



UNIVERSIDAD DE GRANADA

TRABAJO FIN DE MÁSTER
MÁSTER EN INGENIERÍA INFORMÁTICA

Desarrollo de un videojuego estilo Escape Room para Gafas de Realidad Virtual

Autor

Alejandro Guerrero Martínez

Director

Francisco Luis Gutiérrez Vela



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, 06 septiembre de 2018



Desarrollo de un videojuego estilo Escape Room para Gafas de Realidad Virtual

IN-GAME

Autor

Alejandro Guerrero Martínez

Director

Francisco Luis Gutiérrez Vela

Índice

1	Introducción	1
2	Objetivos	2
3	Estado del arte	2
3.1	Escape Rooms	2
3.1.1	Tipos de Escape Rooms	2
3.2	Consideraciones de Diseño de una Escape Room	4
3.2.1	Ambientación, historia y habitación	5
3.2.2	Diseño Puzles	5
3.2.3	Flujo del Juego	6
3.3	Realidad Virtual	7
3.3.1	SDKs/VR Frameworks	11
3.4	Juegos de Escape Room para Dispositivos VR	12
3.5	Conclusión	13
4	Análisis inicial del problema	13
5	Tecnología a usar	16
6	Metodologías a usar en el proyecto	17
7	Plan de entregas	17
7.1	Historias de Usuario	17
7.2	Cálculo temporal del proyecto	19
7.3	Descripción de las entregas	19
8	Desarrollo	20
8.1	Inicio del Proyecto en Unity, Añadir SDK WMR, Explorar la SDK	20
8.2	Modelos manos y Puerta, Añadirlos al Proyecto	24
8.3	Interacción visible con Outline	27
8.4	Interacción con Puerta	27
8.5	Modelos Interruptor y Llaves	29
8.6	Interacción con Interruptor	31
8.7	Posicionamiento de las manos en las Interacciones	32
8.8	Creación del Inventario y objetos Inventariables	34
8.9	Creación de Interactable/Interactor. Interacción de llaves y cerraduras	36
8.10	Interacción con Cajones	38
8.11	Diseño del Menú Inicial y del Primer Nivel	39

8.12	Desarrollo del Puzle del Salón (Caja Fuerte con combinación numérica)	42
8.13	Desarrollo del Puzle de la Librería (Tablet y Librerías)	44
8.14	Desarrollo del Puzle de la Habitación Secreta (Interruptores y Luces)	46
8.15	Sonidos de la escena y Música	47
8.16	Ayudas a los controles, Optimización y Corrección del movimiento	48
8.17	Imágenes con la historia mostradas en la escena	51
9	Conclusiones y Trabajo futuro	53
10	Bibliografía	54
11	Apéndices	55
11.1	Enlace a Código Fuente del Proyecto	55
11.2	In-Game - Game Design Document (GDD)	56

1 Introducción

Las Escape Room son una actividad de ocio de creciente popularidad, en la cual un jugador o un grupo de jugadores son encerrados en una “habitación” en la cual tendrán que recoger objetos, resolver enigmas y puzles para conseguir desenlazar la historia y escapar.

No solo gozan de popularidad esta actividad en los locales físicos dedicados a ello, también son conocidos en el mundo de los videojuegos, teniendo su propio género para clasificar los juegos de este tipo, llamados “Escape Game”.

Igualmente, varias compañías están apostando por Tecnologías de Realidad Virtual las cuales están ganando mucha popularidad, y gracias a ello ahora encontramos en el mercado varios dispositivos avanzados con precios cada vez más asequibles.

Por ello es lógico juntar esta tecnología que nos permite movernos libremente e interactuar con un mundo digital, con trasladar la experiencia de una Escape Room real a un videojuego.

El objetivo de este trabajo es diseñar y desarrollar un videojuego de estilo Escape Room, explorando y haciendo uso de las posibilidades que nos ofrece un sistema de Realidad Virtual y los nuevos dispositivos de interacción que incluyen, para alcanzar una experiencia más real y aumentar la inmersión del jugador.

Con el sistema de Gafas de Realidad Virtual el jugador podrá moverse libremente por la escena trasladando sus propios movimientos al videojuego, así como el movimiento de sus manos a través de unos mandos con los que podrá interactuar con los distintos elementos de la escena.

El resultado de este trabajo ha sido un videojuego en un principio con único nivel, totalmente jugable y con posibilidad de seguir desarrollándose para PC y gafas de realidad virtual compatibles con la plataforma Windows Mixed Reality, de estilo Escape-Game para un solo jugador, al que he llamado In-Game.

Con este proyecto he dado un paso más allá en mi aprendizaje en el paradigma que supone el desarrollo de videojuegos, integrando una tecnología emergente como es la Realidad Virtual y lidiando con los nuevos retos que esto supone.

2 Objetivos

El objetivo principal de este trabajo es el de desarrollar un Juego totalmente jugable de estilo Escape-Game simulando la experiencia de una Escape-Room real haciendo uso de algunos de los sistemas de Realidad Virtual disponibles, tanto como diseñar un juego específico para esta tecnología, aprender a integrar soluciones de realidad virtual en un videojuego y lidiar con los nuevos retos que supone esta nueva forma de interacción.

Como subobjetivos de estos principales me planteo:

- Aprender del diseño de un juego, específicamente del tipo Escape-Room, para resultar entretenido y atractivo para el jugador.
- Analizar e integrar una SDK para una tecnología de Realidad Virtual.
- Innovar en las interacciones con estos dispositivos.
- Conseguir suplir carencias de otros juegos para Realidad Virtual.
- Recrear una atmosfera inmersiva para el jugador.

3 Estado del arte

3.1 Escape Rooms

Las Escape Rooms son juegos de aventuras en grupo o individuales en los que encontraremos puzles y pruebas que deberemos superar para resolver las tareas propuestas y avanzar en la historia de ésta.

Aunque el nombre del juego sea “Escape Room” la misión de este no tiene por qué ser necesariamente la de escapar de un lugar en la que los jugadores se hallan encerrados, si no que los jugadores deben de progresar en las tareas propuestas por la historia creada para dicha Escape Room, como podría ser encontrar las pruebas de un asesinato, perpetuar un robo en una mansión, o escapar de una pirámide en la que nos hemos quedado atrapados.

3.1.1 Tipos de Escape Rooms

El término Escape Room, normalmente hace referencia a este tipo de juego en formato físico en el que a los jugadores se les plantea la historia y la ambientación del juego y posteriormente se encierran en una habitación para comenzar el juego. Típicamente estas Escape Rooms son siempre juegos en grupo con un mínimo de dos personas para

poder jugar, en la que ellos tendrán que coordinarse para registrar las habitaciones, compartir las pistas y resolver los acertijos en un tiempo máximo indicado.

Las Escape Rooms físicas tal y como las conocemos nacieron en Japón en el año 2007, gracias a la compañía SCARP y el creador de la idea Takao Kaoto. Pronto se hicieron populares en toda Asia, donde se fueron desarrollando y posteriormente se exportó la idea al resto del mundo [1].

Actualmente este tipo de actividad consta de una creciente popularidad en todo el mundo, siendo Estados Unidos el país con el mayor número y en Europa el continente con más Escape Rooms [2]. De hecho, son ya tan populares que existen campeonatos de Escape Rooms, en la que el grupo ganador consigue resolver la historia en el menor tiempo posible.



Ilustración 1: Jugadores en una Escape Room

Sin embargo, el origen de las Escape Rooms proviene de los videojuegos, también denominados por el mismo termino o como “Escape Game”, en el cual un solo jugador explora una serie de habitaciones recogiendo pistas y resolviendo puzles para avanzar. El primer videojuego que originó este género fue en 2001 con el juego “MOTAS” (Mystery Of Time And Space) desarrollado por Jan Albartus, aunque no se popularizó hasta el lanzamiento de “Crimson Room” en 2004 y creado por Toshimitsu Takagi.

Este tipo de videojuegos normalmente son en primera persona y de tipo “point-and-click”, bastante popularizado en los conocidos juegos flash, juegos desarrollados para *Flash Player* destinados para jugarlos en webs.

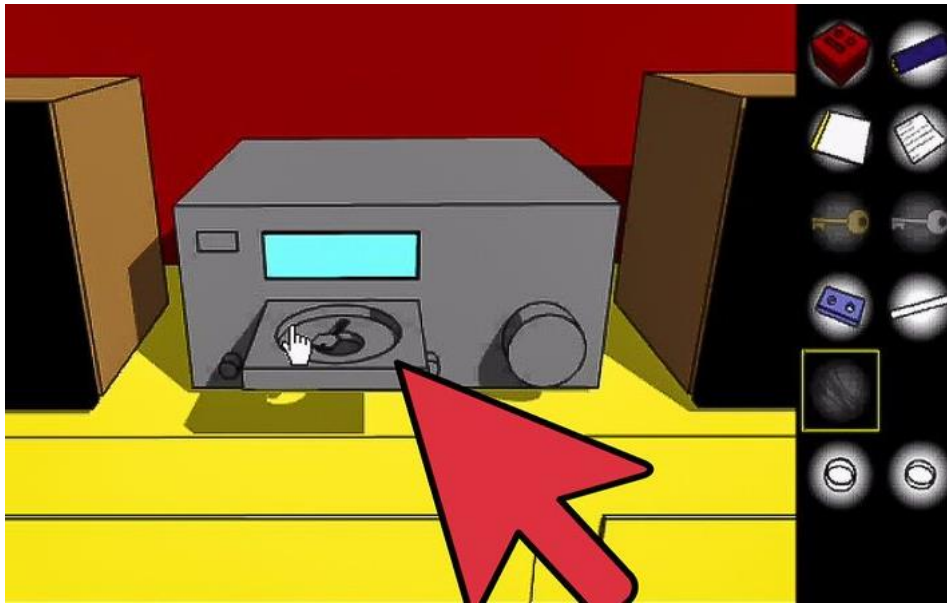


Ilustración 2: Captura del Juego Crimson Room

Sin embargo, los nuevos paradigmas de interacción que están llegando al mercado de los videojuegos nos permiten crear Escapes Rooms en las que nos encontraremos más inmersos y podremos interactuar con los elementos como haríamos en la realidad, con toda la flexibilidad que aporta los videojuegos a la hora de crear escenarios totalmente imposibles en la realidad.

Recientemente también ha llegado al mercado los conocidos como Escape Books, basados en la experiencia Escape Room, proponen un libro de tipo “librojuego” [3], en el cual a la hora de resolver los puzles o realizar acciones en determinados puntos, tomaremos decisiones, la cual nos mandará a una parte distinta del libro, que cambiarán el curso de la historia y nuestro avance en ella. El primer libro de este tipo en España, fue publicado en febrero de 2017, escrito por Iván Tapia Lasaosa.

3.2 Consideraciones de Diseño de una Escape Room

Las siguientes consideraciones son el fruto de mis conclusiones obtenidas del libro *Escape The Game* [4], cuyo autor se trata de un Diseñador de Niveles con experiencia en el diseño de Escape Rooms, así mismo como en webs especializadas en el diseño de experiencias Escape Rooms [5][6].

3.2.1 Ambientación, historia y habitación

A la hora de diseñar una Escape Room (Al igual que en la mayoría de juegos) debemos tener en cuenta una ambientación, la historia que nos lleva al juego y el objetivo que nos motiva a proseguir en el juego.

Los límites de la ambientación solo los impone nuestra imaginación: presente, pasado, futuro, mundos alternativos, etc. Pero debemos de asegurarnos que nuestros puzles, acertijos y elementos que encontremos en la escena concuerden con esta ambientación e historia que propongamos (tanto época como contexto) a no ser que los incluyamos conscientemente para ser parte de un acertijo, ya que los jugadores podrían prestar demasiada atención en estos detalles y considerarlos parte de un puzle que en realidad no existe por lo que los llevaría a frustrarse y no disfrutar del juego.

La historia de nuestra Escape Room, normalmente es introducida antes de empezar el juego e introduce los objetivos a proceder y las reglas, puede ser totalmente concluyente en el momento que la contamos o continuar extendiéndose a medida que progresamos en los puzles del juego, lo cual suele ser un incentivo para el jugador a progresar en el juego para conocer la historia completa.

En cuanto al diseño del juego en si podemos considerar tener una única “habitación” en la cual encontraremos todos los puzles y acertijos, y donde comenzará y acabará el juego, o que el juego discurra en varias habitaciones lo cual da más control para planificar que los puzles sean resueltos en un orden concreto al tener que progresar de una habitación a otra y aumentar la complejidad del juego.

3.2.2 Diseño Puzles

Más allá de esconder elementos en nuestra escena que los jugadores deben de encontrar para progresar, debemos incluir en nuestra Escape Room unos cuantos puzles que supongan un reto para el jugador y serán el núcleo de nuestro juego por lo que es fundamental que estén bien diseñados.

Hacer que los puzles y las mecánicas para resolverlos estén relacionados con la ambientación e historia propuesta le dará un plus de inmersión a los jugadores.

Estos puzles pueden seguir un curso lineal, en el cual encontramos los pasos para resolverlo uno tras otro, o encontrar estas pistas de forma intercalada con pistas de

otros puzzles. En ambos casos estas pistas pueden hacerse reconocibles para el puzzle al que pertenecen (por ejemplo, encontrarse en un objeto relacionado al puzzle) y añadir algunas “posibles pistas” que no lleguen a nada para añadir un plus de dificultad, aunque sin abusar de ellas para no hacer nuestro juego imposible.

Si queremos que nuestra Escape Room llegue a la mayor cantidad de gente y funcione bien y sea entretenida para más jugadores, no podemos presuponer que todos los jugadores saben lo que nosotros sabemos y si nuestros puzzles requieren de cierto conocimiento, puede ser frustrante para los jugadores, por eso si introducimos un puzzle que requiera de algún conocimiento, podemos introducir este conocimiento en elementos de la escena, de forma que el jugador no tenga que buscarlo fuera del juego.

3.2.3 Flujo del Juego

Como en todos los juegos, es importante mantener al jugador en un flujo (Flow), este concepto fue introducido por el psicólogo Mihály Csíkszentmihályi, y determina que este flujo oscila entre la ansiedad producida por la dificultad y el aburrimiento producido por la facilidad y el tiempo transcurrido, si logramos mantener al jugador en el flujo de juego conseguiremos mantenerlo inmerso en él y que disfrute de la experiencia de jugarlo.

De esta forma si nuestros puzzles son muy difíciles de resolver, el jugador acabará por abandonar totalmente frustrado al no poder resolverlo y si estos puzzles son demasiados fáciles para el jugador, acabará aburrido e igualmente decepcionado con el juego.

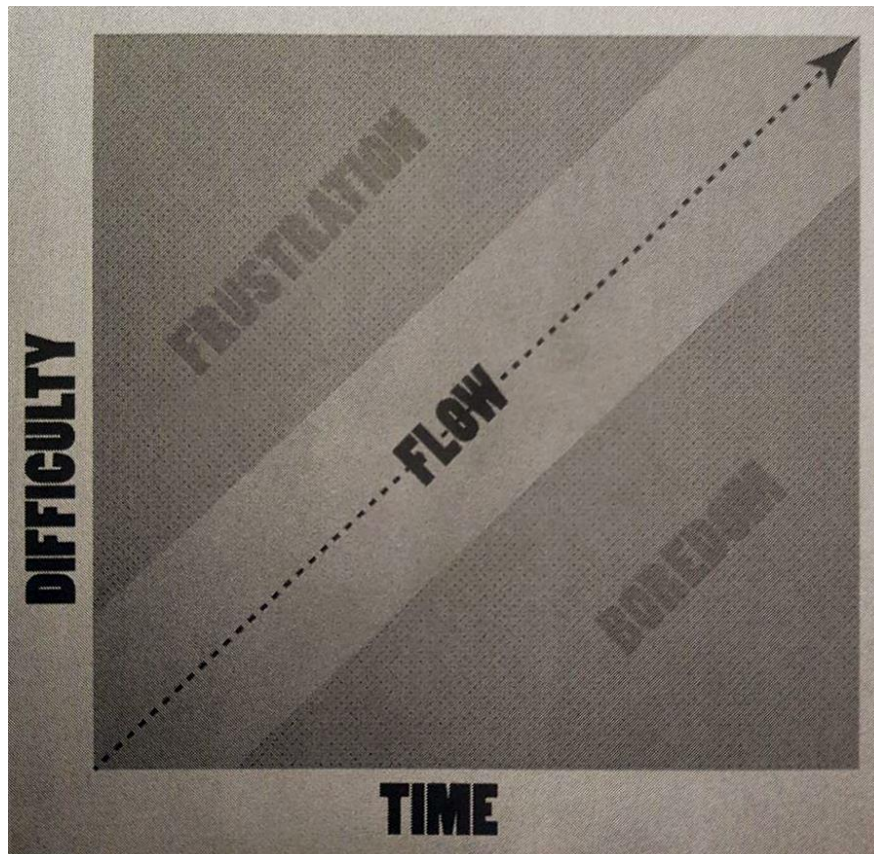


Ilustración 3: Modelo de Flujo de Csíkszentmihályi. Imagen extraída del libro *Escape The Game* de Adam Clare M.Ed

3.3 Realidad Virtual

“La definición de Realidad Virtual viene propiamente por la definición de ambas palabras ‘virtual’ que significa cercano, y ‘realidad’ se refiere al conjunto de experiencias percibidas por los humanos. Por lo que Realidad Virtual básicamente significa ‘cercano a la realidad’.

Esto podría significar cualquier cosa, pero mayormente se refiere a la emulación de la realidad.

En términos técnicos Realidad Virtual es el término utilizado para describir los entornos 3D generados por ordenador que pueden ser explorados e interaccionados por una persona, la cual forma parte de este mundo virtual trasladando interacciones naturales y siendo posible que manipule objetos o lleve a cabo una serie de acciones.” – Descripción de Realidad Virtual por Virtual Reality Society [7].

Aunque ahora se haya afianzado la Realidad Virtual como una Tecnología Emergente, con multitud de dispositivos en el mercado actual y contenido compatible, el concepto de Realidad Virtual ha estado presente desde hace mucho tiempo, desde los primeros

antecedentes en 1838, con la creación de las Fotografías Estereoscópicas y los visualizadores para poder verlas cuyos principios de diseño son los mismos que ahora se usan en los dispositivos de Realidad Virtual más avanzados.

Consistente en una composición de dos imágenes de la misma escena tomadas desde dos posiciones paralelas a una distancia promedio a la de los ojos de una persona, la cual al ser visualizada en un dispositivo con lentes para visualizar cada imagen en cada ojo da una sensación de profundidad en la imagen visualizada.

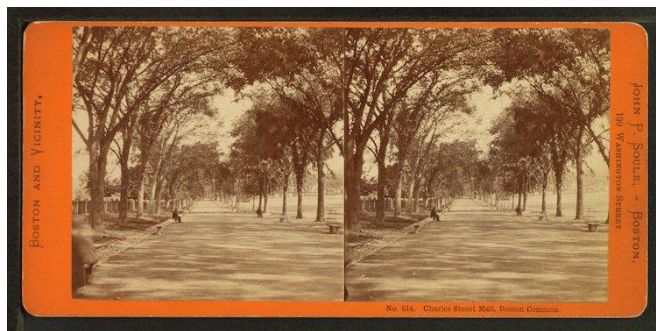


Ilustración 4: Fotografía Estereoscópica Clásica.



Ilustración 5: Dispositivo de visualización estereoscópica anterior al siglo XX.

Pero no fue hasta mediados del Siglo XX cuando se empezó a hablar del término Realidad Virtual, primero con la creación de Sensorama en 1950, una cabina con películas, pensada para estimular los sentidos, contaba con imagen estereoscópica, sonido estéreo, ventiladores, generador de olores y una silla con vibración.



Ilustración 6: Fotografía del dispositivo Sensorama

Posteriormente en los años 60 se empezaron a desarrollar dispositivos de Gafar de Realidad Virtual o Head Mounted Display (HMD), desde la *Telesphere Mask* capaz de reproducir video estereoscópico con audio estéreo, hasta la creación del dispositivo *Sword of Damocles*, el primer dispositivo VR que utilizaba gráficos generados por ordenador además de trasladar el movimiento en la escena.

Este dispositivo era sumamente pesado y necesitaba de un soporte que colgaba del techo para poder utilizarlo:

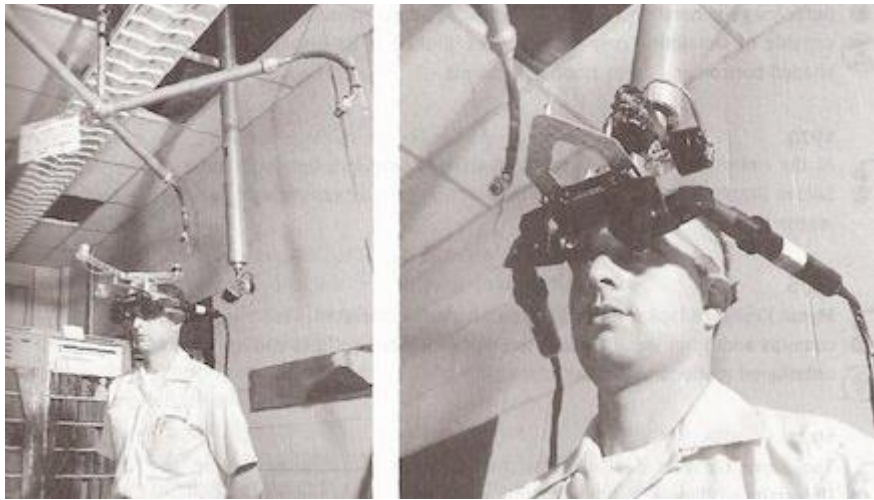


Ilustración 7: Dispositivo VR Sword of Damocles

En los siguientes años nacieron más prototipos y dispositivos de Realidad Virtual, en 1995 Nintendo lanzó la consola Nintendo Virtual Boy, la cual fue un fracaso comercial y no tuvo mucho éxito, por su incomodidad y la limitación de las pantallas monocromáticas en Negro y Rojo.



Ilustración 8: Consola y Controlador Nintendo Virtual Boy

Tras los años 90 la Realidad Virtual pasó a ser una tecnología casi olvidada, hasta que en el año 2012 nació Oculus Rift de una campaña de Kickstarter para crear un nuevo dispositivo de Realidad Virtual la cual recibió un apoyo enorme. Tras el primer kit de desarrollo lanzado por Oculus Rift varias empresas empezaron a sumarse al desarrollo de nuevos dispositivos de Realidad Virtual.

Incluso Google en 2014 anunció las Google Cardboard un soporte de cartón con dos lentes que utiliza un Smart Phone como pantalla y aprovecha sus sensores hacer Head Tracking.

Actualmente se pueden encontrar decenas de versiones mejoradas de las Google Cardboard en el mercado, hechas con materiales más resistentes, lentes más grandes y mayor ergonomía, aun así, un Smart Phone no tiene potencia suficiente para ejecutar juegos de última generación en VR y el Tracking no va más allá de la rotación de la cabeza.



Ilustración 9: Google Cardboard

En cuanto a dispositivos que ofrecen mayor rendimiento principalmente se encuentran disponibles a día de hoy:

- La versión de consumidor de Oculus Rift.
- Las HTC Vive (y las recientes HTC Vive Pro) creadas en colaboración con Steam la mayor plataforma de videojuegos de PC.
- Las Play Station VR creadas específicamente para la consola de Sony Play Station 4.
- Los Windows Mixed Reality Headsets un estándar creado por Microsoft y fabricado en distintas versiones por varias empresas.

Estos dispositivos tienen en común la necesidad de un equipo para ejecutar las aplicaciones y todas ellas cuentan con Tracking del dispositivo, tanto en rotación como movimiento en 3D, además de contar con dispositivos de interacción los cuales también hacen Tracking de su posición y rotación.



Ilustración 10: Oculus Rift



Ilustración 11: HTC Vive Pro



Ilustración 12: Play Station VR



Ilustración 13: Acer Windows MR Headset

3.3.1 SDKs/VR Frameworks

Todos los dispositivos mencionados anteriormente disponen de sus Paquetes de desarrollo, destinadas a crear aplicaciones destinadas a cada uno de los dispositivos en concreto, tanto nativos para crear aplicaciones en nuestro lenguaje de preferencia como para desarrollar aplicaciones en distintos motores gráficos, la gran mayoría con paquetes disponibles para los dos más populares: Unity y Unreal Engine.

Además, podemos encontrar varios Frameworks de desarrollo que nos permiten crear aplicaciones para varios tipos de Dispositivos VR, entre ellos A-Frame, creado por el equipo de Mozilla y pensado para la web, el cual es compatible con prácticamente todos los dispositivos, tanto móviles para usarlos con un visualizador como los principales

dispositivos VR. También encontramos Steam VR en un principio creado por Steam para las HTC Vive pero que se ha abierto para funcionar con los dispositivos que necesitan de un PC para funcionar, una solución muy interesante pero que nos obliga tanto para desarrollar como utilizar las aplicaciones utilizadas por este framework a tener instalada la plataforma Steam y Steam VR para poder funcionar.

3.4 Juegos de Escape Room para Dispositivos VR

Como sabemos es un género que funciona muy bien junto con la Realidad Virtual y que gozan de popularidad por lo que, si buscamos videojuegos de Escape Room para Realidad Virtual, encontraremos una gran cantidad. Nada más que en Steam tenemos disponibles 133 juegos del género para VR, sin embargo, en la tienda de aplicaciones de Windows Mixed Reality apenas podemos encontrar unos cuantos juegos de los cuales solo son 3 los juegos de tipo Escape Room.

Dentro de toda esta multitud de juegos de Escape Rooms para Realidad Virtual, podemos apreciar que, en la mayoría, las interacciones son un poco pobres con los elementos del escenario limitándose a acercar el controlador, pulsar un botón y realizarse automáticamente la acción. Además, los juegos para Realidad Virtual no ofrecen demasiada información sobre cuándo se puede realizar una interacción y que botón de los controladores tenemos que usar para realizar la interacción.



Ilustración 14: Gameplay de The Puzzle Room VR para Steam

3.5 Conclusión

En conclusión, actualmente los dispositivos VR son una tecnología que se está asentando en el mercado y ya podemos encontrar distintas alternativas muy competitivas y con soluciones para los desarrolladores.

Junto a la creciente popularidad de las Escape Room en todo el mundo, ha hecho de este tipo de juegos un imprescindible para plataformas VR en los cuales siempre se puede seguir innovando y aprovechando nuevos mercados que se abren al nacer un nuevo dispositivo VR.

4 Análisis inicial del problema

El videojuego el cual propongo desarrollar debe de hacer uso de una de las Tecnologías VR disponibles y seguir los criterios de diseño de un juego de estilo Escape Game, este será para un único jugador y contará con un menú principal y un nivel jugable, aunque el juego ya presentado sea completo podrá continuarse con más niveles y avanzando en la historia. Los puzzles que plantea el juego deben suponer un desafío para el jugador, sin llegar a ser demasiado complicado como para impedir su avance, planteándole puzzles y escondiendo elementos necesarios para proseguir.

Por ello el juego debe permitir al jugador moverse libremente por el escenario, no solo con el movimiento físico del jugador en la realidad, también permitiéndole desplazarse únicamente en el juego. Sin embargo, debería de impedir ver las zonas sin descubrir o el contenido de almacenes cerrados si el jugador atraviesa con su movimiento una pared virtual, para no romper la experiencia del jugador. El jugador podrá interactuar con pequeños objetos: cogerlos, tirarlos o añadirlos al inventario. E interactuar con elementos de la escena como puertas, cajones, teclados, etc.

El objetivo del jugador no será solo conseguir salir de la escena si no averiguar por qué está ahí encerrado, la historia se contará con noticias en elementos de la escena (ordenadores, televisores, móviles ...). El hilo argumental tratará de que nos hemos quedado atrapados dentro de un juego de realidad virtual del cual no podemos salirnos, poco a poco conoceremos la realidad e intentaremos escapar de este juego.

Se intenta transmitir en este juego una atmosfera de claridad y tranquilidad para que el jugador pueda centrarse en los puzzles y no lo encuentre agobiante, haciendo uso de la luz, el estilo gráfico, la música y los sonidos del juego.

El juego en si constará de un menú principal en el cual podremos iniciar la partida o salir del juego, interactuando con botones e interruptores colocados encima de un

escritorio, en un área pequeña pero totalmente abierta y lo suficientemente oscura para no poder ver más allá de la zona central que será la única iluminada:

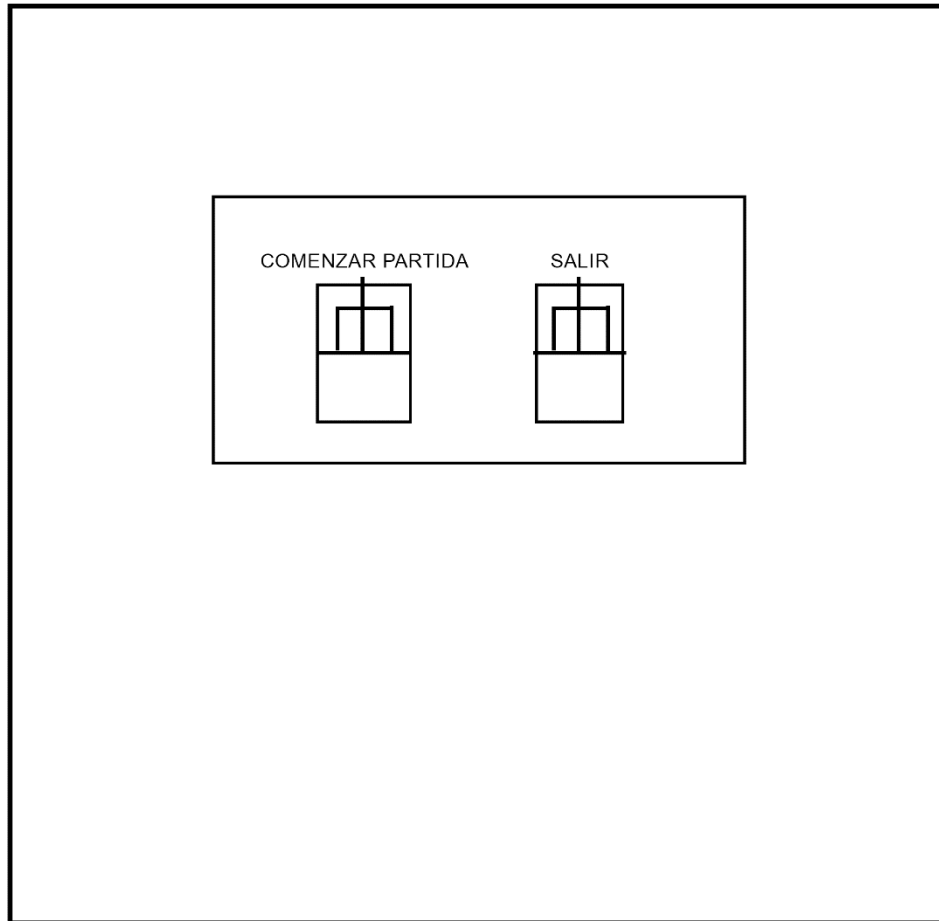


Ilustración 15: Boceto del Menú principal

En cuanto al nivel, empezaremos en la oficina enfrente de la misma mesa que teníamos en el menú principal, tendrá 4 habitaciones distintas, con un puzle que resolver en cada una de ellas para salir, con un total de tres puertas cerradas con sus respectivas llaves y una puerta oculta. Al llegar a la salida volveremos al menú inicial:

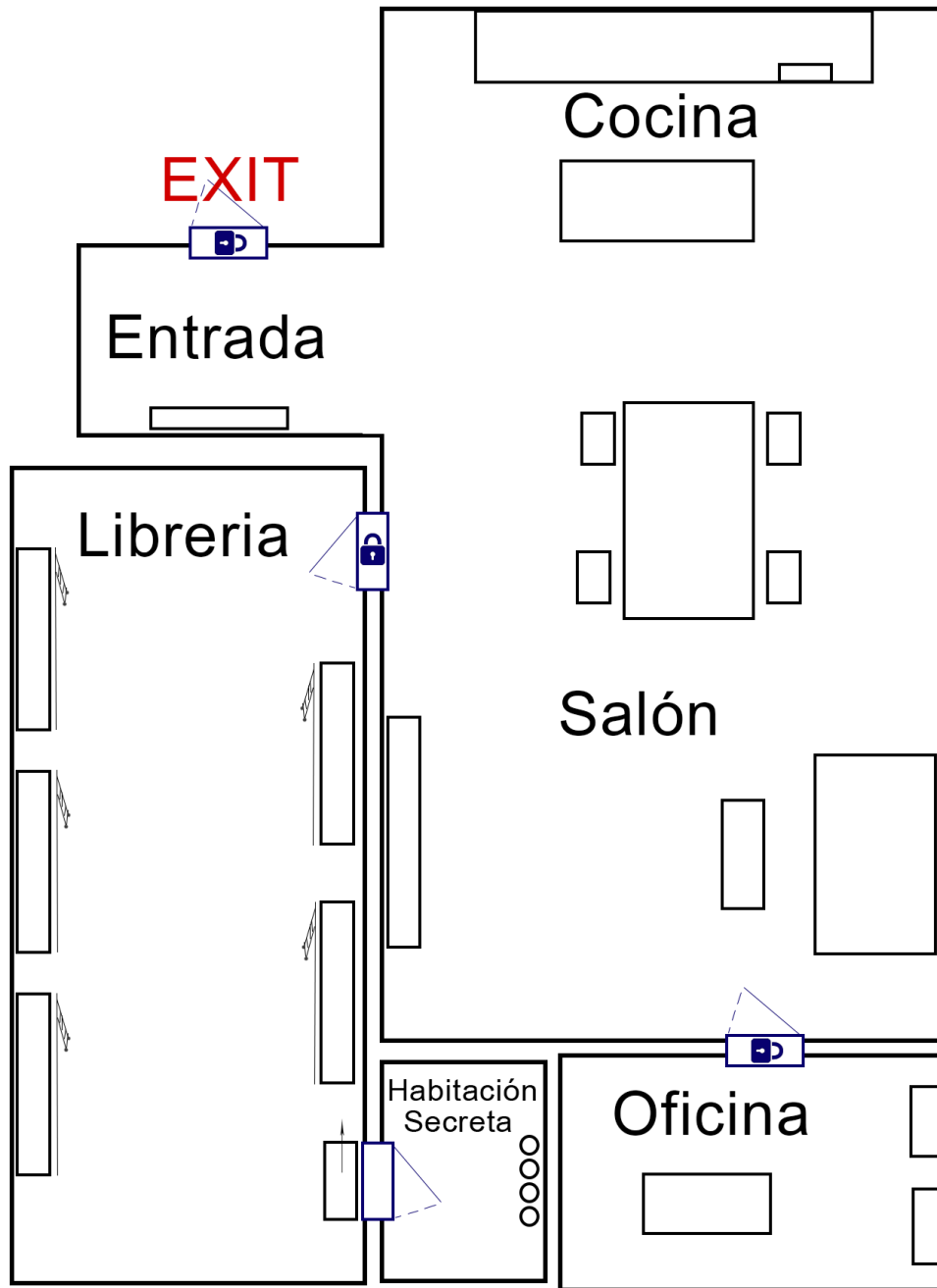


Ilustración 16: Boceto y Diseño del Primer Nivel

Una descripción más extensa y detalla del diseño del juego se encuentra en el Documento de Diseño del Juego (GDD) se encuentra en el correspondiente Apéndice de este documento.

5 Tecnología a usar

Entre los posibles dispositivos VR de los que tengo la posibilidad de trabajar, HTC Vive y Acer WMR Headset, decidí utilizar el HMD de Acer, dado que son una tecnología totalmente reciente que me resulto interesante ya que tienen unas características interesantes como la pantalla de mayor resolución que las Vive además de no necesitar colocar sensores en el área de juego para detectar nuestro movimiento y tiene un precio más competitivo que otras soluciones.

Acerca del SDK a utilizar he preferido utilizar el propio para este dispositivo frente al de Steam VR.

Por una parte, si usase el framework de Steam aunque haría el juego compatible con otros dispositivos VR, el juego queda restringido a Steam que como ya hemos visto su tienda ya encontramos muchos juegos de Escape Room frente a los disponibles en la tienda de Windows, además de ser más difícil publicar un juego en Steam.

Por otro lado, el SDK propio de WMR está solo disponible para Windows bajo C# y para el motor gráfico Unity. Al ser una tecnología muy reciente esta SDK aún no está terminada siendo aún bastante básica, aunque Windows decidió liberar el código publicándolo en GitHub y la comunidad puede participar en su desarrollo, por lo que parte del desarrollo de este trabajo podrá ayudar a mejorar la SDK.

Para realizar el desarrollo de la aplicación se ha realizado con el motor gráfico Unity [8], el cual nos ofrece una solución de alto nivel para desarrollar todo tipo de videojuegos que goza de mucha popularidad, siendo gratuito para proyectos que no generen beneficio o este esté por debajo de cien mil dólares anuales en ingresos brutos, a partir de los cuales deberemos de pagar una suscripción mensual.

Unity es el motor gráfico que he escogido ya que con el que he trabajado antes y tengo cierta experiencia, es gratuito y el único motor gráfico que tiene disponible SDK de WMR. En concreto en su última versión compatible con esta SDK la 2017.3.1f1.

6 Metodologías a usar en el proyecto

A la hora de desarrollar el proyecto se ha seguido una metodología de desarrollo ágil, con un diseño inicial muy básico con el que se crea un prototipo sobre el que se itera para añadir funcionalidad, organizado mediante Scrum se establecen unos objetivos semanales para aumentar esta funcionalidad y el diseño de la aplicación.

Esta suele ser la metodología para trabajar en un videojuego actual, ya que las metodologías clásicas de Desarrollo en Cascada no son tan efectivas en este tipo de proyecto. En el desarrollo de videojuegos suelen surgir nuevos requisitos o algunos requisitos ya contemplados se desechan. Una planificación en cascada supone invertir un tiempo inicial en crear todo el diseño de la aplicación, un cambio en el diseño provocaría volver a esta fase y desechar gran parte de ella provocando una gran pérdida de tiempo, sin embargo, con metodologías ágiles que siguen procesos iterativos este cambio solo supondría un cambio en el diseño de esta iteración.

7 Plan de entregas

En el videojuego a desarrollar que he llamado “In-Game” los principales objetivos a desarrollar son:

- Permitir al jugador agarrar objetos, trasladándolos y rotándolos con sus manos.
- Permitir al jugador guardar objetos en un inventario, pudiendo equipárselos y desequipárselos.
- Permitir al jugador interaccionar con elementos de la escena: puertas, cerraduras, cajones y botones.
- Mostrar ayudas visuales al jugador para conocer los controles del juego.
- Reproducir sonidos en un ambiente 3D para enfatizar las interacciones y las acciones de los elementos del escenario.

7.1 Historias de Usuario

A continuación, se detalla un listado de historias de usuario que se han obtenido de las interacciones del diseño y desarrollo del juego. Incluye una estimación del esfuerzo realizado para el desarrollo y la priorización de las historias.

La estimación (Estim.) está expresada en Puntos de Historia y la prioridad (Prio.) está expresada en un rango de uno a cuatro, siendo 1 el valor que expresa más prioridad.

Ident.	Título	Estim.	Prio.
HU.1	El jugador puede visualizar y desplazarse por la escena.	1/2	1
HU.2	El jugador puede visualizar los Controllers en el juego.	2	1
HU.3	El jugador reconoce cuando puede interactuar con un elemento de la escena.	1	1
HU.4	El jugador reconoce cuando está interaccionando con un elemento de la escena.	2	3
HU.5	El jugador puede agarrar un objeto.	1/2	2
HU.6	El jugador puede abrir una puerta desbloqueada.	4	2
HU.7	El jugador puede desbloquear una puerta.	2	2
HU.8	El jugador puede interaccionar con un interruptor y este realizar una acción/lanzar un evento.	3	2
HU.9	El jugador puede guardar un objeto inventariable en el inventario.	2	2
HU.10	El jugador puede equiparse/desequiparse objetos del inventario.	4	2
HU.11	El jugador puede interaccionar con cajones.	2	2
HU.12	El jugador puede interaccionar con botones.	2	2
HU.13	El jugador puede seleccionar botones en una cerradura con contraseña.	2	2
HU.14	El jugador recibe información del estado de la cerradura con contraseña.	1	3
HU.15	El jugador puede seleccionar números en las estanterías moviendo escaleras.	2	2
HU.16	El jugador percibe información de los controles que debe pulsar para una interacción.	1	4
HU.17	El jugador percibe sonidos en 3D de los elementos de la escena.	2	4
HU.18	El jugador no puede moverse fuera de los límites de la escena.	1	3
HU.19	El jugador no puede ver elementos que deben estar ocultos hasta que los desbloquee.	1	2
HU.20	El jugador puede empezar el juego o salir de este desde el menu.	5	2
HU.21	El jugador puede jugar el primer nivel.	10	2

7.2 Cálculo temporal del proyecto

El desarrollo del proyecto será llevado únicamente por mí, dedicando unas 7 horas diarias de trabajo y estimando que las iteraciones realizadas van a tener una duración de una semana. Por ello considero que una semana completa de trabajo (7 días) y por tanto una interacción equivale entre 5 y 6 días reales de trabajo.

Asignándole un Punto de Historia por día real de trabajo, el ritmo de desarrollo es de 5 a 6 Puntos de Historia por iteración.

Con un Esfuerzo total del proyecto de 50 Puntos de Historia y un ritmo de trabajo de 5 Puntos de Historia por iteración, estimo una duración del proyecto de entre dos meses y medio y tres meses.

7.3 Descripción de las entregas

Entrega	Objetivo	HU
1	Inicio del Proyecto en Unity, Añadir SDK WMR, Explorar la SDK.	1, 5
2	Modelos manos y Puerta, Añadirlos al Proyecto.	2
3	Interacción visible con Outline.	3
4	Interacción con Puerta.	6
5	Modelos Interruptor y Llaves.	8, 7
6	Interacción con Interruptor.	8
7	Posicionamiento de las manos en las Interacciones.	4
8	Creación del Inventario y objetos Inventariables.	9, 10
9	Creación de Interactable/Interactor. Interacción de llaves y cerraduras.	7
10	Interacción con Cajones.	11
11	Diseño del Menú Inicial y del Primer Nivel.	20, 21
12	Desarrollo del Puzle del Salón (Caja Fuerte con combinación numérica).	21, 12, 13, 14
13	Desarrollo del Puzle de la Librería (Tablet y Librerías).	21, 15
14	Desarrollo del Puzle de la Habitación Secreta (Interruptores y Luces).	21
15	Sonidos de la escena y Música.	17
16	Ayudas a los controles, Optimización y Corrección del movimiento.	16, 18, 19
17	Imágenes con la historia mostradas en la escena.	21

8 Desarrollo

8.1 Inicio del Proyecto en Unity, Añadir SDK WMR, Explorar la SDK

Lo primero a tener en cuenta para iniciar el proyecto es tener Unity en la versión compatible [9] (2017.3.1f1) y descargar el SDK del repositorio de GitHub [10], del mismo repositorio podemos descargar una serie de ejemplos para probar el SDK.

Dentro de las escenas de ejemplos que nos ofrece el SDK, encontramos una llamada MotionControllers-GrabMechanics, en la cual podemos hacer una serie de interacciones simples con cubos de la escena, pudiéndolos agarrar de varias formas y lanzarlos.



Ilustración 17: Escena MotionControllers-GrabMechanics

Lo interesante de esta escena es que crea una arquitectura para que podamos agarrar objetos al tocarlo con un Controlador, este sistema sigue la siguiente secuencia, donde el Grabber es el Controlador y el Grabbable al objeto a interaccionar:

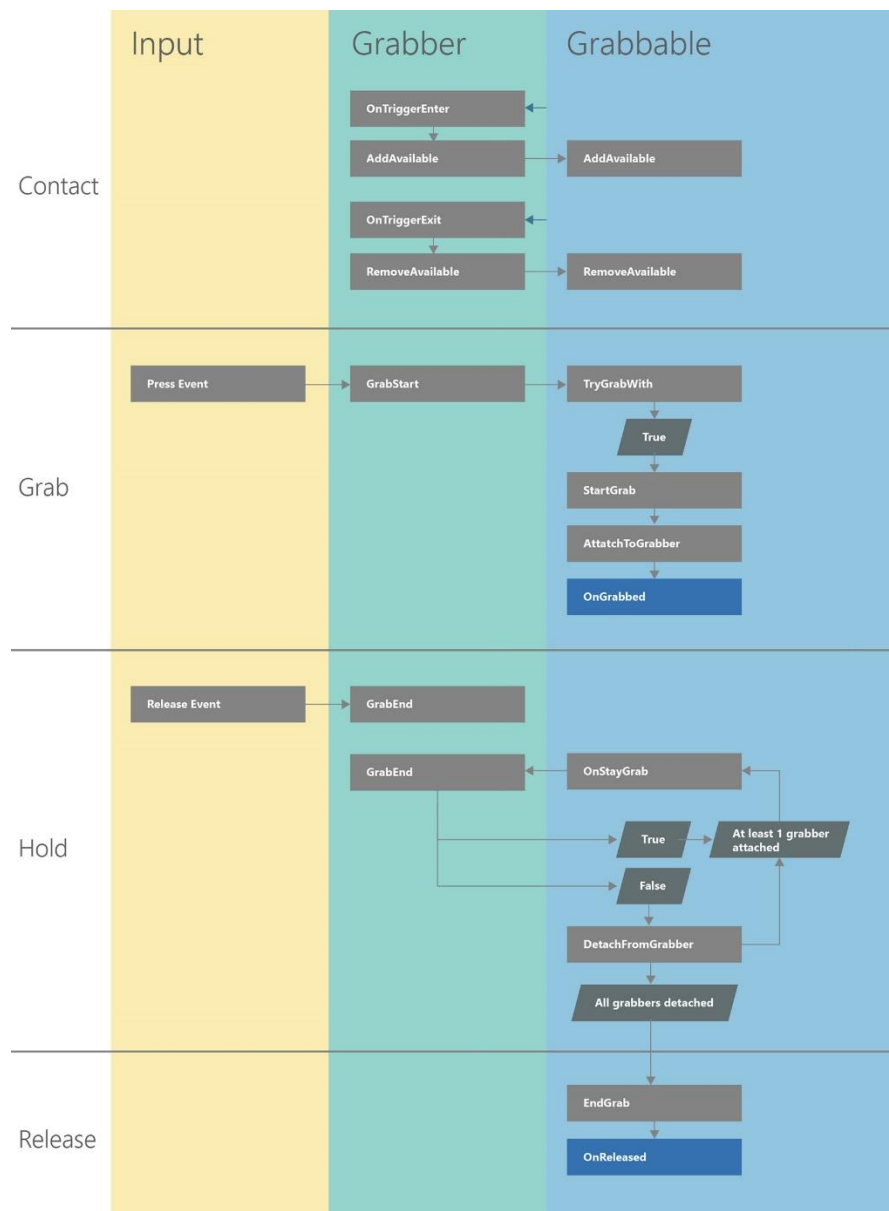


Ilustración 18: Secuencia de interacción Grabber y Grabbable

Además, el ejemplo contiene scripts para indicar el estado de la interacción coloreando el objeto con colores distintos para las siguientes situaciones: está en contacto con un controlador, en contacto con un controlador, siendo “agarrado” por un controlador y siendo “agarrado” por dos controladores.

Explorando los ejemplos y la documentación vemos que la SDK incluye entre muchas más cosas, ya que en la misma SDK sirve para desarrollar aplicaciones con las HoloLens de Microsoft, la conexión con el dispositivo, el control de los Inputs (teclado, controlador de Xbox y los Motion Controller) y el Prefab del jugador que incluye la cámara, un cursor que se sitúa sobre la superficie de lo que miremos o señalemos con un Controlador y un script básico para moverse en la escena.

Con esto comienzo el proyecto en Unity al cual una vez añadida la SDK se configura para funcionar con las Gafas de Realidad Virtual, desde la nueva opción que ha añadido dicha SDK:

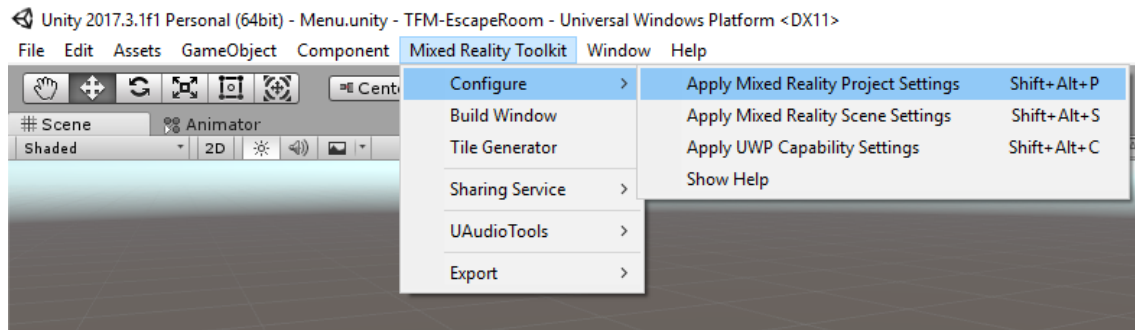


Ilustración 19: Configuración del Proyecto Unity para funcionar con Mixed Reality

Tras marcar esta opción aparece una ventana emergente en la que activamos “Target Occluded Devices” para indicar que el proyecto es para las gafas VR (HMD) y no para las HoloLens:

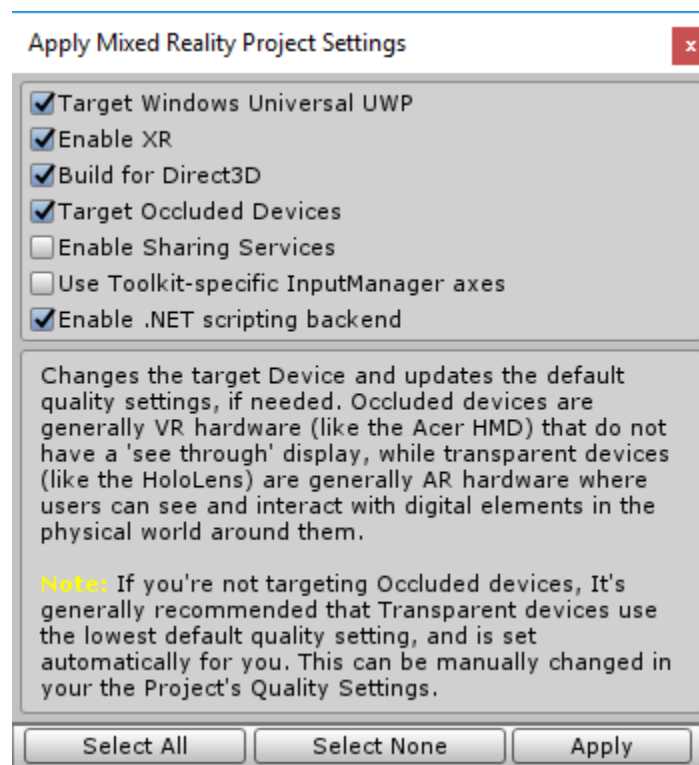


Ilustración 20: Opciones de configuración del proyecto

Una vez configurado el proyecto, podemos configurar la escena para añadirle los elementos necesarios de la Cámara y los Controladores, de nuevo desde el menú de la SDK:

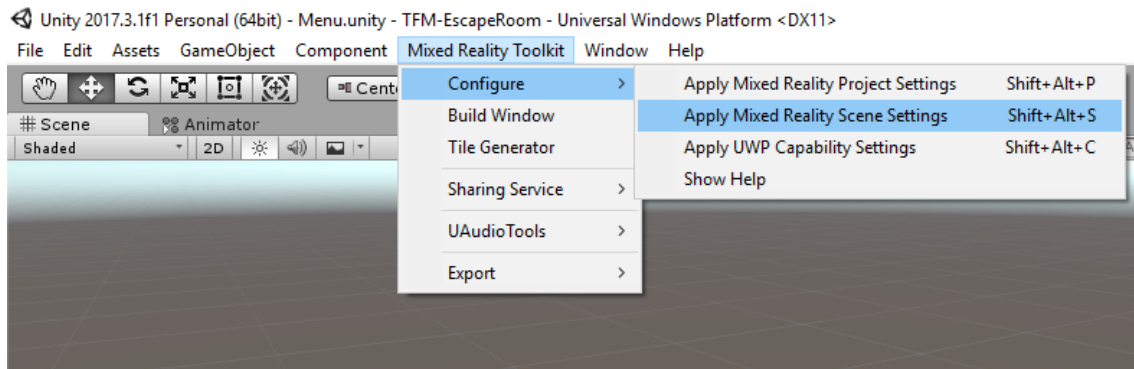


Ilustración 21: Configuración de la Escena para añadir los elementos de Mixed Reality

Y en el menú de configuración dejamos marcadas todas las opciones para que lo añada todo:

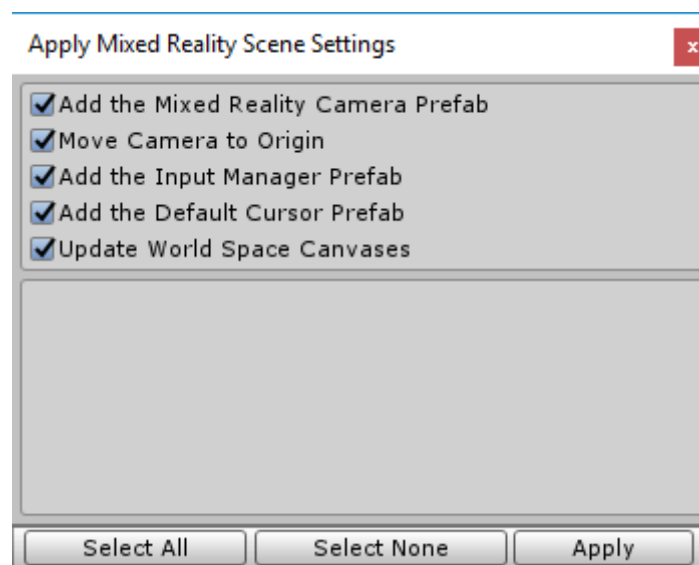


Ilustración 22: Opciones de configuración de la Escena

8.2 Modelos manos y Puerta, Añadirlos al Proyecto

Para representar nuestras manos en el mundo virtual he decidido modelar, texturizar y animar unas manos, de estilo robóticas inspiradas en prótesis biomédicas, con el programa Blender [11] (también gratuito y de código libre).

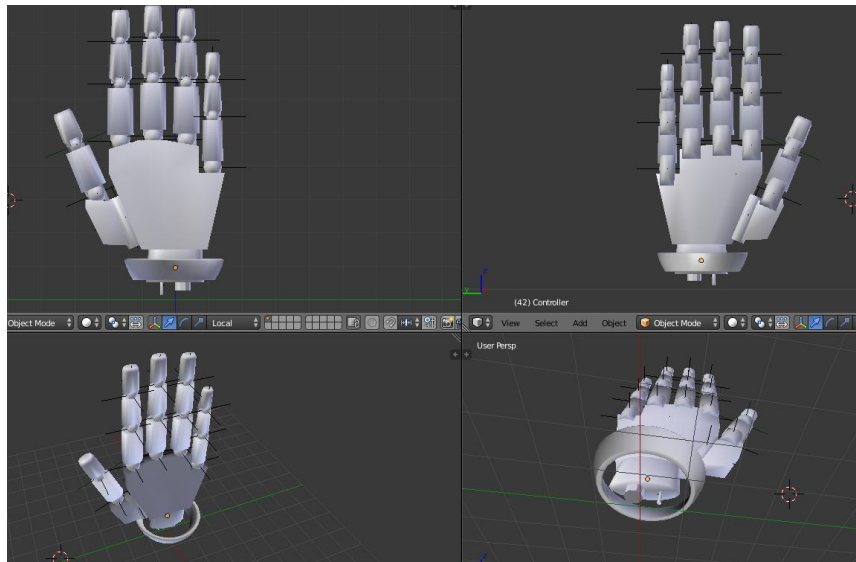


Ilustración 23: Modelo de Mano en blender

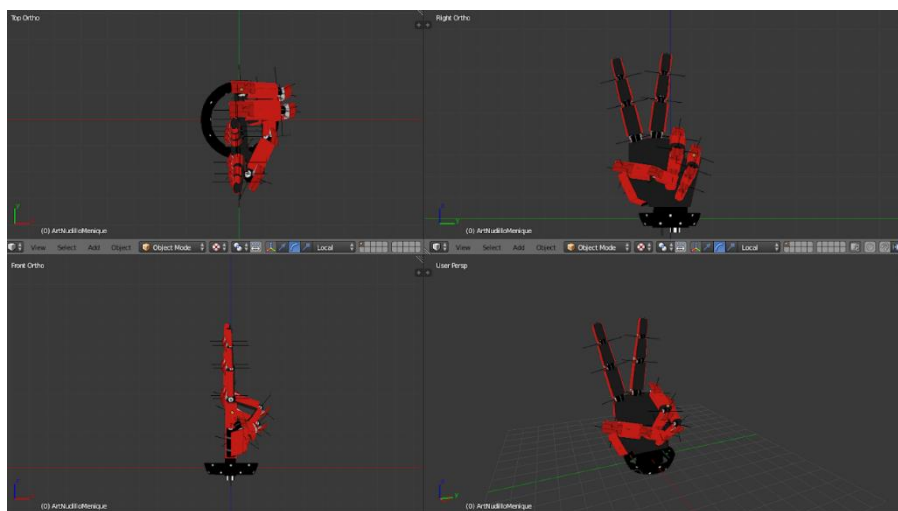


Ilustración 24: Modelo Mano texturizado

A esta mano le he añadido el aro en la muñeca igual que el de los Motion Controller para mostrar la relación que tiene con estos.

Este modelo contiene 9 animaciones:

- **Idle:** una animación larga con un ligero movimiento para cuando la mano no está haciendo ninguna acción.
- **Grab y UndoGrab:** cierra la mano totalmente para agarrar algo y la animación contraria para abrirla.
- **Handle y UndoHandle:** cierra la mano dejando un hueco (como si cogiese una barra) para agarrar objetos con más volumen y la animación contraria.
- **GrabKey y UndoGrabKey:** cierra la mano con el pulgar sobre el índice para coger una llave en esa posición y la animación contraria.
- **Spot y UndoSpot:** deja la mano señalando con el dedo índice y la animación contraria.

Igualmente he Modelado y Texturizado una puerta en Blender, la cual será de los elementos básicos, con los distintos elementos distinguibles en distintas mallas para poder operar con ella en Unity, de forma que sigue la siguiente jerarquía:

- Marco.
 - Bisagras.
 - Puerta.
 - Embellecedor Pomo Delantero.
 - Pomo Delantero.
 - Cerradura Delantera.
 - Embellecedor Pomo Trasero.
 - Pomo Trasero.
 - Cerradura Trasera.

De esta forma con el giro de las bisagras se mueve toda la puerta y los elementos que contiene y permitiendo a los pomos y cerraduras girar sin afectar al resto.



Ilustración 25: Modelo Puerta texturizado

A la hora de importar los modelos en Unity, el formato “.blend” requiere que tengamos Blender instalado en el equipo para poder visualizarlos dentro del editor (una vez en producción no es necesario), además de escalarlos para adaptarlos a la escala de Unity.

En el caso de la mano separar las animaciones y marcar la animación Idle para que se visualice en bucle. Y para ayudarnos a controlar estas animaciones las montamos sobre un *Animator Controller*, en el cual definimos los estados de las animaciones, las transiciones existentes entre ellas y las variables que influyen en estas transiciones:

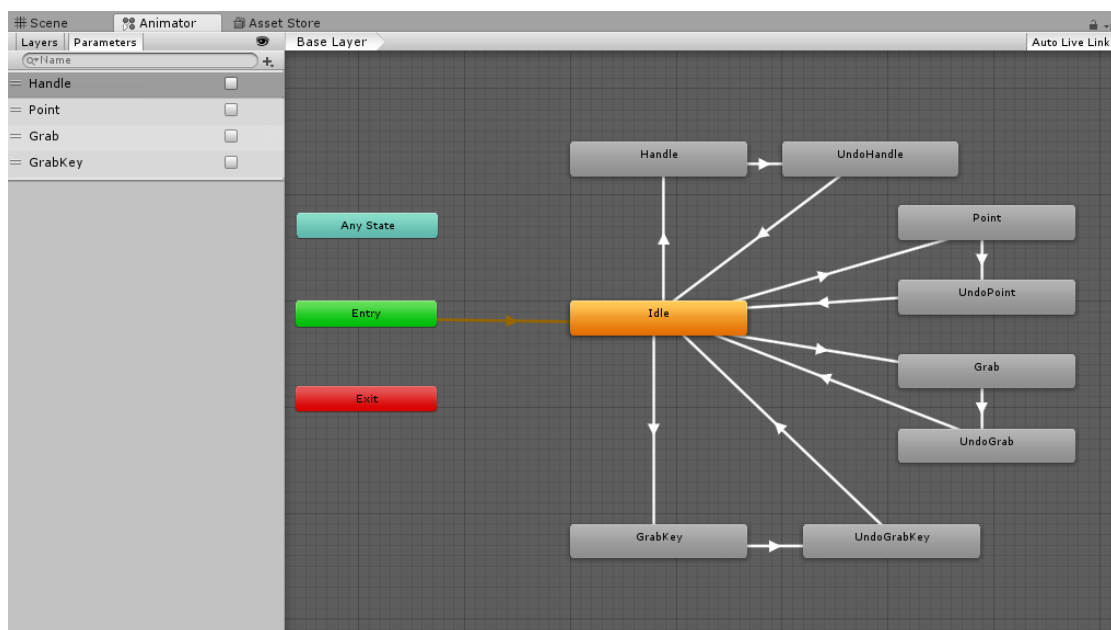


Ilustración 26: Estados del Animator de las Manos

8.3 Interacción visible con Outline

Como ya he mencionado antes al interactuar con los objetos, entre los ejemplos tenemos un script que colorea el objeto a interactuar según el estado del Grabbable. Sin embargo este cambio de color puede no ser claro, dependiendo de la iluminación y de la textura del objeto.

Por ello y porque queda estéticamente mejor, he modificado el script para que en vez de colorear el objeto, dibuje una línea del color por el contorno de este (Outline), haciendo uso de la Herramienta Gratuita de la Asset Store de Unity, Quick Outline [12].

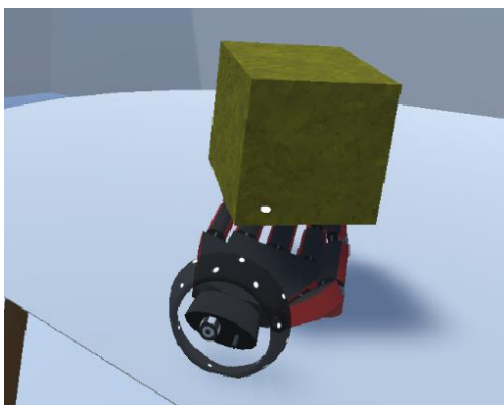


Ilustración 27: Coloreado al interactuar

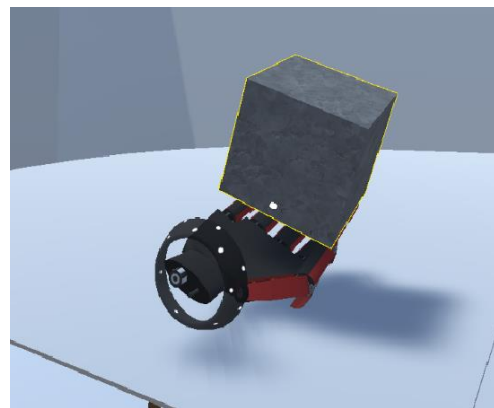


Ilustración 28: Outline al interactuar

8.4 Interacción con Puerta

Para interactuar con las puertas necesito que al agarrar el pomo el jugador pueda moverlo al mover su Controlador y que cuando lo haya girado, la puerta se desbloquee, si no está cerrada con llave, y pueda girarla sobre sus visagras.

Primero para la rotación del pomo, siguiendo la estructura de los Scripts Grabbable, creo un GrabbableRotation que hereda del BaseGrabbable, esta clase es abstracta y gestiona la conexión con los controllers y su estado al entrar en contacto con un controller y si en este se pulsa el botón asignado para interactuar, ofreciendo los metodos a sobrecargar que son llamados cuando estos estados cambian.

Para realizar la rotación, recibe como parámetros:

- La distancia máxima para la interacción.
- El vector que define el eje de Rotación.
- El vector define el Forward de Rotación.
- Los límites de la rotación.

Cuando un Controlador está haciendo Grab sobre el, dentro del bucle de ejecución para cada frame del juego, realiza:

- Obtiene el plano perpendicular al Eje de Rotación definido.
- Calcula la proyección de la posición del Controller sobre el Plano.
- Si la distancia del objeto al Controller es mayor que la distancia máximaPermitida:
 - Termina la Interacción.
- Si es menor:
 - Calcula la rotación entre el vector director del objeto a la posición proyectada del Controller y el vector director del Forward de Rotación definido.
 - Si el ángulo de esta rotación está dentro de los límites:
 - Establece la rotación del objeto como la rotación calculada anteriormente.

Es importante destacar que a los parámetros como Eje y Forward de Rotación, se les aplique antes de usarlos la rotación del padre para que estas sean direcciones locales y al rotar la puerta, roten estas dos direcciones junto con ella .

En la siguiente imagen se muestra la visualización de los parámetros que recibe el script, siendo la esfera azul la máxima distancia de interacción, la línea magenta el eje de Rotación y la línea amarilla el Forward de Rotación:

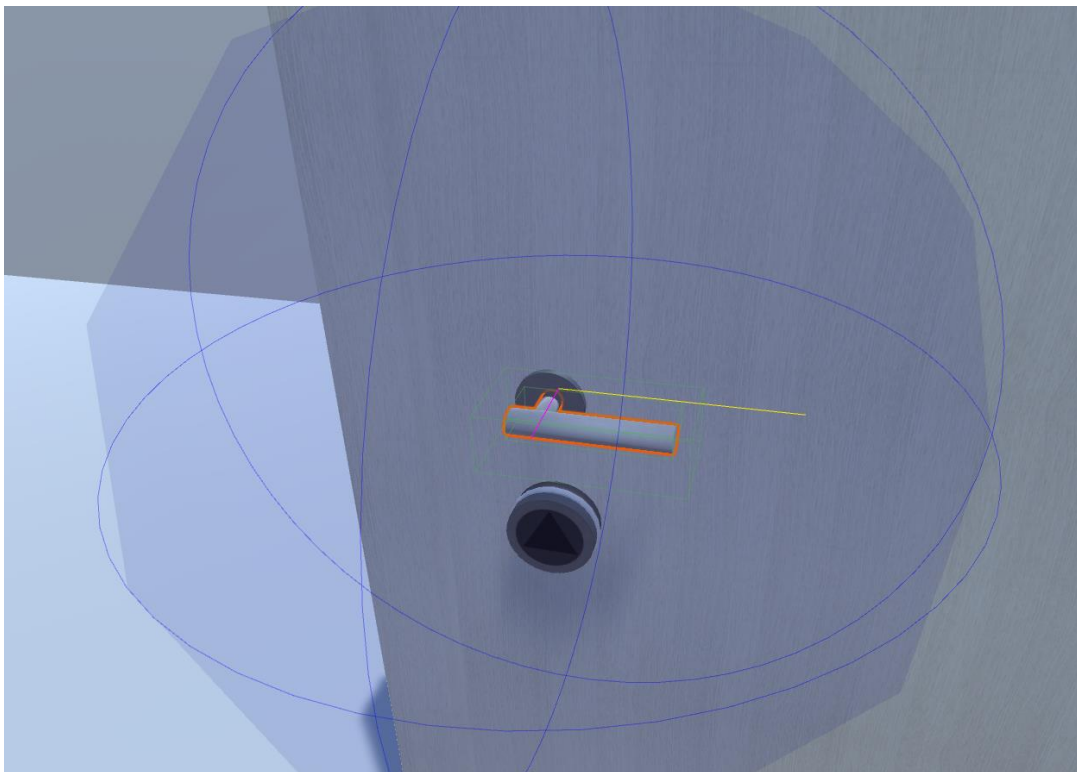


Ilustración 29: Visualización de los parámetros del GrabbableRotation

Una vez agarramos los pomos se debe de poder rotar la puerta sobre las bisagras, para esto creamos un nuevo Script que reaproveche el cálculo anterior, con su propio eje y forward de rotación, pero que esta vez se calcule cuando alguno de los pomos lance el evento de que su estado ha cambiado y este nuevo estado es el de estar siendo agarrado por un Controlador.



Ilustración 30: Visualización de los parámetros en la Bisagra

Por último, para controlar el resto de parámetros de la puerta tenemos el Script Door, el cual contiene el estado de la puerta: bloqueada/desbloqueada con llave, ID de la llave que la abre, puerta cerrada/abierta. Este Script controla la rotación del pomo para que al pasar un umbral se permita rotar abrir la puerta (si está desbloqueada) e impida esta rotación si el pomo está subido y la puerta en su posición inicial.

8.5 Modelos Interruptor y Llaves

Además de lo anterior como elementos básicos, he modelado y texturizado un interruptor clásico de palanca de dos posiciones (conectado, desconectado):

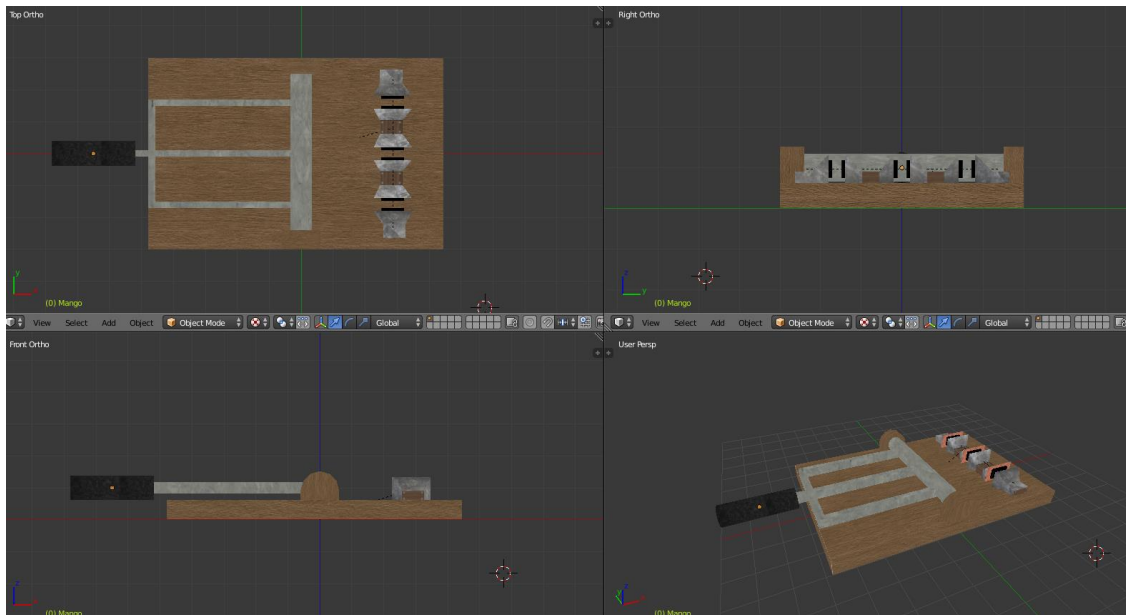


Ilustración 31: Modelo Interruptor Palanca texturizado

Y la llave que que abirán las puertas con cerradura, para darles variedad, se coloreará en unity la cabeza de la llave haciendola concordar con el color de la cerradura de la puerta que abre:

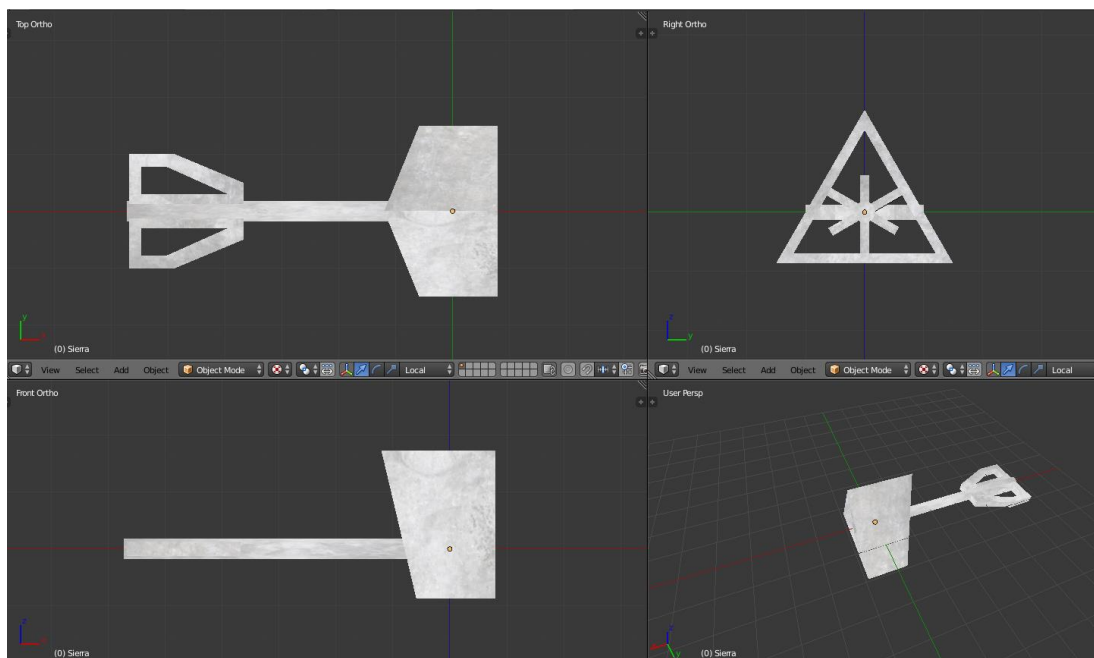


Ilustración 32:Modelo Llave texturizado

8.6 Interacción con Interruptor

Una vez en Unity necesitamos que el interruptor pueda ser agarrado por el mango y girar la palanca, para ello utilizaré el script creado anteriormente GrabbableRotation y añadirlo en el mango, por lo que ahora el script recibirá también como parámetro que objeto rotará, por defecto el propio que contenga el script:

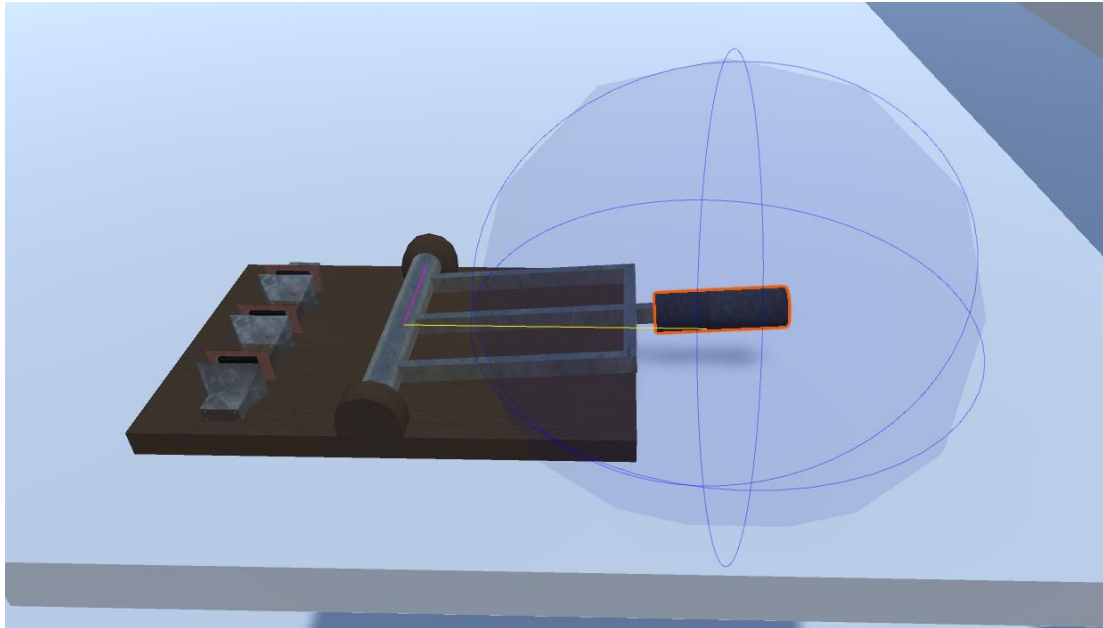


Ilustración 33: Visualización GrabbableRotation en Interruptor

El interruptor en si tiene su propio script Switch, encargado de detectar cuando llega la palanca a la zona de encendido y cuando sale de ella, lanzando eventos en ambas ocasiones, además he añadido una luz naranja y partículas de chispas en los conectores del interruptor que se activan cuando llega a la zona de encendido para dar más información al jugador de que el interruptor ha cambiado de estado:

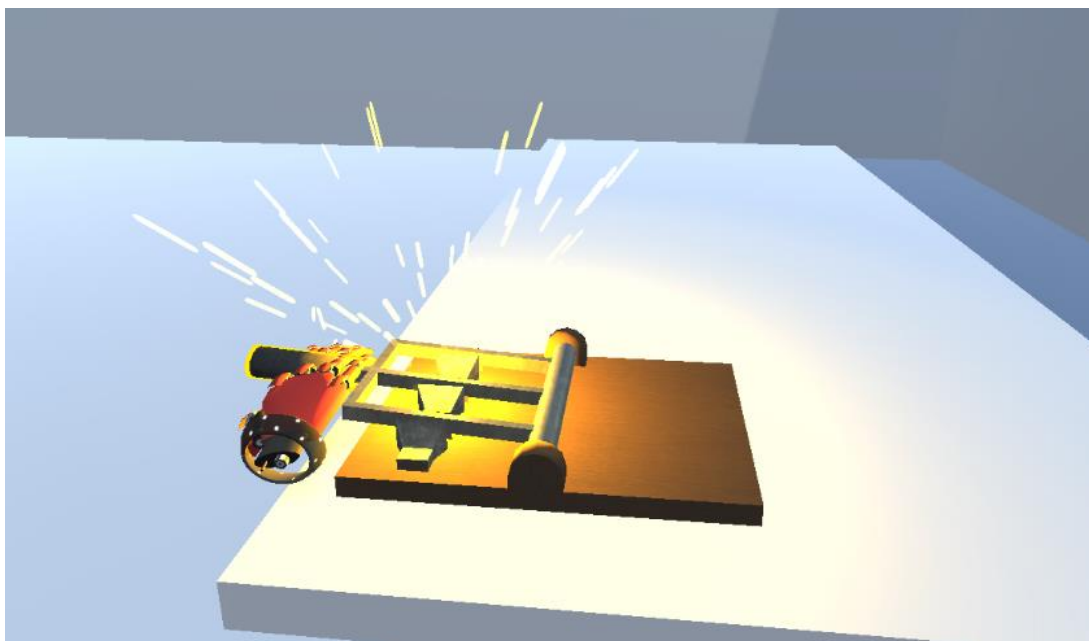


Ilustración 34: Iluminación y Particulas del Interruptor

8.7 Posicionamiento de las manos en las Interacciones

Hasta ahora en las interacciones, el modelo de la mano se movía junto con los Motion Controllers, para hacer más visible la interacción, las manos se posicionarán en el lugar adecuado, sin moverse junto con los Controllers, junto con la animación correspondiente. Así mismo hace falta tener una referencia de donde están los Controllers.

Para ello el modelo de las manos y el aro del Controller estarán separados y serán hijos de un GameObject vacío el cual contendrá los Scripts del Controller, encargado de detectar y manejar las interacciones, y ahora también de quitar el modelo de la mano como hijo para transferirlo a otro GameObject y cambiar el Aro del Controller de rotación, transparencia y color de Outline según distintas acciones.

Por otro lado, los Grabbables que quiera que posicionen la mano en un determinado lugar llevarán un Script HandPositioner, el cual recibe el modelo de la mano que interactúa, lo coloca en una posición y rotación determinada como su hijo y activa la animación correspondiente:

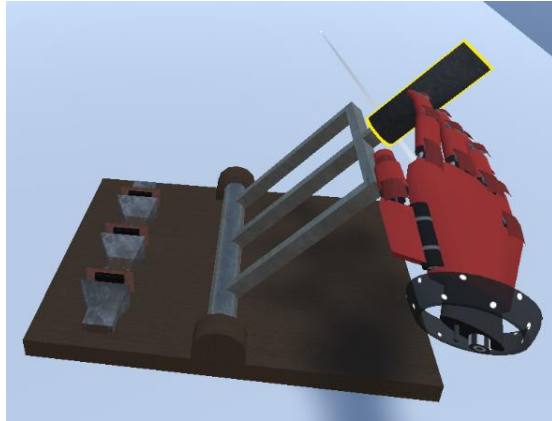


Ilustración 35: Acción Grab sin posicionar mano

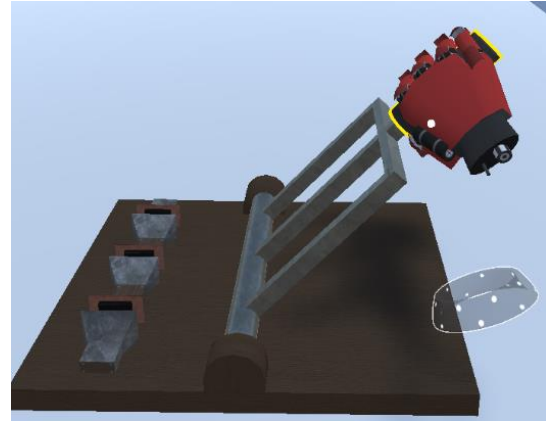


Ilustración 36: Acción Grab con posicionamiento y animación de mano

En algunos casos como el mango del interruptor o el pomo de la puerta, la mano se posicionará en una posición concreta, aunque interactuemos desde el otro lado como en la imagen:

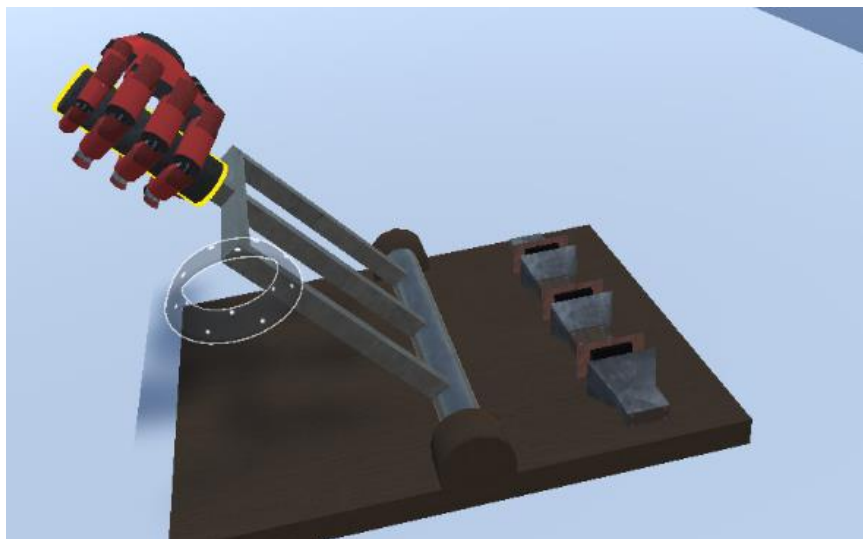


Ilustración 37: Posicionamiento estático de la mano

Esto queda solucionado con la creación del Script HandRotation añadido en el Grabbable, el cual al estar en una interacción aprovecha el algoritmo de GrabbableRotation para definir un eje de rotación y rotar sobre él, el modelo de la mano, así siempre esta rotada de acuerdo a la posición del Controller:

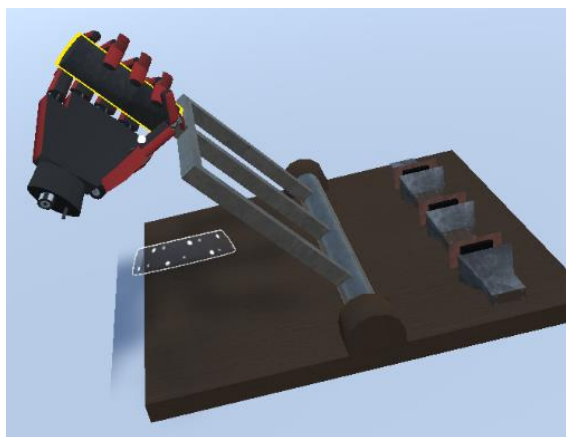


Ilustración 38:Acción Grab con Rotación de la Mano

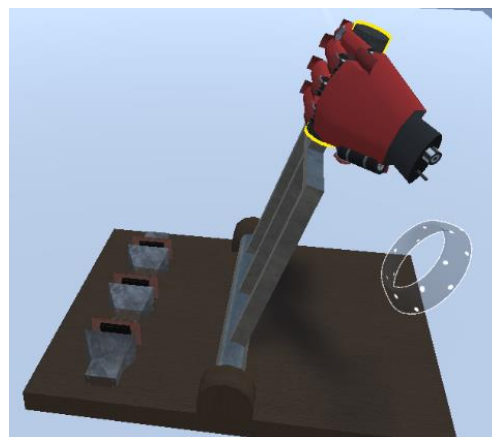


Ilustración 39:Acción Grab con Rotación de la Mano

8.8 Creación del Inventario y objetos Inventariables

Siguiendo el tutorial de Mixed Reality sobre los Motion Controllers [13], podemos desarrollar una aplicación de dibujo 3D simple, en el apartado “*Brush tools with radial layout*” enseña a crear un menú radial en 3D con scripts que incluye el SDK y un script para los Items del menú:



Ilustración 40: Menú Radial de pinceles

En este ejemplo, el menú está pensado para contener una cantidad estática de elementos y los ítems siempre son pinceles, por lo que el script que contienen es más específico además de lo necesario para el menú.

Para este proyecto utilizaré este tipo de menú adaptándolo a las necesidades y los modelos del juego. En el inventario tenemos los scripts usados en el ejemplo: Ellipse el cual crea una elipse de puntos en el espacio, LineObjectCollection que recibe una lista de objetos y los posiciona entre los puntos creados por la elipse conforme a unos parámetros e InventorySelector el cual recibe la entrada del Controller, posiciona el menú en la mano, mueve de posición los elementos para cambiar el seleccionado. Sobre el Inventory Selector he modificado para intercambiarlo entre ambos Controllers cuando recibe un input de menú y comunicarse con la mano para ocultarla y mostrar el aro del controlador de forma distintiva cuando entra en el menú.

Por otro lado, un GameObject Inventory guardará como hijos los elementos del Inventario, al que he desarrollado su correspondiente Script Inventory en cual se encarga de comunicarse con la parte anterior para añadir o eliminar elementos del inventario, estableciendo el offset en el LineObjectCollection para que los muestre de forma que el elemento seleccionado esté en el medio.

Cada elemento que puede guardarse en el Inventario lleva un Script GrabbableToInventory, el cual hace que el objeto se añada al inventario y desactive sus propiedades de Grabbable al ser agarrado, además de un script Inventory Item, el cual es llamado cuando un ítem es seleccionado moviéndolo a la posición indicada y poniendo la mano en la animación indicada.



Ilustración 41: Menú con 3 elementos



Ilustración 42: Menú con 7 elementos



Ilustración 43: Llave seleccionada equipada

8.9 Creación de Interactable/Interactor. Interacción de llaves y cerraduras

Con la intención de poder diferenciar totalmente la acción de Agarrar e Interaccionar, he recreado la estructura y flujo de los Grabbers/Grabbables, con los nuevos Interactors/Interactables. Estos los usaré para interaccionar con cerraduras y botones, asignándole un nuevo botón en los Controladores para realizar la interacción.

Esta separación permite desactivarlos independientemente, de forma que cuando tenemos un objeto del inventario equipado queda desactivado el Grabber y no es posible coger o agarrar Grabbables hasta que lo desequipemos.

En cuanto a las llaves, estas guardan una ID que se compara con los de la cerradura en el momento de entrar en contacto, si coincide nos permitirá interaccionar con ella, para lo que deben de girar sobre un eje al girar el Controlador, esto es distinto al GrabbableRotation ya que este giraba un objeto conforme a la posición del Controlador y no a su rotación.

Por lo que he desarrollado el Script InteractableTransferRotation, el cual recibe como parámetros:

- La distancia máxima para la interacción.
- El vector que define el eje de Rotación.
- Los límites de la rotación.

Cuando un Controlador está haciendo Grab sobre el, dentro del bucle de ejecución para cada frame del juego, realiza:

- Si es el primer ciclo de la interacción, guarda la rotación de la cerradura inicial y de el angulo de rotación del Controlador sobre el eje de Rotación inicial.
- Si la distancia del objeto al Controller es mayor que la distancia máximaPermitida:
 - o Termina la Interacción.
- Si es menor:
 - o Calcula la diferencia en ángulos entre la angulo inicial del controller y el actual sobre el eje de rotación.
 - o Si el angulo de rotación más el ángulo de la rotación inicial de la cerradura esta dentro de los límites:
 - Establece la rotación del objeto como la rotación calculada anteriormente.
 - o Si sobrepasa alguno de los límites, llama a los eventos correspondientes por haber pasado uno de los límites y ajusta el ángulo de rotación al límite.

De esta forma conseguimos rotar un objeto conforme a la rotación del controller, sin reiniciar la rotación a la del Controlador cada vez que se interacciona con el. En las siguientes imágenes se muestra el Eje de rotación en la cerradura (línea roja) y la interacción con ellas:

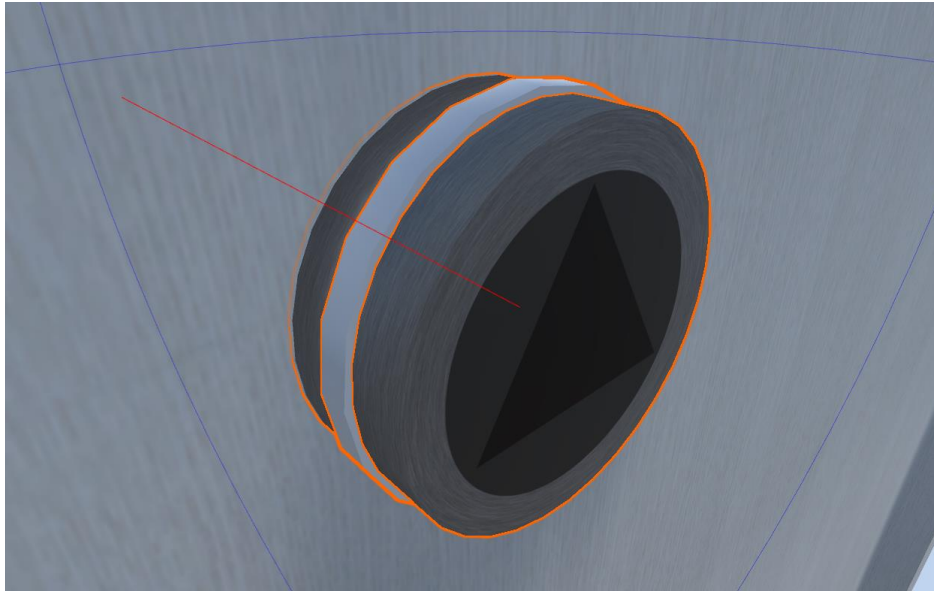


Ilustración 44: InteractableTransferRotation en la cerradura

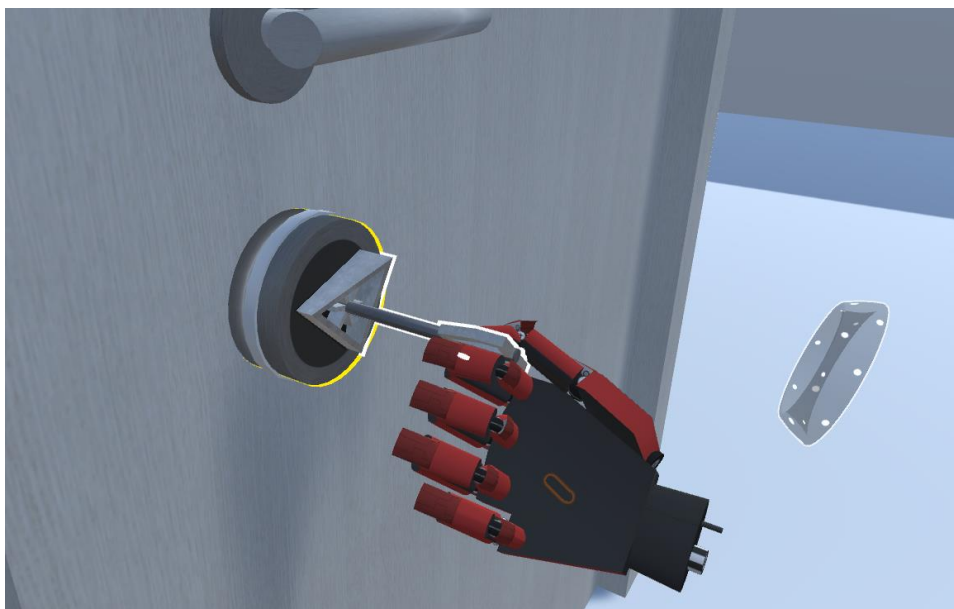


Ilustración 45: Interacción con cerradura

8.10 Interacción con Cajones

Para el juego, tanto con los cajones como para otras interacciones, necesito poder agarrar un objeto y moverlo en una línea recta dentro de unos límites, para ello he creado el Script GrabbableLinealMovement.

Este Script recibe como parámetros:

- La distancia máxima para la interacción.
- El vector que define la Línea de Movimiento.
- Los límites de desplazamiento.
- Objeto a Transladar.

Cuando un Controlador está haciendo Grab sobre el, dentro del bucle de ejecución para cada frame del juego, realiza:

- Calcula el desplazamiento realizado desde el punto de Grab hasta el Controller, solo tomando valores en la Línea de Movimiento.
- Si la distancia del objeto al Controller es mayor que la distancia máximaPermitida:
 - o Termina la Interacción.
- Si es menor:
 - o Si la posición inicial del objeto a mover y el desplazamiento está dentro de los límites:
 - Establece el movimiento en el objeto a transaladar.

En la visualización de los parámetros que recibe el script, la esfera azul es la máxima distancia de interacción, la línea amarilla la línea de Movimineto y los cubos amarillos los límites del movimiento:

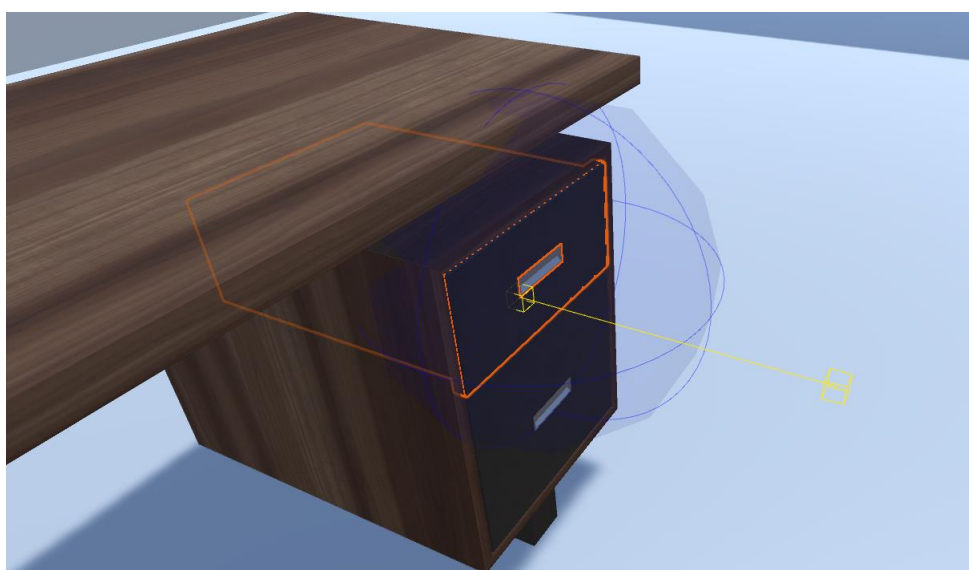


Ilustración 46: Visualización de GrabbableLinealMovement en cajón

Y de esta forma queda la interacción con uno de los cajones, pudiéndolos mover linealmente sin sobrepasar ninguno de los límites:

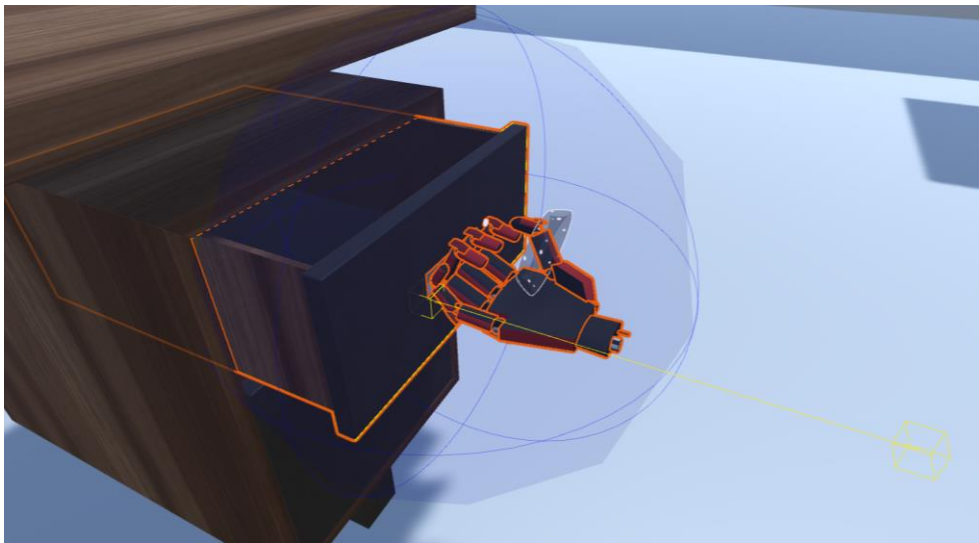


Ilustración 47: Interacción movimiento lineal con cajón

8.11 Diseño del Menú Inicial y del Primer Nivel

Una vez terminadas las interacciones principales y la forma de realizar estas, siguiendo el diseño del primer nivel especificado en el GDD, he creado el nivel del juego además de un menú principal. Este menú consta del mismo escritorio que encontramos al principio del juego con dos Interruptores de Palanca para elegir si comenzar el juego o salir de él con la intención, como especifico en el GDD de no usar interfáces clásicas para la realidad virtual, si no elementos físicos con los que interactuar:



Ilustración 48: Escena del Menú principal

Tanto para este menú y el primer nivel he necesitado varios modelos más y varias texturas, los cuales he conseguido de forma gratuita en la Asset Store de Unity, la web de modelos 3D cgtrader [14], la web de modelos 3D del programa de modelado Sketchup [15] 3Dwarehouse [16] y la web de texturas y materiales Textures.com [17].

Para modelar y texturizar la estructura del nivel (suelo, paredes y techo) además de algunos elementos simples necesarios, he usado la herramienta ProBuilder [18] para Unity, la cual añade funciones de modelado simples al editor del motor de juego.

Parte de estos modelos de internet ha sido necesario editarlos en Blender o Sketchup para adaptarlos a las necesidades del nivel, separar elementos de una misma malla, agrupar objetos o editar alguna malla para hacer practicalbes ciertas partes.



Ilustración 49: Escena completa del Primer Nivel

Siguiendo el flujo de juego diseñado el primer puzzle consiste en buscar una llave dentro de los libros colocados en las dos estanterías de la oficina, el libro se ha editado con ProBuilder para agrandarlo y hacer un hueco dentro de él, aparte de crear nuevos lomos para dar variedad a los libros.



Ilustración 50: Primera llave escondida en un libro hueco

8.12 Desarrollo del Puzle del Salón (Caja Fuerte con combinación numérica)

El segundo puzle del nivel consiste en una caja fuerte escondida dentro de la mesa principal del salón, la cual agarrando un mango que tiene por debajo puede abrirse por la mitad desplazando ambas mitades, para dicha interacción he usado el Script GrabbableLinealMovement mencionado anteriormente además de crear otro Script basado en este, el cual realiza un movimiento lineal determinado al hacer Grab sobre un grabable. Este Script sirve para mover la otra mitad de la mesa en dirección contraria al tirar de un lado de ella.

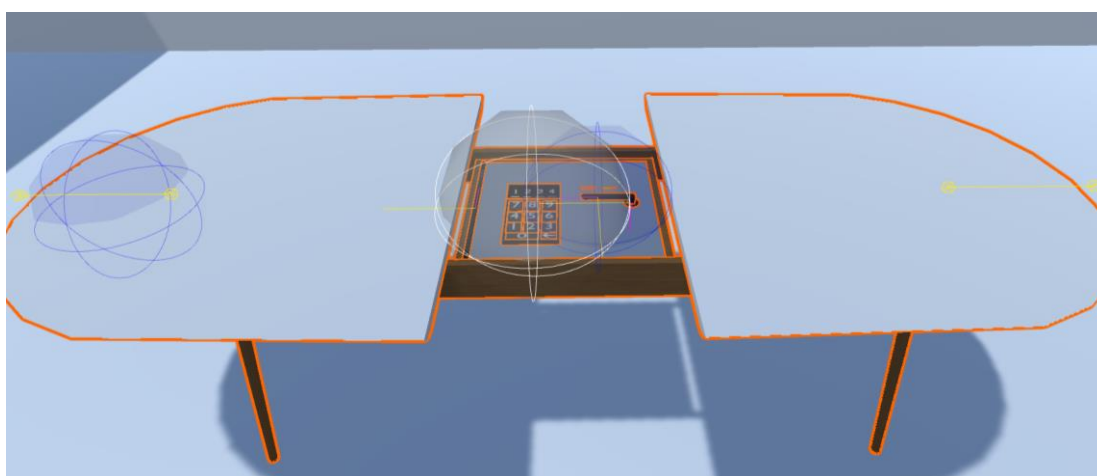


Ilustración 51: Elementos de la mesa con caja fuerte

La caja Fuerte está bloqueada por una contraseña numérica de 4 dígitos, para poder interactuar con la cerradura numérica, al acercarse una Mano al panel, esta activa la animación Spot y reduce su Collider al dedo índice para facilitar la interacción con los botones, si la mano sale de un radio concreto (esfera blanca en la imagen anterior) se devuelve a su estado normal.

Cada uno de los botones tiene un Script InteractableKey, el cual lanza un evento al ser interaccionado y finaliza la interacción, sin posicionamiento de la mano. Este evento puede ser configurado desde el inspector del editor para cada objeto, y llama al Script NumericScreen para comunicarle el valor del botón que ha sido pulsado.

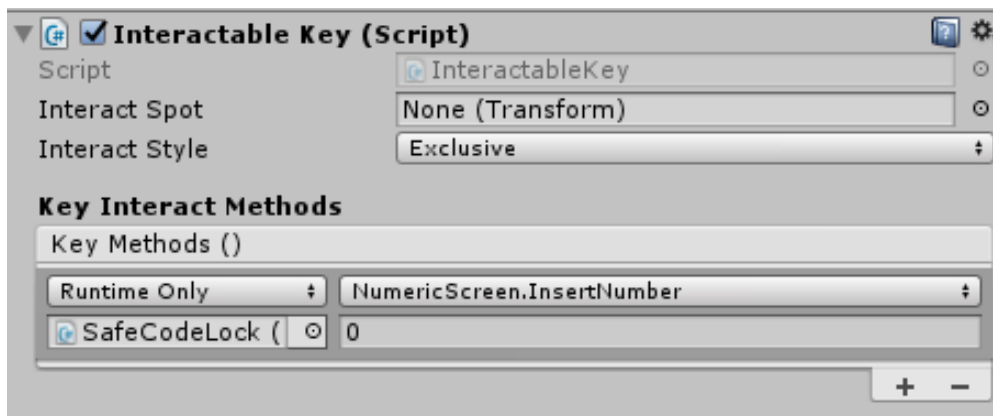


Ilustración 52: Asignación de valores en el Script InteractableKey en el inspector

Este Script NumericScreen se encarga de recibir los botones pulsados y añadir los números a la contraseña introducida (o eliminarlos si se pulsa el botón de borrar), además de comprobar si la contraseña introducida es correcta, en cuyo caso desbloquea la puerta de la caja fuerte y enciende la luz verde, si es incorrecta hace parpadear la luz roja para indicarlo.

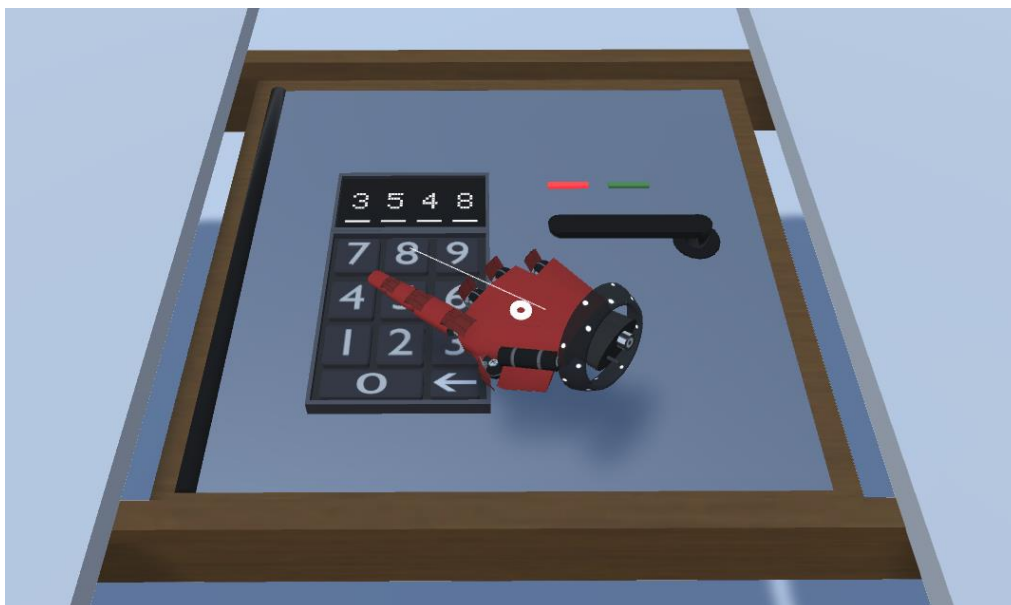


Ilustración 53: Interacción con el teclado de la caja fuerte

El contenido de la caja fuerte no debe de ser visualizado ni interactuado si esta no se ha desbloqueado, para asegurarme de ello, el Script ContentController recibe una serie de objetos los cuales desactiva su visualización y lógica al comenzar la ejecución de la escena y el cual es llamado para activar su contenido cuando la caja fuerte se desbloquea.

La contraseña necesaria para abrir la caja fuerte la encontramos en el horno de la cocina, para poder verla debemos empujar ambos botones del horno, los cuales tienen una variable Interactable de LinealMovement, y al pulsarse muestran la mitad de la contraseña cada uno con una luz de color en el interior del horno, al dejar de interactuar con ellos los botones vuelven a su posición inicial desactivando las luces del interior y la visualización de la contraseña.



Ilustración 54: Contraseña mostrada en el horno al pulsar ambos botones

8.13 Desarrollo del Puzle de la Librería (Tablet y Librerías)

En la librería dispondremos de cinco estanterías, cada una con números romanos del uno al cinco en el estante superior. Cada estantería tiene una escalera de mano montada sobre un raíl la cual podremos mover para ubicarla debajo de uno de los números romanos, estas escaleras se mueven haciendo uso del Script GrabbableLinealMovement. El número seleccionado se obtiene con la colisión del colider en el centro superior de las escaleras con el colider de los números, estos contienen un Script NumberKey el cual a que número corresponde y es leído por las escaleras al colisionar por el, comprueba si el número seleccionado es igual al número correcto para dicha escalera y lo comunica al Script LibraryCodeControler.



Ilustración 55: Estantería de la biblioteca, interacciones y colliders

El Script LibraryCodeControler es el encargado de recoger los estados de las escaleras, si están en el número correcto o no, y si todos los números son correctos, desplaza la estantería disitnta al resto para mostrar la puerta a la Sala Oculta.

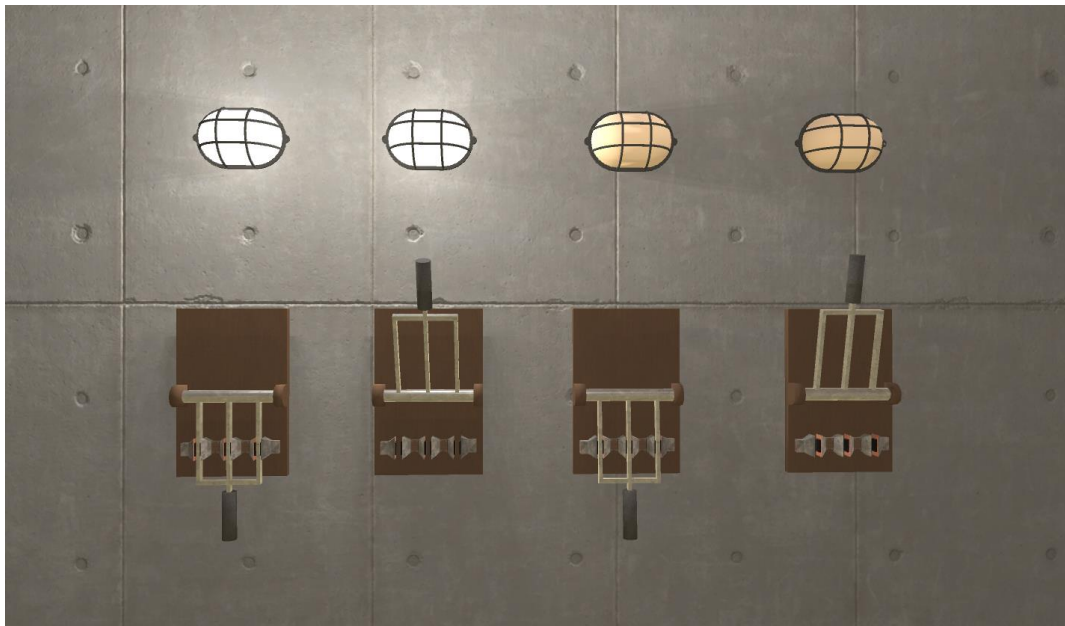
El código de las escaleras lo encontramos en la tablet de la oficina, si esta la llevamos al cargador inalámbrico el Script ChangeScreenOnCharge de la tablet cambiará la textura de la pantalla para mostrar la pista, si se vuelve a alejar del cargador vovlerá a la pantalla que muestra el icono de batería baja.



Ilustración 56: La tablet muestra la pista al acercarse al cargador

8.14 Desarrollo del Puzle de la Habitación Secreta (Interruptores y Luces)

Dentro de la habitación secreta está el puzle de los interruptores y luces, cada uno de los interruptores cambia el estado de algunas de las luces, llamando a su Script LightKey, el cual cambia su estado encendiendo/apagando una luz y activando/desactivando el componente Emission del material para que parezca que la luz sale del objeto.



LightKey registra el estado de las luces al Script LightPuzzleController el cual comprueba con un frame de retraso si todas las luces están encendidas o no. Este retraso se debe al hecho de que al intercambiar el estado de las luces secuencialmente puede ocurrir que una posición incorrecta de interruptores de positivo al estar todas las luces encendidas antes de empezar a apagar algunas de ellas. Si todas las luces están encendidas, abrirá una puerta secreta en el escritorio de la oficina mostrando la llave que abre la salida.

Igualmente, que ocurría con la mesa del Salón, para proteger el contenido de la puerta secreta del escritorio hago uso el Script ContentController.



Ilustración 57: Puerta secreta del escritorio de la oficina

Al desbloquear la puerta de la salida, la cámara hará una transición a pantalla en blanco y vuelve al menú principal.

8.15 Sonidos de la escena y Música

Para enriquecer el ambiente, enfatizar las acciones y proporcionarle al jugador información de lo que pasa, se han añadido los siguientes sonidos:

- Puerta:
 - Bloquear/Desbloquear cerradura de la puerta.
 - Intentar abrir la puerta estando bloqueada.
 - Girar pomo para abrir la puerta.
 - Puerta al cerrarse.
- Interruptor:
 - Activar Interruptor.
- Horno:
 - Activar/Desactivar botón.
 - Funcionamiento del horno al estar alguno de los botones pulsados.
- Caja Fuerte:
 - Pulsar un botón.
 - Introducir un código incorrecto.
 - Introducir un código correcto.

- Estantería de la Librería:
 - Estantería moviéndose al desbloquear el código de la librería.
- Puerta Secreta Escritorio:
 - Puerta secreta abriéndose al encender todas las luces.

Todos estos sonidos se han colocado cada uno en su fuente de sonido y permitido el posicionamiento del sonido en 3D, para dar información al jugador de la posición de la fuente de sonido.

En cuanto a la música, se ha hecho una selección de canciones con un sonido bajo y tranquilo de una lista de música Chill sin Copyright de Youtube [19], reproducidas en orden aleatorio, sin posicionamiento 3D, por el Script MusicController desde la misma cámara donde se encuentra el AudioListener.

8.16 Ayudas a los controles, Optimización y Corrección del movimiento

En los juegos de realidad virtual es importante mostrar de alguna forma cuales son los controler del juego, ya que los controladores con los que jugamos son distintos en cada plataforma y el jugador no considera aún ningún comportamiento predeterminado por parte de un botón.

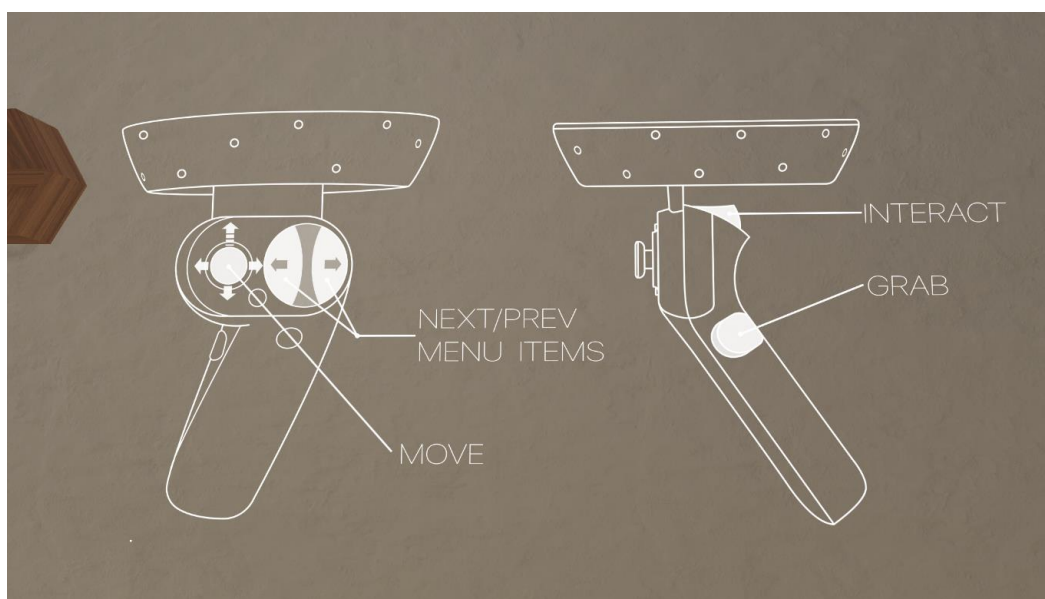


Ilustración 58: Panel explicativo de controles del juego

Además de una imagen explicativa en la pared de la primera sala del juego, se ha añadido en la propia mano una animación que indica que botón del Controller debemos pulsar para realizar la interacción con la que estamos haciendo contacto en ese momento.

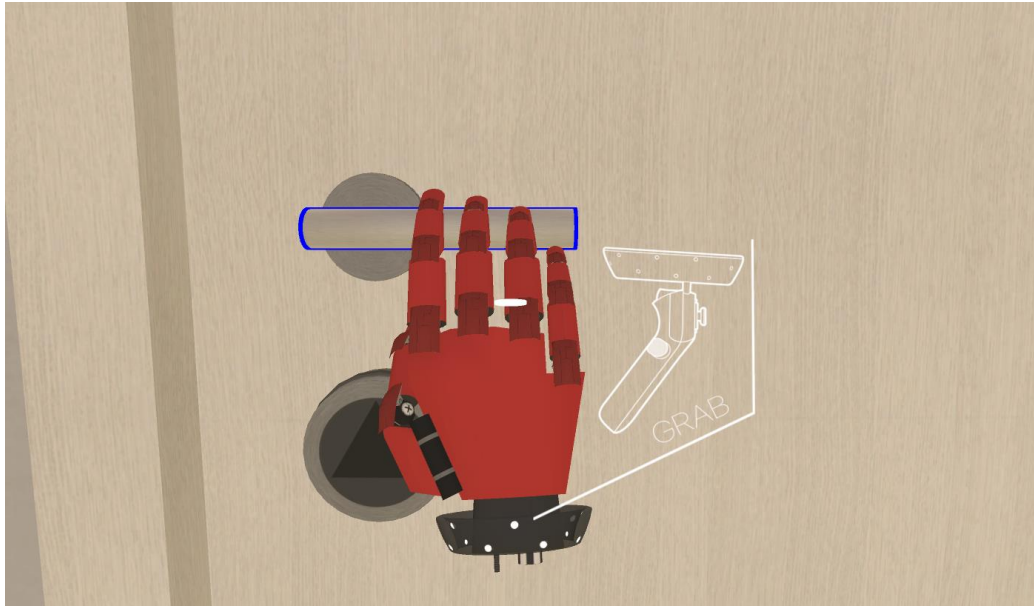


Ilustración 59: Ayuda interacción Grab

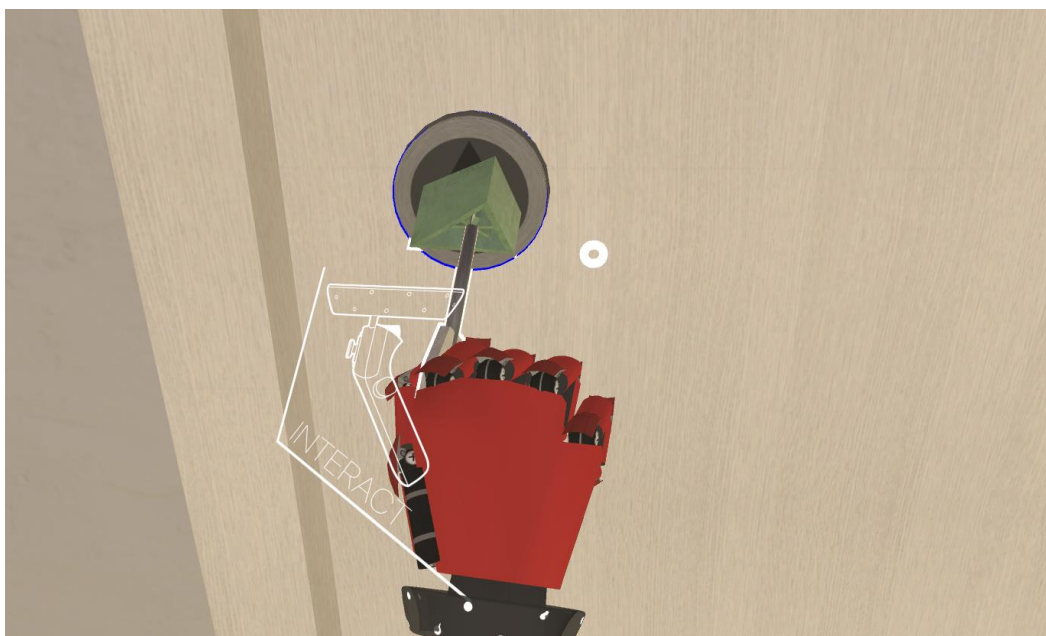


Ilustración 60: Ayuda interacción Interact

Por el hecho de usar modelos no creados específicamente para este juego, algunos de ellos que se repiten varias veces tienen una gran cantidad de polígonos en una escena pequeña, lo que acaba haciendo al juego pesado para ejecutarse, para que el motor gráfico solo dibuje los objetos que estamos viendo y no todos los que entran en el frustum de la cámara he calculado el Occlusion Culling en esta escena. Para ello debemos de tener objetos estáticos en la escena que sepamos que no se van a mover en ningún momento de la ejecución (paredes, suelo, techo y muebles estáticos), definir el tamaño mínimo de objeto que va a tapar lo que tenga detrás y de tamaño del agujero a través del cual vamos a poder ver lo que tenga detrás.

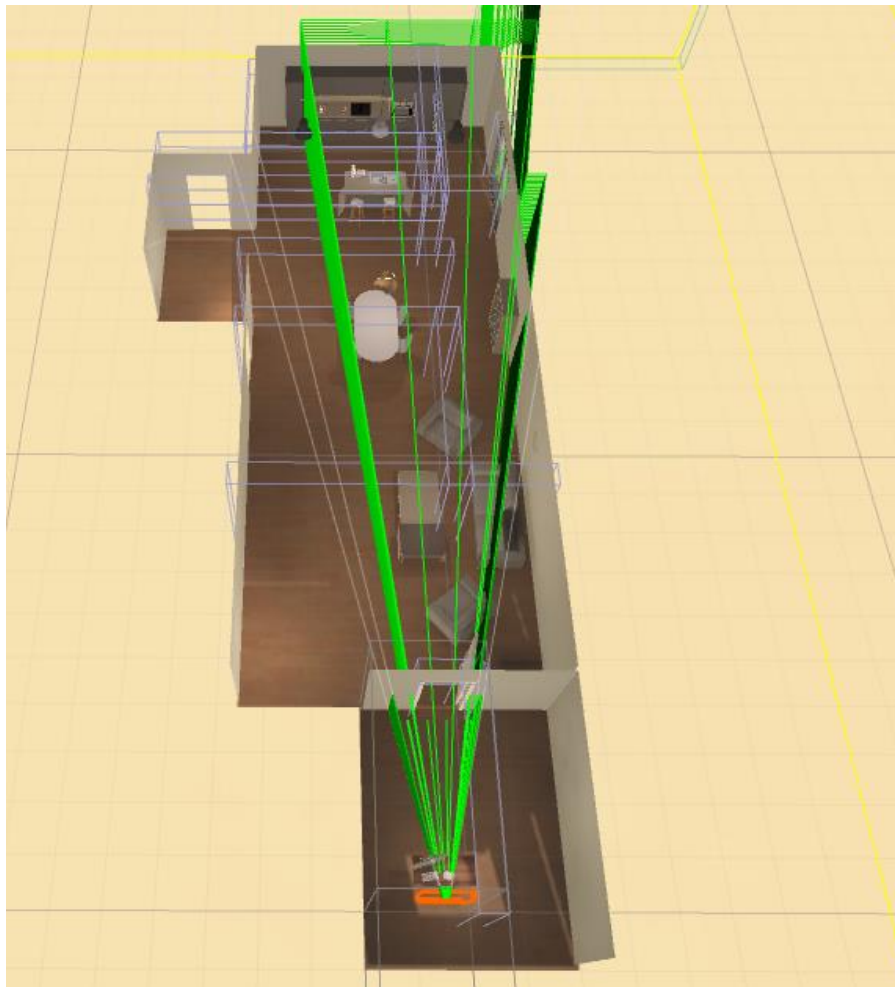


Ilustración 61: Objetos dibujados en la escena con el Occlusion Culling

En cuanto al movimiento que incorporaba la SDK de Mixed Reality presentaba el problema de poder atravesar todo al desplazarse hacia atrás con el Joystick, aunque no haya suelo, para ello he modificado la función de movimiento que implementa para comprobar si la posición destino del movimiento deja la cámara encima de suelo lanzando un RayCast hacia el suelo en dicha posición y comprobando las colisiones de este.

Además, tanto para evitar este movimiento hacia atrás como para las teletransportaciones atravesando paredes y llegando a lugares no desbloqueados, las habitaciones que no han sido desbloqueadas desactivan todos sus objetos (o al menos el suelo) y únicamente son activados al desbloquear dicha habitación. Aunque no resuelva el hecho de poder atravesar paredes, solo lo podremos hacer entre salas ya desbloqueadas y tampoco podremos ver lo que hay en una sala bloqueada, aunque nos “asomemos”.

8.17 Imágenes con la historia mostradas en la escena

A lo largo de la escena encontraremos imágenes de noticias que desvelarán parte de la historia, esta imagen se encuentra en el ordenador de la oficina, la cual anuncia un nuevo tipo de tecnología VR muy innovadora:



Ilustración 62: Primera imagen de la historia en la Oficina

En la televisión del salón, encontramos una noticia sobre sucesos sobre adictos a videojuegos hospitalizados:



Ilustración 63: Segunda imagen de la historia en el Salón

En el ordenador de la librería, muestra que estos sucesos están ocurriendo frecuentemente y se abre una investigación oficial:



NET NEWS

Exponential Increase Hospitalization Cases Related With Large Video Game Sessions

Thousands of people have been hospitalized and even have lost their lives this year and this numbers still increasing. All of them in the same circumstances, they were alone in their houses for several days without doing any thing except playing games. Inanition and sleep deprivation produce serious physical and mental injures to this gamers.

An investigation is openned to determine the cause of this continous increase, which points that is another dangerous viral internet challenge.

Ilustración 64: Tercera imagen de la historia en la Librería

En un portátil en la Habitación Secreta, desvela la relación entre la tecnología VR y los sucesos y que el jugador se encuentra igualmente atrapado:



NET NEWS

Neural VR Technology Behind Gamers Injuries A Bug Make You Get Trapped Inside Games



Official investigation and Neural VR Company have reported that a bug in this technology get some people trapped inside the game being impossible to ESCAPE and causing inanition and sleep deprivation.

Ilustración 65:Cuarta imagen de la historia en la Sala Secreta

9 Conclusiones y Trabajo futuro

Con el trabajo realizado durante este proyecto he trabajado con un dispositivo distinto a lo que estoy acostumbrado a trabajar en entornos gráficos, el cual ofrece mucha más libertad al jugador y plantea a los desarrolladores nuevos paradigmas de interacción y elementos más resistentes a las diferentes acciones del jugador.

La Realidad virtual es una plataforma con mucho futuro que plantea una nueva forma de jugar y de hacer juegos que ofrece nuevos retos a los desarrolladores, mi experiencia tras trabajar con ellas e investigar en el diseño de un tipo de juego tan concreto como son las Escape Rooms ha sido muy enriquecedora con la que no dudaría en continuar su desarrollo.

Como trabajo futuro de este proyecto, necesita seguir diseñando y desarrollando todos los niveles y puzles del juego completo, además de refinar el movimiento del jugador para que este no pueda traspasar los muros con su movimiento físico, además de hacer la dirección de teletransporte parabólico como suele usarse en juegos para estas plataformas. También sería necesario que la SDK utilizada avance un poco, ya que es muy básica aún, por ejemplo, aún no permite hacer que los Controllers vibren lo cual es bastante interesante como retroalimentación para el jugador al realizar alguna interacción.

En cuanto al estilo del juego sería necesario tener modelos propios pensados para este juego, los descargados de internet a parte de tener su propio estilo suelen tener gran cantidad de polígonos lo que ha resultado en una escena muy pesada y difícil de mover para las tarjetas gráficas.

10 Bibliografía

- [1] History and origin of Escape Games - Lock Academy:
<https://lockacademy.com/en/history-and-origin-of-escape-games/>
- [2] Facts about escape rooms around the world - escape rooms: <https://www.escape-rooms.com/articles/facts-about-escape-rooms-around-the-world>
- [3] Aclaración por la Fundación del Español Urgente - librojuegos.org:
<http://librojuegos.org/2015/01/aclaracion-por-la-fundacion-del-espanol-urgente-fundeu/>
- [4] Adam Clare M.Ed (2015). *Escape The Game: How to Make Puzzle and EscapeRooms*.
- [5] Blueprint for crafting your first Escape Room – Lock Paper Scissors:
<https://lockpaperscissors.co/craft-1st-escape-room>
- [6] Top 11 puzzle ideas for Escape Rooms - Escape Room Tips:
<https://escaperoomtips.com/design/escape-room-puzzle-ideas>
- [7] What is Virtual Reality? - Virtual Reality Society: <https://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html>
- [8] Unity: <https://unity3d.com/es>
- [9] Mixed Reality - Install the Tools: <https://docs.microsoft.com/es-es/windows/mixed-reality/install-the-tools>
- [10] MixedRealityToolkit - Unity: <https://github.com/Microsoft/MixedRealityToolkit-Unity>
- [11] Blender: <https://www.blender.org/>
- [12] Quick Outline - Unity Asset Store:
<https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488>
- [13] Mixed Reality Motion Controllers Unity Example - Centro de desarrollo de Windows: <https://docs.microsoft.com/es-es/windows/mixed-reality/mixed-reality-213>
- [14] 3D Models - cgtrader: <https://www.cgtrader.com/>
- [15] Sketchup: <https://www.sketchup.com/es>
- [16] 3D Models - 3Dwarehouse: <https://3dwarehouse.sketchup.com/>
- [17] Textures.com: <https://www.textures.com/>
- [18] ProBuilder - Unity:
<https://unity3d.com/es/unity/features/worldbuilding/probuilder>
- [19] Copyright Free Music - ChilledCow Youtube:
<https://www.youtube.com/watch?v=HTNXz6WspCU&list=PL6NdkXsPL07IbXr-jMNJk0WUFej0-XDzY>

11 Apéndices

11.1 Enlace a Código Fuente del Proyecto

<https://drive.google.com/open?id=1EpxQflfTzTWzcFL4vm-2Q8JFaHLkaBL8>



Game Desing Document

Nombre del Juego:

In-Game

Alejandro Guerrero Martínez

Curso 2017/2018

Versión:

1.0.4

Control de Versiones

Versión	Fecha	Cambios
1.0.0	4/08/2018	Rasgos generales del Juego
1.0.1	11/08/2018	Mecánicas y reglas del Juego
1.0.2	16/08/2018	Diseño del nivel del Juego
1.0.3	22/06/2018	Historia del Juego
1.0.4	29/06/2018	Experiencia,Limitación e Información

Tabla de contenido

CONTROL DE VERSIONES	2
1 INFORMACIÓN GENERAL	5
1.1 RESUMEN DEL JUEGO	5
1.2 OBJETIVOS A ALCANZAR POR EL JUEGO	5
1.3 JUSTIFICACIÓN DEL JUEGO	5
1.4 CORE GAMEPLAY	6
1.5 CARACTERÍSTICAS DEL JUEGO	6
1.5.1 GÉNERO	6
1.5.2 NÚMERO DE JUGADORES	6
1.5.3 PLATAFORMAS DE DESTINO	6
1.5.4 ESTÉTICA Y ARTE DEL JUEGO	7
1.5.5 RESUMEN DE HISTORIA	7
1.6 CARACTERÍSTICAS DEL JUGADOR	7
2 MECÁNICAS	7
2.1. ELEMENTOS JUEGO	7
2.2 REGLAS DE INTERACCIÓN	9
2.3 ELEMENTOS DE REGISTRO Y PROGRESO	10
3 DINÁMICA	10
3.1 DETALLES DEL JUEGO EN TEMÁTICA	10
3.2 HISTORIA DETALLADA	11
3.3 DESCRIPCIÓN DE MISIONES, NIVELES O CAPÍTULOS	11
3.3.1 PRIMER NIVEL	12
3.3.1.1 Objetivos	13
3.3.1.2 Flujo y Puzles	13
3.3.1.3 Historia	15
3.4 INTERFAZ DEL JUEGO	17
3.5 APRENDIZAJE DEL JUEGO	17

4	EXPERIENCIA	17
4.1	JUGABILIDAD INTRÍNSECA	17
4.2	JUGABILIDAD MECÁNICA	17
4.3	JUGABILIDAD INTERACTIVA	18
4.4	JUGABILIDAD ARTÍSTICA	18
4.5	JUGABILIDAD INTRAPERSONAL	18
5	LIMITACIONES	18
5.1	LIMITACIONES TÉCNICAS	18
6	INFORMACIÓN DEL DOCUMENTO	19
6.1	DEFINICIÓN, ACRÓNIMOS Y ABREVIATURAS	19

1 Información general

In-Game es un juego de tipo Escape Room, para Dispositivos de Realidad Virtual (VR) con controladores (Motion Controllers), ambientado en el presente o un futuro muy próximo (sin grandes cambios en los objetos cotidianos y la tecnología), el jugador deberá de encontrar objetos y pistas para resolver los distintos puzles propuestos en el escenario y conseguir salir de las habitaciones.

En este juego se tratará de llevar la experiencia de una Escape Room real a la inmersión que nos proporciona los dispositivos VR. Haciendo que la interacción con los elementos de la escena sean lo más realistas posibles (tendrás que girar el pomo de una puerta para poder abrirla, introducir y girar la llave, girar interruptores ...)

El juego transcurrirá a lo largo de distintos escenarios cíclicos (el mismo escenario, pero con distintos puzles y modificaciones en cada iteración hasta ser algo totalmente irreal) en los cuales tendremos que escapar de la casa, durante estos ciclos encontraremos elementos en las habitaciones que nos contará la historia del juego (revistas, ordenadores, diálogos en la radio ...).

1.1 Resumen del juego

Juego de estilo Escape Room con tecnología de VR, el jugador partirá de una habitación en la que está encerrado. Tendrá que buscar elementos y pistas que le ayuden a salir resolviendo varios puzles de las sucesivas habitaciones y completar los escenarios que se planteen.

En lo referente a la historia, el “personaje” se haya en un bug de un juego del cual no puede escapar, por lo que el escenario se repite cíclicamente con distintas variaciones y puzles, hasta que el jugador consiga resolver todos los puzles y escapar completamente del juego.

1.2 Objetivos a alcanzar por el juego

El objetivo del jugador será salir de cada escenario, conocer por qué esta encerrado en el juego y por último conseguir escapar completamente del juego.

1.3 Justificación del Juego

Frente al resto de Escape Room de Realidad Virtual disponibles, In-Game presenta una historia que transcenderá a lo largo del juego, y no solo se conoce al principio como suele ser, por lo que además pretende ofrecer más tiempo de juego, consiguiendo objetivos y cambiando escenarios mientras mantiene la consistencia de la historia, para que el jugador no se frustre o aburra por un juego largo en el que parece no avanzar.

Además frente a otros Juegos del género para VR, In-Game será una experiencia más inmersiva por su interacción más realista con los elementos del juego.

1.4 Core gameplay

El jugador podrá realizar las siguientes acciones:

- Coger objetos, moverlos y tirarlos.
- Agarrar elementos móviles y moverlos/girarlos con el movimiento de las manos.
- Interaccionar con elementos, algunos con interacción estática y otros interaccionarán con movimientos/giros de las manos.
- Recoger objetos clave y guardarlos en el inventario.
- Seleccionar objetos clave recogidos del inventario y equiparlo en la mano.
- Interaccionar con el elemento clave equipado en la mano y elementos de la escena.

Además, habrá elementos que reaccionarán a otros elementos de la escena, al entrar en contacto, al introducir una clave, o al posicionar correctamente uno de los elementos.

1.5 Características del juego

1.5.1 Género

El juego pertenece a los géneros Escape Game y Aventura, ya que el género en si Escape Game define las mecánicas principales del juego y engloba al género puzzle del que también trate este juego. Además, por la progresión de la historia a o largo del juego se considera aventura.

1.5.2 Número de jugadores

El juego será para un solo jugador, sin modo online.

En el futuro podría añadirse un modo cooperativo online para dos, tres o cuatro jugadores, ya que los dispositivos VR consumen demasiados recursos para tener dos conectados al mismo equipo.

1.5.3 Plataformas de destino

La plataforma será PC para el sistema operativo Windows 10 y será necesario uno de los dispositivos VR que funcionan bajo la tecnología Windows Mixed Reality.

1.5.4 Estética y arte del juego

El juego transcurrirá en interiores de una casa de la actualidad/futuro cercano, con una estética realística.

1.5.5 Resumen de Historia

El “personaje” se haya en un bug de un juego del cual no puede escapar, por lo que el escenario se repite cíclicamente con distintas variaciones y puzles, hasta que el jugador consiga resolver todos los puzles y escapar completamente del juego.

1.6 Características del jugador

Los jugadores objetivo de este juego son los de tipo Exploradores, aquellos que les gusta explorar lugares y descubrir zonas nuevas, por lo que se verán recompensados al cumplir los objetivos del juego.

2 Mecánicas

2.1. Elementos Juego

- **Jugador:** el jugador estará totalmente inmerso en el mundo virtual, hubicando la camara en la cabeza al igual que en la realiad. Los controladores (MotionControllers) serán sustituidos por un modelo de unas manos las cuales se moverán siguiendo las acciones que puedan realizar o las que estén realizando.
- **Escenarios:** los escenarios contarán con distintas habitaciones a las que no se podrá acceder a menos que desbloqueemos las puertas que las separan. Cada escenario mantendrá la estructura principal, al menos la habitación inicial, el resto irá modificandose a cada iteracción del juego y encontraremos puzles distintos cada vez para no resultar repetitivo.

Dentro de los escenarios encontraremos los siguientes elementos:

- **Elementos Estáticos:** aquellos elementos de la escena con los que no podrémos interaccionar y serán méramente decorativos (sillas, armarios, jarrones ...), o servirán de obstaculo para impedirnos pasar (paredes, estanterías ...).

- **Objetos “Agarrables”:** objetos de la escena que podemos agarrar con nuestras manos, lanzarlos y llevarlos de un lado a otro. Podrán interaccionar con otros elementos de la escena por contacto/proximidad o colocación concreta. A este tipo de elementos pertenecen:
 - **Elementos Decorativos Pequeños** (Jarrones, Platos, Libros, Teclado, Touchpad): elementos de la escena que no interactúan con otros elementos. Aunque podrán contener otros elementos en su interior, como el libro hueco para esconder una llave dentro.
 - **Tablet:** una tablet que marca en su pantalla que no tiene batería. Al acercarla al elemento estático Cargador Inalámbrico encenderá su pantalla mostrando el contenido.
- **Elementos “Agarrables”:** partes de un objeto que podremos agarrar para girarlo o moverlo. A este tipo de elementos pertenecen:
 - **Puerta:** agarrando el pomo el jugador lo girará al igual que una puerta real y posteriormente empujará/tirá del pomo para mover la hoja de la puerta la cual rotará sobre sus bisagras.
 - **Cajón:** agarrando el mango podremos sacar y meter el cajón en la cajonera.
 - **Escaleras Correderas:** al agarrar las escaleras correderas podremos moverlas de un lado a otro siguiendo sus railes.
 - **Mesa Desplegable:** al agarrar el mango de la mesa desplegable las dos partes que conforman la mesa se separarán o juntarán al moverlas.
 - **Interruptor de Palanca:** agarrando el mango de la palanca podremos moverlo para que gire sobre su eje y encender o apagar el interruptor.
- **Elementos Interaccionables:** partes de un objeto con los que podremos realizar una interacción que puede conllevar o no, un movimiento o rotación. A este tipo pertenecen:

- **Cerradura Puerta de Llave:** con la llave adecuada en la mano podremos interaccionar con la cerradura, la cual giraremos para desbloquear o bloquear la puerta.
- **Botón Pulsador:** interaccionando con el botón podremos empujarlo para activarlo.
- **Botón Pequeño:** al interactuar sobre él se activará la función del botón.
- **Objetos de Inventario:** parecidos a los objetos agarrables, pero estos al ser agarrados desaparecerán para irse a nuestro inventario, luego pueden ser equipados para posicionarse en la mano y poder observarlos o interaccionar con ellos. A este tipo de elementos pertenecen:
 - **Llave:** llave que al equiparse nos permitirá interaccionar con las cerraduras correspondientes.

2.2 Reglas de interacción

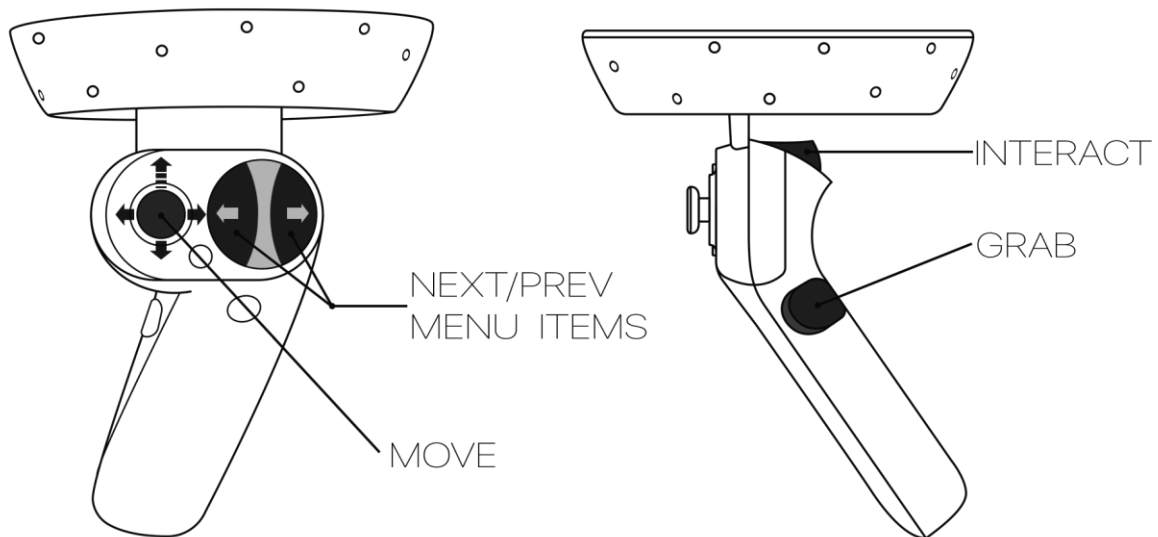
El jugador interaccionará con su entorno, pudiéndose mover tanto físicamente el dentro de la habitación real, cuyo desplazamiento será trasladado al juego, como ubicarse dentro de la escena virtual, haciendo uso del Joystick de los controladores, siempre teniendo en cuenta que cualquier desplazamiento nunca nos puede situar fuera del suelo de la habitación:

- Movimiento hacia Delante: teletransporta al jugador al punto al que apunte el controlador en el momento de soltar el Joystick.
- Movimiento hacia Atrás: Desplazará al jugador cierta distancia hacia atrás.
- Movimiento Horizontal: girará al jugador en la dirección del Joystick.

A la hora de interaccionar con los Objetos y Elementos Agarrables/Interaccionables y de Inventario, al acercarse el jugador las manos a un objeto de este tipo, este indicará que se puede interaccionar con él resaltándose con un contorno de color, al mismo tiempo en el controlador aparecerá una instrucción con el botón que podemos pulsar para interaccionar con él, en el caso de agarrarlo el botón "Gasp" y de interacciones el botón "Select".

Para acceder a los elementos del inventario, pulsaremos el "Touchpad" en un lado u otro para seleccionar el siguiente o anterior. Este elemento del inventario aparecerá en la mano desde la que hayamos accedido al inventario, si se accede al inventario desde otra mano se intercambiará el objeto seleccionado de mano.

Al igual que necesitaremos tener en la mano algún objeto del inventario para poder interaccionar con ciertos elementos de la escena, si tenemos el objeto en la mano no podremos interaccionar con el resto de elementos de la escena con esa mano hasta que no desequipemos el objeto.



2.3 Elementos de registro y progreso

El juego se autoguardará entre cada ciclo y en cada uno de estos podremos observar el tiempo que llevamos en el en relojes situados en habitaciones.

3 Dinámica

3.1 Detalles del Juego en temática

En nuestro entorno debemos encontrarnos elementos conocidos para casi todos con un estilo realista pero sin necesidad de tener muchos detalles, aunque estemos atrapados en la casa los espacios serán amplios para que el jugador pueda moverse y no se sienta incomodo.

En cuanto a la música se buscará una atmosfera relajada con un sonido bajo, con varios temas que no la haga repetitiva, debe de dejar pensar al jugador y no hacerle sentir que siempre se repite, dejando más presencia con un sonido más alto a los sonidos producidos por las acciones del jugador.

3.2 Historia detallada

Una empresa de periféricos para jugar ha desarrollado una tecnología con interfaz neuronal que nos permite una inmersión total en los juegos prácticamente trasladando a los jugadores dentro de ellos, es una tecnología de gran éxito que ha conseguido copar gran parte del mercado de jugadores. Al mismo los casos de personas hospitalizadas por pasarse hora jugando sin comer ni dormir empieza a crecer alarmantemente, hasta ser un problema de escala internacional. Tras varias sospechas e investigaciones oficiales encargadas por varios gobiernos la empresa creadora de la tecnología VR acaba admitiendo un BUG aparentemente aleatorio en sus sistemas que hace que el jugador quede atrapado dentro del juego sin poder desconectarse de la interfaz neuronal, lo cual podía resultar mortal en personas que pasan mucho tiempo solas y queden atrapadas sin que nadie las desconecte. Inmediatamente se procedió a la retirada del mercado y de las ya vendidas, por desgracia para nuestro personaje no conoce las noticias y ha quedado atrapado dentro de su dispositivo VR.

Entraremos en el juego sin conocer nada, desde un menú con el mismo escritorio que encontraremos al empezar el juego, a partir de aquí iremos conociendo las noticias del suceso de forma lineal en la que descubriremos que esta conectado a este dispositivo y aparentemente condenado a repetir el mismo juego sin poder escapar a la realidad, sin embargo, a cada iteración el “juego” en el que estamos atrapado será mas inestable e irreal hasta que consigamos desbloquear el dispositivo y escapar de él.

3.3 Descripción de misiones, niveles o capítulos

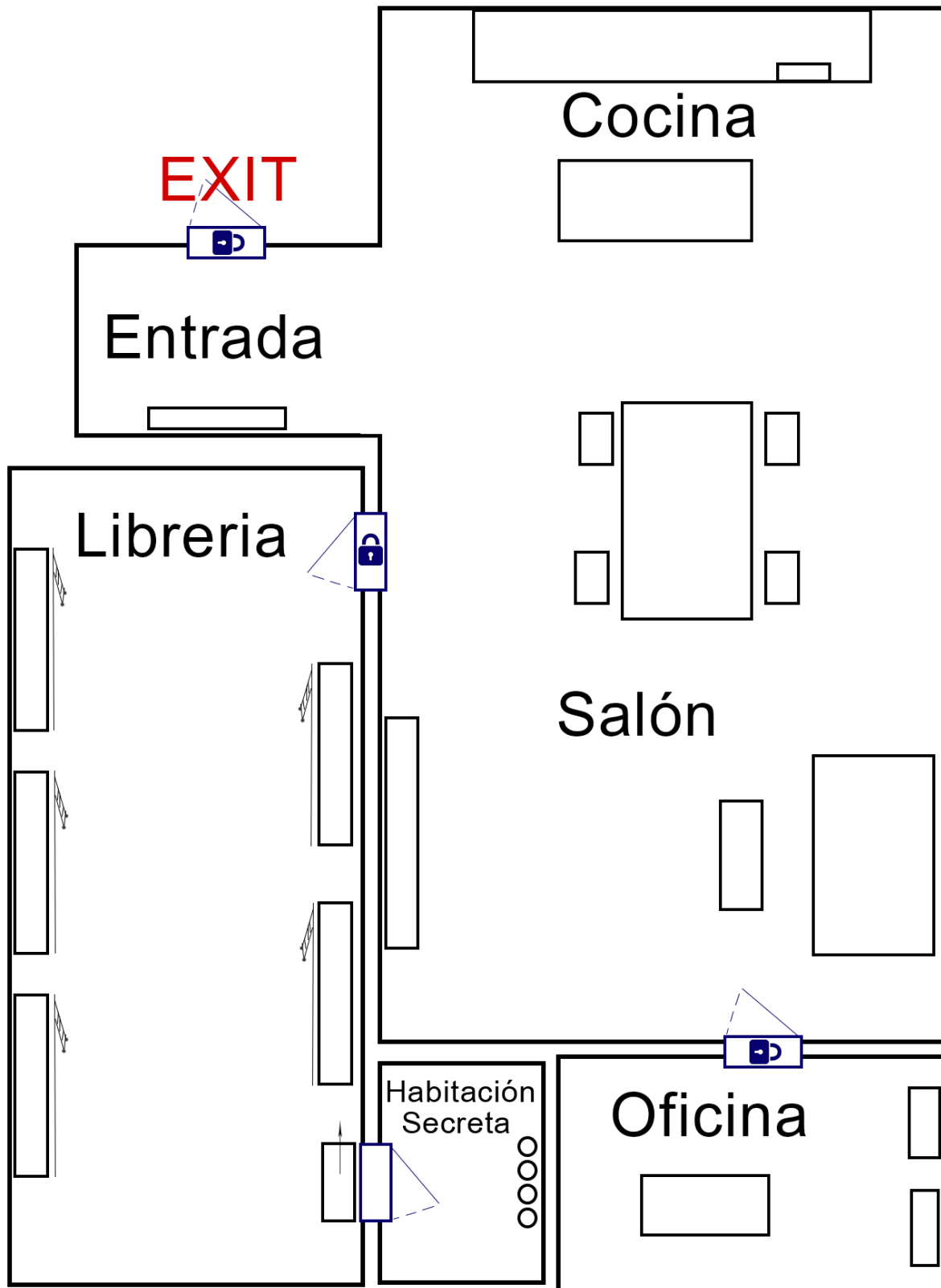
La historia será lineal dándose a conocer en elementos de la escena, noticias en el periodico, en ordenadores, videos en la televisión audios, en la radio.

Los niveles serán cíclicos siempre cambiando todos los puzles y con algunos cambios en las habitaciones y los elementos que hay en ellas.

En esta versión del GDD solamente desarrollaré el menú y el primer nivel:

3.3.1 Primer Nivel

Diseño del escenario del primer nivel, consta de 4 habitaciones, con el Salón dividido en 3 zonas:



3.3.1.1 Objetivos

Nuestro objetivo en este nivel será alcanzar la salida.

Como objetivo adicional el jugador podrá descubrir la historia leyendo las noticias en las pantallas.

3.3.1.2 Flujo y Puzzles

- ➔ En el primer nivel empezaremos en la Oficina, con la intención de salir de esta habitación y pasar al Salón, para ello el jugador debe de buscar la llave en la habitación que contará con un Escritorio con cajones y dos estanterías de libros. La llave se encuentra dentro de uno de los libros es cual está hueco y en el lomo se puede leer “The KEY to Success” para dar una pista al jugador sobre el libro correcto.
- ➔ Desde el Salón pasaremos a la Librería para ello el jugador debe:
 - Abrir la Mesa Desplegable del centro del salón, la cual nos mostrar una caja fuerte con código numérico de cuatro dígitos.
 - Ir a la cocina e interaccionar con los dos botones del horno, que nos mostrará en el cristal de este los 4 dígitos de la caja fuerte: **3849**.
 - Abrir la caja fuerte introduciendo los dígitos correctos. En el interior se encontrará la llave de la Librería.
- ➔ En la librería pasaremos a la Habitación Oculta, de la cual el jugador no conocerá la ubicación de la puerta al estar tapada con una estantería. Para ello el jugador deberá:
 - Encontrar en Cargador Inalambrico en la Librería.
 - Recoger la Tablet del cajón del Escritorio de la Oficina y llevarla al cargador. Al acercarse esta mostrará un dibujo de las estanterías con escaleras y un número del 1 al 5 al lado de cada una:



- El jugador debera colocar las escaleras de las estanterías de forma que estas estén debajo del número correspondiente que representaba en la tablet. Estos números estarán en la parte de arriba de las estanterías escritas en números romanos.
 - Al colocarlas todas correctamente la estantería que tapa la Puerta a la Habitación Secreta se desplazará dejandola ver.
- ➔ En la Habitación Secreta hay un puzzle que consta de cuatro luces y cuatro interruptores, el jugador activando y desactivando los interruptores debe de averiguar a que luces afecta cada interruptor y encontrar una combinación para que todas las luces queden encendidas. Enumerando los interruptores y las luces de izquierda a derecha, seguirán la siguiente lógica:
- Primer Interruptor: intercambiará el estado de las bombillas 2º y 3º.
 - Segundo Interruptor: intercambiará el estado de las bombillas 1º, 2º y 3º.
 - Tercer Interruptor: intercambiará el estado de las bombillas 1º y 3º.
 - Cuarto Interruptor: Intercambiará el estado de las bombillas 2º, 3º y 4º.

De esta forma la combinación válida de interruptores para encender todas las luces, es tener activados el 1º, 2º y 4º, y el 3º desactivado.

Este puzzle abrirá un compartimento secreto en la parte posterior del Escritorio de la Oficina, en la que encontraremos la última llave que abrirá la puerta de la Salida.

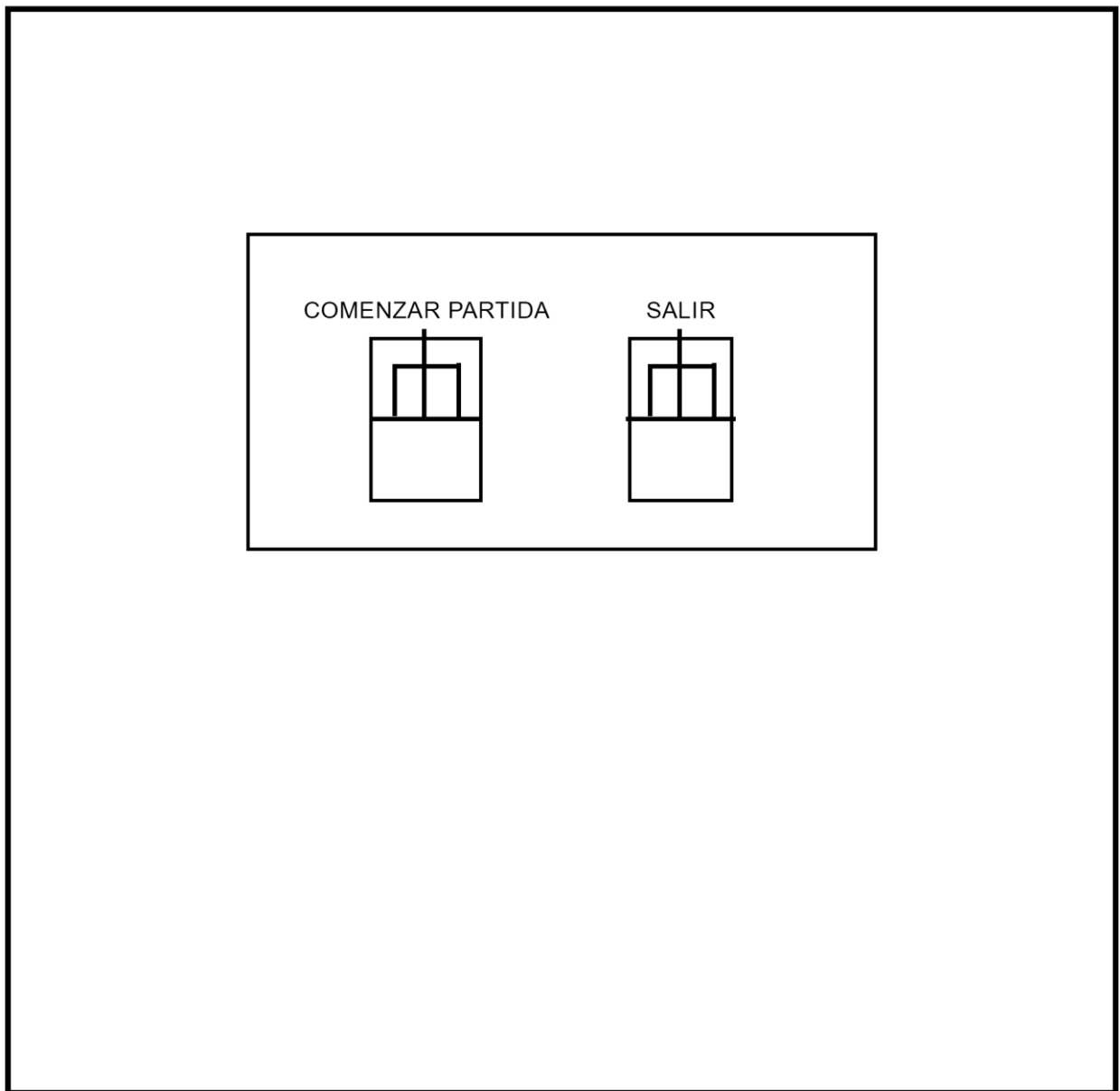
3.3.1.3 Historia

En este nivel el jugador comenzará sin conocer nada de la historia y con el flujo del nivel podrá descubrirla:

- ➔ En La Oficina encima del escritorio un monitor mostrará la noticia de la creación de una nueva tecnología VR con Interfaz Neuronal que revolucionará la industria de los videojuegos.
- ➔ En el Salón, encontraremos en la televisión una notica sobre un caso de un joven hospitalizado por pasar demasiadas horas seguidas jugando.
- ➔ En la Librería, podremos leer en otro monitor el problema que suponen estos casos de gente que no para de jugar para comer o dormir y que ha comenzado una investigación para ello.
- ➔ En la Habitación Secreta conoceremos la verdad en un portátil, con el titular enunciando que esta tecnología es peligrosa y podemos quedarnos atrapados sin poder salir.

3.3.2 Meú Principal

En el menú principal encontraremos la misma mesa del inicio del primer nivel, en la cual dispondremos de dos interruptores para inicial el juego o salir de el. Esta escena tendra una pequeña porción de suelo en la que nos podremos desplazar y estará iluminado el centro de esta donde se encuentra la mesa y empieza el jugador.



3.4 Interfaz del juego

En este juego se evitará el uso de un Canvas con las interfaces clásicas, todos los elementos y menús estarán representados por elementos físicos en el escenario que podamos tocar e interaccionar con los controladores.

Así pues, en el menú principal constará de dos Palancas señalizadas para iniciar el juego y salir.

Los elementos del inventario, cuando intercambiemos entre ellos en el juego, irán apareciendo en orden y en el sentido que el jugador indique en el lugar de la mano y el que tengamos resaltado, tras un tiempo sin cambiar de objeto en el inventario, se equipará en la mano correspondiente.

3.5 Aprendizaje del juego

Tanto en el menú inicial como en la primera habitación, se mostrará una imagen con todos los controles que puede realizar en el juego.

Además, en cuanto se pueda realizar algún Agarre/Interacción con algún Controlador aparecerá al lado de este el botón que debemos de pulsar para realizar la acción.

La mayor dificultad de aprendizaje en este Juego esta al inicio de este y si no hemos jugado a más juegos con los mismos controladores, pues al no visualizar los Controladores dentro del juego, un jugador totalmente novato no conoce la posición exacta de los botones.

4 Experiencia

4.1 Jugabilidad Intrínseca

Los puzles del juego deben de ser un desafío para el jugador, los cuales irán creciendo en dificultad conforme avance en el juego. A resolver estos puzles se recompensa con nuevas zonas desbloqueadas, ítems de inventario, pistas para otro puzle y/o algún elemento nuevo de la historia.

4.2 Jugabilidad Mecánica

En In-Game es muy importante la interacción con el mundo virtual y que este responda con fidelidad, por lo que es necesario que las colisiones respondan correctamente, y sobre todo en el caso de los objetos claves, no atraviesen elementos de la escena y puedan desaparecer.

Además, las interacciones de mundo virtual deben de acompañarse con sonidos que le correspondan y sean fieles a la realidad para aumentar considerablemente la inmersión del jugador. Como son los sonidos al desbloquear una cerradura, girar el pomo y abrir una puerta.

4.3 Jugabilidad Interactiva

Como ya se ha indicado es importante en este juego que las interacciones con elementos del menú no sean con las interfaces clásicas en 2D, si no con elementos del propio escenario con los que interaccionemos.

4.4 Jugabilidad Artística

El arte de juego no debe de ser demasiado sobrecargado de detalles, con objetos claros y bien iluminados, tanto los modelos que nos rodean como la música debe dejar al jugador pensar y centrarse en el puzle, excepto en momento que se busque crear tensión, lo cual se hará con una música más alta, con un tempo más rápido y espacios cerrados pequeños.

4.5 Jugabilidad Intrapersonal

Se busca que el Jugador se sienta relajado y tranquilo por los escenarios que le rodea, pero a medida que avanza el juego se busca generar cierta angustia por la idea de no poder salir de esta serie de ciclos e incluso aborrecer la habitación inicial, su aparente tranquilidad y belleza.

5 Limitaciones

5.1 Limitaciones técnicas

Las limitaciones técnicas de In-Game derivan de la tecnología necesaria para poder jugarlo, el quipo de prestaciones medias-altas necesario para poder usar esta tecnología y las librerías que han sido utilizadas para su desarrollo, por ahora haciéndolo solo compatible con gafas de Realidad Virtual de Microsoft Mixed Reality ejecutadas en el sistema operativo Windows 10.

Además, tenemos la limitación de necesitar al menos un Controlador (Motion Controller), sin ser posible jugar sin ellos.

6 Información del documento

6.1 Definición, acrónimos y abreviaturas

- VR: Siglas en inglés de Realidad Virtual.
- HDM: Siglás en inglés de Head Mounted Display, haciendo referencia a las Gafas de Realidad Virtual.
- MotionController/Controlador: Controladores específicos para juegos de VR, los cuales trasladan su posición y rotación al juego además de contener varios elementos de interacción como botones, touchpad y joystick.