UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Diseño y producción de un videojuego en realidad virtual. Mistborn, Ghostbloods contracts

Trabajo Fin de Grado

Grado en Tecnologías Interactivas

AUTOR/A: Grau Giannakakis, Jorge

Tutor/a: Pérez López, David Clemente

CURSO ACADÉMICO: 2022/2023

Resumen:

En este trabajo final del grado se presenta el diseño y producción de un videojuego para plataformas de VR de género *Shooter* y *Rogue-Lite*, basado en la saga literaria Mistborn de Brandon Sanderson. Aunque existe alguna adaptación de la saga, ésta es la primera adaptación a un entorno VR. Se trata de una adaptación enfocada a la acción y la estrategia, inspirada en las narrativas, el universo y las mecánicas de los libros.

Asimismo, se analizan los videojuegos más significativos relacionados con la temática planteada, para identificar las prácticas que más gustan al público, así como las carencias de éstos para realizar un diseño único que satisfaga las demandas del público actual.

Además, se presenta un estudio de las herramientas para el desarrollo de VR, identificando las opciones más adecuadas. Se analizan las características, funcionalidades y ventajas de cada herramienta, considerando aspectos como la compatibilidad, la facilidad de uso, el rendimiento y las capacidades de interacción.

También se presenta un estudio del hardware actual para VR, donde se analizan y comparan las características técnicas, capacidades de *tracking*, resolución, comodidad y compatibilidad de diferentes cascos. Este estudio permite seleccionar el casco que se ajusta mejor a las necesidades del proyecto.

Abstract:

This final degree project presents the design and production of a Shooter and Rogue-Lite video game for VR platforms, based on the Mistborn literary saga by Brandon Sanderson. Although there is some adaptation of the saga, this is the first adaptation to a VR environment. It is an adaptation focused on action and strategy, inspired by the narratives, the universe and the mechanics of the books.

Likewise, the most significant video games related to the proposed theme are analyzed, to identify the practices that the public likes the most, as well as the shortcomings of these to carry out a unique design that satisfies the demands of the current public.

In addition, a study of the tools for VR development is presented, identifying the most appropriate options. The characteristics, functionalities and advantages of each tool are analyzed, considering aspects such as compatibility, ease of use, performance and interaction capabilities.

A study of the current hardware for VR is also presented, where the technical characteristics, tracking capabilities, resolution, comfort and compatibility of different helmets are analyzed and compared. This study allows selecting the VR headset that best suits the needs of the project.

Palabras clave:

Videojuego, Unity, Realidad Virtual, Disparos, XR Interaction Toolkit

Palabras clave (inglés):

Videogame, Unity, Virtual Reality, Shooter, XR Interaction Toolkit

Índice

1 Introducción

1.1 Preámbulo

La Realidad Virtual (RV o VR en inglés) ha experimentado un crecimiento significativo en los últimos años, brindando a los usuarios una experiencia inmersiva y envolvente en entornos virtuales. El desarrollo de videojuegos para esta plataforma ofrece nuevas oportunidades para explorar y ampliar los límites de la interacción y la diversión en el ámbito de los juegos (Stecula, 2022). Este trabajo se centra en la creación de un videojuego de VR que aprovecha al máximo las capacidades y características de los cascos de VR. Este tipo de dispositivos, proporcionan una inmersión única al permitir a los jugadores sumergirse por completo en un mundo virtual tridimensional. El objetivo es crear una experiencia de juego emocionante y cautivadora que aproveche las posibilidades de la VR para ofrecer una sensación de presencia.

Durante el desarrollo del videojuego, se prestará especial atención al diseño de la experiencia, los controles intuitivos y la interacción fluida con los elementos del juego. Se espera que este TFG contribuya a la exploración de nuevas formas de entretenimiento y narrativa a través de la VR. También, se busca proporcionar a los jugadores una experiencia de juego inmersiva y emocionante, aprovechando al máximo las capacidades de los cascos de VR y explorando los límites de la interacción virtual.

En el presente documento se pretende explicar todas las partes del desarrollo previas al lanzamiento y realizar un estudio de mercado, en el que se explorará el estado de la industria VR (opciones en cuanto a hardware y sistemas de desarrollo) y también el estado de la industria del videojuego en cuanto a los géneros a los que pertenece el videojuego presentado en esta memoria.

Asimismo, se pretende explorar cómo se podría adaptar la saga literaria Mistborn (Sanderson, 2022), para crear un videojuego de VR. El título que se ha elegido es Mistborn: Ghostbloods Contracts, y está siendo desarrollado actualmente en solitario, aunque es importante destacar que no todos los elementos en el juego son de creación propia. No somos expertos en modelado, texturizado o sistemas de audio, pero hemos intentado desarrollar por nosotros mismos algunos de los *props* o elementos de la parte visual y sonora. La gran mayoría de los elementos visuales han sido adquiridos en la Asset Store de Unity (Unity Technologies, 2010) o itch.io (Corcoran, 2013), verificando siempre que las licencias y los términos de uso de cada elemento permitan su uso en un trabajo como éste.

El juego es una experiencia de un solo jugador, en el que el objetivo es desbaratar las operaciones criminales que ocurren en el planeta, pasando por ciudades, aldeas e incluso mazmorras de la época de leyenda.

El jugador podrá elegir qué armas o poderes llevar en cada misión, teniendo en cuenta las características de cada nivel para adaptarse lo mejor posible. En los niveles deberá derrotar a todos los enemigos que se encuentre, para ello podrá utilizar el entorno y el equipamiento que haya seleccionado. Una vez completada una misión, el jugador vuelve a la sala de preparación y es recompensado con desbloqueos de nuevo equipamiento. Si por desgracia el jugador muriese en alguna de sus incursiones, se le daría la oportunidad de mantener parte del equipamiento desbloqueado, gastando un recurso conseguido al completar misiones.

1.2 Objetivos

- Diseñar y producir un videojuego de géneros *Rogue-Lite* y *Shooter* en Realidad Virtual llamado Mistborn: Ghostbloods Contracts.

Los objetivos específicos son:

- Realizar un análisis del sector de la VR, plataformas de desarrollo/dispositivos.
- Realizar un análisis de los géneros y sus juegos más famosos en el mercado.
- Diseñar los sistemas principales de juego, así como implementar un ciclo jugable.
- Transportar el concepto de la saga de libros al entorno VR.

1.3 Estructura del documento

En los siguientes capítulos se desarrollarán todos los elementos nombrados anteriormente en la presentación de la memoria. El documento se ha dividido en 6 capítulos, Introducción, Materiales y métodos, Conclusiones y trabajos futuros, Referencias bibliográficas, Glosario de términos y Anexos, con la siguiente estructura.

- En el **capítulo 1, Introducción**, se describe la metodología empleada y las herramientas de trabajo, además se exponen los objetivos del trabajo y las fases del desarrollo de un videojuego, primero dando una descripción general y luego una específica sobre el proyecto realizado.
- En el **capítulo 2, Materiales y métodos**, se presenta el contexto del trabajo, se muestra el documento de diseño de juego (GDD en inglés) empleado, se hace un análisis del sector, del hardware y las plataformas de desarrollo y más importante, se muestra la implementación final junto a las pruebas realizadas.
- En el **capítulo 3, Conclusiones y trabajos futuros**, se resume el trabajo y se valora el proyecto desde un punto de vista personal, además de aportar ideas para la mejora del videojuego a futuro.
- En el **capítulo 4, Referencias bibliográficas**, se lista la bibliografía consultada para la realización de este trabajo.
- En el **capítulo 5, Glosario de términos**, se explican algunos de los tecnicismos o palabras extranjeras usadas en la redacción de esta memoria.
- En el **capítulo 6, Anexos**, se encuentran varios videos referentes al desarrollo del juego, el ejecutable del juego y un enlace a un *gampeplay* sin cortes.

1.4 Metodología

A continuación, explicaremos cómo nos hemos organizado para completar el trabajo y que pasos han sido los más importantes de cara al proyecto.

Antes de lanzarnos a desarrollar prototipos complejos sin tener un rumbo fijo, nos dedicamos a realizar un documento de diseño de juego, este paso fue muy importante porque, en general, marca el inicio del desarrollo como tal, además, mientras avanzábamos en la escritura de este documento, comenzamos el análisis del estado de la industria VR, valorando qué opciones existían para resolver los distintos sistemas de juego.

Al trabajar en solitario el proyecto, la organización fue un pilar fundamental. Conocer todas las tareas a realizar, su estado o errores, nos permitía tener una visión general del proyecto en todo momento.

Para organizar el avance del trabajo se ha utilizado Trello (Johnson, 2017), una herramienta para la gestión de tareas en equipo.

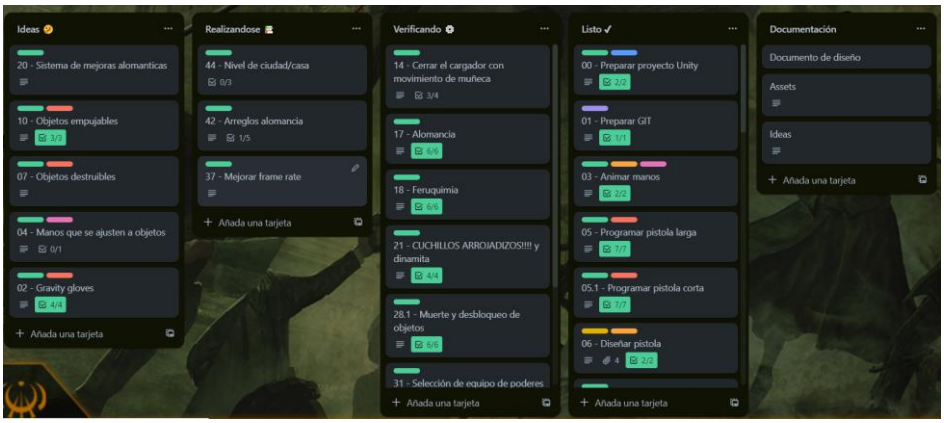

Figura 1 Tablero de Trello

Cómo se puede ver en la Figura 1, el tablero de Trello se divide en las secciones Ideas, Realizándose, Verificando, Listo y Documentación. Cada sección cumple una función a la hora de manejar tareas, que en este caso son:

Ideas

Aquí se colocan las tareas a realizar e ideas que podían ser interesantes para implementar en el ejecutable final.

Realizándose

Aquí se colocan las tareas que se están realizando.

Verificando

En esta sección se dejan todas las tareas que están acabadas, y se siguen probando hasta saber a ciencia cierta que no provocan ningún error.

Listo

Aquí se colocan las tareas que no tienen errores de ningún tipo.

Documentación

Este apartado lo usamos para guardar referencias de recursos externos que se utilizaban y documentación necesaria para la realización de la memoria.

Además de organizar las tareas, también es importante mantener copias de seguridad del código por si algo fallase o se perdiese, la herramienta de control de versiones que se utiliza es GitKraken (Axosoft, 2000).

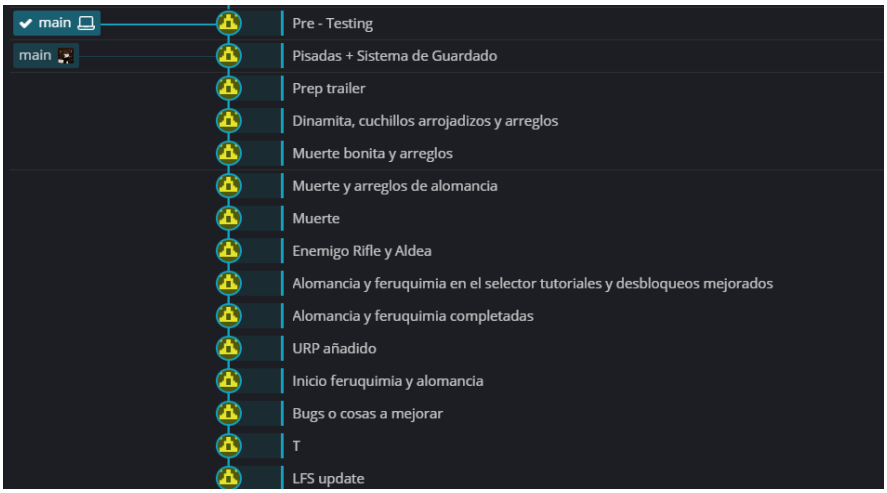

Figura 2 Interfaz de GitKraken

Es importante comentar que se han realizado practicas correctas hacia el trabajo en equipo, por ejemplo, el mantenimiento del Git, realizando todos los días de trabajo un *commit*, para mantener una copia de seguridad. En la Figura 2 se puede ver un ejemplo de los *commits*.

Gracias a mantener el Git activo y las tareas de Trello, se podían generar prototipos o probar ideas de forma rápida y segura, estos añadidos se mantenían en el proyecto si nos gustaban o se descartaban si no nos parecían suficientemente buenos.

1.5 Fases del desarrollo de un videojuego

Las fases de desarrollo de un videojuego son: la planificación, la preproducción, la producción, la fase de pruebas (Testing), el pre-lanzamiento, el lanzamiento y el post-lanzamiento. En este proyecto se han realizado todas menos las referentes al lanzamiento. En la figura 3 se puede ver un cronograma en el que se muestra cuanto tiempo se empleó en cada apartado del desarrollo.

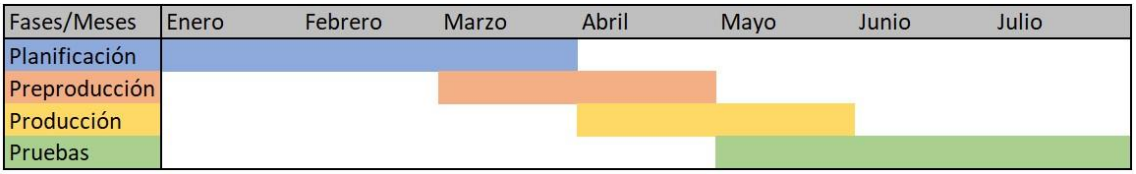

Figura 3 Cronograma

1.5.1 Planificación

En esta etapa del desarrollo se resuelven cuestiones respecto a presupuestos, fechas de entregas, tipo de cámara, género, etc. Esta fase acaba con una prueba de concepto.

En nuestro caso la planificación nos llevó a valorar cómo se quería abordar el videojuego, el mundo de Mistborn es extenso y realmente se pueden hacer juegos de géneros muy distintos, en un inicio queríamos replicar los sucesos de alguna de las novelas y hacer un desarrollo tipo Half Life Alyx (Valve Corporation, 2020), pero al darnos cuenta de la inmensa cantidad de trabajo que había que hacer para simplemente realizar de forma correcta una zona, decidimos cambiar de idea.

El juego pasó de una historia con carga narrativa a un *Shooter, Rogue-Lite*, donde la historia no es tan importante y hay que priorizar más las mecánicas e interacciones entretenidas. Los comentarios de desarrolladores en Half Life Alyx junto al libro Video Game Design (Salmond, 2016) nos ayudaron a dar los primeros pasos en el desarrollo VR.

Durante esta fase probamos varios juegos de VR, ver la Figura 4 (que se adquirieron en una oferta de Humble Bundle), e intentamos replicar sus mecánicas básicas. Mientras jugábamos nos fijamos en cómo se resolvían ciertos problemas y por suerte nos habituamos al juego en VR.

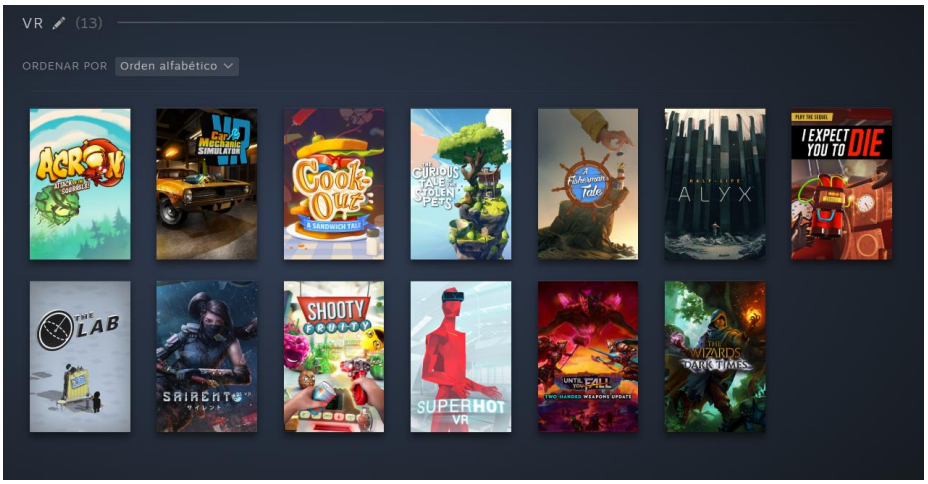

Figura 4 Juegos probados en VR

La fase de planificación acabó con una prueba de concepto, esta prueba de concepto puede ser visualizada en el Anexo Video I Prueba de concepto.

1.5.2 Preproducción

En esta etapa se refina la definición del videojuego y se empiezan a realizar las primeras interfaces y prototipos jugables.

Por lo tanto, en esta fase, se usó como herramienta de trabajo el documento de diseño de juego, esto nos ayudó a definir bien los sistemas y elementos que queríamos que tuviese la versión final, asimismo, creamos la gran mayoría de tareas en el Trello basadas en lo descrito en el documento de diseño de juego. En esta etapa se definieron aspectos importantes del diseño como UI, poderes, tipos de arma, enemigos, misiones, etc.

Para nuestro proyecto, la preproducción acabó cuando conseguimos completar la sala de preparación, el menú radial, algunas armas y la creación del nivel de la mazmorra, el cual se generaba de forma semi-procedural y transmitía muy bien la idea del juego.

La fase de preproducción acabó con un prototipo, este prototipo puede ser visualizado en el Anexo Video II Prototipo.

1.5.3 Producción

La etapa de producción es la más extensa y compleja, en esta etapa se terminan todos los pequeños detalles de la etapa anterior y se ensambla todo para formar un conjunto final. Es muy común que secciones enteras se descarten o rediseñen para satisfacer los nuevos objetivos del proyecto.

Para la fase de producción nos centramos en completar sistemas de juego y añadir todos los objetos de equipamiento, también objetos de recuperación de salud, de investidura (magia), objetos arrojadizos, objetos explosivos, selección de niveles y un nivel nuevo, la aldea.

La producción acabó cuando lanzamos el tráiler del juego, aunque había elementos por pulir y añadir, la mayor parte estaba lista y esto permitía que los *testers* lo probasen. Esto nos dio pistas para saber que era lo más importante que debíamos retocar antes de la fecha de entrega final.

El tráiler puede ser visualizado en el Anexo Video III Tráiler.

1.5.4 Fase de pruebas

Normalmente, la fase de pruebas con *testers* ocurre durante el desarrollo del propio juego, dado que solucionar problemas en ciertos sistemas es más sencillo al principio del desarrollo. En nuestro caso, como no teníamos disponibles *testers* durante el desarrollo, nos preocupamos de intentar perfeccionar el juego al máximo cada vez que se añadía una funcionalidad.

Una vez acabado todo el desarrollo y para ver si el videojuego está listo para ser lanzado, debemos someterlo a las pruebas de los *testers*, con el fin de ver qué bugs o problemas no se tuvieron en cuenta durante el desarrollo.

Una vez pudimos traer a *testers* a probar el juego, pedimos a amigos y personas interesadas que lo jugasen y diesen una opinión de cada aspecto importante. Viendo su reacción y opiniones modificamos el juego para arreglar los fallos o añadir las sugerencias que tenían.

Como se comentará más adelante en la sección 2.7, las pruebas de los *testers* nos ayudan a garantizar que ciertos elementos que nosotros como desarrolladores podemos pasar por alto o permitir, se corrijan, evitando que estos puedan influir negativamente en la experiencia de juego.

2 Materiales y métodos

2.1 Introducción e inspiración

A continuación, realizaremos un análisis del estado del género *Shooter* y *Rogue-Lite* fuera y dentro de la VR, además de hablar de las distintas plataformas de desarrollo para videojuegos VR y del hardware que se está utilizando en este proyecto.

Antes de eso, es conveniente también hablar de la idea del propio proyecto.

Brandon Sanderson fue el escritor que nos devolvió las ganas de leer e interesarnos por la fantasía, siempre hemos leído muchos libros, sobre todo de este género, pero hubo un punto en nuestra vida en el que dejamos de leer, no había libro que nos interesase o no conocíamos a los autores.

El primer libro de Sanderson que leímos fue El imperio final (Sanderson, 2008), y nos gustó tanto que en poco menos de un año habíamos leído todo lo que había sacado al mercado, lo que en resumen son unos 18 libros. Este escritor crea unos mundos que se sienten realistas, la "magia", llamada Investidura en su obra, no desentona con el mundo o sociedad, como puede llegar a ocurrir en otros libros, y además el desarrollo de personajes y *worldbuilding* que presenta en cada novela es increíble. Leer estos libros nos hacía sentir inmersos en los mundos que planteaba el autor y si algo nos ha gustado siempre de los videojuegos es como llegan a conectar con el usuario.

Desde pequeños nos han gustado los videojuegos y aunque habíamos desarrollado anteriormente prototipos, no ha sido hasta este año que probamos a desarrollar en VR. Al empezar a trabajar en VR-Comrades, nuestro proyecto del grado, tuvimos claro que queríamos hacer un juego para nuestro TFG. Los sistemas en VR, aunque a veces pueden provocar mareos, ayudan mucho con la inmersión, es increíble lo rápido que olvidas que estas jugando a algo que no existe en el mundo real.

Así que cuando pensamos en qué juego hacer para nuestro TFG, se nos ocurrió que un videojuego basado en Mistborn, podía ser algo factible y divertido de hacer. Es más, hace no mucho, apareció un intento de juego, no para sistemas VR, que gustó mucho a los fans llamado Mistborn: Ashes Project (Mistborn fan community, 2021). Y como recientemente había salido el último libro de la tetralogía de la segunda era de Mistborn, nos decidimos a empezar este proyecto.

2.2 VR y videojuegos. Análisis del género

La Realidad Virtual es una tecnología que crea una simulación computerizada de un entorno tridimensional interactivo, en el que los usuarios pueden experimentar una sensación de presencia y participación activa. Utilizando dispositivos de visualización y seguimiento de movimiento, la VR inmersiva busca proporcionar una experiencia multisensorial, que incluye la vista, el oído y, en algunos casos, el tacto, con el objetivo de transportar a los usuarios a entornos virtuales convincentes y realistas (Chen, 2021). Por lo tanto, es una tecnología muy adecuada para el desarrollo de este trabajo.

Hoy en día, los consumidores de juegos tienen un catálogo muy amplio sobre el que elegir. Nuevos desarrolladores aparecen cada día, proponiendo mezclas de géneros o ideas que tienen el potencial de desaparecer entre la centena de juegos nuevos que salen cada mes, o establecerse como un nuevo estándar.

El proyecto que se ha desarrollado en esta memoria se encuentra dentro de los géneros *Shooter* y *Rogue-lite*. A continuación, resumiremos el estado del arte de estos dos géneros en la industria del videojuego, viendo algunos de los títulos más destacados en VR.

2.2.1 Shooter

El género *Shooter* engloba todo videojuego en el que el jugador puede controlar un personaje que dispone de un arma que puede ser disparada a voluntad.

En cuanto a este género, se vive una situación curiosa, es un género que es muy intuitivo para el jugador y muy fácil de combinar con otros sistemas, la saga de disparos más conocida y popular estos últimos años ha sido Call of Duty (Game World Observer, 2022), y no han sido pocos los competidores como Titanfall 2 (EA, 2016), Overwatch 2 (Blizzard Entertainment, 2022) o Battlefield (Dice, 2021), que han llegado al top 1, y meses más tarde han desaparecido del mercado. Es decir, el género *Shooter* se presta a ser maleable y generar ideas distintas, pero lo más popular y lo que más gusta al público general son los juegos que menos modifican la esencia.

Aun así, cada vez salen más *Shooters* al mercado con ideas distintas, por ejemplo, Splatoon (Ninentdo, 2017) el *Shooter* de Nintendo donde el objetivo no es acabar con tus enemigos sino pintar más mapa que ellos, o la vuelta de clásicos como DOOM (Bethesda Softworks, 2016) o Wolfenstein (Bethesda Softworks, 2014) que reviven las campañas frenéticas para un solo jugador (PCGamesN, 2020). Por ello, lo que se está viviendo en la industria hoy en día, es la transición de todos los jugadores casuales a otros subgéneros del *Shooter*.

Si hablamos de subgéneros, hay que comentar los más famosos últimamente, que han provocado que las sagas mejor establecidas como Call of Duty, mencionada anteriormente, creen un modo similar, Call of Duty: Warzone (Activison, 2020), para volver a captar la atención de sus jugadores.

Los más populares últimamente son el *BattleRoyale* y los *Extraction Shooters*, que son modificaciones de la formula clásica. En los *BattleRoyale* los jugadores se enfrentan entre si hasta que únicamente queda uno en pie. En los *Extraction Shooter* los jugadores deben entrar a un mapa a conseguir armas y objetos útiles, si mueren durante la partida pierden todos los objetos que llevaban encima o habían recogido, por otro lado, si consiguen salir del mapa ganan ese equipamiento y lo pueden emplear en próximas partidas.

Desde el boom del *BattleRoyale* (Eurogamer, 2020), gracias a Fortnite (Epic Games, 2017) o los *Extraction Shooter* (más o menos realistas), con Escape From Tarkov (Battlestate Games, 2017), se ha vivido una oleada de títulos con estos modos de juego o características similares.

En el mercado VR, hasta la popularización de estos nuevos modos de juego, la mayoría del catálogo disponible eran simuladores o *Shooters* muy genéricos, alguno copiaba el estilo táctico de juegos tipo Counter Strike Global Offensive (Valve Corporation, 2012) o Valorant (Riot Games Inc, 2020), pero en esencia eran muy similares.

Algunos ejemplos de juegos VR de disparos son:

Contractors (Figura 5)

Contractors (Caveman Studio, 2018) es un Shooter VR clásico que pretende imitar el estilo de juego de Call Of Duty. El sistema de recarga simple nos sirvió como inspiración.

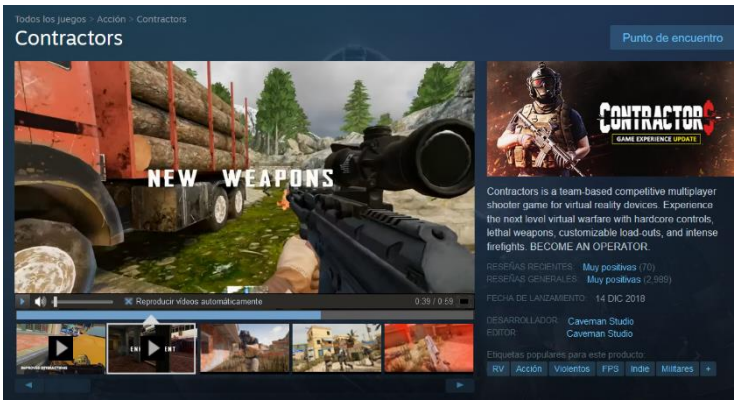

Figura 5 Ficha de Contractors en Steam

POPULATION ONE (Figura 6)

Population One (Big Box VR, 2020) es un Shooter Battle Royal, que ha copiado el sistema de Fortnite.

Figura 6 Ficha de Population One en Meta Quest

Ghosts of Tabor (Figura 7)

Ghost of Tabor (Combat Waffle Studios, 2023) es un Extraction Shooter, se conoce a este juego, principalmente por ser la versión VR de Escape From Tarkov.

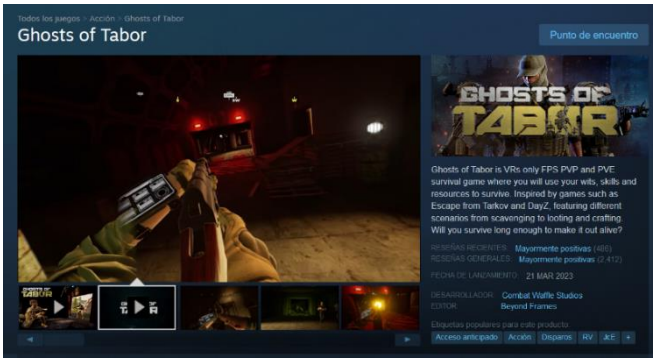

Figura 7 Ficha de Ghosts of Tabor en Steam

Pistol Whip (Figura 8)

Pistol Whip (Cloudhead Games, 2019), es un Shooter VR centrado en partidas de un solo jugador muy dinámicas y frenéticas. Uno de los Shooters más originales del mercado VR actual. La claridad con la que se muestran las balas y los enemigos nos sirvió de inspiración.

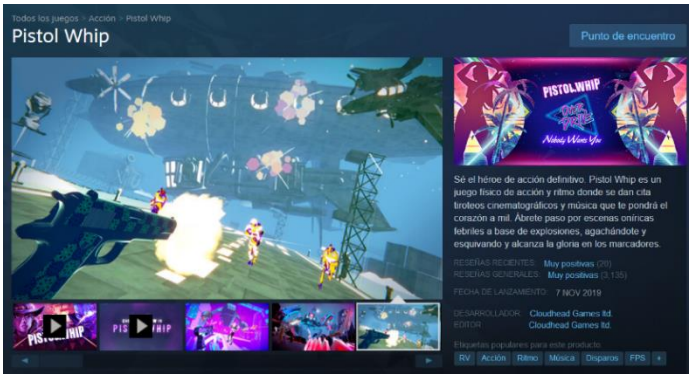

Figura 8 Ficha de Pistol Whip en Steam

2.2.2 Rogue-Lites

Para hablar de los *Rogue-Lite* conviene conocer un poco su historia. En 1980 se lanzó al mercado Rogue (Epyx, 1980), un juego en el que, el jugador debía atravesar mazmorras generadas proceduralmente. Esta idea innovadora hizo que el juego se popularizase entre algunos jugadores, hasta crear una pequeña pero fiel comunidad de fans. Con el tiempo se fue recreando la idea inicial y fue necesario un estándar, para definir el género de los juegos tipo Rogue, es decir *Rogue-Likes*. La "interpretación de Berlín" (Laid, 2008) dividió todos los elementos del juego original en factores de alto valor y bajo valor con los que definir si un juego realmente era un *Rogue-Like*. Los factores para tener en cuenta se ven en la Figura 9.

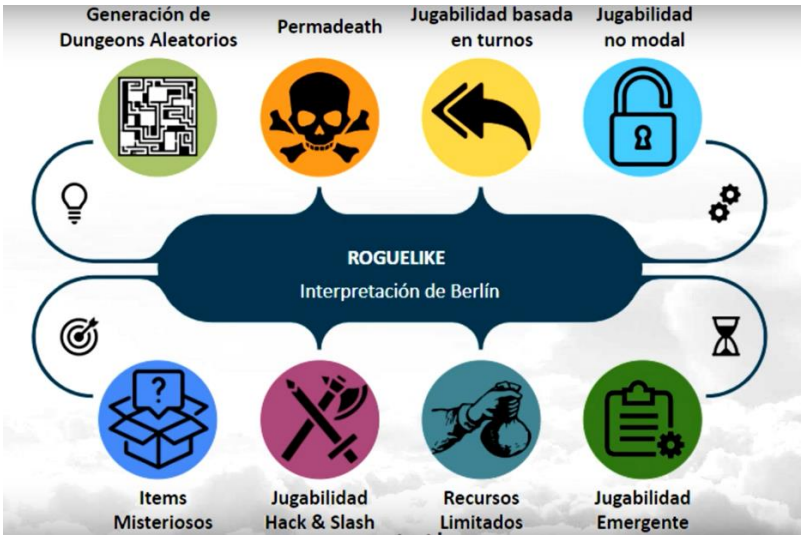

Figura 9 Interpretación de Berlín

Por otra parte, el género continuó progresando, dando origen a nuevos subgéneros, como el que fue creado por Rogue Legacy (Cellar Door Games, 2013). Este videojuego comparte los elementos fundamentales de la "Interpretación de Berlín", lo cual llevó a sus desarrolladores a denominarlo como "*Rogue-Lite*", una interpretación "ligera" del género en la que solo se deben cumplir dos características principales: la generación aleatoria y la muerte permanente (Jiménez, 2018). Gracias a esta nueva definición como *Rogue-Lite* el género se expandió y ahora es mucho más conocido y jugado globalmente.

COMPOUND (Figura 10)

Compound (notdead LLC, 2022) además de ser un *Rogue-Lite* es un *Shooter* que revive la estética retro aportando mucha diversión y nostalgia, sus desbloqueos son principalmente modificadores para próximas partidas. El *gunplay* nos sirvió de inspiración para nuestro videojuego.

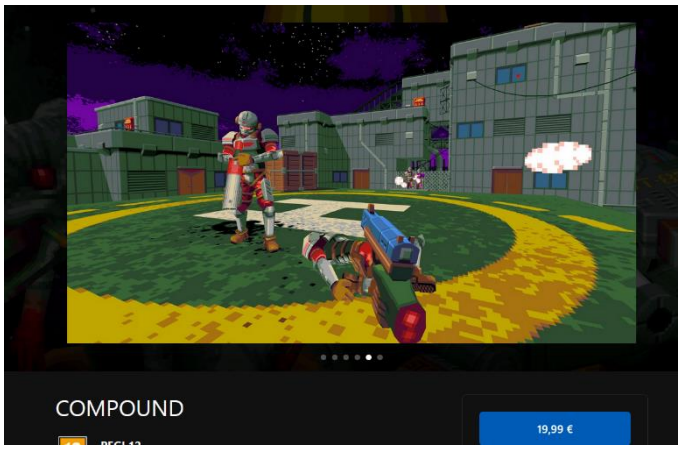

Figura 10 Ficha de Compound en Meta Quest.

Ancient Dungeon (Figura 11)

Ancient Dungeon (Thullen, 2020) nos pone en la piel de un aventurero que entra en una mazmorra cambiante, sus desbloqueos son salas nuevas para la mazmorra y modificadores de armas. La mazmorra cambiante nos sirvió de inspiración para nuestro juego.

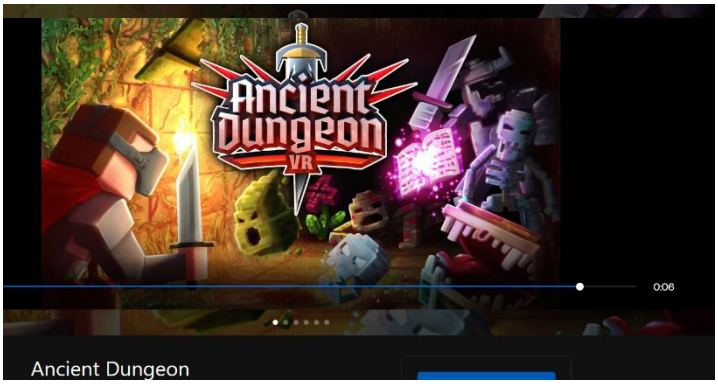

Figura 11 Ficha de Ancient Dungeon en Meta Quest

Dead Hook (Figura 12)

Dead Hook (Joy Way, 2023) es una experiencia VR en el mundo de DOOM (2016), este juego será muy frenético, y las mejoras que desbloquearemos serán aptitudes para el personaje y armas.

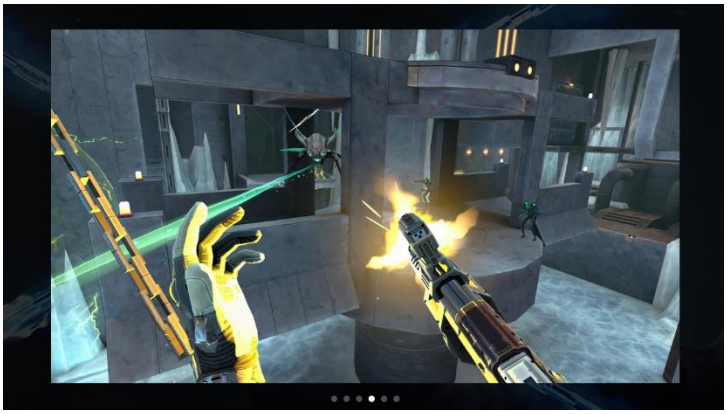

Figura 12 Ficha de Dead Hook en Meta Quest

Until You Fall (Figura 13)

Until You Fall (Schell Games, 2020) es un juego de peleas rítmicas, en el que se integra también una progresión después de morir. Al finalizar cada partida se desbloquearán mejores armas y hechizos que permiten al jugador, empezar nuevas partidas siendo más poderoso.

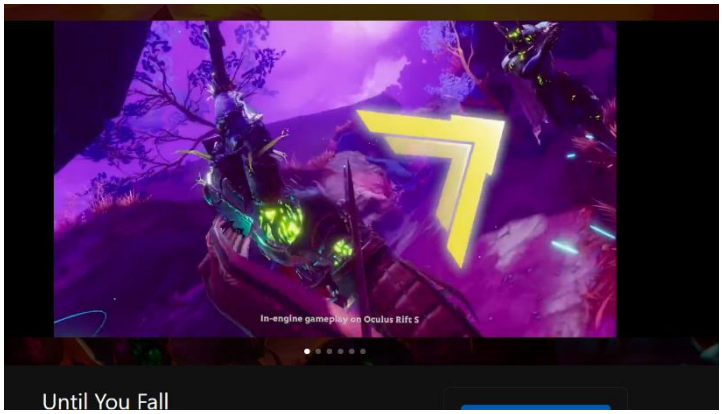

Figura 13 Ficha de Until You Fall en Meta Quest

2.3 Hardware disponible y utilizado

A continuación, se discute sobre algunos de los más importantes sistemas de VR disponibles en el mercado. En esta sección compararemos las Meta Quest 2, las Pico 4, las HP Reverb G2 y las HTC Vive Pro-2 entre sí. Al final de la comparación expondremos cual hemos escogido para el desarrollo del proyecto y las principales razones. Para realizar la comparación de forma precisa se empleará la página VR-Compare (Brown, 2023).

2.3.1 Meta Quest 2

Las Meta Quest 2 (Figura 14) ofrecen una resolución buena, 1832x1920 pixeles por ojo y la mejor tasa de refresco de las 4, junto a las Vive Pro 2, 120 Hz. Un punto muy positivo es su peso de 503 gramos, siendo una gafa muy ligera que no molesta demasiado en sesiones largas de uso.

Es un sistema *standalone*, lo que significa que no necesita de conexión a un PC para funcionar, tiene integrada una tienda de aplicaciones llamada Meta Quest que permite acceder a todo tipo de contenidos. También cuenta con conexión opcional a PC a través de un USB-C, con la funcionalidad de Quest Link, o empleando el método Wireless (Air Link), esta conexión pasa todo el trabajo de procesamiento al dispositivo conectado y nos permite controlar el PC desde las gafas, a través de un escritorio virtual. Las tiendas de aplicaciones disponibles para ésta son Meta Quest de forma nativa, y Steam (Valve Corporation, 2003) empleando los métodos de conexión anteriores.

Por lo que respecta al tracking, usa el método de seguimiento "*inside-out*", lo que significa que los sensores están integrados en el propio dispositivo y no requieren sensores externos ni estaciones base (Holzwarth, 2021).

Sus controladores son ergonómicos y emplean pilas para recargarse, dando una duración de batería bastante larga. También incluye seguimiento de manos, por lo que es posible usar la gafa sin los controladores.

Como puntos negativos del sistema tenemos, el *strap* de sujeción, que no es demasiado estable, provocando molestia en juegos o contenidos donde hay que moverse mucho. Aun así, estos fallos se compensan con su bajo precio de 400 euros.

Figura 14 Componentes de las Meta Quest 2

2.3.2 Pico 4

Las Pico 4 (Figura 15) ofrecen una muy buena resolución de 2160x2160 pixeles por ojo, y una tasa de refresco de 90 Hz. Las Pico 4, pesan 600 gramos y su sistema de sujeción ayuda a que no se sientan pesadas distribuyendo el peso de forma adecuada. Además, destacan por ser unas de las primeras gafas con lentes *Pancake*, que otorgan beneficios como ligereza o menor volumen (Hou, 2022).

Son un sistema *standalone* y se pueden conectar a un PC a través de un USB-C o wifi, como las Quest 2, aunque no incluye un escritorio virtual. En cuanto a los controladores, éstos son cómodos y ergonómicos, estas gafas son las más nuevas de las 4 y tanto en la distribución de pesos y construcción de sus controladores se nota que han aprendido de los fallos de sus competidores.

Al igual que Quest 2, utiliza el método de seguimiento "*inside-out*", sin estaciones base.

Su punto más negativo es que es un sistema poco desarrollado, es decir, si comparamos su software con su principal competidor, Meta Quest 2, se nota enseguida que es un dispositivo nuevo y le faltan actualizaciones, además su catálogo de aplicaciones en *standalone* no es tan diverso como otros, y esa poca optimización al conectarse a un PC provoca que muchos usuarios sigan usando Quest 2 como gafa para conectar a Steam. Aun así, este sistema es uno a tener en cuenta, dado que de no ser por el software sería una opción mucho mejor a otras en el mercado, su precio es de 430 euros.

Figura 15 Componentes de las PICO 4

2.3.3 HP Reverb G2

Las HP Reverb G2 (Figura 16) son la propuesta de HP para entrar al mercado VR. Es un sistema VR que necesita estar conectado a un PC para funcionar, esta conexión se realiza a través de un puerto *DisplayPort* y un USB-C. Como tienda de aplicaciones emplea Windows Mixed Reality y Steam.

Su resolución es muy buena, 2160x2160 pixeles por ojo, y su tasa de refresco es de 90 Hz. Su peso es de 498 gramos, resultando la gafa más ligera de las 4 y por lo tanto ideal para sesiones largas de uso. Además, el casco es muy ergonómico y tiene la posibilidad de ajustarse mediante velcros.

Tanto la gafa como los controladores emplean el método de seguimiento "*inside-out*", sin estaciones base. Esta gafa no cuenta con un tracking tan preciso como las Pico 4 o las Oculus, esto ocurre por el posicionamiento de las cámaras y por el sistema de seguimiento de los mandos que está basado en luminosidad y no en infrarrojos, lo que provoca que al tener poca o mucha luz varie la precisión con la que sigue los controladores.

En cuanto al rendimiento, este visor requiere de un PC potente para funcionar, por lo que dependeremos de éste, además de la optimización de Windows Mixed Reality. Este visor tiene un precio de 599 euros.

Figura 16 Componentes de la HP Reverb G2

2.3.4 HTC Vive Pro-2

Las HTC Vive Pro-2 (Figura 17) tienen la mejor resolución de pixeles por ojo de las 4, 2448x2448 y cuentan con una buena tasa de refresco, 120 Hz. Un elemento que destacar en estas gafas es su campo de visión, éstas tienen el mayor de las 4 con 116º horizontales, 96º verticales y 113º diagonales.

Son las más pesadas de las 4, pesando 850 gramos. Su sistema de seguimiento difiere de los otros cascos al emplear estaciones base. Además, debe conectarse con USB-C y *DisplayPort* al PC para funcionar. Como tienda de aplicaciones y juegos tiene Viveport (HTC, 2016) y Steam, lo que da más variedad a los usuarios de este dispositivo.

El aspecto más débil del sistema son sin duda sus controladores, éstos se quedan atrás respecto a la competencia, no llegan a ser tan cómodos como otros. Aun así, estas gafas están consideradas como unas de las mejores del mercado por las características de alta gama de su visor. Sin duda una buena compra si estás dispuesto a pagar el precio de 1700 euros.

Figura 17 Componentes de la HTC Vive Pro-2

2.3.5 Conclusión

El dispositivo que hemos elegido para desarrollar el proyecto ha sido la gafa de Meta, las Meta Quest 2. Se eligió el sistema por las siguientes razones:

- Dispositivo *standalone*.
- Tracking autónomo.
- Posibilidad de utilizar las manos como controladores.
- Publicación en Meta Quest y Steam.
- Precio económico.
- Comunidad de desarrolladores activa.
- Accesorios de terceros útiles.

Respecto al último punto mencionado, encontramos unos accesorios para las Meta Quest 2 que mejoraban la experiencia de uso, y solucionaban algunos de los fallos que presenta la gafa, los cambios que realizamos al dispositivo fueron:

- Cambiar la correa de sujeción por un halo configurable, que nos daba más estabilidad y repartía mejor el peso.
- Cambiar el *facial cover* para dar más espacio a los ojos y una experiencia más cómoda.
- Se añadieron dos lentes graduadas que se ajustaban a nuestras dioptrías, lo que nos permitió ver de forma nítida los contenidos en VR sin emplear gafas o lentillas.

2.4 Análisis plataformas de desarrollo

A continuación, realizaremos un pequeño análisis de las características que ofrecen las principales plataformas de desarrollo de aplicaciones VR. Cabe destacar que el análisis será sobre SteamVR (Valve Corporation, 2016), Oculus SDK (Meta, 2013) y XR Interaction Toolkit (Unity Technologies, 2021), dado que el dispositivo que se seleccionó para el desarrollo son las Meta Quest 2 y como motor de videojuegos Unity (Unity Technologies, 2005).

SteamVR

SteamVR es una plataforma de Realidad Virtual desarrollada por la empresa de videojuegos Valve, la cual creó entre otros el mítico Half Life y su secuela de VR Half Life Alyx. Destaca frente a otros sistemas por la amplia compatibilidad y los recursos disponibles, además tiene una fácil integración con Unity e incluso deja preparadas algunas escenas de prueba con elementos que puedes aplicar a tu juego de forma instantánea.

La gran desventaja de este sistema es que si quieres publicar un juego utilizando SteamVR debes hacerlo en Steam, con todo lo que ello conlleva, es decir, comisión de ingresos del 30% y alta competencia, aparte de eso, algunos usuarios se quejan de que no está del todo bien optimizado.

Por otra parte, sólo es posible publicar juegos en versión PCVR, no permitiendo la publicación directa para cascos *standalone*.

Oculus SDK

El SDK de Oculus tiene una gran ventaja y desventaja al mismo tiempo, es un sistema especialmente desarrollado para sistemas Oculus, lo que permite una optimización muy buena y el acceso a las características de las gafas como la detección de manos, su desventaja es que necesitas precisamente este tipo de gafa para desarrollar, no puedes emplear este SDK con otras como por ejemplo cualquier casco de HTC.

Otra ventaja sería el acceso a la tienda de Meta, una plataforma específicamente desarrollada para ofrecer juegos y aplicaciones VR, por lo que todos los usuarios de Meta podrían llegar a ver y comprar tu producto.

Además, permite la publicación del juego tanto en versión *standalone* como en PCVR.

XR Interaction Toolkit

XR Interaction Toolkit ofrece compatibilidad multiplataforma, además de una sencilla implementación de los controles básicos y un gestor de eventos implementado, a parte, cuenta con una de las comunidades más activas del desarrollo en VR. Una de las mayores ventajas es que, este SDK esta desarrollado por Unity, lo que significa que la compatibilidad con el motor está integrada de base y las actualizaciones son continuas.

Además, permite la publicación del juego tanto en versión *standalone* como en PCVR.

Las desventajas de este sistema serían las limitaciones de cara a las características únicas de cada gafa y que la curva de aprendizaje es más elevada que otros SDK.

Para el desarrollo de este proyecto elegimos XR Interaction Toolkit, principalmente por las siguientes razones.

- Compatibilidad entre diferentes plataformas de Hardware.
- Posibilidad de publicar en cualquier tienda.
- Desarrollar en una plataforma diferente a la aprendida en el grado.
- Implementación sencilla y comunidad de desarrolladores activa.

2.5 Requerimientos, documento de diseño de juego

2.5.1 Concepto

2.5.1.1 Descripción general

En este juego controlarás a un miembro de la organización secreta de los Sangre Espectral, tu deber será proteger a los ciudadanos de Scadrial de las peligrosas organizaciones criminales y los planes de dioses malignos.

Conforme avances en el juego obtendrás poderosas artes metálicas y mejoras de armamento con las que dar la vuelta a las situaciones más peligrosas.

2.5.1.2 Puntos clave

- Descubrir las salvajes aldeas terrisanas, las bonitas calles de Elendel o las catacumbas de la época de las leyendas.
- Emplear tus poderes para dar la vuelta a los tiroteos.
- Cambiar de armas según te convenga con el ágil sistema de la rueda de armas.

2.5.1.3 Genero

Acción, *Rogue-Lite* y *Shooter* en primera persona.

2.5.1.4 Audiencia

PEGI 7

2.5.1.5 Plataforma

Distribución: Steam y Meta Quest.

Ejecución: PCVR Windows y Oculus *standalone* (Android)

2.5.1.6 Requerimientos técnicos

Es un juego de un jugador sin funcionalidades en línea.

2.5.1.7 Descripción de tipos de gameplay

Rogue-Lite: Los juegos de tipo *Rogue-Lite* son un subgénero del tipo *Rogue-Like* y se caracterizan por tener niveles generados de forma aleatoria, donde la muerte del jugador es permanente y para volver a jugar tiene que iniciar una nueva partida, desde cero. El jugador progresa desbloqueando nuevos objetos o formas de iniciar el nivel, como modificadores especiales u objetos de inicio.

Shooter: Conforman un género que engloba un amplio número de subgéneros que tienen la característica común de permitir controlar un personaje que, por norma general, dispone de un arma (mayoritariamente de fuego) que puede ser disparada a voluntad.

2.5.1.8 Historia y narrativa

Nos situamos en el contexto de la saga de libros Mistborn, concretamente en la etapa temporal de la segunda era, el jugador pertenece a una organización secreta conocida como los Sangre Espectral y su objetivo será desbaratar las organizaciones criminales que ponen en peligro al planeta.

Al iniciar la partida, el jugador se encuentra en una sala de planificación donde puede empezar una misión o revisar su equipamiento.

Las misiones se completan al derrotar a todos los enemigos, recoger algún objeto o aguantar ciertas oleadas. Una vez acabada la misión, el jugador volverá a la sala de planificación y podrá emprender otra misión cuando quiera.

2.5.1.9 Controles

Los controles de los distintos sistemas de juego se pueden ver en las Figuras 18, 19, 20 y 21

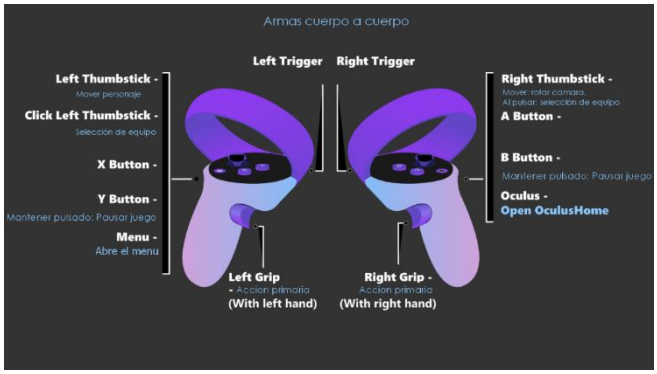

Figura 18 Diseño de Controles Cuerpo a cuerpo

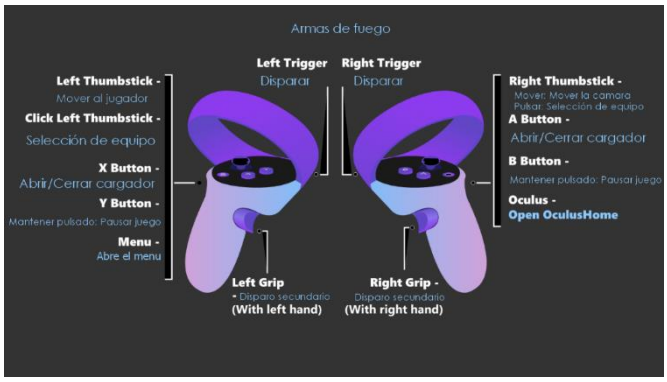

Figura 19 Diseño de Controles Armas de fuego

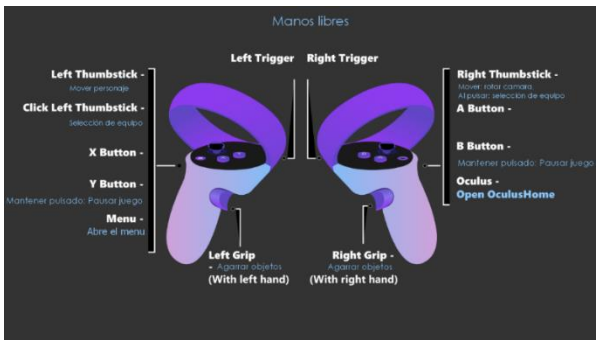

Figura 20 Diseño de controles Manos libres

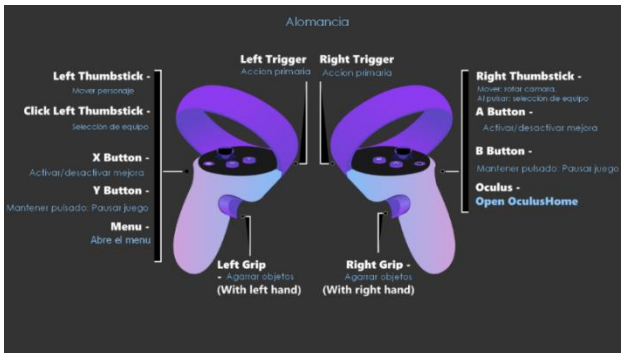

Figura 21 Diseño de controles Alomancia

2.5.1.10 Cámara

Cámara en primera persona, usando un casco o headset VR

2.5.1.11 UI/HUD (Head-Up Display)

- Líneas azules, para los objetos metálicos.
- Indicador de reserva de metales.
- Indicador de salud.
- Ruleta de armas.
- Objetivo de la misión.

2.5.1.12 Wireframe de la progresión jugable

En el diagrama de la Figura 22 se puede ver el ciclo jugable propuesto.

Figura 22 Ciclo Jugable

2.5.2 Interfaz de jugador a nivel alto

2.5.2.1 Menú

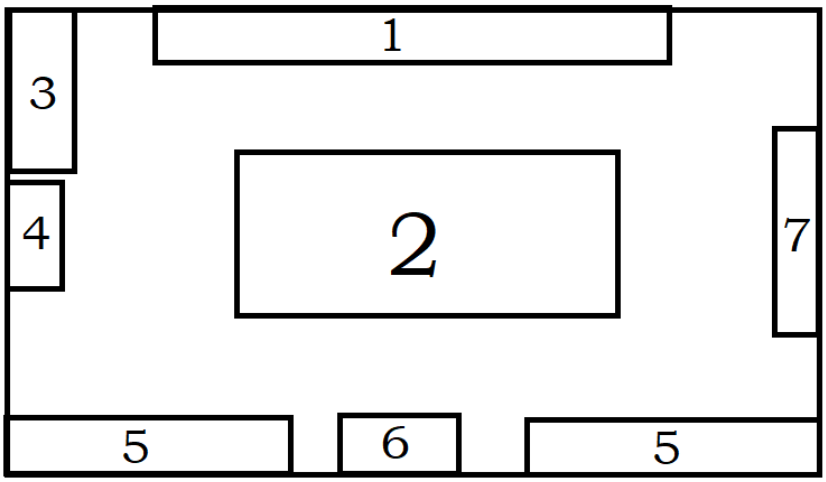

Figura 23 Diseño Sala de preparación

Al desarrollar un juego para dispositivos VR, el menú puede ser un elemento físico del escenario, en la Figura 23 se ve un diseño de sala con las siguientes características:

- La marca 1, representa las vitrinas con los desbloqueables de la partida, aquí el jugador encontrará el equipamiento desbloqueado después de cada misión.
- La marca 2, representa el mapa, donde el jugador elige el próximo nivel colocando una representación de sí mismo en una de las marcas del mapa.
- La marca 3, representa la selección de equipamiento, el jugador deberá llevar los objetos de las vitrinas hasta aquí para poder utilizarlos en la misión, esto incluye tanto armas como alomancia.

- La marca 4, representa la puerta, es la salida del nivel y el inicio de la misión.
- La marca 5, representa la zona de estadísticas o noticias como el desbloqueo de nuevo equipamiento.
- La marca 6, representa el altar de explicación de equipamiento donde el jugador puede dejar equipamiento y se le proporciona una explicación de qué hace y cómo se usa.
- La marca 7, representa los elementos de decoración de la sala.

2.5.2.2 Logos

Deberán aparecer el logo de Unity, el diseñado para el juego y el logo de Mistborn/Brandon Sanderson.

2.5.2.3 Opciones

La única opción de sonido es la integrada por las Meta, que permiten subir y bajar el volumen.

2.5.2.4 Pantallas de carga

Al cambiar de escena hay un fundido a negro que oculta la pantalla, una vez carga el nivel se deshace el fundido a negro y el jugador puede actuar con normalidad.

2.5.2.5 Puntuación

Después de cada misión el jugador desbloqueará progresivamente equipamiento, según la cantidad de enemigos derrotados y niveles completados, además, recibirá una recompensa en forma de puntos, atium, el metal divino de Ruina, que podrá utilizar para desbloquear permanentemente equipamiento, cada vez que sea derrotado.

El equipamiento desbloqueado permanentemente se mantiene entre partidas, igual que el atium. Todo objeto que no haya sido desbloqueado de forma permanente se deberá volver a desbloquear jugando.

2.5.3 Game Overview

2.5.3.1 Mecánicas de juego

Las mecánicas de juego se pueden dividir en 3 grupos, Preparación, Exploración y Combate.

Preparación

En la sala de preparación, antes de empezar cada nivel, el jugador deberá seleccionar el próximo nivel a completar y que equipamiento quiere llevar a la misión. Cuenta con dos espacios para armas/poderes en cada mano.

Exploración

Dado que se trata de un juego VR, prácticamente cualquier objeto del escenario puede ser recogido o interactuado, esto nos servirá para que el jugador pueda emplear de forma creativa el escenario, recogiendo objetos que le ayuden en el progreso del nivel o incluso encontrando munición o paquetes de salud para reabastecerse.

Por ejemplo: El jugador puede agarrar un escudo y bloquear balas o lanzar objetos para dañar un poco a los enemigos.

Combate

En combate se empleará el uso de armas a distancia, cuerpo a cuerpo o arrojadizas. Existen tres tipos de armas:

- El revólver, que cuenta con el disparo normal y un disparo secundario que lanza una ráfaga de disparos.
- La escopeta de una mano, que al disparar lanza metralla y cuenta con un disparo secundario que dispara las dos balas al mismo tiempo.
- Armas arrojadizas, como dinamita o dagas que se recogen en los propios niveles.

Las armas cuerpo a cuerpo a elegir son la daga y los bastones de duelo.

- La daga tiene poco rango y buen daño.
- El bastón tiene más rango y es igual de dañino que la daga.

Además de estas armas más convencionales se pueden emplear los poderes alománticos y feruquímicos para darle la vuelta al combate. Los poderes disponibles se pueden consultar en la Tabla 1.

Tabla 1 Habilidades Alománticas y Feruquímicas

Metal	Alomancia (Activa)	Feruquimia (Pasiva)	Mejoras alomancia
Hierro	Atraer objetos	Mayor velocidad	Mayor daño al golpear a un enemigo con un objeto atraído.
Acero	Empujar objetos, empujar las propias balas	Esquivar balas	Escudo deflector de balas
Estaño	Aumenta los sentidos (enemigos)	Aumenta los sentidos (objetos)	Aumento de alcance
Peltre	Resistencia mejorada	Vida aumentada + fuerza aumentada	Aumento de resistencia
Zinc	Un enemigo se vuelve más agresivo	Tiempo bala	Afecta a más de un enemigo
Latón	Un enemigo se vuelve menos agresivo	----------	Afecta a más de un enemigo
Oro	----------	Regeneración	Mejor regeneración

2.5.3.2 Mundo

Se puede acceder a todos los niveles desde el inicio, pero algunos son más difíciles que otros, la progresión normal sería mazmorra, ciudad y aldea. Cada uno da una cantidad distinta de atium al completarse.

2.5.3.3 Niveles

En el juego habrá 3 tipos de niveles.

Las mazmorras de la época de leyenda son niveles generados de forma semi-aleatoria, donde no hay muchos enemigos y están divididos entre salas. El jugador puede enfrentar a los enemigos poco a poco y los enemigos que aparecen llevan armas de corta distancia.

La ciudad es un nivel donde el objetivo es asaltar una casa. En estas misiones, generalmente, hay muchos enemigos. Además, hay que derrotar a los enemigos que están dentro de la casa y una vez derrotados, el jugador deberá resistir una oleada. La casa cuenta con munición, por lo que, si el jugador consigue limpiar la casa, tendrá los recursos suficientes para aguantar más enemigos.

La aldea, en este nivel el jugador tendrá que defender una aldea de las actividades de una banda. Este tipo de niveles se caracterizan porque el jugador se encuentra en un espacio abierto y van apareciendo enemigos constantemente. Será muy importante el posicionamiento y acabar con los enemigos que sean de larga distancia. En esta misión hay barriles explosivos que ayudarán al jugador a acabar con grupos de enemigos rápidamente.

2.5.3.4 Progresión del jugador

Cuando un jugador completa una misión se cuentan los niveles completados y los enemigos derrotados y se va desbloqueando equipamiento progresivamente, por ejemplo: Cuando el jugador completa 3 misiones de tipo mazmorra desbloquea la escopeta.

Si el jugador ha desbloqueado un arma o arte metálica en la partida, puede gastar atium, recurso que se obtiene completando misiones, y que sirve para desbloquear permanentemente ese equipamiento y así contar con dicha pieza una vez se empieza una nueva partida al ser derrotado.

2.5.3.5 Audio

Se pretende utilizar una banda sonora de música tipo western y efectos de sonido, dentro de lo que cabe, realistas.

- Cada nivel tendrá una música de fondo distinta.
- Al seleccionar nivel o equipamiento en la sala de preparación se escuchará un sonido característico.
- Al sacar un arma de la rueda se escuchará un sonido de desenfundado.
- Las armas tendrán sonido al ser disparadas, recargadas y al abrir o cerrar el cargador.
- Los golpes cuerpo a cuerpo también tienen sonido, navajazo o golpe sordo, depende del tipo de arma.
- Los enemigos tendrán sonidos al disparar, al recargar o al recibir daño.

2.5.4 El jugador

2.5.4.1 Matriz del jugador

A continuación, se detallarán las capacidades y estadísticas del jugador, además de sus armas, alomancia y el estado de muerte. Las armas que puede equipar el jugador, así como su daño y munición se encuentran en la Tabla 2.

Tabla 2 Daño y capacidad de munición de cada arma

Arma	Daño	Munición
Revólver normal	35	6
Revólver de cañón largo	50	6
Escopeta	30 x perdigón	2
Daga	65	-
Bastón de duelo	100	-

El jugador tiene 200 unidades de vida y esta cantidad está representada por una gema que se coloca en la mano izquierda.

Cuando la vida del jugador está al máximo, la gema aparece de color verde, cuando la vida está a la mitad, la gema aparece con un color anaranjado o amarillo y cuando la vida está baja, la gema aparece roja. El color se modifica dinámicamente, al recibir daño o sanación.

El jugador tiene 100 unidades de reserva metálica, con esta reserva puede utilizar sus poderes alománticos, en el juego está representada con una pulsera, esta pulsera se coloca en la mano derecha y el indicador es una perla que se apaga conforme se gastan los poderes.

En cuanto al equipamiento relacionado a la Alomancia y a la Feruquimia se representan con clavos.

Cada clavo representa un metal y poder distintos, se han elegido colores distintivos para cada clavo, similares a sus colores reales. Identificarlos según color y nombre puede ser complicado para quien no conozca los libros, por lo que pasarlos por el altar de explicación será necesario para comprender que habilidades otorga cada uno. El listado de metales y poderes se presenta más adelante.

Respecto al estado de muerte del jugador, cuando éste muere es transportado al reino cognitivo, donde puede adquirir equipamiento de forma permanente gastando atium y, además, puede volver a la vida.

El sistema de movimiento elegido es el de movimiento continuo, la elección de este sistema en contra del típico movimiento en VR por teletransporte se debe a que el movimiento continuo ofrece al jugador una experiencia de juego más fluida, como se ha visto en otros juegos durante la fase de estudio.

2.5.4.2 Progreso del jugador

Los requisitos para desbloquear cada pieza de equipamiento pueden consultarse en la Tabla 3.

Tabla 3 Desbloqueos de cada pieza de equipamiento

Arma	Requisito	Arma	Requisito
Revólver	De inicio	**Revólver Largo**	Número total de enemigos muertos
Escopeta	Mazmorra completada	**Daga**	De inicio
Bastón	Número total de enemigos muertos	**Hierro alomántico**	Revólver
Acero alomántico	Rifles	**Estaño alomántico**	Número total de niveles completados
Peltre alomántico	Ciudad completada	**Zinc alomántico**	Mazmorra completada
Latón alomántico	Mazmorra completada	**Hierro feruquímico**	Aldea completada
Acero feruquímico	Número total de niveles completados	**Estaño feruquímico**	Aldea completada
Peltre feruquímico	Ciudad completada	**Zinc feruquímico**	Número total de enemigos muertos
Oro feruquímico	Escopeteros		

Coste en atium: El equipamiento más básico cuesta 2 unidades de atium, las armas como el revólver largo o el bastón de duelo 3 y todas las mejoras Feruquímicas pasivas cuestan 3 también. Esto creará un dilema cuando el jugador muera, ya que tendrá que decidir cuál es el equipamiento a mantener permanentemente.

2.5.4.3 Armas

El jugador cuenta con pocas opciones cuando comienza una nueva partida. Para ser exactos tiene el revólver normal y la daga. Estas dos armas no hacen mucho daño, pero permiten al jugador tener enfrentamientos intensos.

El revólver corto tiene un daño de 35, con lo que se necesitan 3 disparos de los 6 que tiene por cargador para matar a un enemigo normal. Por otro lado, la daga tiene un daño de 65 y está pensada para poder acabar con los enemigos que se echen encima del jugador o como última opción, en caso de no tener balas. Estas dos armas iniciales son una buena pareja porque permiten acabar con un enemigo normal de un tiro y una puñalada.

El revólver largo cuenta con la misma capacidad de balas que el revólver corto, pero añade un buen incremento de daño, tiene un daño de 50, lo que permite al jugador acabar con enemigos básicos de dos disparos.

La escopeta solo tiene 2 balas de capacidad, pero lo compensa con una gran cantidad de perdigones y un daño moderado de 30. Al disparar aparecerán 6 perdigones que se repartirán según una variable de dispersión, hay que tener cuidado con la dispersión porque si el jugador no se encuentra suficientemente cerca, puede fallar el disparo contra el objetivo y quedarse expuesto. La escopeta tiene un rango efectivo de 4 o 5 unidades si se apunta al cuerpo.

Los estados de las armas son los siguientes:

- El revólver normal cuenta con la capacidad de disparar, recargar y hacer trucos.
- El revólver largo cuenta con la capacidad de disparar, recargar y hacer un disparo en ráfaga.
- La escopeta cuenta con la capacidad de disparar, recargar y hacer un disparo doble.
- Y la daga cuenta con la capacidad de cambiar el agarre para apuñalar con un movimiento distinto.

2.5.4.4 Utilidades

Las utilidades son todos aquellos objetos que el jugador no se puede equipar en la sala de preparación. Estos objetos se encuentran tanto en el escenario como en los *drops* de los enemigos. Los objetos se dividen en tres categorías.

- Munición: Cargadores para los revólveres o la escopeta.
- Recuperación: Objetos para recuperar salud o frascos de metales para recargar alomancia.
- Armas arrojadizas: Tanto la dinamita como los cuchillos arrojadizos, todo lo que se pueda lanzar y haga daño al enemigo.

2.5.4.5 Modificadores al jugador o los enemigos

Aquí entran en juego la Feruquimia y la Alomancia.

Primero trataremos el efecto de la Alomancia y la Feruquimia en el jugador. Los distintos metales feruquímicos modifican estas estadísticas o capacidades. En la Tabla 4 se puede ver en detalle qué estadísticas se modifican.

Tabla 4 Modificaciones de la Feruquimia a las estadísticas del jugador

Metal Feruquímico	Vida	Velocidad	Esquive	Fuerza	Regeneración	Ver objetos	Parar el tiempo
Ninguno	200	1.1	False	False	False	False	False
Hierro	200	1.5	False	False	False	False	False
Acero	200	1.1	True	False	False	False	False
Peltre	300	1.1	False	True	False	False	False
Oro	200	1.1	False	False	True	False	False
Estaño	200	1.1	False	False	False	True	False
Zinc	200	1.1	False	False	False	False	True

En cuanto a los efectos alománticos sobre el jugador solo influyen el peltre y el estaño, ya que el hierro y el acero son para objetos del entorno y el zinc y el latón para enemigos.

- El peltre permite resistir más golpes, se recibe la mitad del daño.
- El estaño permite ver a los enemigos a través de las paredes.

En cuanto a los enemigos, éstos pueden ser afectados por alomancia mental, es decir el zinc y el latón. El zinc provoca furia en los enemigos lo que les hace no distinguir a sus aliados y tratarlos como enemigos, el latón disminuye el radio de detección de los enemigos hasta el punto de anularlo, por lo que si el jugador tiene muchos enemigos encima puede usar el poder para dejar alguno fuera de combate.

2.5.4.6 Estados de jugador o enemigos

Todos los enemigos tienen los mismos estados, aunque las animaciones son distintas. En el juego hay tres tipos de enemigos, los que llevan revólver, los que van con escopeta y los que llevan rifles. En la figura 24 se puede observar los distintos estados de los enemigos y cómo cambian de uno a otro.

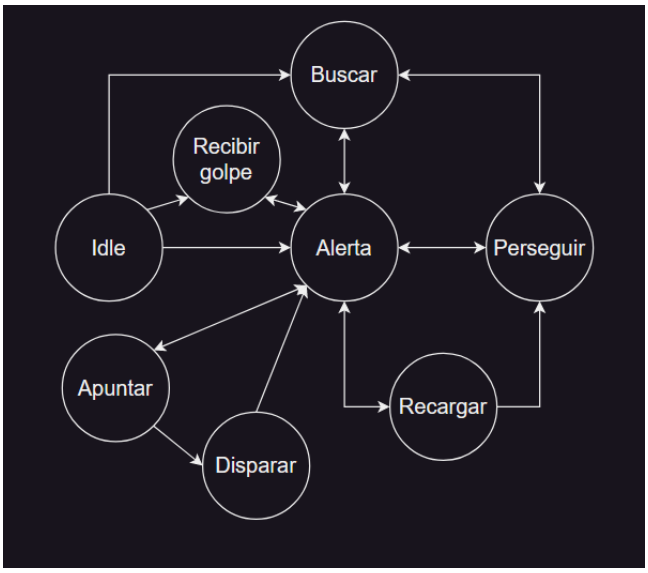

Figura 24 Estados de los enemigos

2.5.4.7 Muerte del jugador o enemigos

El jugador es trasladado a la escena de muerte cuando su vida baja a 0, donde podrá utilizar el atium conseguido durante la partida para desbloquear equipamiento de forma permanente y volver a empezar a jugar. La escena de muerte está inspirada en el reino cognitivo de los libros.

Cuando un enemigo muere, se activa el modo muñeco de trapo, dejando encima de él, un objeto de su pool de ítems, el pool normalmente es munición del arma que lleve, u otro objeto útil. Los objetos que suelta cada enemigo al morir, junto con sus probabilidades, se pueden consultar en la Tabla 5.

Tabla 5 Probabilidades de drops de los enemigos

Enemigo	Objeto	Probabilidad
Revólver	Munición revólver	50
Revólver	Salud	25
Revólver	Investidura	25
Escopeta	Munición escopeta	75
Escopeta	Salud	25
Rifle	Salud	50
Rifle	Investidura	50

2.5.5 Pantallas de juego

2.5.5.1 Sistema de guardado

Se guarda partida de forma automática después de completar o fallar un nivel, y volver a la sala de preparación. Los datos que se guardan son los objetos desbloqueados, el atium y todo lo referente a estadísticas y desbloqueos.

Estos datos se guardan en un archivo que se encripta en binario y se escribe en una ruta segura de Unity, haciendo que no se puedan modificar fuera del juego.

2.5.5.2 Economía

El atium se consigue después de completar un nivel de forma exitosa, la cantidad de atium que se recibe es diferente dependiendo del nivel que se haya completado.

La mazmorra es un nivel corto con pocos enemigos y el jugador consigue una unidad de atium por completarla.

La casa es un poco más compleja porque hay bastantes enemigos y después aparece una oleada, por lo que el jugador obtiene dos unidades de atium al completar este nivel.

La aldea es el nivel más difícil de todos, el jugador debe derrotar 5 oleadas de enemigos, si lo consigue obtiene 3 unidades de atium.

Cuando el jugador muere es llevado al reino cognitivo donde se le brinda la oportunidad de comprar el equipamiento que haya desbloqueado, por lo general, el equipamiento básico cuesta 2 unidades de atium y las piezas de armamento más potentes o Feruquimia 3.

Para que el jugador pueda comprar un objeto, esté debe colocarlo en el altar de la zona de muerte. Si el jugador tiene suficiente atium para comprarlo, entonces el objeto desaparece y se ejecuta una explosión de partículas. A partir de ese momento dispondrá del objeto permanentemente.

2.5.6 NPCs (non-player characters), Enemigos, IA

2.5.6.1 Tipos de enemigos

En el juego hay 3 tipos de enemigos, que se diferencian en tamaño, apariencia y uso de armas. Las características de los enemigos se pueden consultar en la Tabla 6, aunque las más destacadas son:

- El enemigo con revólver tiene vida y tamaño medio, es el enemigo básico.
- El enemigo con escopeta es el más grande y con más vida, además se mueve más rápido que sus compañeros.
- El enemigo con rifle es más pequeño, tiene menos vida y se mueve más lento.
- También cada uno tiene un rango de ataque distinto, siendo la escopeta el menor rango y el rifle el mayor.

Tabla 6 Estadísticas de los enemigos

Enemigo	Vida	Daño	Velocidad de ataque	Rango de ataque	Munición total	Velocidad
Revólver	100	20	1	8	6	3.5
Escopeta	150	10xperdigón	2	4.5	4	4
Rifle	50	40	4	15	4	2

2.5.6.2 Reglas de los enemigos

Los enemigos detectan al jugador cuando éste les dispara, o cuando éste entra en su radio de detección, además, si un enemigo detecta a un jugador, pero no lo tiene a tiro, buscará un ángulo desde el cual poder dispararle.

Cuando un enemigo es derrotado, tiene la posibilidad de soltar objetos útiles para el jugador, como curas, munición o recargas de reserva metálica.

El nivel de agresividad varía en función del enemigo. El escopetero, si no tiene tiro, irá corriendo a por el jugador. El que usa revólver espera unos segundos y luego intenta buscar ángulo para disparar al jugador. Por último, el del rifle espera bastante y si el jugador no sale de cobertura también intenta flanquearle, como el revólver.

Utilizando el zinc y el latón se puede variar la conducta de los enemigos. Uno los aturde y hace que no se muevan, es muy útil para dejar a enemigos fuera de combate. El otro hace que se vuelvan furiosos y no distingan aliados de enemigos, haciendo que sus compañeros los detecten como una amenaza y les disparen también.

2.5.6.3 Enemigos que solo aparecen en ciertas áreas

- En el nivel de la mazmorra aparecen el revólver y el escopetero.
- En el nivel de la ciudad aparecen dentro de la casa revólveres y rifles. Cuando el jugador los derrota, aparece una oleada adicional de revólveres y escopeteros.
- En el nivel de la aldea aparecen revólveres y rifles.

2.6 Implementación

2.6.1 Configuración básica del proyecto

Antes de empezar con el desarrollo del proyecto hay que dejar configurados ciertos apartados. Se empleó Unity para realizar el proyecto, la versión 2021.3.11f1, y se instalaron los módulos de

compilación para sistemas Android y Windows. Esto nos permitirá realizar *builds* para las gafas Meta de forma nativa y un .exe para poder ejecutar el juego desde un PC con cualquier gafa.

Además, Unity ofrece tres sistemas de renderizado, el Standard, que es el predeterminado y más extensamente utilizado, URP (Universal Render Pipeline), está basado en el trazado de rayos, lo que significa que utiliza técnicas avanzadas de sombreado y renderizado para lograr resultados visuales más realistas y precisos, optimizando los recursos de renderizado sin perder fidelidad gráfica y HDRP (High Definition Render Pipeline) que está diseñado específicamente para ofrecer gráficos de alta calidad y fidelidad visual en proyectos que requieren un aspecto más realista y detallado, a costa de requerir más recursos al sistema.

Como nuestro juego va a requerir de alguna técnica de renderización avanzada para representar algunos materiales y está pensado desde una perspectiva multiplataforma, se optó por URP. Al añadir URP desde el Packet Manager debemos establecerlo como sistema de renderizado y utilizar la pestaña de conversión para transformar todos los materiales a URP.

2.6.2 Setup del XR Interaction Toolkit

A continuación, es preciso configurar el proyecto para incorporar la funcionalidad de XR Interaction Toolkit, para ello, dentro de Unity, en los ajustes del proyecto, instalamos el XR Plugin Management, dentro de éste debemos elegir un sistema para la versión de PC y la versión de Android. En la versión de Android elegimos el sistema Oculus, dado que planteamos el juego para que funcionase de forma nativa en las gafas, y para la versión de PC escogimos el traductor Open XR (Khronos Group, 2019), que registra los inputs de cualquier controlador y los traduce a los del juego. Por último, tanto en Open XR y Oculus seleccionamos el *render mode* en *Multipass*, lo que hace que el juego se renderice en las dos lentes de la gafa cuando utilizamos ciertos efectos visuales.

Ahora que tenemos las configuraciones previas, accedemos al Packet Manager de Unity y descargamos el paquete de XR Interaction Toolkit para el proyecto, también descargamos el paquete adicional llamado Default Input Actions, que nos ahorra tiempo al preconfigurar las acciones más básicas de los controladores.

Con estos dos paquetes instalados podemos empezar a crear cualquier tipo de aplicación o videojuego en VR.

2.6.3 Revólveres y sistema de acciones

Una vez preparado XR Interaction Tookit buscamos modelos y *Assets* que nos permitiesen prototipar. Buscando armas encontramos un revólver que encajaba bastante bien en el proyecto, pero aun necesitábamos otro más, así que empezamos a modelar un revólver de cañón largo, parecido a *Vindicación*, un arma de la saga de Mistborn. Lo que en nuestro mundo sería un Smith and Wesson model 3, en la Figura 25 hay una imagen de referencia.

Figura 25 Revólver de referencia Smith & Wesson Model 3

Cuando importamos los revólveres y creamos los cargadores, nos pusimos a desarrollar el sistema de disparos y animaciones, estos elementos se controlan a través de un sistema de manejo de armas, que funciona de la siguiente manera; al realizar el jugador alguna acción, se registra el botón pulsado en el script correspondiente y se llama a una animación, la cual tiene asignado un evento que ejecuta una acción. Por ejemplo:

Al pulsar Trigger con el revólver, se ejecuta la animación de disparo, y cuando el gatillo llega a la parte más baja se ejecuta un evento, el evento es una función llamada Disparo, que provoca que el arma dispare una bala, al final de la animación se ejecuta otro evento que hace rotar la recamara en 60 grados, simulando que se ha movido la recamara y está lista para volver a disparar. En la figura 26 se ve un ejemplo de una animación, empleando la herramienta de animación de Unity. Esta herramienta nos permite establecer unos puntos denominados *keyframes* con los que especificar la posición y rotación del objeto que se esté animando, también, nos permite añadir eventos que ejecutan funciones en el código.

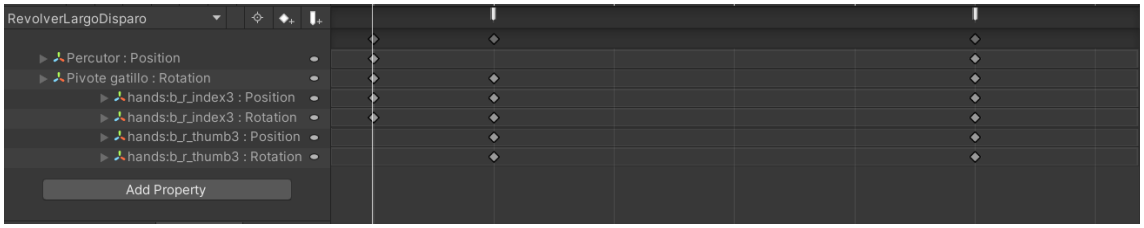

Figura 26 Animación de disparo del revólver largo

Después de eso se implementó el sistema de recarga, empezando por las animaciones que era lo más sencillo y rápido de añadir. Para el revólver pequeño hicimos que el cargador se abriese rotando a un lado, y en el caso del revólver largo, toda la parte del cañón se mueve hacia abajo. También, antes de hacer la recarga como tal, añadimos el cambio de armas, al pulsar el botón de movimiento o girar la cámara se equipaba o desequipaba el revólver.

Aquí ocurrió un problema, si el jugador abría la recamara del revólver y cambiaba a la mano, cuando volvía al revólver, éste se cerraba y abría constantemente, para solucionarlo se realizaron dos operaciones, la primera fue que cuando se pulsase el botón de cambiar de arma, si la recamara estaba abierta, se iniciase la animación de cerrar y luego se ejecutaba el cambio de arma, la segunda fue poner el estado de cerrar la recamara como idle.

Hicimos los dos revólveres iguales, lo único que cambiamos fue el daño que hace cada uno y las acciones secundarias. Pulsando Grip, el revólver largo lanza una ráfaga de tres balas y el revólver pequeño realiza trucos de forma aleatoria.

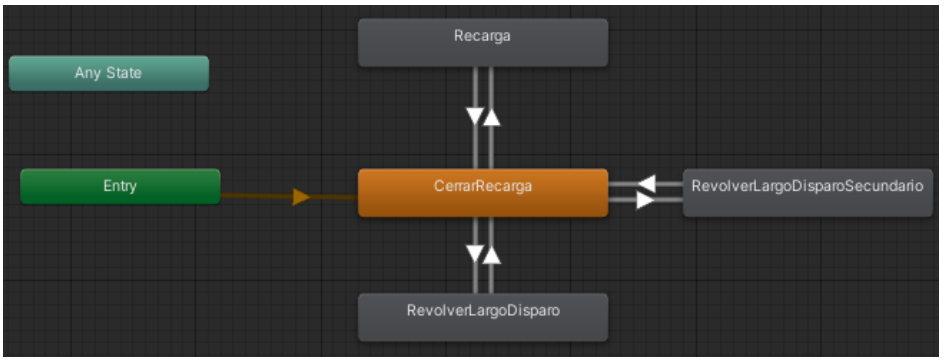

Figura 27 Estados de animación del revólver largo

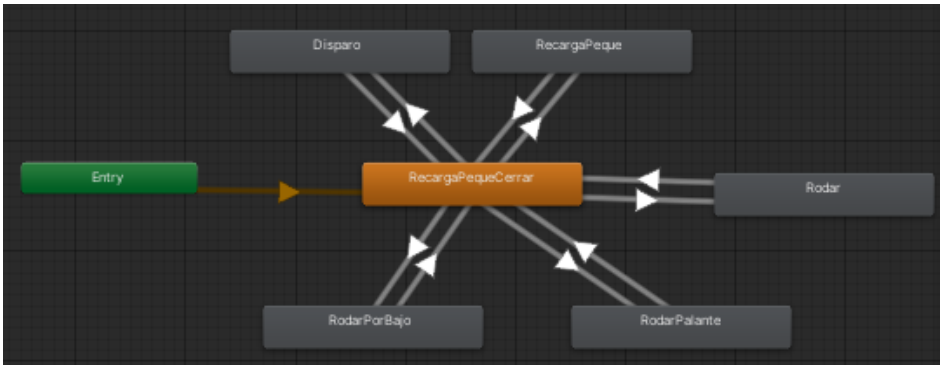

Figura 28 Estados de animación del revólver

En la figura 27 y 28 se muestra la máquina de estado de cada revólver y cómo se relacionan los estados entre ellos.

En este momento del desarrollo solo faltaba añadir que el arma se quedase sin munición y que fuese capaz de recargar. El contador de munición fue sencillo, en el código añadimos un número que definía cuantas veces se podía disparar el arma antes de recargarse y un contador que se incrementa cuando el arma se dispara, si el contador es igual al número máximo de disparos, el arma no se puede disparar y debe ser recargada.

Para el sistema de recarga empleamos los *Sockets*, que son un elemento muy conveniente de XR Interaction Toolkit. Estos elementos nos permiten guardar dentro de ellos otros objetos, es por así decir un bolsillo, a ese bolsillo se le puede asignar que solo entren determinados objetos, así que lo que hicimos fue que mientras el revólver tenía la recamara abierta el bolsillo estuviese también abierto y solamente aceptase objetos de tipo CargadorRevolver. Al añadir un cargador, éste se elimina y pone el contador de disparos a cero.

También añadimos que al abrir la recamara los casquillos del arma saliesen despedidos, y al recargarse volviesen a aparecer dentro, por temas de optimización, hicimos que los casquillos que quedaban fuera volviesen a su posición original, de esta forma evitábamos crear y destruir de forma innecesaria esos objetos.

2.6.4 Sistema cuerpo a cuerpo

Una vez implementado todo el sistema de combate a distancia nos propusimos hacer un sistema rudimentario de cuerpo a cuerpo. Para ello buscamos un modelo de una daga y empezamos a planear un sistema de vida y daño para el jugador y los enemigos.

El sistema de daño cuerpo a cuerpo se implementó calculando la velocidad a la que se movía la daga, si al impactar contra un enemigo el impacto era suficientemente rápido el enemigo recibía daño. También añadimos la capacidad de cambiar el tipo de agarre de la daga, para que el jugador pudiese apuñalar de la forma más cómoda a sus enemigos. Dado que ahora sabíamos cómo manejar la velocidad se añadió una funcionalidad más al sistema de recarga de armas. Si el cargador estaba abierto el jugador podía cerrarlo con un movimiento de muñeca.

Figura 29 Estados de animación de la daga

En la figura 29 se muestran los estados de la daga y como se relacionan.

2.6.5 Sistema de salud y enemigos

Para el sistema de salud definimos una clase que se puede asignar tanto al jugador como a los enemigos, esta clase detecta si está asignada a un jugador o a un enemigo y recibe daño de las fuentes que le correspondan, es decir, si te disparas a ti mismo, o disparas con una pistola dentro de tu *Collider* no te haces daño, pero si disparas a un enemigo sí que le haces daño.

Para el comportamiento de los enemigos creamos un sistema basado en detección radial, es decir, el enemigo tiene un radio de detección, si el jugador entra en él, el enemigo lo detecta y se dirige hacia a su posición. Como queríamos que el sistema fuese simple, porque no había un modelo para el enemigo, lo único que hicimos fue que éstos detectasen al jugador y al entrar en el rango de ataque, iniciasen la co-rutina de atacar, disparándole.

Si el jugador les golpeaba o los enemigos le golpeaban a él, aparecía un salpicón de sangre donde la bala o puñalada había impactado, este salpicón se hizo con el sistema de partículas de Unity. Si la vida de los enemigos bajaba a 0, éstos desaparecían, en el caso del jugador se recargaba la escena. Cabe destacar que al salir del rango de detección de los enemigos, éstos se quedaban quietos, y además, si el jugador les disparaba no reaccionaban.

2.6.6 Menú radial

Seguidamente se implementó un cambio de armas más complejo. Al pulsar el botón de moverse o el botón de mover la cámara, aparece un menú radial que contiene todas las armas equipadas, al seleccionar un arma, la imagen se empequeñece para dar confirmación visual y la sensación de estar seleccionando esa pieza de equipamiento. Al volver a pulsar este botón se equipa el arma elegida.

Este sería el diseño del menú y sus características.

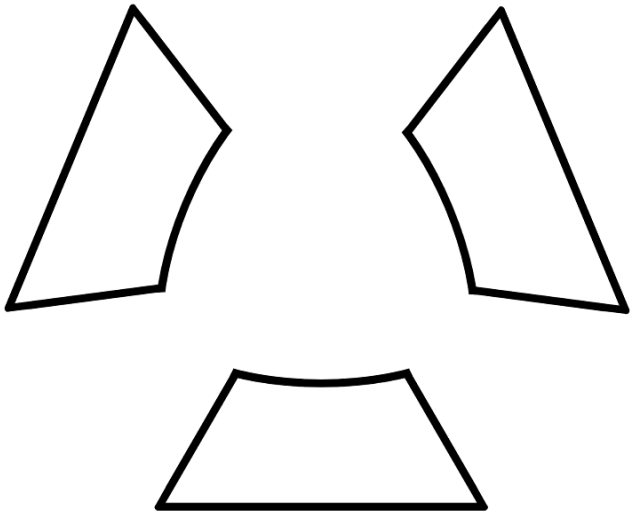

Figura 30 Diseño del menú radial

Las características del menú mostrado en la Figura 30 son:

- Se puede seleccionar dos armas que están situadas a la izquierda y a la derecha.
- La opción de bajo siempre será una mano libre.
- Si el jugador solo lleva un arma los demás huecos serán manos libres.
- En los huecos de la izquierda y de la derecha se equipan las armas que elija el jugador.
- El jugador puede llevar 4 armas 2 en la mano izquierda y 2 en la mano derecha.

2.6.7 Mazmorra, primer nivel

Al finalizar estas tareas el jugador era capaz de atacar a sus enemigos y cambiar de arma, era el momento de crear un nivel donde poder enfrentarse a peligros reales. Para la creación del primer nivel empezamos buscando *Assets* de villas desérticas o pueblos del lejano oeste, todos los que encontramos no nos convencían o eran de pago, por lo que optamos por modelar una mazmorra. Las mazmorras de la primera era son un punto clave en la historia de Mistborn y no nos pareció que desentonase con el tono. La idea era crear una guarida para criminales y que el jugador irrumpiese y limpiase el sitio de malhechores.

Para generar el mapa optamos por un diseño semi-procedural, y para ello se diseñaron 8 salas distintas, que son:

- Sala de inicio.
- Pasillo.
- Sala de alquimia.
- Sala de entrenamiento.
- Sala común.
- Cocina.
- Pasillo derruido.
- Habitación del líder.

La generación del nivel funciona de la siguiente manera, primero se asigna en una lista las salas a crear, entre sala y sala siempre hay un pasillo, la primera sala siempre será la de inicio y nos llevara a una sala de alquimia, entrenamiento o común, estas 3 salas tienen pasillos a izquierda y derecha y éstos nos pueden llevar a una cocina o un pasillo derruido, si nos llevan al pasillo

hay una probabilidad del 50% de que haya una habitación extra, como si fuera la habitación del jefe, conectada al pasillo.

Una vez generado el mapa en la lista, se instancian las salas y el *manager* del nivel genera enemigos en ciertas áreas marcadas como puntos de aparición.

En el proceso de creación del nivel se encontró un problema, el componente de manejo de movimiento de la *IA NavMesh* no creaba terreno para que se pudiesen mover los enemigos, este error se solucionó con una llamada desde código al componente *NavMesh Surface*, esta llamada se realizaba al crearse todas las salas.

2.6.8 Enemigos actualizados

Para la creación de los enemigos empleamos un modelo de bandido y para animarlo se utilizó Mixamo (Adobe, 2008), que es una base de datos de animaciones en la que puedes importar tus propios modelos para animarlos. Descargamos un set de animaciones de un pistolero y las importamos en Unity. Aquí ajustamos los huesos del modelo y modificamos las animaciones para poder ajustarlas, Mixamo no te permite editar sus animaciones, pero si las duplicas puedes editarlas.

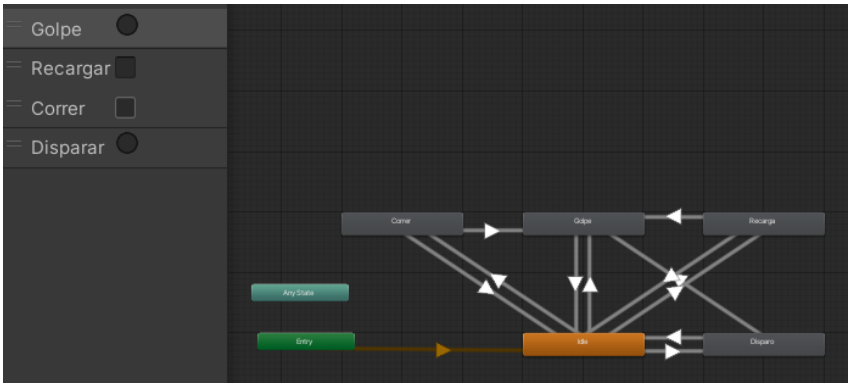

Figura 31 Máquina de estados del enemigo revólver

En la figura 31 podemos ver los *Trigger* de los estados y cómo se relacionan entre sí para el enemigo revólver.

Dentro del código establecimos los disparadores de las animaciones. Las capacidades de los enemigos son las siguientes:

- Detección radial.
- Disparan sí tienen a un jugador a tiro y además se encuentra en su rango de ataque.
- Deben recargar cuando realizan 6 disparos.
- Si el jugador sale de su rango de detección, los enemigos van a su última posición conocida.
- Al golpearles, inician una animación de golpeo donde no se pueden mover ni disparar.
- Al dispararles, avisan a sus compañeros para que vayan a por el jugador que disparó.
- Cuando mueren se activa el modo muñeco de trapo.
- Cuando mueren aparece un objeto encima del cadáver, que puede ser munición u objetos de utilidad.
- Si el jugador no está a tiro, pero está dentro de su rango de ataque, esperan a que salga de cobertura o le intentan flanquear.

En cuanto al enemigo escopeta, realizamos los mismos pasos que para la creación del primero, buscamos una escopeta e implementamos animaciones de armas a dos manos. Su árbol de estado es idéntico al del revólver.

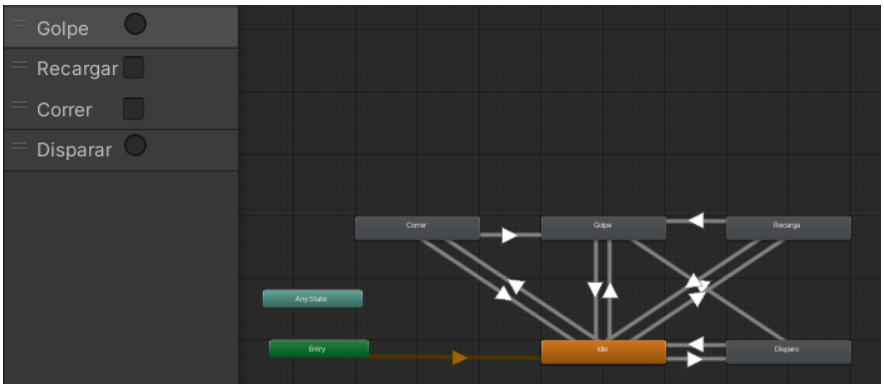

Figura 32 Máquina de estados del enemigo escopeta

En la figura 32 podemos ver los *Trigger* de los estados y como se relacionan entre sí. Los cambios que introduce este enemigo son:

- Apariencia distinta, cambio de color en la textura y modificación del modelo a mano.
- Escopeta, disparo en cono con varios perdigones.
- Deben recargar cuando realizan 4 disparos.
- Movimiento más rápido y agresivo hacia el jugador.
- Menor rango de ataque y detección.

Una vez creados estos dos tipos de enemigos hicimos que apareciesen de forma aleatoria en cada punto de *spawn* y para que no apareciesen siempre con la misma orientación, le asignamos a cada uno una rotación aleatoria.

2.6.9 Ciclo jugable primitivo

Desarrollados los enemigos y un mapa para enfrentarlos se implementó el ciclo jugable, es decir, que el jugador pudiese elegir el nivel que quisiera jugar, que se pudiese equipar y que al completar el nivel o morir en el intento volviese a la sala de preparación para volver a empezar.

Creamos la sala de preparación según el documento de diseño de juego y empezamos a desarrollar los sistemas de selección de equipamiento y misión. Aquí nos fueron muy útiles los *Sockets*, básicamente cada hueco de equipamiento tenía un *Socket* asociado y dejando un arma sobre ese *Socket* se asignaba a la ruleta de armas. Para el selector de misión hicimos lo mismo, al colocar un objeto que representa al jugador en un punto del mapa se seleccionaba un nivel.

Los problemas que surgieron en esta parte los solucionamos empleando los eventos de los *Sockets*, al sacar un objeto de un *Socket* algunas veces no detectaba que había salido, añadiendo eventos de *OnExit* lo pudimos solucionar.

También preparamos el sistema de estadísticas que cuenta cuántos enemigos y niveles ha derrotado y completado el jugador en la partida, al morir estos contadores se reinician. Además, creamos los iconos para cada arma y pieza de equipamiento que tenía el juego, así serían más reconocibles tanto en la ruleta de armas como en las estanterías de equipamiento. Finalmente teníamos un ciclo jugable primitivo.

Como añadido final se corrigió un error, derivado de usar el movimiento continuo como paradigma de desplazamiento, el personaje podía atravesar paredes y ver a través de ellas. Para solucionar el problema de atravesar paredes se añadió un *CharacterController* al personaje y un script llamado *Character Controller Driver*.

Para el problema de ver a través de ellas, se añadió un plano de color negro delante de la cámara, que detectaba si el jugador estaba en contacto con algún objeto del entorno como paredes o muebles. Si la cabeza del jugador colisionaba con alguno de estos elementos, el plano se activaba y oscurecía la pantalla.

Con estos dos simples cambios los jugadores no podían atravesar paredes moviendo al personaje o moviéndose ellos por el mundo real.

2.6.10 Escopeta

Antes de implementar todo el sistema de Alomancia y Feruquimia decidimos añadir la escopeta de mano para el jugador, el modelo en el que nos basamos para crearla fue una Diablo 12 Gauge Pistol, se puede observar una imagen de referencia en la figura 33.

Figura 33 Imagen de referencia de la escopeta

Su árbol de animaciones, Figura 34, es un poco distinto a los revólveres, en este caso creamos una animación idle que forzaba a la escopeta a estar cerrada.

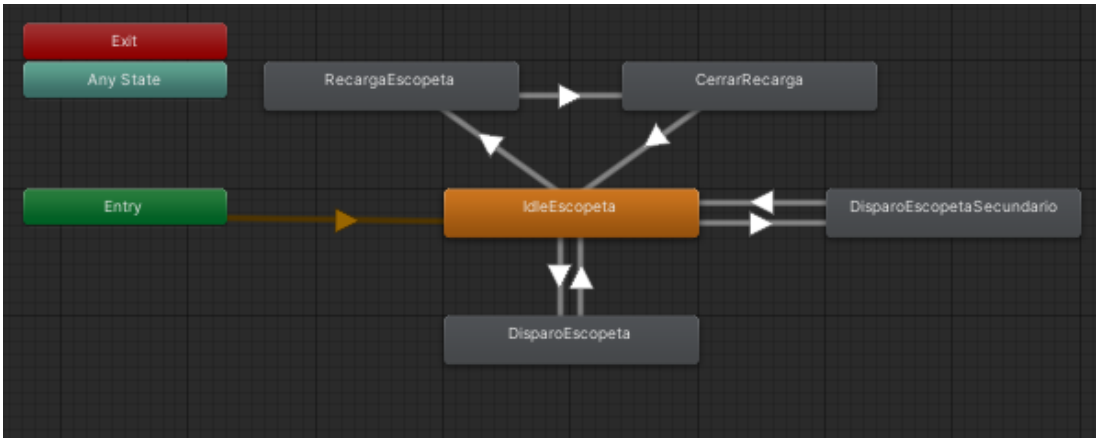

Figura 34 Máquina de estados de la escopeta

En cuanto a la programación de la escopeta, funciona igual que los revólveres y sus únicas diferencias son:

- Disparo en cono con perdigones.
- Doble disparo con Grip.
- Capacidad de dos balas.
- Alta dispersión, funcional en distancias cortas.
- La munición de la escopeta se encuentra en los enemigos con escopeta.

2.6.11 Alomancia, Feruquimia y bastón de duelo

Las alomancias son poderes activos que otorgan al jugador ventaja momentánea sobre las situaciones en el campo de batalla, las feruquimias dan ventajas pasivas que se mantienen durante todo el nivel y que ayudan enormemente a poder completarlo. El jugador puede equipar únicamente una Alomancia y una Feruquimia. Las capacidades Alománticas y Feruquímicas pueden consultarse en la Tabla 1.

Primero buscamos los modelos y recogimos los iconos de cada metal, los modelos elegidos son clavos, esto es una referencia a la Hemalurgia de los libros, cada metal tiene una variación de un mismo material provocando que sean suficientemente distintos. Una vez creados los clavos y añadidos los iconos, empezamos a implementar la programación de la Feruquimia, dado que era más sencillo que todo el sistema alomántico.

Cuando el jugador coloca un clavo feruquímico, el *Socket* detecta qué tipo de clavo se ha añadido y según el tipo modifica unas variables guardadas en el *game manager*. Cuando ocurren eventos como matar a un enemigo o recibir un balazo, se consulta el *game manager* y se ve el estado de las variables.

En cuanto a la alomancia, funciona como un arma más. Cada poder está pensado para una situación específica y el jugador tiene unos usos limitados, si quiere seguir empleando el poder, deberá recargar su investidura. Para activar el poder que se haya seleccionado se emplea una llamada a una animación, que varía según el poder elegido y en esta animación hay un evento que dispara la función con el poder de la alomancia escogida.

Las alomancias más importantes a comentar son el hierro, el zinc y el estaño. Para el hierro, el cual permite atraer objetos a distancia, se realizaron cálculos de parábolas, ya que muchos juegos VR implementan un sistema parecido, incluso XR Interaction Toolkit lo trae implementado, aunque no sea realmente una parábola. Como inspiración nos fijamos en los ***Gravity Gloves*** *de Half Life Alyx.* El zinc supuso un rediseño del sistema de daños y de detección de los enemigos, la idea es que el jugador pueda marcar a un enemigo para que entre en furia, al hacerlo, los demás enemigos lo detectan como una amenaza y le atacan, este poder correctamente usado puede ayudar a distraer a los enemigos unos segundos. Para añadir el estaño se empleó la herramienta de oclusión de URP. Con esta funcionalidad el jugador puede ver a través de objetos a los enemigos.

Cuando planteamos cómo mostrar al jugador cuánta reserva alomántica le quedaba, tuvimos una idea que nos ayudó también a mostrar la vida actual del personaje. Tanto la vida como la alomancia siguen el mismo sistema, el personaje tiene dos brazaletes, uno en cada brazo, estos brazaletes varían su color e intensidad para mostrar el nivel de salud o alomancia. En cuanto a salud, al tenerla al máximo el cristal será verde y al ir reduciéndose cambia a rojo, en cuanto a la alomancia la perla será de un azul brillante cambia a una perla negra.

Una vez acabado todo el sistema alomántico, nos decidimos a realizar una tarea un poco más sencilla, añadir una nueva arma cuerpo a cuerpo, el bastón de duelo. Esta arma es muy usada

por uno de los personajes principales de los libros así que debíamos añadirla. El bastón de duelo se comporta como la daga y simplemente es un arma que tiene más rango.

2.6.12 Ciclo jugable completo, desbloqueos permanentes

Primero empezamos por registrar las muertes de cada enemigo y nivel completado de forma individual y total, es decir, si el jugador había completado 2 veces la mazmorra y había matado 10 revólveres y 5 escopeteros, internamente se guardarían los datos de niveles totales, mazmorras completadas, enemigos totales, revólveres derrotados y escopeteros derrotados. Al completar el nivel, se revisaban los datos recogidos y se desbloqueaba equipamiento acorde a los requisitos de desbloqueo de cada pieza.

Por ejemplo, si el jugador volvía de la mazmorra y había derrotado a 5 escopeteros, desbloqueaba la escopeta de mano, esta pieza de equipamiento la podía usar hasta que le matasen y reiniciasen su progreso. Además, dentro del juego se comunica al jugador a través de un cartel los requisitos para desbloquear otras piezas de equipamiento, ya sea cuántos niveles faltan por completar o qué enemigos ha de derrotar. También cada vez que el jugador completa un nivel se añade atium, que es una "moneda" que se mantiene entre partidas y sirve para desbloquear equipamiento de forma permanente.

Cuando el jugador muere es transportado a un plano etéreo donde puede ver la infinidad del cosmos, el planeta girando bajo sus pies. También puede ver el equipamiento que ha desbloqueado durante la partida. Si el jugador tiene suficiente atium, podrá sacrificarlo para desbloquear permanentemente equipamiento.

2.6.13 Aldea, enemigo con rifle y tutoriales

Acabado el ciclo de juego completo y añadido todo el arsenal, había que dar más variedad a las misiones, así que nos propusimos hacer la aldea terrisana. En este nivel el jugador se enfrenta a hordas de enemigos mientras defiende una aldea, también hay un efecto día y noche entre oleadas por lo que cuanto más avance el jugador más de noche se hará, dando la sensación de que ha estado peleando todo el día. Para este nivel se añadió al último tipo de enemigo del juego, que es un enemigo con rifle, con mucho daño y rango de ataque, pero con baja vida y velocidad de ataque lenta. La aldea es uno de los niveles más difíciles y da 3 puntos de atium cada vez que se completa.

Acabado el nivel adicional empezamos a trabajar en la información, es decir, descripciones de las misiones a las que se envía al jugador o explicaciones del equipamiento. Para ello colocamos un altar como el de la escena de muerte, si el jugador deja un arma en ese altar el cartel de los desbloqueos se llena con los controles del arma e información adicional. En cuanto a los niveles, delante de la puerta de salida hay un cartel que muestra información del nivel seleccionado.

2.6.14 Sonidos, objetos arrojadizos y sistema de guardado

A continuación, se añadieron los sonidos que faltaban al juego, es decir, sonidos a las armas y niveles, música de fondo, etc. Básicamente casi todo tenía un sonido, también se incluyó el sonido 3D, que permite al jugador escuchar con más precisión desde donde vienen los sonidos. También se incluyeron los sonidos de pasos, indispensables para la aldea, ya que permiten detectar por donde se aproximan los enemigos.

Llegados a este punto añadimos algunos elementos arrojadizos, como dinamita o cuchillos, que funcionan realmente bien con el hierro y el acero alománticos, permitiendo tirar y empujar objetos. Estos elementos se encuentran en las misiones, desperdigados por el mapa, los jugadores más avispados podrán recogerlos y usarlos para ganar ventaja.

Con el juego prácticamente acabado se añadió el sistema de guardado. Se empleó un sistema de serialización binaria, lo que nos permite asegurar que los datos no se pueden modificar fuera del proyecto. Otros tipos de guardado como los *PlayerPrefs* o la serialización en json son vulnerables a modificaciones externas.

Se implementó de la siguiente manera, en primer lugar, creamos una clase donde seleccionamos los elementos a guardar, en nuestro caso todo elemento referente a estadísticas. Una vez seleccionados estos datos, se transforman a binario y se almacenan en un fichero que se guarda en una ruta persistente de forma local. Para cargar los datos se recoge el archivo guardado localmente y los datos que contiene son convertidos de vuelta a sus valores en Unity. Si se encontrase este archivo y se tratase de modificar los datos guardados quedarían corruptos. Por último, el sistema guarda datos cada vez que el jugador llega a la escena de preparación, y sea porque completó un nivel o porque murió.

2.7 Pruebas

Todas las implementaciones anteriormente mencionadas se han sometido a una fase de pruebas. Para que las implementaciones se diesen como correctas o válidas, primero debían cumplir los requisitos de verificación de la tarea creada en Trello.

Una vez todos los requisitos estaban satisfechos, se seguía probando durante un tiempo para ver si había casos que no se habían tenido en cuenta y se producían errores inesperados. Si en este tiempo surgía algún error, la prioridad era solucionarlo e intentar ver si ese error podía desencadenar otros errores, solucionando así errores futuros. Por otro lado, si después de cierto tiempo no surgían más errores, la tarea se daba por finalizada.

Una vez terminada cada fase de desarrollo, se realizaba una *build* para Windows y Oculus *standalone*. Estas pruebas de compatibilidad nos servían para garantizar el funcionamiento adecuado en las dos plataformas.

También se realizaron pruebas con *testers* para que valorasen los aspectos de jugabilidad, como la dificultad, la progresión, la variedad de escenarios, etc. Se escogió a un grupo de 10 personas de distintos rangos de edad y experiencia en el ámbito de los videojuegos. Todo para conseguir la mejor experiencia de usuario alcanzable. Además, los *testers* rellenaron un test de usabilidad (Brooke, 1995) donde podían expresar su opinión del juego de forma anónima. En este test se pide a los usuarios que respondan a 10 preguntas con rangos del 0 al 4, una vez acabado se suman todas las puntuaciones obtenidas y se multiplica el resultado de la suma por 2.5, el resultado es una escala del 0 al 100 que determina el desempeño general de la experiencia.

Por lo que se puede ver en los resultados la puntuación final media esta entrono al 79 sobre 100, así que, si bien el juego tiene espacio para pulirse y mejorarse, hemos creado una buena versión inicial.

El test y los resultados pueden verse en Anexos Prueba de usabilidad SUS (System Usability Scale). El pdf de los resultados contiene las puntuaciones de cada pregunta y las puntuaciones finales del 0 al 100 de cada usuario.

2.8 Resultados

En esta sección mostraremos los elementos finales del juego desde Unity, que son, las armas, las alomancias, los enemigos, los niveles y algunos de los elementos de UI.

Armas

Como se ha comentado en la sección de implementación, hay tres armas a distancia y dos cuerpo a cuerpo, siendo éstas, el revólver, el revólver de cañón largo, la escopeta, la daga y el bastón de duelo. Los modelos empleados se pueden ver en la figura 35.

Figura 35 Modelos de las armas vistas desde Unity

Alomancia y Feruquimia

Los clavos de la Alomancia y Feruquimia se pueden ver en la siguiente Figura 36. Los metales de los que están compuestos los clavos de izquierda a derecha son: Peltre, hierro, latón, oro, zinc, acero y estaño.

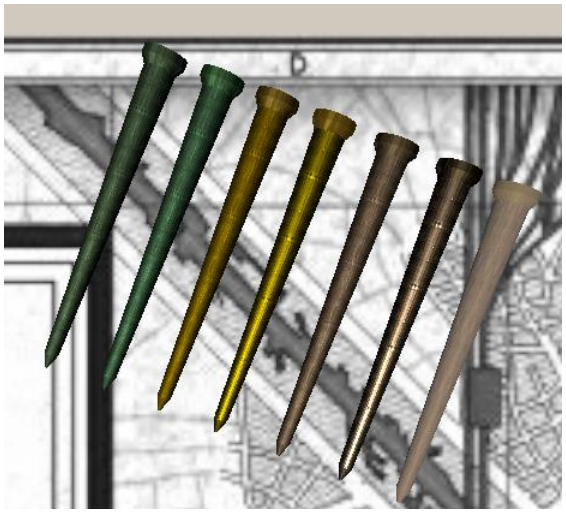

Figura 36 Clavos para la Alomancia y la Feruquimia

Enemigos

Los tres enemigos implementados en el juego se pueden ver en la Figura 37. El de la izquierda es el escopetero, en medio tenemos al que lleva revólver y a la derecha el tirador con rifle. También se incluye una imagen de los enemigos disparando al jugador desde distintas distancias.

Figura 37 Enemigos en Unity

Niveles

Los distintos escenarios por los que se moverá el jugador son la sala de preparación (Figura 38), la mazmorra de leyenda (Figura 39), la aldea (Figura 40) y la muerte (Figura 41).

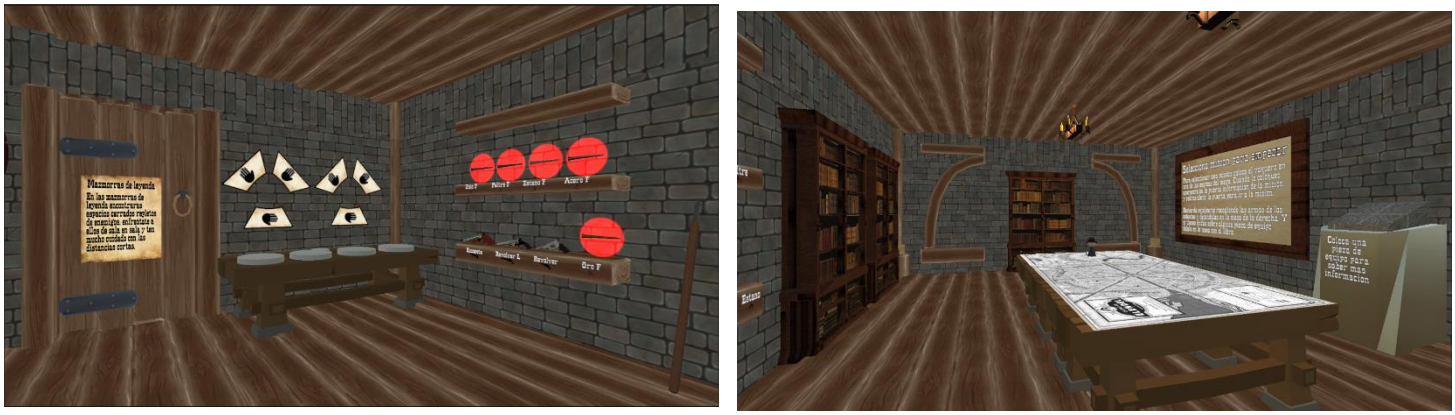

Figura 38 Sala de preparación en Unity

Figura 39 Mazmorra en Unity

Figura 40 Aldea en Unity

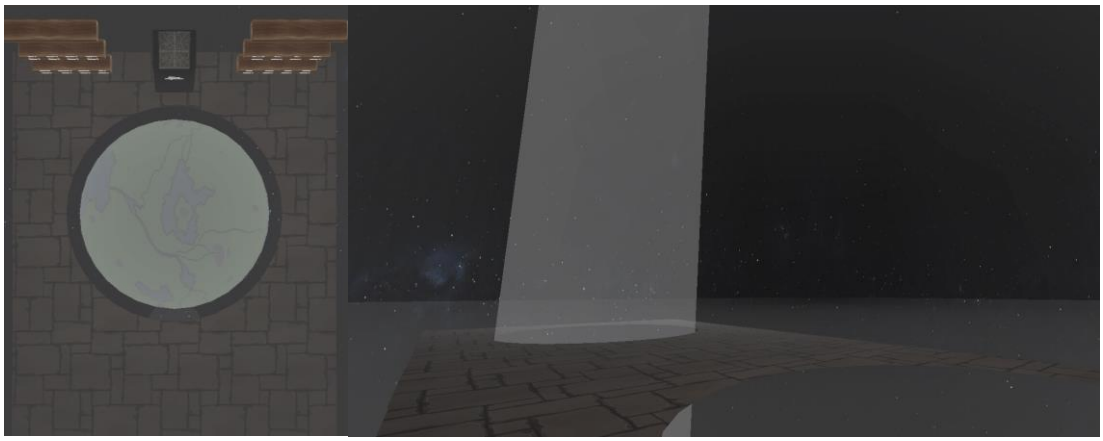

Figura 41 Zona de muerte en Unity

Elementos de UI

Los elementos de interfaz de usuario como los indicadores de vida y alomancia (Figura 42 y 43), los diferentes textos (Figura 44) y las partículas (Figura 45).

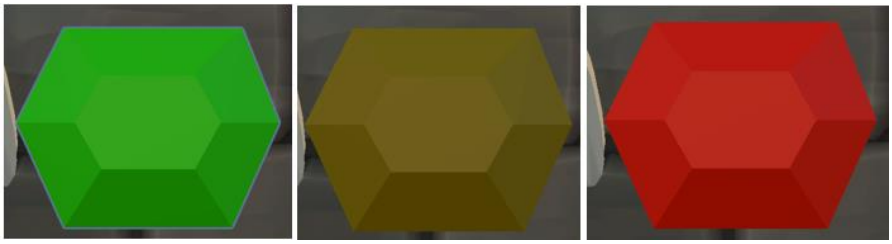

Figura 38 Representación de la salud Unity

Figura 39 Representación de la investidura Unity

Figura 40 Ejemplos de UI en Unity

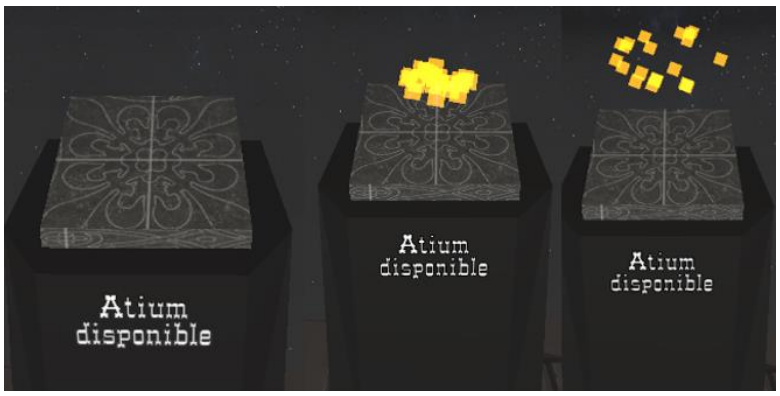

Figura 41 Partículas en Unity

Las utilidades

Todos los elementos extra que se deben encontrar durante la partida, explorando los mapas o recogiéndolos de los enemigos. Estos elementos son cargadores (Figura 46), salud o investidura y armas arrojadizas (Figura 47).

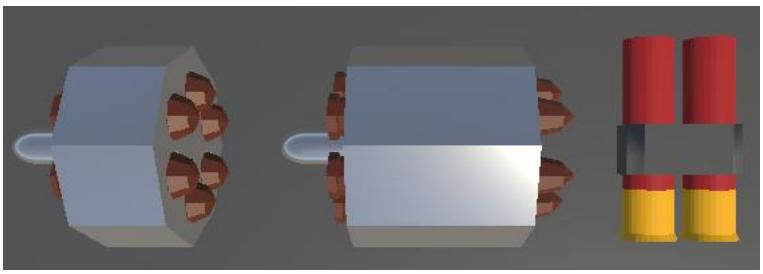

Figura 42 Cargadores en Unity

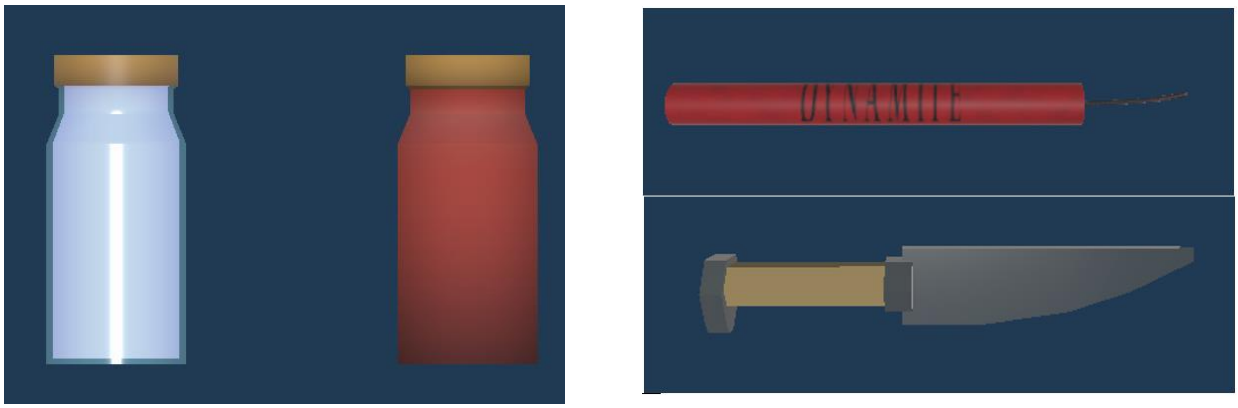

Figura 43 Elementos arrojadizos y de recuperación en Unity

3 Conclusiones y trabajos futuros

Este proyecto ha cumplido con los objetivos propuestos, logrando el diseño y producción de "Mistborn: Ghostbloods Contracts", un videojuego de géneros *Rogue-Lite* y *Shooter* en VR.

En primer lugar, se realizó un análisis exhaustivo del sector de la VR, pudiendo así seleccionar el hardware y las herramientas software más adecuadas para el desarrollo del videojuego, es decir Meta Quest 2 y XR Interaction Toolkit.

Además, se llevó a cabo un análisis detallado de los géneros *Rogue-Lite* y *Shooter*, el cual permitió identificar elementos clave y tendencias dentro de estos géneros, como los sistemas de cambio de armas, el sistema de recargas, la progresión en cuanto a desbloqueos, la dificultad de enfrentamientos, etc.

También se logró trasladar con éxito el concepto de la saga de libros Mistborn a un entorno VR, siendo éste el primer videojuego de la saga, que acaba su fase de producción.

En términos de diseño, se trabajó en la creación de los sistemas principales de juego, enfocándose en la implementación de un ciclo jugable atractivo y desafiante.

Por otra parte, este trabajo ha supuesto un gran reto, todo aquel que haya desarrollado un juego en solitario sabe lo difícil que es encargarse de todos los aspectos que conlleva, más aún si se utiliza alguna herramienta con la que no se ha trabajado antes. Gracias a nuestra experiencia previa con desarrollo de videojuegos, las cuestiones referentes al diseño de éste, no han supuesto un reto, pero trasladarlas a el sistema VR y con una estética tan definida sí ha resultado complejo.

Además, al trabajar siguiendo las guías de un mundo preestablecido, hemos tenido la sensación de estar trabajando en un proyecto de mayor escala, los problemas no eran tanto buscar ideas para saber qué sistemas implementar, sino traducir los sistemas que existían en los libros a la experiencia VR. Como en toda adaptación este trabajo no es fiel al 100% pero sí que traduce la idea y las sensaciones que transmite la saga.

Sobre este juego se podrían realizar diversas mejoras que se escapan del alcance de este proyecto. Las mejoras propuestas son:

- Crear *props* propios que encajen mejor con la estética.
- Añadir mejoras de poderes alománticos.
- Incluir mayor variedad de equipamiento.
- Crear nuevos niveles, con la posibilidad de que algunos de ellos sean desbloqueables
- Añadir diferentes tipos de misión, por ejemplo, recuperar objetos, rescatar ciudadanos, etc.
- Preparar el sistema para que pueda cargar contenido descargable (DLC, por sus siglas en inglés)

Por último, también se plantea la posibilidad de publicar el juego en una plataforma de distribución de videojuegos.

4 Referencias bibliográficas

Activision. (2020). Call of Duty: Warzone. https://store.steampowered.com/app/1962663/Call_of_Duty_Warzone/ [Consulta: 20 de Junio de 2023].

Adobe. (2008) Mixamo. https://www.mixamo.com/ [Consultado: 5 Marzo 2023]

AlehandoroVR. (2022, 18 de octubre) QUEST 2 vs PICO 4 ¿CUÁL me COMPRO? ¿Qué MERECE LA PENA? Mis conclusiones. https://youtu.be/FNHaQcsmH6E [Video File]. Youtube.

Axosoft. (2000). GitKraken. https://www.gitkraken.com. [Consultado: 5 Marzo 2023]

Barnett, J. (2022, 5 de Enero) How to Make VR Games in 2022 - Updated Unity VR Tutorial. https://youtu.be/yxMzAw2Sg5w[Video File]. Youtube.

Barnett, J. (2022, 15 de Diciembre) How to use VR Sockets (and XR Interaction Toolkit) in Unity 2021.2. https://youtu.be/rRNvq09Itdw [Video File]. Youtube.

Battlestate Games. (2017). Escape From Tarkov. https://www.escapefromtarkov.com [Consulta: 20 de Junio de 2023].

Bethesda Softworks. (2016). Doom. https://store.steampowered.com/app/379720/DOOM/ [Consulta: 20 de Junio de 2023].

Bethesda Softworks. (2014). Wolfenstein: The New Order. https://store.steampowered.com/app/201810/Wolfenstein_The_New_Order/ [Consulta: 20 de Junio de 2023].

BigBox VR. (2020). Population One. https://store.steampowered.com/app/691260/POPULATION_ONE/ [Consulta: 20 de Junio de 2023].

Blizzard Entertainment. (2022). Overwatch 2. https://overwatch.blizzard.com/es-es/ [Consulta: 20 de Junio de 2023].

Brown, R. (2023). HP Reverb G2 vs Oculus Quest 2 vs Pico 4 vs HTC Vive Pro 2 (Comparison). VR Compare. Available from: https://vr-compare.com/compare?h1=HdfjN_0UlPY&h2=pDTZ02PkT&h3=1qnvNfJKq&h4=mLbW9G7f4 [Consultado: 10 Marzo 2023]

Brooke, John. (1995). SUS: A quick and dirty usability scale. Usability Eval. Ind.. 189.

Caveman Studio. (2018). Contractors. https://store.steampowered.com/app/963930/Contractors/ [Consulta: 20 de Junio de 2023].

Cellar Door Games. (2013). Rogue Legacy. https://store.steampowered.com/app/241600/Rogue_Legacy/ [Consulta: 20 de Junio de 2023].

Chen, W., & Lu, Y. (2021). Virtual Reality in Education: A Comprehensive Review. IEEE Transactions on Education, 64(1), 19-34.

Cloudhead Games. (2019). Pistol Whip. https://store.steampowered.com/app/1079800/Pistol_Whip/ [Consulta: 20 de Junio de 2023].

Combat Waffle Studios. (2023). Ghost of Tabor. https://store.steampowered.com/app/1957780/Ghosts_of_Tabor/ [Consulta: 20 de Junio de 2023].

Corcoran, L. (2013). itch.io platform. https://itch.io/. [Consultado: 5 Marzo 2023]

Dice. (2021). Battlefield 2042. https://store.steampowered.com/app/1517290/Battlefield_2042/ [Consulta: 20 de Junio de 2023].

EA. (2016). Titanfall 2. https://store.steampowered.com/app/1237970/Titanfall_2/ [Consulta: 20 de Junio de 2023].

Epic Games. (2017). Fortnite. https://store.epicgames.com/es-ES/p/fortnite [Consulta: 20 de Junio de 2023].

Epyx. (1980). Rogue. https://store.steampowered.com/app/1443430/Rogue/ [Consulta: 20 de Junio de 2023].

Eurogamer. (March 21, 2020). Number of users of Fortnite worldwide in the first year of release in 2017-18 (in millions) [Graph]. In Statista. Retrieved July 07, 2023, from https://www.statista.com/statistics/1110008/fortnite-players-first-year/

Game World Observer. (June 9, 2022). Lifetime unit sales generated by Call of Duty series worldwide as of June 2022 (in millions) [Graph]. In Statista. Retrieved June 15, 2023, from https://www.statista.com/statistics/1244224/cod-lifetime-unit-sales/

Holzwarth, V., Gisler, J., Hirt, C., & Kunz, A. (2021). Comparing the accuracy and precision of steamvr tracking 2.0 and oculus quest 2 in a room scale setup. In 2021 the 5th International conference on virtual and augmented reality simulations (pp. 42-46).

Hou, Q., Cheng, D., Li, Y., Zhang, T., Li, D., Huang, Y., ... & Wang, Y. (2022). Stray light analysis and suppression method of a pancake virtual reality head-mounted display. Optics Express, 30(25), 44918-44932.

HTC. (2016) Viveport. https://www.viveport.com [Consultado: 5 Marzo 2023]

Indierama. (2022, 11 de Enero) Sistema de GUARDADO SIMPLE en Unity. https://youtu.be/V9bfdlyEHx4 [Video File]. Youtube.

Jiménez, L. (2018). Historia del roguelike: Nacimiento de lo aleatorio. GamingGuardian. Available from: https://gaminguardian.com/articulo-historia-del-roguelike-nacimiento-de-lo-aleatorio [Consultado: 15 Julio 2023]

Johnson, H. A. (2017). Trello. Journal of the Medical Library Association: JMLA, 105(2), 209.

Joy Way. (2023). Dead Hook. https://store.steampowered.com/app/2342360/Dead_Hook/ [Consulta: 20 Junio 2023].

Khronos Group. (2019). OpenXR. https://www.khronos.org/openxr/ [Consultado: 5 Marzo 2023]

Laid, J. (2008, 16 de Septiembre). Interpretación de Berlín. Available from: https://groups.google.com/g/rec.games.roguelike.development/c/Orq2_7HhMjI?pli=1

Matt (2023, 21 de Marzo) Show LOOT & ENEMIES Behind Walls (Unity URP). https://www.youtube.com/watch?v=NXXc9xXzqnk [Video File]. Youtube.

Meta. (2013). Oculus SDK. https://developer.oculus.com/resources/ [Consultado: 5 Marzo 2023]

Mistborn fan community, (2021). Mistborn: Ashes Project. https://ashesproject.carrd.co/ [Consultado: 15 Noviembre 2022]

Nintendo. (2017). Splatoon 2. https://www.nintendo.es/Juegos/Juegos-de-Nintendo-Switch/Splatoon-2-1173295.html [Consulta: 20 de Junio de 2023].

notdead LLC. (2022). Compound. https://store.steampowered.com/app/615120/COMPOUND/ [Consulta: 20 de Junio de 2023].

PCGamesN. (March 21, 2020). Peak concurrent players of DOOM and DOOM Eternal during first week of release on Steam worldwide as of March 2020 (in 1,000s) [Graph]. In Statista. Retrieved July 07, 2023, from https://www.statista.com/statistics/1109975/doom-eternal-players-steam/

Riot Games Inc. (2020). Valorant. https://www.leagueoflegends.com/es-es/ [Consulta: 20 de Junio de 2023].

Salmond, M. (2016). Video game design: Principles and practices from the ground up. Bloomsbury Publishing.

Sanderson, B. (2022). The Final Empire, The Well of Ascension, The Hero of Ages, Mistborn: Secret History, The Alloy of Law, Shadows of Self, The Bands of Mourning, The Lost Metal. Tor Books.

Sanderson, B (2008). El imperio final. Nova.

Schell Games. (2020). Until You Fall. https://store.steampowered.com/app/858260/Until_You_Fall/ [Consulta: 20 de Junio de 2023].

Stecuła, K. (2022). Virtual Reality Applications Market Analysis—On the Example of Steam Digital Platform. In Informatics (Vol. 9, No. 4, p. 100).

Stringer, D. (2023, 30 de Agosto) STOP PEEKING THROUGH WALLS IN VR - Unity XR tutorial. https://youtu.be/WyqjGW7euVc [Video File]. Youtube.

Thullen, E. (2020). Ancient Dungeon. https://store.steampowered.com/app/1125240/Ancient_Dungeon/ [Consulta: 20 de Junio de 2023].

Unity Technologies. (2010). Unity Asset Store - The Best Assets for Game Making. https://assetstore.unity.com/. [Consultado: 5 Marzo 2023]

Unity Technologies. (2005). Unity. https://unity.com/es. [Consultado: 5 Marzo 2023]

Unity Technologies. (2021). XR Interaction Toolkit – Unity – Manual. https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.3/manual/index.html [Consultado: 5 Marzo 2023]

Valve Corporation. (2003). Steam. https://store.steampowered.com/?l=spanish. [Consultado: 1 Febrero 2023]

Valve Corporation. (2016). SteamVR. https://store.steampowered.com/app/250820/SteamVR/?l=spanish. [Consultado: 1 Febrero 2023]

Valve Corporation. (2020). Half Life Alyx. https://store.steampowered.com/app/546560/HalfLife_Alyx/ [Consulta: 20 de Junio de 2023].

Valve Corporation. (2012). Counter-Strike: Global Offensive. https://store.steampowered.com/app/730/CounterStrike_Global_Offensive/ [Consulta: 20 de Junio de 2023].

Virtual Insider. (2021, 29 de Abril) A Guide To Guns In Virtual Reality. https://youtu.be/3aNxXKBsl7k [Video File]. Youtube.

We Are Social, & DataReportal, & Meltwater. (January 26, 2023). Most popular video game genres among internet users worldwide as of 3rd quarter 2022, by age group [Graph]. In Statista. Retrieved July 07, 2023, from https://www.statista.com/statistics/1263585/top-video-game-genres-worldwide-by-age/

5 Glosario de términos

A

Asset
Activo digital, como imágenes, videos, sonidos o archivos, utilizados en proyectos multimedia., 4

B

battleroyale
El Battle Royale es un género de videojuegos donde un gran número de jugadores luchan entre sí hasta que solo queda uno en pie., 11, 12
bugs
Los bugs son errores o fallos en un software que causan comportamientos inesperados o incorrectos., 9

E

estaciones base
Las estaciones base son componentes físicos que se colocan en la habitación donde se pretende emplear el dispositivo VR. Estos aparatos permiten el seguimiento de los mandos y el casco funcionando como puntos de referencia., 16
extraction shooters
Los extraction shooters son videojuegos en los que los jugadores deben rescatar o extraer a personajes o elementos importantes mientras se enfrentan a desafíos y enemigos., 11

F

FOV
El FOV (Field of View) es el campo de visión que abarca el área visible desde un punto de vista específico, como en un videojuego o dispositivo de realidad virtual., 16

G

gampeplay
Gameplay se refiere a la experiencia interactiva de jugar un videojuego, incluyendo mecánicas, controles y la forma en que el jugador interactúa con el juego., 5
GDD
GDD (Game Design Document) es un documento que describe los aspectos fundamentales de diseño y desarrollo de un videojuego., 5, 6, 8, 40
GitKraken
GitKraken es una interfaz gráfica de usuario (GUI) para Git, que facilita la gestión y visualización de repositorios de código., 7

I

itchio
itch.io es una plataforma de distribución y venta de videojuegos independientes y proyectos creativos., 4

P

props
Elementos utilizados en producciones teatrales o audiovisuales para ayudar a la narrativa o ambientación de una escena., 4

R

rogue-lite
Rogue-lite es un género de videojuegos que presenta elementos de roguelike, como la generación procedimental y la muerte permanente, pero con ciertos elementos persistentes o de progresión entre las partidas., 5, ***10***, ***11***, ***21***

S

SDK
SDK (Software Development Kit) es un conjunto de herramientas y recursos de desarrollo que facilitan la creación de aplicaciones y software para una plataforma o sistema específico., 19, 20
shooter
Shooter es un género de videojuegos que se centra en la acción de disparar y eliminar enemigos utilizando armas de fuego., 5, 8, 10, 11, 12, 13
spawn
En juegos para ordenador, el término spawn hace referencia a la creación de una entidad dentro del juego, como un personaje jugador, un personaje no jugador u otro ítem. Los puntos de spawn, son los lugares donde se aparece el elemento generado., 38

T

Trello
Trello es una herramienta de gestión de proyectos en línea que utiliza tableros y

U

V

6 Anexos

6.1 Ejecutable

Build para PC:
https://drive.google.com/drive/folders/1pXvMb72VvpnJMB8VGT8uj_zPKwWIC_8u?usp=sharing

Build para Meta Quest 2:
https://drive.google.com/drive/folders/1od5tH__NyfX2wboRQmgqOOJsHqJ7oAzb?usp=sharing

6.2 Video I Prueba de concepto

Video: https://youtu.be/gF6sb9es8TI

6.3 Video II Prototipo

Video: https://youtu.be/G1yvE3KlncA

6.4 Video III Tráiler

Video: https://youtu.be/qcxUZz9Xbgk

6.5 Video IV Gameplay sin cortes

Video: https://youtu.be/DSaH7CWc_QM

6.6 Prueba de usabilidad SUS

URL al cuestionario: https://forms.gle/DUjeMTaRyVv8R4Jn9

Anexo adjuntado como pdf.

6.7 Repositorio Git

URL del repositorio Git: https://github.com/JorgeGGeasy/Mistborn

6.8 ODS

Anexo adjuntado como pdf.