: Sus armas son unas dagas que lanzará a distancia. Realizará un ataque giratorio si te acercas demasiado.

- **2ª fase (50% HP)**: Realiza una serie de ataques con orbes de energía mientras gira. Habrá que evitarlos para acercarse a ella y atacar cuando sea vulnerable.

- **3ª fase (25% HP)**: Se subirá un balcón y habrá que subir unas plataformas para llegar a ella y atacar. Mientras tanto, lanzará rayos en la dirección del jugador.

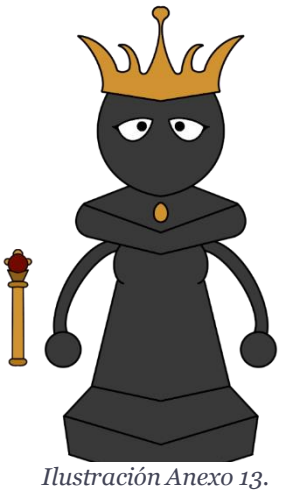


Ilustración Anexo 13. Diseño de la dama

Rey:

El rey lleva una larga capa roja y una corona de oro, es orgulloso y narcisista. Cree firmemente que la vida de los demás solo vale para proteger la suya, pero cuando no queda nadie para defenderlo, este huye, ya que en el fondo es un inútil y un cobarde, dejando atrás su corona y, por ende, renunciando a su reinado.

Ilustración Anexo 14. Diseño del rey

9. Jugabilidad

9.1 Descripción de los niveles:

Nivel 1	Presentación del juego y mecánicas de este. Habrá que enfrentarse a oleadas de enemigos y al primer jefe. Primera recompensa: Lanza.
Nivel 2	Cambio de jugabilidad, modo "carrera". Avanzamos por el escenario hasta llegar a un caballo, comenzando la parte en que se está montado a caballo y habrá que saltar y esquivar. Tras esta carrera aparecerá el segundo jefe. Segunda recompensa: Escudo.
Nivel 3	Vuelve las oleadas, esta vez con nuevos enemigos. Después seguirá el tercer jefe. Tercera recompensa: Doble salto.
Nivel 4	Último modo carrera, esta vez en sentido ascendente. Se terminará en la cima de una torre con el cuarto jefe. Cuarta recompensa: Cañón.
Nivel 5	Última fase de oleadas. Esta vez serán 3, después de cada una aparecerá un jefe. Quinta recompensa: cada jefe derrotado otorgará un fragmento de llave, al juntarlas se obtendrá la llave que abre las puertas del castillo.
Nivel 6	Clímax y resolución. Pelea contra el último jefe del juego. Tras vencerlo, dará paso al epílogo.

9.2 Metas y recompensas:

El objetivo principal de Pawn es avanzar por los diferentes niveles del juego hasta llegar al Rey, así que por el camino tendrá que ir eliminando a todo aquel que lo obstaculice. Estos enemigos no soltarán ningún objeto como recompensa, a excepción de los jefes de cada nivel. Por tanto, la vida que se pierda por el camino se recuperará solamente con los paquetes de vida que estarán repartidos de manera estratégica y solamente se podrá usar una vez, ya que cuando recuperes vida con uno de ellos este desaparecerá.

En cuanto a recompensas del juego, los jefes serán los encargados de proporcionarlos.

- **Peón**: Al derrotarlo te recompensará con una lanza.
- **Caballo**: Se obtendrá un escudo con el que se podrá protegerte de los ataques de los enemigos. Además, tu vida aumentará.
- **Alfil**: Conseguirás una mitra que dará la habilidad de doble salto, pudiendo acceder a zonas más altas que antes eran inaccesibles.
- **Torre**: Cañón que se podrá usar como arma secundaria para atacar a distancia.

9.3 Mecánicas y HUD:

Para la comodidad del jugador, existirán botones del teclado o mando para pelear o acceder de manera más rápida y sencilla a menús.

- **WASD / Joystick izquierdo**: Nos moveremos con las teclas “W” (delante), “A” (izquierda), “S” (detrás) y “D” (derecha); y con el eje del *joystick*.

- **Esc / botón pausa**: Se abrirá el menú de pausa, con un estilo parecido al de los recursos de debajo.

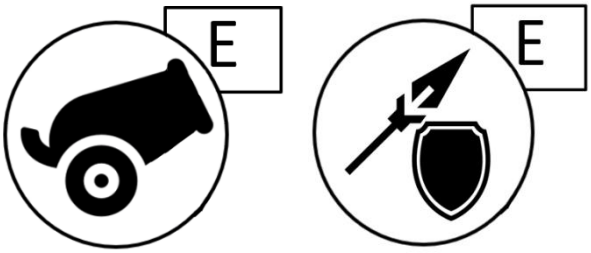
Ilustración Anexo 15. Ejemplo de diseño de menú

- **Espacio / X**: Si lo presionamos una vez pegaremos un salto simple, pero si ya hemos obtenido la habilidad de doble salto del alfil jefe podremos saltar más alto presionando este botón dos veces seguidas.

- **Clic izquierdo del ratón / R1**: Un solo clic realizará un barrido con la espada o lanza. En caso de tener la lanza, se añadirá otro ataque que podremos realizar dando doble clic, de esta manera podremos hacer una estocada contra algún enemigo. En caso de equiparse el cañón, lanzaremos la munición (limitada) presionando una vez el clic izquierdo. Esta munición se te dará al principio de cada nivel.

- **Clic derecho del ratón / L1**: Una vez equipado el escudo, podremos usarlo presionando una vez esta tecla, pero solo se usará si lo mantenemos pulsado, una vez lo soltemos, el escudo volverá a posicionarse tras la espalda. En caso de tener equipado el cañón, este botón nos servirá para apuntar.

- **E / R2**: Cambio de arma. Podremos cambiar de lanza y escudo a cañón, o viceversa. Los ejemplos que podemos ver a continuación estarían situados en la parte inferior derecha de la pantalla



- **Desplazar el ratón / Joystick derecho**: La cámara seguirá en todo momento al personaje y se moverá a la par con el ratón o *joystick*.

La **vida** está situada abajo a la izquierda y representada por la silueta de Pawn, que irá desapareciendo conforme recibimos daño, parecido a la barra de vida usada en *Minecraft*.

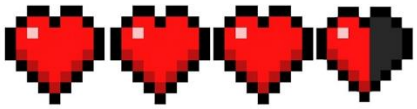


Ilustración Anexo 16. Ejemplo de vida

En cuanto a la de los **jefes finales**, se encontrará en la parte superior central de la pantalla, representado con el color rojo y encima, el nombre del enemigo, como en este ejemplo de *Borderlands 3*.

Ilustración Anexo 17. Captura del videojuego Bordelands 3

Al **apuntar** con el cañón a un enemigo la mira se volverá de color rojo cuando el enemigo pueda ser herido con el disparo, si sale en blanco significa que no está dentro del alcance o que no hay ningún enemigo en la mira. Todo esto se hará en tercera persona. La mira seguirá al ratón. Se pueden tomar como ejemplo las imágenes siguientes del juego *Worms 3D*, pero cambiando la primera persona por una cámara en tercera.

Ilustración Anexo 18. Captura del juego Worms 3D

10. Diagrama de estados:

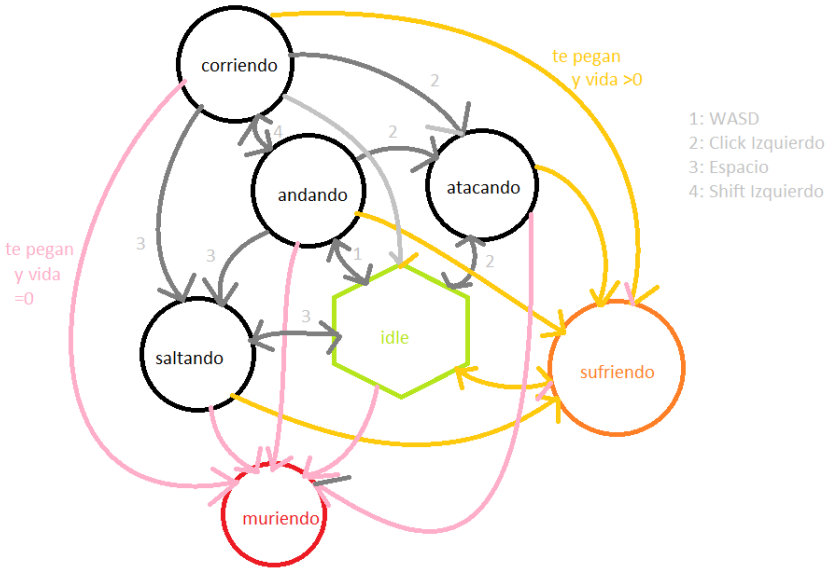


Ilustración Anexo 19. Máquina de estados

11. Reglas del juego:

- El jugador tendrá que jugar dentro del escenario, no podrá rebasar los límites de este, aunque lo intente.
- Si muere, empezará desde el último punto de guardado.
- No se puede mover ni usar la lanza o cañón al mismo tiempo que el escudo.
- Una vez se utilice un paquete de vida, este desaparecerá y no se podrá volver a usar.
- No se podrá volver a los niveles anteriores.

12. Menús y navegabilidad

12. 1 Diagrama de navegación

El juego contará con 2 menús: el inicial y el de pausa. La navegabilidad entre cada uno de ellos la podemos ver de manera esquematizada en la siguiente imagen.

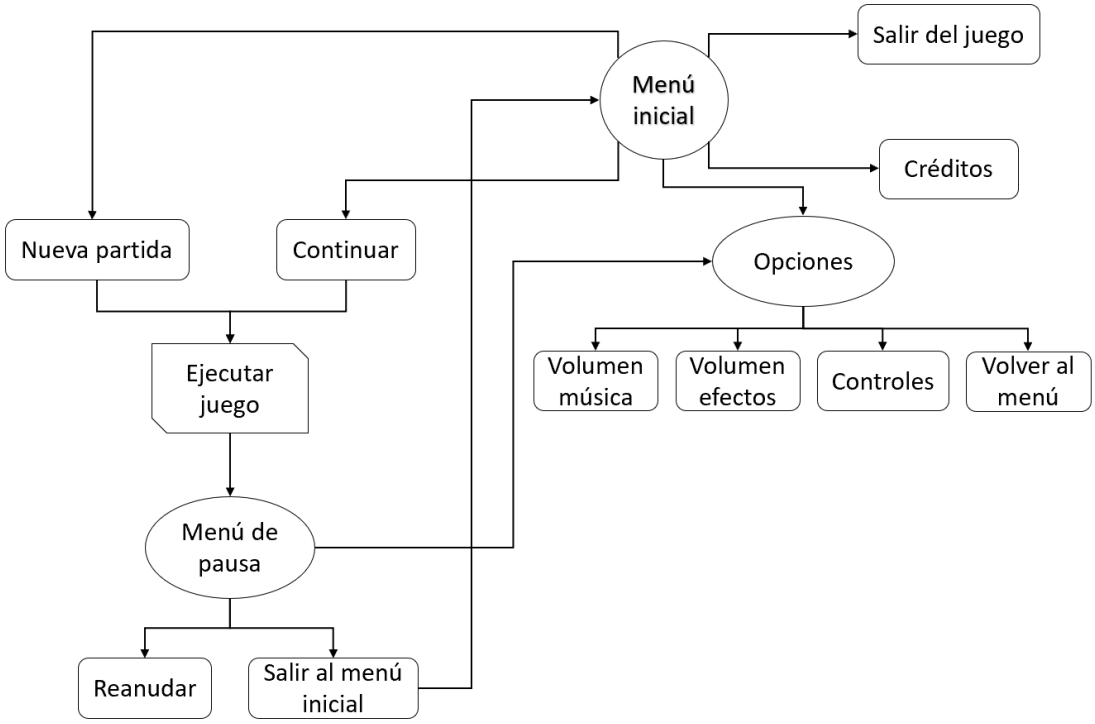


Ilustración Anexo 20. Navegación entre menús

12.2 Menú inicial:

Es el que veremos nada más abramos el juego. Una vez dentro, se observará la animación de un libro abriéndose, para a continuación mostrar las diferentes opciones del menú:

- **Nueva partida**: Te mostrará un mensaje preguntando si deseas comenzar partida o no. En caso de que la respuesta sea afirmativa, te llevará directamente al inicio del juego, específicamente en la narrativa de la historia.
- **Cargar partida**: Continuaremos por donde nos quedamos en el juego (el último punto de guardado).
- **Multijugador**: Se abrirá el *lobby* del este modo de juego
- **Opciones**: Se abrirá un menú para poder configurar varias opciones.
- **Créditos**: Se mostrará por pantalla los créditos del juego.
- **Salir del juego**: Se saldrá del menú.

12.3 Menú de pausa:

Se accederá a él presionando la tecla esc. Una vez abierto, el fondo del juego se verá difuminado y un poco más oscuro, de esta manera se mostrará varias opciones:

- **Reanudar**: Se cerrará el menú y podremos seguir jugando.
- **Cargar partida**: Cargará una partida guardada.
- **Opciones**: Compartirá el mismo menú de opciones del menú inicial.
- **Menú principal**: Nos llevará al menú inicial.

12. 4 Menú opciones:

Esta ventana será la que nos permitirá configurar algunas cosas del juego si es que no estamos a gusto con la configuración que viene por defecto. Por ello, contará con las siguientes opciones:

- **Gráficos**
- **Sonido**
    - Volumen global
    - Volumen de la música
    - Volumen de los efectos de sonido (SFX)
    - Volumen del ambiente del juego
- **Controles**
- **Volver:** Se vuelve al menú anterior

13. Escenarios

13.1 Estética de la ambientación

El juego se desarrolla en una guerra medieval en un mundo de juegos de mesa. Existen dos entornos dentro de este videojuego, el primero es el mundo jugable y nuestro contexto más cercano, y en el segundo veremos el mundo externo a nuestra guerra, donde podremos observar representaciones de otros juegos de mesa.

Como hemos comentado anteriormente, la estética visual será una guerra medieval donde los colores predominantes son el marrón y el verde, pero también podremos observar otros colores algo más vibrantes cuyo papel es darles vida a los escenarios, puesto que el color predominante de los personajes es el negro y queremos generar cierto contraste entre ellos.

Tanto escenarios como personajes serán low poly, ya que nuestro objetivo es lograr una estética cartoon y minimalista. Utilizaremos colores similares a los de la siguiente paleta.

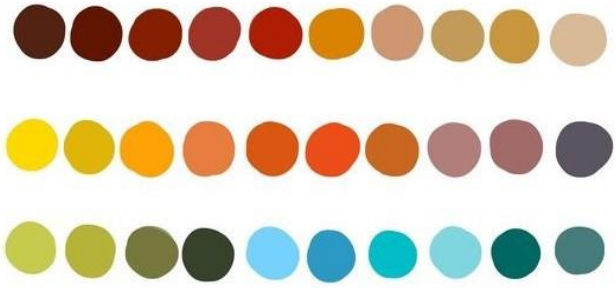


Ilustración Anexo 21. Paleta de colores

13.2 Escenarios existentes por niveles y escenas

Existen 4 escenarios donde se desarrolla el videojuego. A su vez, en estos pueden desarrollarse varios niveles y habrá diferentes escenas dentro de cada escenario. A continuación, detallaremos qué engloba a cada uno de ellos:

- **Escenario 1**, **el campo de batalla**: engloba los niveles 1, 2 y 3 del juego. En este escenario habrá 7 escenas diferentes, una escena inicial de cada nivel donde tendremos que avanzar y derrotar a los enemigos, y otra escena para cada uno de los jefes finales. En el nivel 2 se divide en tres escenas.

- **Escenario 2, la torre**: corresponde al nivel 4 del juego. En este escenario habrá 2 escenas, una para el nivel y otra para el *boss*.

- **Escenario 3, el patio de armas**: corresponde al nivel 5 del juego. En este escenario habrá 2 escenas, una para el nivel y otra para el jefe.

- **Escenario 4, interior del castillo**: corresponde al nivel 6 del juego. En este último escenario habrá 3 escenas, la primera será una presentación de la Dama y el Rey, la segunda será la batalla y derrota de la Dama, y la última será el final del juego.

13.3 Breve descripción de los escenarios

- **Escenario 1**, **el campo de batalla**: este escenario tiene diferentes escenas que representan cada nivel, la primera será el campo de batalla, en una línea casi recta donde habrá enemigos, además de diferentes obstáculos como pueden ser: árboles, cajas, trincheras,

barricadas, puestos de arma, vallas, etc. que también serán los que delimitarán el escenario.

La segunda escena será un camino en línea recta, donde iremos a caballo y tendremos que saltar obstáculos y evitar o golpear enemigos. Estos obstáculos serán los mismos que en la escena anterior y tendrán la misma función. El jefe final estará en un área circular delimitada.

La tercera escena estará a las afueras del muro del castillo y será un área rectangular, al igual que las escenas anteriores contaremos con obstáculos y enemigos, que limitan o cortan el paso de nuestro protagonista.

- **Escenario 2, la torre:** este escenario será circular y tendremos que ascender por las escaleras de la torre que suben en espiral hasta el matacán donde nos encontraremos al jefe de este nivel. El escenario estará compuesto por enemigos y obstáculos como cajas de armas, huecos entre escalones que tendremos que saltar, etc. En la escena del matacán encontraremos escombros, cajas, muros, etc.

- **Escenario 3, el patio de armas**: será un área circular que abarca desde las puertas, murallas y torres defensivas hasta la entrada al castillo. Los obstáculos serán escasos, ya que es un nivel en el que estaremos luchando constantemente.

- **Escenario 4, interior del castillo**: tendrá forma rectangular y representa la sala del trono, en él podremos apreciar objetos que servirán para delimitar el escenario, así como para protegernos y ambientar la escena, puede ser: bancos, muros y columnas de piedra, etc.

14. Elementos jugables

Pawn es un videojuego cuyos elementos jugables son limitados, estos podrían resumirse en:

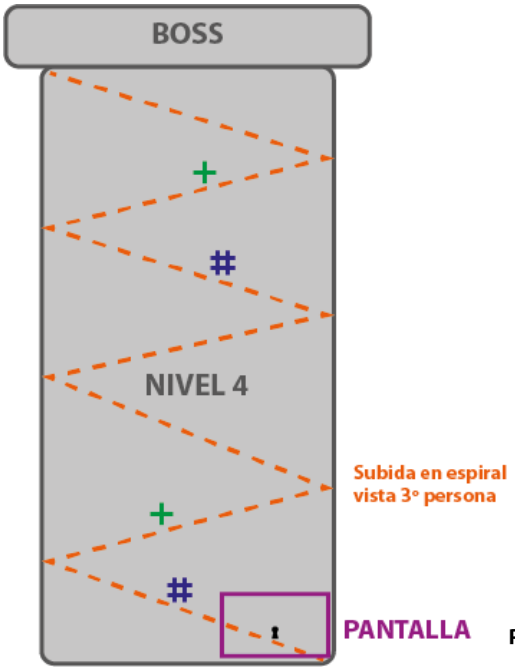
- **Los enemigos**: cuya interacción se resume en ser golpeados o golpearnos hasta que son derrotados.

- **Paquetes de vida**: los cuales estarán repartidos estratégicamente por los escenarios y al pasar por encima nos recuperarán automáticamente un porcentaje de vida. Una vez utilizados estos desaparecen, por lo que son de un solo uso, además, si nuestra vida está al 100% no podremos curarnos más.

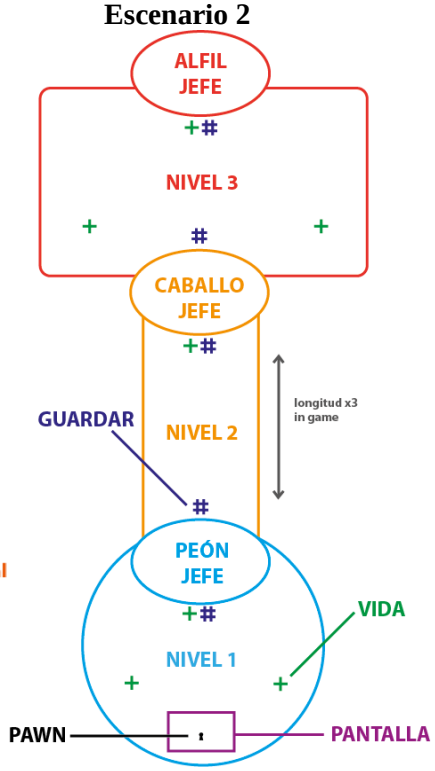
- **Obstáculos**: estos no generan una interacción automática como los demás, pero podemos saltar encima y acceder a una zona más alejada, o para cubrirnos del fuego enemigo, para delimitar el terreno o simplemente para decorar el escenario. Los *prefabs* de los que estamos hablando son los siguientes: Árboles (existen 3 tipos), Carros (existen 2 tipos), hogueras, tiendas de campaña, camas y cojines, barriles (dos tipos), rocas (al menos 5 tipos diferentes), carteles orientativos, troncos caídos y tocones, edificios (5 diferentes), un pozo, vallas, hierba, caballos, huerta y un puente que cruza el río.

15. Bocetos

Escenario 1

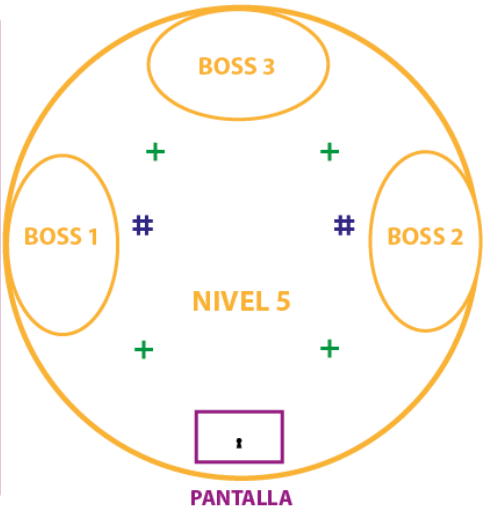


Escenario 2



Escenario 3

Escenario 4



Anexo II: ODS

ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

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

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

El trabajo de este proyecto ha consistido en el desarrollo de un videojuego 3D, sin ningún tipo de relación con el medio ambiente o la salud, pero sí que hay objetivos de desarrollo sostenible que pueden tener algo que ver con este trabajo. Se explicarán cada uno de ellos, comenzando por los más bajos, hasta terminar con los que más relación tienen.

En primer lugar, se ha señalado la igualdad de género a nivel bajo debido a que, a pesar de que la mayoría de los enemigos se pueden catalogar de género masculino, la Dama, que podría representar al género femenino, se muestra como un enemigo fuerte y poderoso, es más, comparada con el resto, incluso con el Rey, es la que más fuerza tiene y la que resulta más difícil de derrotar. Esta relación no es muy notable, por eso se ha catalogado con un nivel bajo, pero sí es importante recalcar que, a pesar de no haber peones, caballos, alfiles o torres del género masculino, el femenino no es inferior a ellos, ni mucho menos.

El siguiente punto para comentar es el ODS 9. Hoy en día existen multitud de videojuegos de todo tipo, pero cada uno de ellos acaban siendo una innovación a su manera. Pawn no es una excepción, la temática de ajedrez que contiene resulta innovador y se podría decir que no hay ningún juego en el mercado que haya metido fichas de ajedrez en un juego del género de plataformas o arcade, mucho menos en uno con una ambientación bélica o medieval, con infraestructuras típicas de la época. El juego se basa en avanzar por el camino que hay frente a los ojos del jugador, siendo que todo el juego se desarrolla en una vista 3D para el usuario que juegue hasta llegar al castillo que se ve al fondo, encontrándose por el camino enemigos con los que habrá que pelear. Estos enemigos estarán en algunas ocasiones situados en campamentos inspirados en los que había en la época en la que está ambientado.

Por último, los dos puntos más importantes que se han tratado son los de “Reducción de las desigualdades” y “Paz, justicia e instituciones sólidas”, más concretamente en la justicia. Para el primer caso, la historia de Pawn está bastante relacionada con este aspecto, teniendo en cuenta que el objetivo del protagonista es acabar con las desigualdades que existen respecto al rango de “peón”. En el juego original del ajedrez, las primeras fichas que se suelen sacrificar son estas, y lo mismo ocurre en el juego. El Rey es un personaje débil y cobarde, y no se lo piensa dos veces antes de mandar a los peones a la guerra. El objetivo del protagonista es luchar para acabar con esta desigualdad, desea que todas las categorías del ejército (caballo, alfil y torre) sean tratadas de la misma manera, sin sacrificios innecesarios, busca que los derechos de cada uno sean iguales, sin importar su rango.

Aquí entraría la relación que hay respecto al punto 16. Al mismo tiempo que busca desigualdad, también desea hacer justicia por todos sus amigos caídos a causa de las órdenes de ese Rey y por todos los que están por caer.