



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

CAMPUS D'ALCOI

# UNIVERSITAT POLITÈCNICA DE VALÈNCIA

## Escuela Politécnica Superior de Alcoy

Desarrollo de aplicación para generar salas formativas en  
realidad virtual.

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Mengual Barroso, Juan José

Tutor/a: Orta López, Jorge

Cotutor/a: Izquierdo Doménech, Juan Jesús

CURSO ACADÉMICO: 2021/2022

## 1.1 Resumen en castellano

Actualmente, vivimos una de las situaciones más complicadas a nivel personal para todo el mundo, el COVID-19. Prácticamente todos los sectores se han visto perjudicados, y la educación no se ha salvado. Esto nos ha llevado a nuevas formas de enseñanzas y algunas de ellas han llegado para quedarse.

En el siguiente trabajo, se pretende desarrollar una aplicación de realidad virtual, que busca que la enseñanza a distancia no sea una desventaja, dando soluciones a esas largas videollamadas en las que, tanto profesores como alumnos, no estaban cómodos.

La idea es crear una aplicación capaz de ser usada por personas que no necesariamente tengan conocimientos de programación, para crear unas salas virtuales de formación, donde puedas seleccionar qué información mostrar e incluso qué elementos te gustaría que fuesen interactivos, como, por ejemplo, un microchip o un músculo. Así, sería posible poder manipularlo en cuanto a escala y posición, y así apreciar de una mejor forma los detalles más pequeños.

Una vez creada la sala, los estudiantes y el profesor se unirán en un entorno virtual e inmersivo de situaciones reales o irreales, en la que apreciaremos de mejor forma situaciones que en la vida real serían imposibles de recrear.

## 1.1 Palabras Clave

Aplicación. Realidad virtual. Educación. Multijugador. Unity

# 1.1 Resumen en inglés

Currently, we are experiencing one of the most complicated situations on a personal level for the whole world, COVID-19. Virtually all sectors have been affected, and education has not been spared. This has led us to new forms of teachings and some of them are here to stay.

In the following work, it is intended to develop a virtual reality application, which seeks that distance learning is not a disadvantage, providing solutions to those long video calls in which both teachers and students were not comfortable.

The idea is to create an application capable of being used by people who do not necessarily have programming knowledge, to create virtual education rooms, where you can select what information you want to show and even what elements you would like to be interactive, such as, for example, a microchip or a muscle. Thus, it would be possible to be able to manipulate it in terms of scale and position, and thus better appreciate the smallest details.

Once the room is created, the students and the teacher will join in a virtual and immersive environment of real or unreal situations, in which we will better appreciate situations that in real life would be impossible to recreate.

# 1.1 Palabras Clave

Application. Virtual reality. Education. Multiplayer. Unity

## 1.1 Resumen en valenciano

Actualment, vivim una de les situacions més complicades a nivell personal per a tothom, el COVID-19. Pràcticament tots els sectors s'han vist perjudicats, i l'educació no s'ha salvat. Això ens ha portat a noves formes d'ensenyaments i algunes d'elles han arribat per a quedar-se.

En el següent treball, es pretén desenvolupar una aplicació de realitat virtual, que busca que l'ensenyament a distància no siga un desavantatge, donant solucions a aqueixes llargues \*videollamadas en les quals, tant professors com alumnes, no estaven còmodes.

La idea és crear una aplicació capaç de ser usada per persones que no necessàriament tinguen coneixements de programació, per a crear unes sales virtuals de formació, on pugues seleccionar quina informació mostrar i fins i tot quins elements t'agradarien que anaren interactius, com, per exemple, un microxip o un múscul. Així, seria possible poder manipular-ho quant a escala i posició, i així apreciar d'una millor forma els detalls més xicotets.

Una vegada creada la sala, els estudiants i el professor s'uniran en un entorn virtual i immersiu de situacions reals o irreal, en la qual apreciarem de millor forma situacions que en la vida real serien impossibles de recrear.

## 1.1 Palabras Clave

Aplicació. Realitat virtual. Educació. Multijugador. \*Unity

# 02 Índice

## Contenido

1.1 Resumen en castellano .....	1
1.1 Palabras Clave .....	1
1.1 Resumen en inglés .....	2
1.1 Palabras Clave .....	2
1.1 Resumen en valenciano.....	3
1.1 Palabras Clave .....	3
Índice .....	4
Introducción y motivación .....	8
3.1 Estructura del documento .....	9
3.2 Motivación .....	11
3.3 Objetivos .....	12
Estado del arte.....	13
4.1 Introducción a la realidad virtual .....	14
4.1.1 ¿Qué es la realidad virtual? .....	14
4.1.2 El mercado actual sobre esta tecnología.....	14
4.1.3 ¿Cómo desarrollar aplicaciones? .....	19
4.1.4 ¿Qué es el metaverso? .....	20
4.1.5 ¿Estudio de mercado, aplicaciones similares? .....	21
4.2 Herramientas para el desarrollo.....	23
4.2.1 ¿Qué es Unity? .....	23
4.2.2 ¿Qué es Oculus SDK? .....	26
4.2.3 ¿Qué es Netcode? .....	26
4.2.4 ¿Qué es Itween?.....	26
4.2.5 ¿Qué es Sketchbook?.....	27
4.2.6 ¿Qué es Figma? .....	27
Desarrollo del proyecto .....	28

5.1 “Roadmap” para el proyecto .....	29
5.2 Primeros pasos.....	30
5.2.1 Viabilidad del proyecto. ....	30
5.2.2 Dispositivos usados en el proyecto.....	31
5.2.3 Instalación de Unity. ....	31
5.2.4 Instalación de paquetes de Unity.....	33
5.2.5 Instalación de “assets” de Unity.....	34
5.2.6 Activación del modo desarrollador de las Oculus. ....	36
5.2.7 Preparación del entorno de Unity para Oculus RV. ....	37
5.2.8 Primeras pruebas y toma de contacto. ....	39
5.3 Creación del personaje (jugador) .....	41
5.3.1 Componentes utilizados de Oculus SDK. ....	41
5.3.1 Movimiento continuo. ....	45
5.3.2 Movimiento con teletransportación.....	46
5.4 Creación de menús e interfaces.....	49
5.4.1 Componentes de los menús .....	50
5.4.2 Diseño final de los menús .....	53
5.5 Creación de los paneles .....	59
5.6 Creación de los objetos interactivos.....	61
5.6.1 Componentes de los objetos.....	62
5.6.2 Listado de los objetos.....	66
5.6.3 Soporte de los objetos .....	70
5.7 Creación de salas .....	72
5.7.1 Planteamiento y estructura.....	72
5.7.2 Creación de las salas.....	75
5.8 Creación del sistema multijugador.....	78
5.9 Implementación del visualizador de videos.....	79
Posibles actualizaciones.....	82

6.1 Sistema multijugador .....	83
6.2 Avatar personalizado .....	83
6.3 Importar objetos 3D personalizados.....	84
6.4 Creación del sistema de comunicación.....	84
Conclusiones .....	85
Bibliografía y referencias.....	87

Introducción y

03 motivación

## 3.1 Estructura del documento

- *Capítulo 3: Introducción y motivación:*

*Se explica cuál ha sido el motivo de la elección para realizar este TFG. También se define cual ha sido el planteamiento inicial y los objetivos propuestos para completar con éxito este trabajo.*

- *Capítulo 4: Estado del arte:*

*En este capítulo se expone toda la investigación que se ha realizado sobre la situación actual de la realidad virtual, y se explica que herramientas han sido utilizadas para el desarrollo de este TFG.*

- *Capítulo 5: Desarrollo del proyecto:*

*Se detalla cómo se ha organizado el trabajo y se exponen todos los procedimientos o problemas encontrados a la hora de completar cada uno de los objetivos.*

- *Capítulo 6: Posibles actualizaciones:*

*En este capítulo se presenta un listado de ideas, que debido a limitaciones de conocimiento, compatibilidad o tecnología no se han podido implementar a día de hoy.*

· *Capítulo 7: Conclusiones:*

*Se redactan las conclusiones después de haber realizado dicho trabajo.*

· *Capítulo 8: Bibliografía y referencias:*

*En este capítulo se muestran las fuentes y las referencias donde se ha obtenido la información para el desarrollo de este proyecto.*

## 3.2 Motivación

La motivación que ha llevado a realizar este TFG “DESARROLLO DE APLICACIÓN PARA GENERAR SALAS FORMATIVAS EN REALIDAD VIRTUAL”, es la de ponerme un último reto antes de acabar mis estudios en esta universidad. En esta universidad, se han enseñado la base de muchos conceptos para poder adaptarse a diferentes situaciones en la vida laboral, es por eso por lo que he decidido hacer mi TFG sobre un tema que para mí es completamente nuevo y desconocido, ya que no he tenido ninguna asignatura relacionada con la realidad virtual ni algo similar. Algo que me gustaría que se tuviese en cuenta a la hora de evaluar.

Otro de los antecedentes para este nuevo reto, ha sido la situación actual en la que hemos tenido que vivir estos últimos años, debido a la pandemia mundial. Con esta pandemia hemos aprendido muchas nuevas formas de trabajar y cumplir con nuestras obligaciones, de formas que nunca hubiésemos pensado, y aunque todo el mundo ha puesto de su parte, no en todos los ámbitos, se ha conseguido que las nuevas soluciones estuviesen a la altura de las actividades cotidianas. Es por eso por lo que se me ha ocurrido aportar una idea, en concreto para el sector de la educación y formación, que consiste en ofrecer una alternativa a las reuniones de Teams o similares, donde el profesor o el narrador no deja de hablar sin ningún tipo de “feedback” y puede llegar a ser frustrante, en algunas situaciones, tanto para los oyentes como para el narrador debido a la monotonía de estas.

Además, considero que es una idea atemporal, ya que podría emplearse siempre que se tuviese que dar cursos de forma remota o para dar formación

en campos que puedan verse beneficiados de las características que se ofrecen.

## 3.3 Objetivos

El objetivo final de este TFG consiste en crear una aplicación de realidad virtual, que consiga generar una sala de exposiciones con diversos elementos de forma que la enseñanza sea más interactiva, y además puedas seleccionar algunos objetos para poder manipularlos en cuanto a tamaño, posición y escala para tener una visión más amplia y detallada de estos. Una de las características principales de esta idea, también es que las salas, puedan ser generadas por personas que no necesariamente necesiten tener conocimientos de programación y puedan aportar su propio material al nuevo mundo virtual, así mismo, un médico podría crear una sala con una serie de músculos, para poder enseñárselos a sus alumnos de forma más real y con mayor detalle, que a través de una foto.

Para realizar este trabajo me he propuesto una serie de subobjetivos para facilitarme el desarrollo de este:

1. Informarme sobre el estado actual de la realidad virtual y que herramientas existen hoy en día.
2. Estudiar la viabilidad del proyecto e instalar materiales necesarios.
3. Crear un personaje: Movimiento, cámara, interacción con los mandos.
4. Crear los mecanismos para conseguir objetos interactivos.
5. Crear menús y una experiencia de usuario agradable.
6. Crear sistema de escenas o salas.
7. Crear sistema multijugador.
8. Desarrollar algoritmo para ubicar los elementos seleccionados.
9. Crear visualizador de videos de Youtube.

# 04 Estado del arte

## 4.1 Introducción a la realidad virtual

### 4.1.1 ¿Qué es la realidad virtual?

La realidad virtual, tal y como la conocemos hoy en día, apareció por primera vez de la mano de Morton Heilig. Este inventor de nacionalidad norteamericana desarrolló lo que podríamos denominar el primer dispositivo de realidad virtual. Este dispositivo se basaba en simular un paseo en motocicleta por la ciudad de Nueva York.

Aunque es cierto que este innovador invento no permitía la interacción con la simulación que se ejecutaba, conseguía que el usuario, a través de los sentidos, tuviese la sensación de estar en un lugar que realmente no estaba.

Por ello podríamos decir que la realidad virtual consiste en, mediante la tecnología informática, trasladar al usuario a un escenario virtual y hacerle creer a través de los sentidos que ha sido desplazado a un entorno completamente inmersivo (Levis, 2006).

### 4.1.2 El mercado actual sobre esta tecnología.

Actualmente, existen muchas empresas que compiten por ser las primeras en estar a la última en la tecnología y desean tener los dispositivos más potentes del mercado. Es por ello que gracias a esta competitividad nosotros como consumidores podemos aprovecharnos de esta situación y tenemos un amplio abanico de opciones en el mercado a la hora de seleccionar los dispositivos de realidad virtual que más se ajusten a nuestras necesidades.

Entre las empresas más potentes del mercado hoy en día podemos encontrar que sus últimos dispositivos son los siguientes:

#### 4.1.2.1 Oculus



**Modelo:** Oculus Quest 2 (Meta, 2020).

**Fecha de lanzamiento:** 13 de octubre de 2020.

#### 4.1.2.2 Sony



**Modelo:** PlayStation VR (Sony, 2016).

**Fecha de lanzamiento:** 13 de Octubre de 2016.

### 4.1.2.3 Google



**Modelo:** Google Day Dream View.

**Fecha de lanzamiento:** Año 2016 (actualmente no disponibles).

### 4.1.2.4 Steam



**Modelo:** Visor de Valve Index (Valve Corporation, 2019).

**Fecha de lanzamiento:** 28 de junio de 2019.

#### 4.1.2.5 Vive



**Modelo:** HTC Vive Cosmos Elite (HTC Vive, 2020).

**Fecha de lanzamiento:** Año 2020.

#### 4.1.2.6 Samsung



**Modelo:** Samsung Gear VR (Samsung, 2015).

**Fecha de lanzamiento:** noviembre 2015.

#### 4.1.2.7 HP



**Modelo:** HP Reverb G2 (HP, 2021).

**Fecha de lanzamiento:** 17 de enero de 2021.

Como se puede observar estas tan solo son algunas de las gafas más sofisticadas del mercado, pero no son las únicas. Y es que de una forma u otra todas las marcas con cierta repercusión en el mundo de la tecnología han querido realizar su aportación e invertir en la realidad virtual.

Otra característica notable de estos dispositivos es que el precio entre algunos de ellos es muy diferente, y esto puede ser causado por distintos factores, como podría ser el desconocimiento de los usuarios frente a lo que realmente deberían pagar por adquirir un dispositivo de estas características, o que simplemente al ser un mercado relativamente nuevo, esté todavía por estabilizar.

### 4.1.3 ¿Cómo desarrollar aplicaciones?

El auge de esta tecnología se debe a una nueva corriente llamada metaverso, que está en el punto de mira de muchas empresas. La inquietud de muchas personas, sobre todo programadores, por saber cuál es la base de todo esto y qué es lo que realmente es necesario para poder incorporarse al mercado, hace plantearse cómo se pueden desarrollar aplicaciones para este tipo de dispositivos.

Actualmente, en el mercado podemos encontrar dos grandes empresas que nos permiten crear aplicaciones o juegos de una forma muy flexible para esta nueva tecnología.

En primer lugar, tenemos Unity. Esta herramienta se está convirtiendo en la preferida por los desarrolladores ya que, gracias a su facilidad de uso y su versatilidad, permite crear juegos y aplicaciones. Además, cuenta con una comunidad muy grande que se ayuda entre sí y la propia empresa facilita y proporciona mucha documentación y materiales para el desarrollo de tus proyectos.

Por otro lado, encontramos a Unreal Engine. Esta aplicación, a diferencia de Unity, está enfocada a programadores más experimentados o a empresas donde se trabaje de forma conjunta, ya que el nivel de dificultad es bastante más complejo (Rodríguez, 2022).

#### 4.1.4 ¿Qué es el metaverso?

Como se ha mencionado anteriormente, el metaverso es una palabra muy nombrada en la actualidad.

Su origen proviene del griego donde “meta” significa después o más allá y “verso” significa universo. Con este término se hace referencia a un nuevo mundo virtual en 3D sin ser exclusivamente propiedad de ninguna empresa, sino simplemente, un nuevo concepto (Santander, 2022).

Y es que, dados todos los avances tecnológicos de hoy en día, hacen que distinguir qué es real y qué no lo es, se convierta en una tarea cada vez más compleja.

Es por ello que, muchas empresas están tratando de instaurarnos que a través del nuevo mundo de realidad virtual podríamos llegar a tener una realidad paralela a nuestras vidas, donde se podría hacer exactamente lo mismo que se hace en el mundo real, pero sin las limitaciones que este conlleva (Iberdrola, 2021).

Las ideas que se proponen para realizar dentro de este mundo del más allá son tan diversas que nadie quiere quedarse fuera de este universo paralelo. Por ello, empresas de todos los sectores quieren estar al día de este nuevo mundo virtual, ya sea para contratar trabajadores dentro de este, crear publicidad dentro de esta nueva realidad, virtualizar sus empresas o incluso instaurar un sistema bancario de forma descentralizada.

#### 4.1.5 ¿Estudio de mercado, aplicaciones similares?

Antes de iniciar este proyecto se tuvieron en cuenta aplicaciones que ofrecían al usuario la posibilidad de crear sus propias salas virtuales.

El objetivo de este estudio fue encontrar los puntos más débiles y cuáles son las características que hoy en día no están disponibles en las aplicaciones existentes.

Las aplicaciones que se tuvieron en cuenta para el desarrollo de este proyecto fueron:



En primer lugar, Engage. Esta sin duda es la aplicación más similar al objetivo de este proyecto, y por el momento es la que más flexibilidad le ofrece al usuario para crear sus propios entornos. Cuenta con una gran librería de escenarios y de objetos, pero aquí es donde reside su punto más débil, y es que todos los objetos provienen de una librería, y por el momento no te permite importar contenido personalizado del exterior. Una desventaja que se planteará en el desarrollo de este trabajo.

Por otro lado, se encontraron muchas aplicaciones como: Arthur, Glue, Spatial, Horizon Workrooms y muchas más que comparten su propósito, que únicamente es el de funcionar como una sala de reuniones donde solo puedes comunicarte con otras personas, pero no puedes compartir tus propios espacios virtuales.

## 4.2 Herramientas para el desarrollo

### 4.2.1 ¿Qué es Unity?

Unity es una de las plataformas de creación de contenido en tiempo real más utilizadas hoy en día. Es cierto que comenzó siendo una herramienta destinada a la creación de videojuegos tanto en 2D como en 3D para diversas plataformas, pero hoy en día se utiliza para desarrollar todo tipo de experiencias, como aplicaciones (como es el caso de este proyecto), animaciones, videojuegos, etc.

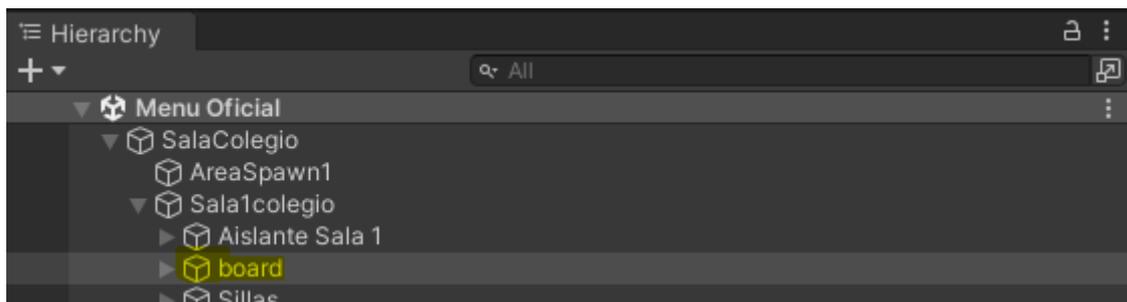
Mediante un editor visual y utilizando la programación basada en scripts se puede conseguir dar forma a los proyectos en mente, y es por eso por lo que yo he utilizado esta herramienta, ya que la polivalencia es una de las virtudes de esta plataforma y además es una de las plataformas que más se enfoca en la educación, facilitando licencias gratuitas, como una gran cantidad de material didáctico y una documentación muy completa la cual ha sido de gran ayuda para progresar en este proyecto.

En Unity se utiliza un lenguaje de programación basado en objetos y es por ello por lo que es necesario, entender que son y para que se utilizan los siguientes elementos:

#### 4.2.1.1 GameObjects

*“Los GameObjects son objetos fundamentales en Unity que representan personajes, props, y el escenario. Estos no logran nada por sí mismos, pero funcionan como contenedores para los Componentes, que implementan la verdadera funcionalidad. Por ejemplo, un objeto Light es creado al adjuntar un componente Light a un GameObject.*

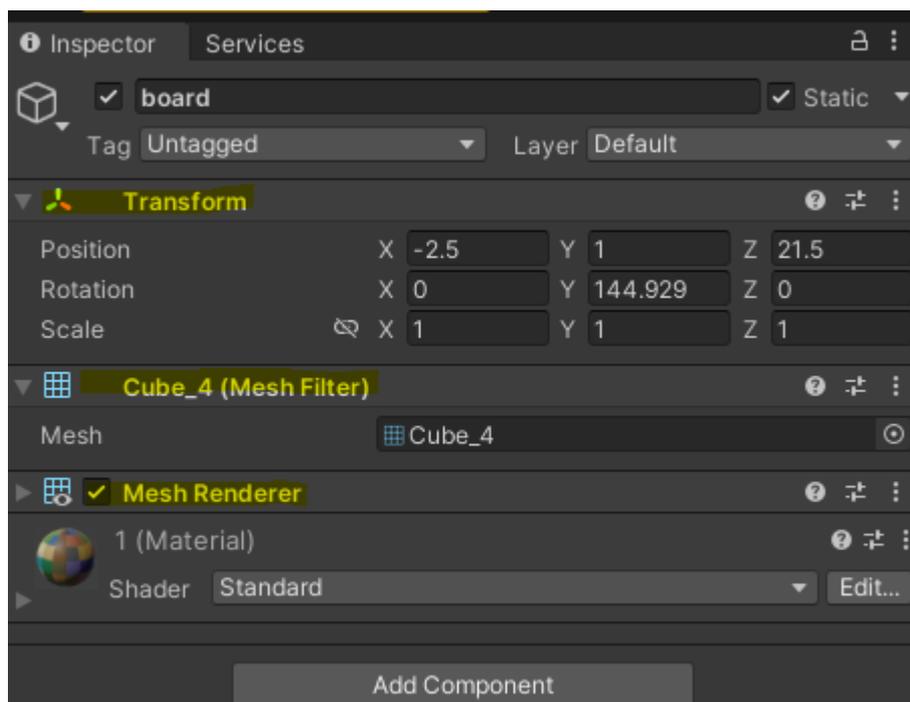
*Un GameObject siempre tiene el componente Transform adjunto (para representar la posición y orientación) y no es posible quitar esto. Los otros componentes que le dan al objeto su funcionalidad pueden ser agregados del menú Componentes del editor o desde un script.” (Unity Technologies, 2016).*



#### 4.2.1.2 Componentes

*“Los Componentes son las tuercas y tornillos de los objetos y comportamientos de un juego. Son las piezas funcionales de cada GameObject.” (Unity Technologies, 2016).*

Dicho de otra forma, los componentes son las propiedades que se le aplican a los GameObjects, desde su comportamiento el cual es manejado mediante scripts, hasta que material, peso, interacción... Va a tener finalmente el objeto.



#### 4.2.1.3 Prefabs

*“Unity contiene un tipo de objeto al que le llama Prefab, el cual le permite almacenar un objeto GameObject junto a sus componentes y propiedades. El prefab actúa como una plantilla a partir de la cual se pueden crear nuevas instancias del objeto en la escena. Cualquier edición hecha a un prefab será inmediatamente reflejada en todas las instancias producidas por él, pero, también se puede anular componentes y ajustes para cada instancia individualmente.”* (Unity Technologies, 2016).

Esto es un concepto realmente útil para mi proyecto, ya que, al generar contenido dinámico, era necesario establecer unos Prefabs (plantillas) para que a la hora de instanciar y manipular el contenido se realizase de forma automática.

## 4.2.2 ¿Qué es Oculus SDK?

Oculus SDK es una librería de componentes, herramientas y recursos, destinado a la creación de aplicaciones o juegos nativos para los dispositivos Oculus, que, en mi caso concreto, es el que he utilizado para desarrollar este proyecto.

Esta librería es especialmente útil ya que una vez conoces cual es la funcionalidad de cada componente y las herramientas te simplifica el desarrollo de las interacciones más básicas con el entorno, como podría ser el movimiento o la interacción con los mandos.

## 4.2.3 ¿Qué es Netcode?

Netcode es una librería de redes creadas especialmente para el motor de Unity. Su intención es ser de uso libre y están haciendo lo posible para que sea una biblioteca muy versátil y se pueda ajustar a las necesidades de cada usuario.

Decidí implementar esta tecnología ya que era muy innovadora y todavía no había escuchado a mucha gente a hablar de ella.

Finalmente, la ambición no dio el resultado esperado, ya que esta biblioteca está en una fase muy temprana y por el momento no ha sido posible implementarla de forma correcta en la realidad virtual.

## 4.2.4 ¿Qué es Itween?

Itween es un asset de Unity, el cual contiene los componentes necesarios para poder animar a los objetos instanciados en nuestro mundo virtual de una manera muy sencilla y fácil.

Es una herramienta muy práctica la cual te ayuda a darle vida a tus proyectos de una forma rápida y sin tener que invertir demasiado tiempo en las animaciones, ya que contiene una gran librería de ellas.

#### 4.2.5 ¿Qué es Sketchbook?

Sketchbook es una aplicación disponible en varios dispositivos, en mi caso concreto para iPad destinada para realizar bocetos, pinturas o ilustraciones.

Esta aplicación me ha servido de gran ayuda para desarrollar una de las partes más importantes del proyecto y que más he trabajado, los “wireframes” de baja fidelidad de los menús con el que el usuario finalmente tendrá que interactuar.

#### 4.2.6 ¿Qué es Figma?

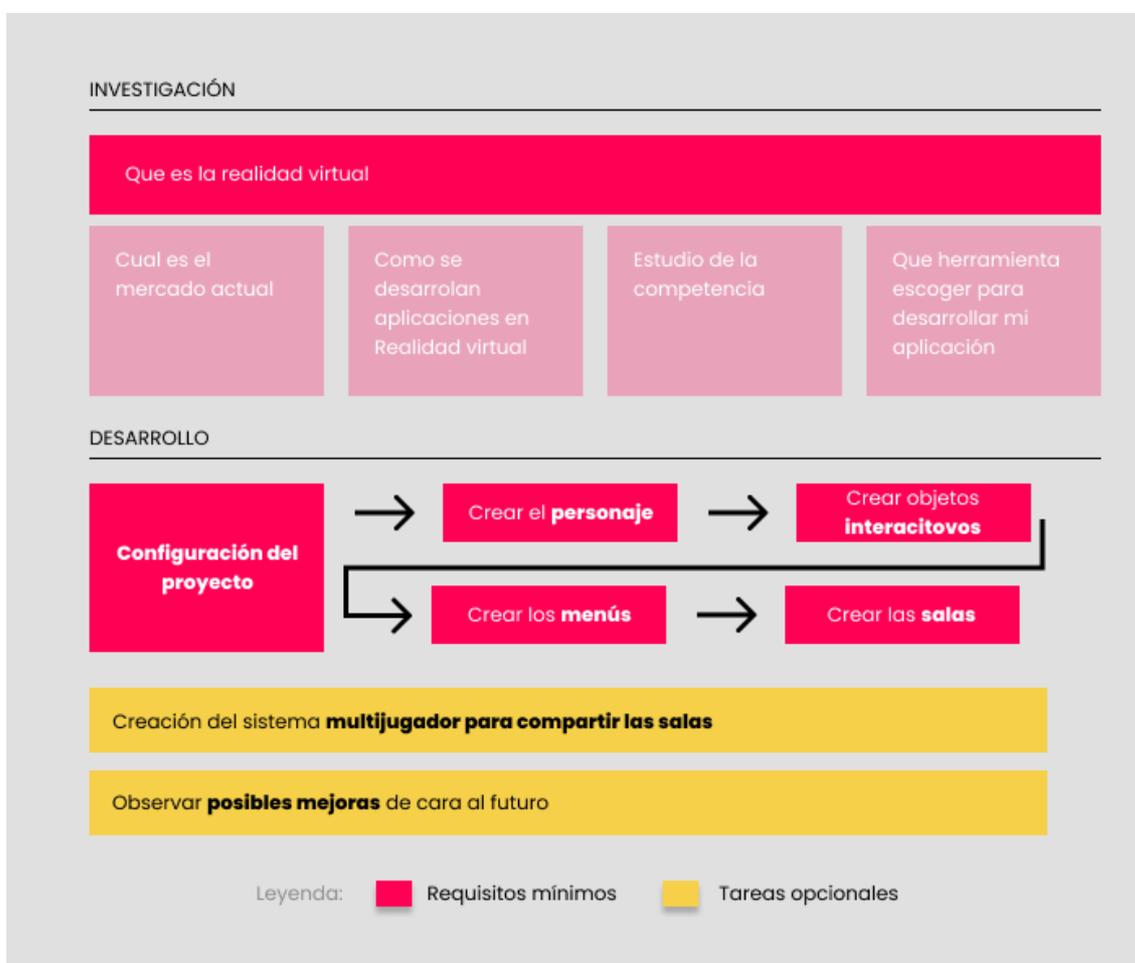
Figma es un editor de gráficos vectorial y una herramienta de prototipado web, la gran ventaja de esta herramienta es que todos los documentos se alojan en la web.

He utilizado esta aplicación, ya que profesionalmente me dedico al prototipado web y es una herramienta con la que trabajo diariamente y considero que es muy útil, tanto como para crear contenido gráfico como puede ser el roadmap o para realizar ajustes o montajes de imágenes.

# Desarrollo del 05 proyecto

# 5.1 “Roadmap” para el proyecto

El “roadmap” (hoja de ruta) de un proyecto es una vista general de los logros clave y los objetivos en general del proyecto.



Esta hoja de ruta ha resultado de gran ayuda, debido a que, al ser la realidad virtual, un entorno desconocido para mí y del cual se necesitan tantas cosas, resultó el guion perfecto para desarrollar el proyecto con éxito.

## 5.2 Primeros pasos

### 5.2.1 Viabilidad del proyecto.

Para este proyecto se requiere tener unos requisitos mínimos si se desea desarrollarlo de forma óptima.

El material necesario (para las Oculus Quest 2) es:

- Ordenador con los siguientes requisitos mínimos:
  - ✓ Procesador: Intel Core i5-4590 o un AMD Ryzen 5 1500X o superior
  - ✓ Memoria RAM: 8 GB de RAM
  - ✓ Sistema operativo: Windows 10
  - ✓ Tarjeta gráfica NVIDIA GeForce GTX 970 o AMD Radeon 400 Series o superior, y con al menos 3 GB de VRAM, con soporte para píxel shader 5.1 y vertex shader 5.1.
- Gafas de Realidad virtual.
- Conexión a internet.
- Licencia de Unity

## 5.2.2 Dispositivos usados en el proyecto.

En mi caso, compré las gafas de realidad virtual Oculus Quest 2 ya que tienen una gran ventaja respecto al resto de opciones del mercado, y es que la conexión entre el ordenador y las gafas se realiza de forma muy sencilla mediante un USB C y a la hora de desarrollar la aplicación no es necesario tener que compilar todas las pruebas que se realizan, ya que te permite ver los cambios realizados en tiempo real activando el modo de desarrollador de las gafas.

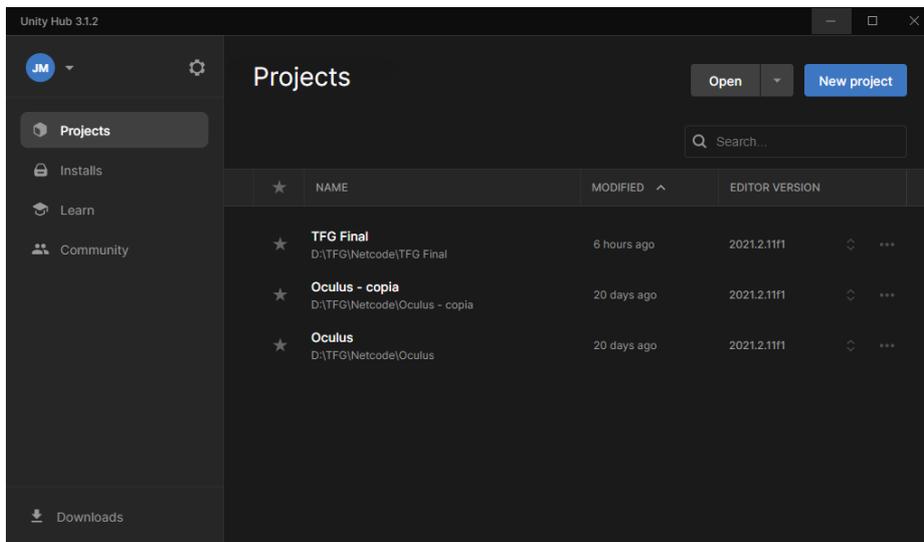
En cuanto a los componentes de mi ordenador son los siguientes:

- Procesador: 11th Gen Intel(R) Core(TM) i5-11400F @ 2.60GHz 2.59 GHz
- Memoria RAM: 16GB de RAM
- Tarjeta Gráfica: NVIDIA GeForce RTX 3050
- Sistema operativo: Windows 10

## 5.2.3 Instalación de Unity.

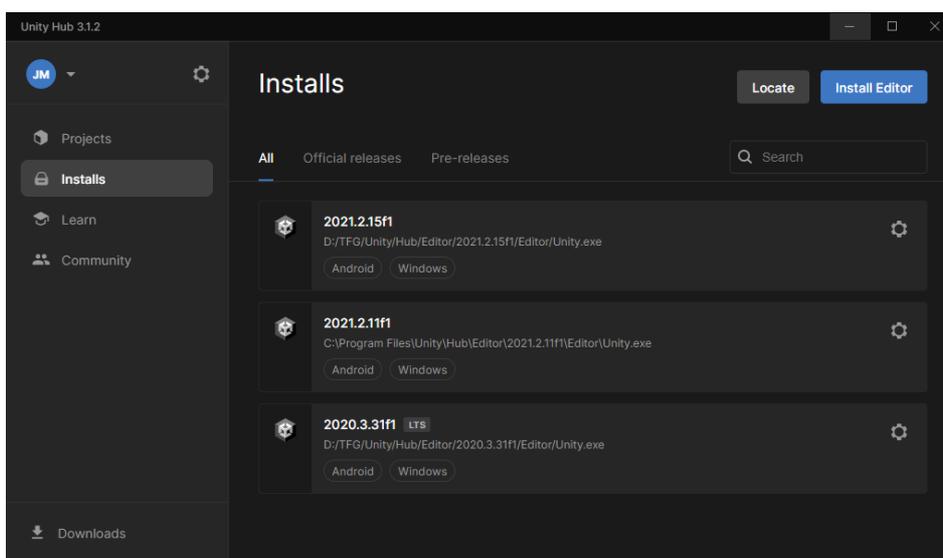
A la hora de preparar el entorno de desarrollo, el primer paso era descargar Unity. En este caso, la empresa Unity te facilita un "Hub" llamado Unity Hub, el cual es un lanzador desde el que puedes descargar la versión del entorno de programación necesitas y además te permite instalar diferentes versiones de manera simultánea. Esto es un punto muy favorable, ya que no todas las versiones soportan todos los paquetes o extensiones que puedes llegar a requerir en un proyecto. Gracias a este Hub, Unity incluso permite que puedas migrar los proyectos ya iniciados en una versión a una nueva versión nueva deseada, ya sea más moderna (update) o más antigua (downgrade).

El Unity Hub podemos obtenerlo de la página oficial de Unity.



Como podemos observar en la imagen, en el Unity Hub aparecen los tres proyectos que actualmente tengo yo creados, es aquí donde podemos crear nuevos proyectos y editar la versión en la que se lanzaran dichos proyectos, siempre y cuando tengamos dichas versiones instaladas previamente.

En la pestaña de "Installs" es donde podemos instalar las diferentes versiones del editor, que a la hora de crear un nuevo proyecto deberemos seleccionar cual es la deseada.

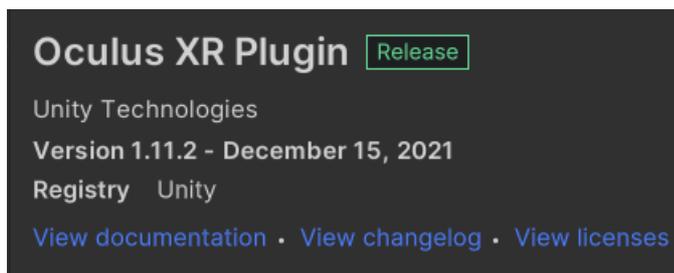


Como se puede observar tengo tres versiones diferentes del editor de Unity, debido que a medida que iba realizando las pruebas de rendimiento y observando que necesitaba nuevos paquetes o extensiones que no tenían soporte en algunas versiones determinadas tuve que ir actualizando hasta encontrar la versión óptima para este proyecto.

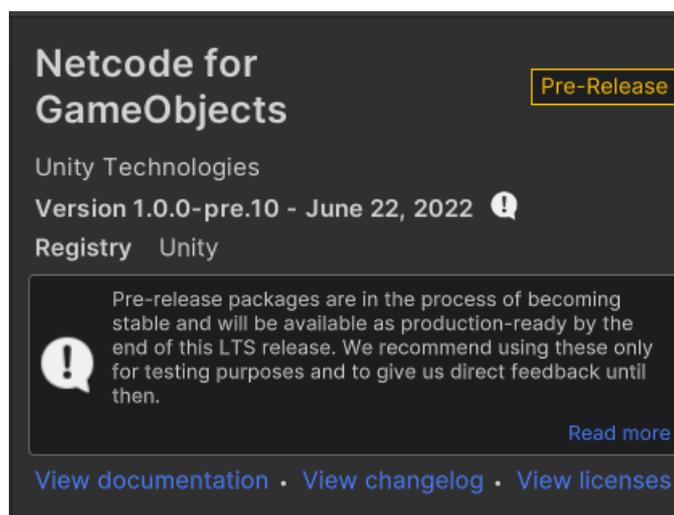
Es por ello por lo que la versión: 2021.2.11f1 es la escogida, ya que es la versión con mayor compatibilidad con los paquetes que en la siguiente sección se detallan.

## 5.2.4 Instalación de paquetes de Unity.

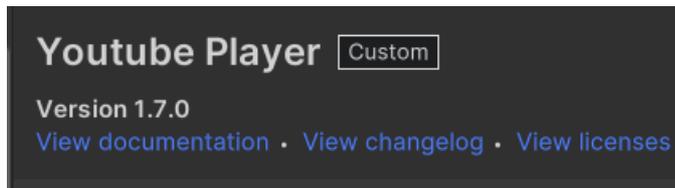
### 5.2.4.1 Oculus XR



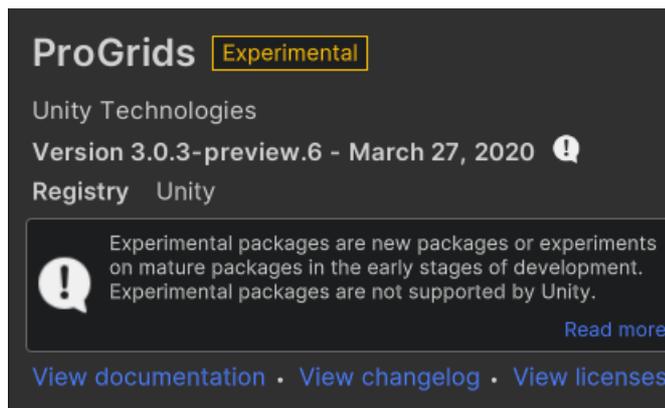
### 5.2.4.2 Netcode



### 5.2.4.3 Youtube Player

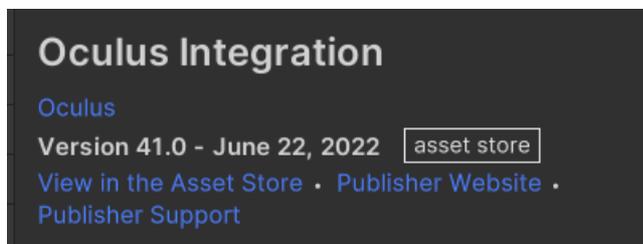


### 5.2.4.4 ProGrids

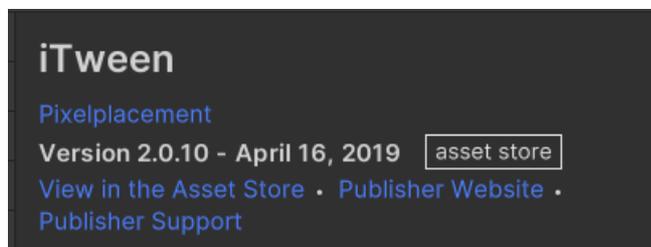


## 5.2.5 Instalación de “assets” de Unity.

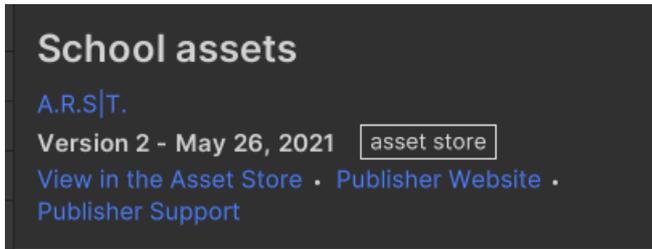
### 5.2.5.1 Oculus integration SDK



### 5.2.5.2 Itween



### 5.2.5.3 School assets



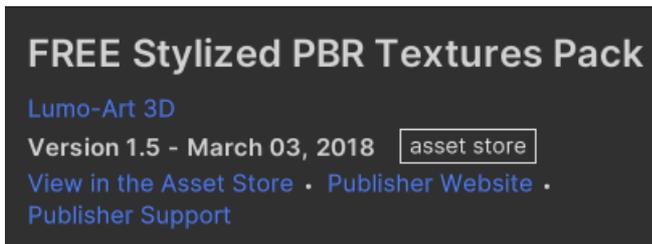
**School assets**

A.R.S|T.

Version 2 - May 26, 2021 [asset store](#)

[View in the Asset Store](#) · [Publisher Website](#) · [Publisher Support](#)

### 5.2.5.4 FREE Stylized PBR Textures Pack



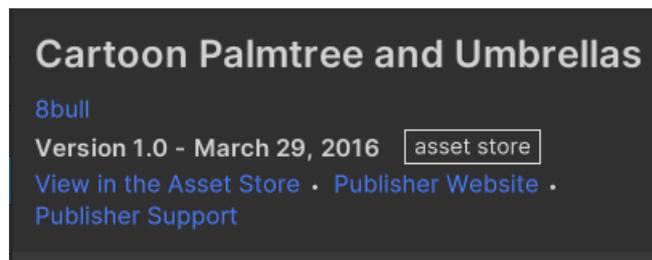
**FREE Stylized PBR Textures Pack**

Lumo-Art 3D

Version 1.5 - March 03, 2018 [asset store](#)

[View in the Asset Store](#) · [Publisher Website](#) · [Publisher Support](#)

### 5.2.5.5 Cartoon Palmtree and Umbrellas



**Cartoon Palmtree and Umbrellas**

8bull

Version 1.0 - March 29, 2016 [asset store](#)

[View in the Asset Store](#) · [Publisher Website](#) · [Publisher Support](#)

## 5.2.6 Activación del modo desarrollador de las Oculus.

Antes de empezar el proyecto se recomienda activar el modo de desarrollador de Oculus en una primera instancia. Esta característica le otorga al usuario la capacidad de probar, debuguear o testear las aplicaciones propias que desarrolle.

Para poder completar dicha activación en primer lugar es necesario que el usuario cuente con una cuenta de Oculus como desarrollador. Como el dispositivo utilizado en este proyecto es de la marca Oculus, la cual pertenece a Meta Platforms, Inc (anteriormente llamada Facebook) esta tarea se facilita, ya que se nos permite iniciar sesión con una cuenta de Facebook ya existente.



Una vez se ha configurado correctamente la cuenta de desarrollador se debe utilizar esta misma cuenta para iniciar sesión en la aplicación de Oculus. Dicha aplicación se encuentra disponible para varias plataformas, en este caso en concreto se ha utilizado la de la versión para el ordenador ya que cuando se conectaron las gafas, se descargó automáticamente.

Una vez tengamos conectadas las gafas, deberemos ir a el apartado de dispositivos > (seleccionar el dispositivo) > configuración del dispositivo, y será

en este mismo apartado donde se permitirá activar la opción de desarrollador.

Finalmente, una vez se haya activado esta característica, nos saldrá un aviso en las gafas, donde se avisará de que dicha opción ha sido activada y deberemos aceptar los carteles emergentes que nos aparezcan.

### 5.2.7 Preparación del entorno de Unity para Oculus RV.

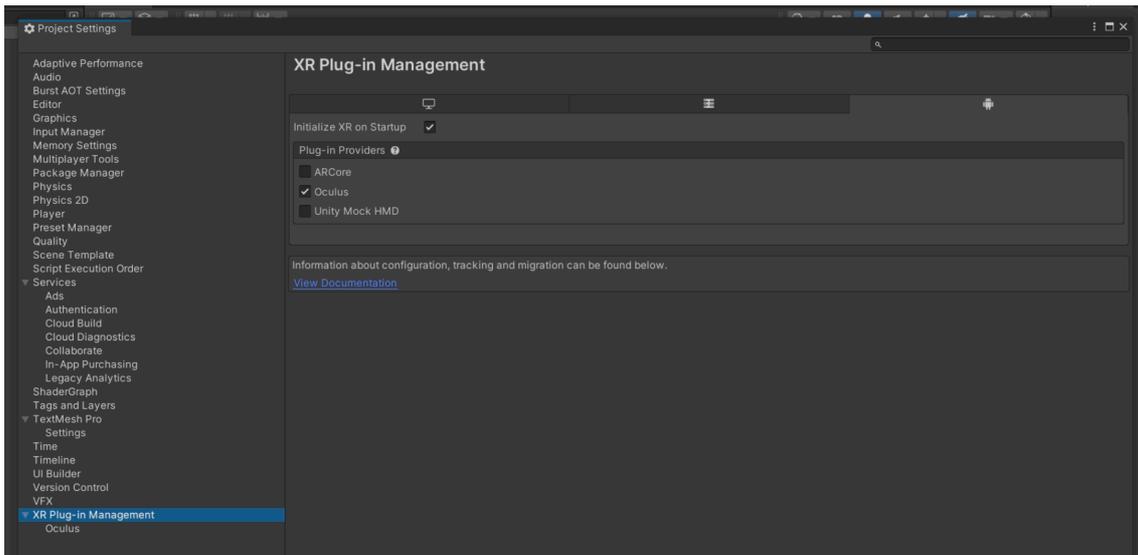
A la hora de preparar el entorno de Unity, será necesario realizar unos ajustes previos dentro del proyecto de Unity, ya que si no se siguen los siguientes pasos cuando se trate de lanzar las aplicaciones desarrolladas, pueden aparecer infinidad de inconvenientes.

Personalmente pude experimentar problemas como:

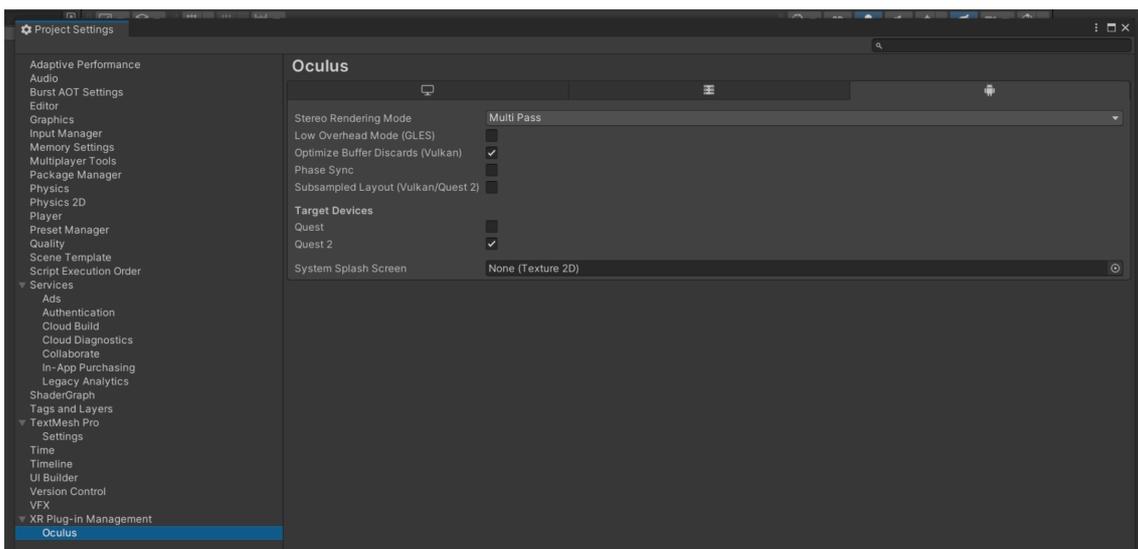
- Imposibilidad de que el programa Unity detectase las gafas.
- Fallo en los sensores de las gafas y en la estabilización.
- Imagen difusa y funcionamiento exclusivo por una lente.

Para preparar el entorno, debemos ir al menú editar > opciones del proyecto.

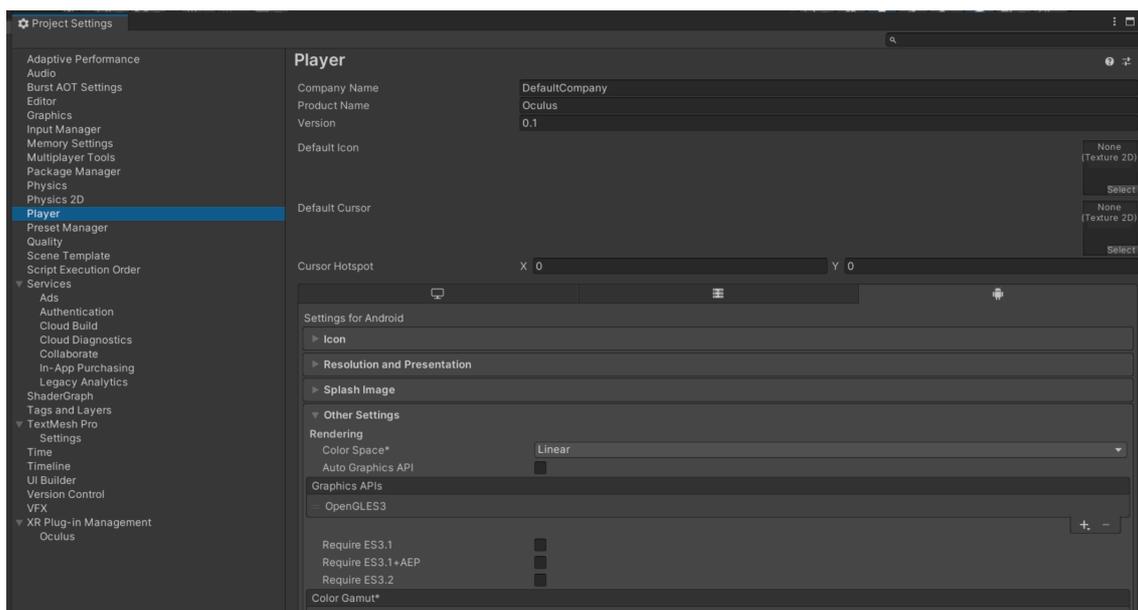
En la parte izquierda de este panel, deberemos seleccionar el paquete que se ha instalado previamente llamado "Oculus XR" y que en el menú aparece con la opción XR Plug-in Management. En esta sección seleccionaremos las siguientes opciones.



A continuación, en la opción de Oculus deberemos configurarlo de dicha forma para evitar los problemas mencionados anteriormente.



Por último, para acabar de configurar el dispositivo deseado, será necesario seleccionar la opción de jugador en el menú izquierdo y modificar el espacio de color a "Linear" ya que por defecto estará en "Gamma"

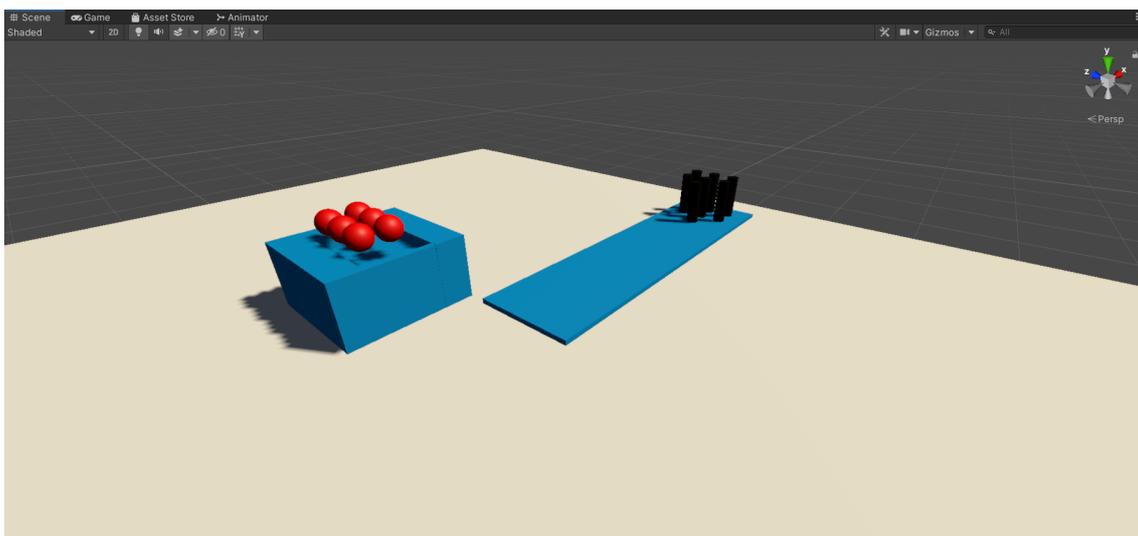


### 5.2.8 Primeras pruebas y toma de contacto.

Una vez completado todos los pasos de configuración anteriores, fue el momento de crear mi primer proyecto de Unity, el cual fue utilizado como una sala de pruebas, donde el objetivo principal fue conseguir configurar el entorno para que las pruebas se reprodujesen en las gafas como dispositivo predeterminado, y aprender todas las utilidades que nos proporciona el “asset” Oculus SDK integration.

Desde la configuración del personaje y sincronización de la cabeza con el escenario, como todas las interacciones que podrá hacer el usuario con su entorno. También aproveché este momento para probar como se comportarían algunos objetos al ser manipulados dependiendo de variables como el peso, la gravedad o las mallas de colisión.

El resultado final fue una bolera con elementos muy simples y geométricos donde el jugador podía derribar unos cilindros con una esfera que podía agarrar de una mesa tanto con las manos como con los mandos de Oculus.



## 5.3 Creación del personaje (jugador)

Nos encontramos con en el primer elemento clave de este proyecto, el jugador. Este `GameObject` será el encargado del movimiento, visión e interacción del personaje virtual.

Aunque es cierto que el personaje es posible crearlo completamente de forma independiente y manual desde cero, como se ha explicado anteriormente se ha utilizado el “asset” de Unity Oculus SDK que nos proporciona la plataforma de desarrolladores de Oculus y se puede obtener de forma gratuita.

Finalmente, gracias a este asset y debido a que se ha adquirido unas gafas de esta marca, conseguiremos tener compatibilidad completa con todos los dispositivos de dicha marca y no será necesario reprogramar ningún parámetro, aunque el modelo de las gafas difiera del que se ha utilizado en este proyecto (Oculus Quest 2) siempre y cuando la marca del dispositivo sea Oculus.

### 5.3.1 Componentes utilizados de Oculus SDK.

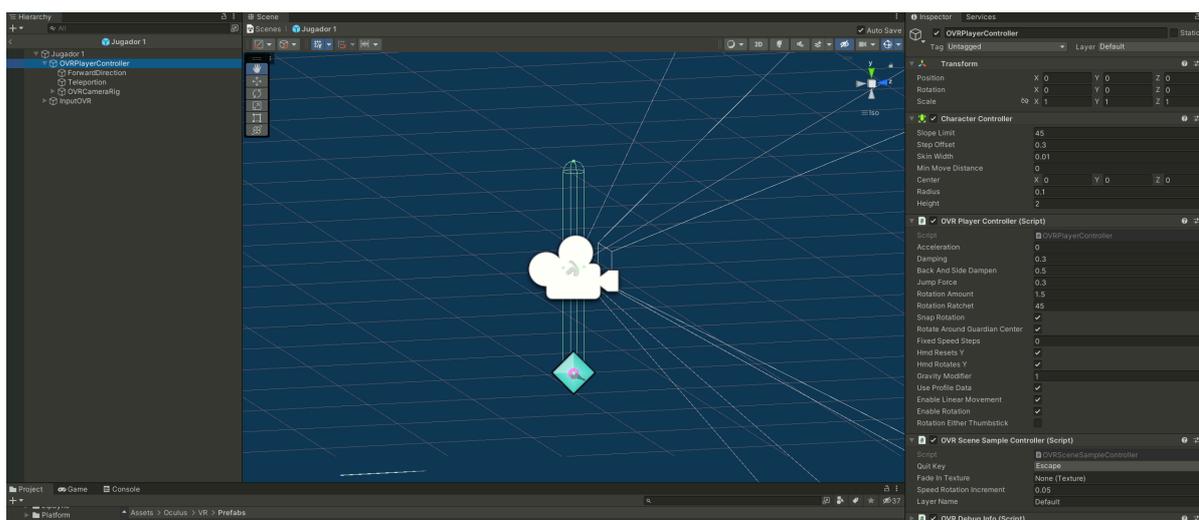
Para poder utilizar los elementos necesarios y esenciales, es importante leer previamente toda la documentación que nos proporciona Oculus sobre todos los componentes que contiene su librería. Tras el estudio completo lo aconsejable es recopilar los componentes esenciales para que más tarde se puedan introducir en sus `GameObjects` correspondientes.

En el propio asset de Oculus se pueden encontrar prefabs creados que ya tienen los componentes necesarios, para un funcionamiento básico de ciertos elementos, estos se han modificado para incorporarle nuevas características.

A continuación, se muestra el prefab Jugador, el cual contiene todo lo necesario para que este funcione. Se decidió crear un prefab, debido a que iba ser necesario para poder instanciar un único objeto cuando en el sistema multijugador necesitara crear un nuevo personaje.

El prefab Jugador contiene:

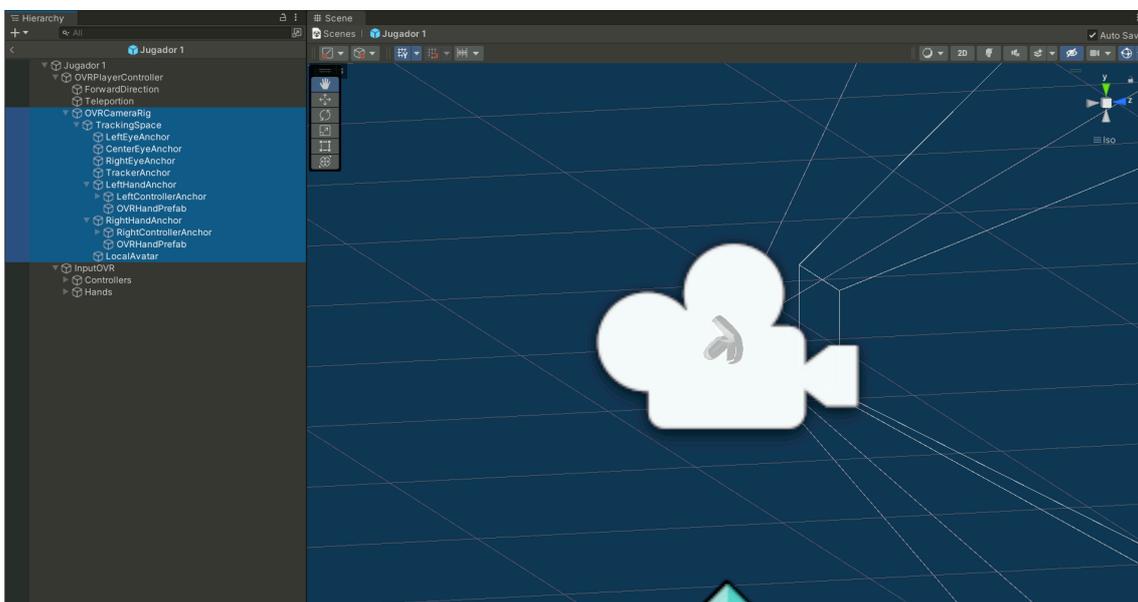
- El prefab **OVRPlayerController** : Este objeto es el que contiene la capsula del jugador, necesaria para detectar las colisiones con los demás objetos del escenario. Además, es aquí donde se controla mediante el OVR Player Controller las propiedades del jugador como la velocidad de movimiento, los ángulos de rotación, si se le permite saltar o no y varios parámetros más, los cuales son los encargados del movimiento del personaje dentro del mundo virtual.



- El prefab **OVRCameraRig** : En este objeto podemos encontrar todo lo relacionado con la cámara y todos los sensores del dispositivo de Oculus. Como podemos observar se han introducido un par de objetos que difieren al prefab original que Oculus nos proporciona.

Estos objetos son los encargados de renderizar en la posición correcta los modelados de las manos y de los mandos de Oculus para poder visualizarlos dentro del juego. Por último encontramos el objeto LocalAvatar donde se ha cargado un modelo 3D el cual simula una cabeza de un personaje para darle mayor realismo.

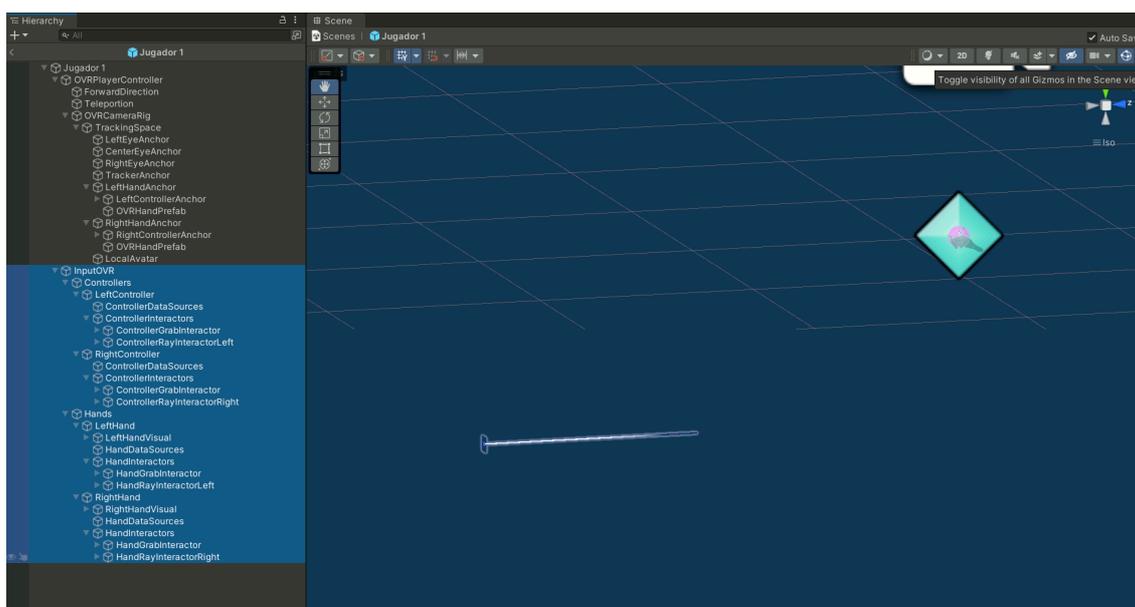
Es este mismo objeto el que debería modificar si finalmente se incorpora la mejora del avatar personalizado, explicada en la última sección del proyecto.



- El prefab OVR inputs: Aquí se encuentra toda la interacción de los métodos de entrada, es aquí donde se configura la información deseada para poder transmitir la información de los mandos o las manos del jugador a los elementos del mundo virtual.

Se puede observar que el tratamiento de las manos y de los mandos de Oculus se tiene que configurar de forma individual, por lo que es una consideración a tener en cuenta, ya que podemos establecer controles y comportamientos diferentes, según el dispositivo de entrada que el jugador escoja utilizar.

Como podemos ver en la parte inferior de la imagen también podemos apreciar que es aquí donde se ha generado el elemento con forma de rayo, el cual se visualiza cuando el jugador trata de interactuar con algún panel de interfaz o menú y cuyo objetivo es mostrar con una mayor precisión donde está apuntando el jugador dentro del panel.



Por último, se detallará como se ha conseguido implementar dos tipos de movimiento distinto para el personaje dentro del mundo virtual, para que el usuario pueda utilizar el que más cómodo le resulte.

### 5.3.1 Movimiento continuo.

Por defecto el prefab de `OVRPlayerController` del jugador está configurado para que mediante el uso de los joysticks de los mandos de Oculus el personaje pueda desplazarse en el mundo virtual. Tras realizar un estudio se comprendió que el método por defecto para el movimiento en un entorno de realidad virtual no debería ser este, ya que para la gente que no está acostumbrada y familiarizada con esta tecnología, podría llegar incluso a causarle mareos y náuseas el hecho de que su mente este visualizando un movimiento continuo y sin embargo su cuerpo este parado o sentado en una silla.

Es por ello por lo que se creó el siguiente script, el cual es el encargado a través del menú de opciones del menú principal, el que le permite al usuario intercambiar este modo de movimiento.

Lo que hace técnicamente es modificar la velocidad de movimiento del personaje a un valor de 0, lo cual consigue que, al utilizar el joystick el jugador no se desplace dentro del mundo virtual y por consecuencia no pueda moverse. Seguidamente se activa el objeto de teletransportación explicado en el siguiente apartado el cual contiene todos los componentes para que el jugador pueda desplazarse a una nueva ubicación usando este nuevo mecanismo de movimiento.

Si se desea volver al movimiento continuo, se vuelve a establecer la velocidad inicial y se desactiva el objeto teletransportacion dejándolo inactivo.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class cambioDeModo : MonoBehaviour
6  {
7      private void Start()
8      {}
9
10
11     public GameObject teleportacion;
12     public OVRPlayerController script;
13     //public GameObject jugador = this.gameObject.transform.GetChild(0).gameObject;
14
15     public void movimientoContinuoModo(){
16         //OVRPlayerController script = jugador.GetComponent<OVRPlayerController>();
17         script.Acceleration = 0.1f;
18         teleportacion.SetActive(false);
19     }
20
21
22     public void teleportacionModo(){
23         //OVRPlayerController script = jugador.GetComponent<OVRPlayerController>();
24         script.Acceleration = 0.0f;
25         teleportacion.SetActive(true);
26     }
27 }
28 }

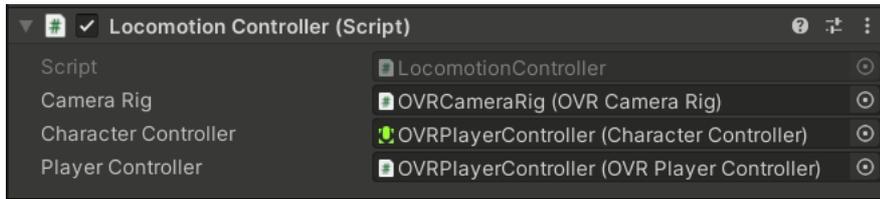
```

### 5.3.2 Movimiento con teletransportación.

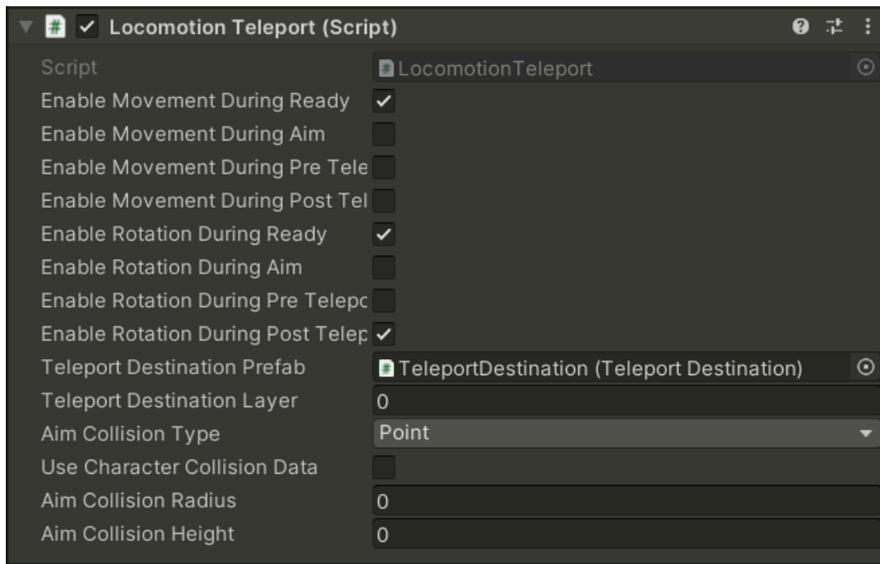
Para conseguir activar este método de movimiento, se desarrolló un nuevo objeto el cual contiene mediante componentes, todo lo necesario para crear un rayo con un indicador que le permite al usuario apuntar a cualquier sitio permitido del mundo virtual cual será su nueva posición dentro de este. Para hacerle más cómoda esta interacción al usuario, se ha generada una capa en la cámara principal de las gafas, con la que la transición a la nueva posición no se hace de forma instantánea y tiene una transición de desvanecimiento, para hacer este salto mucho más sutil y suave.

A continuación, se muestran todos los componentes de este nuevo objeto:

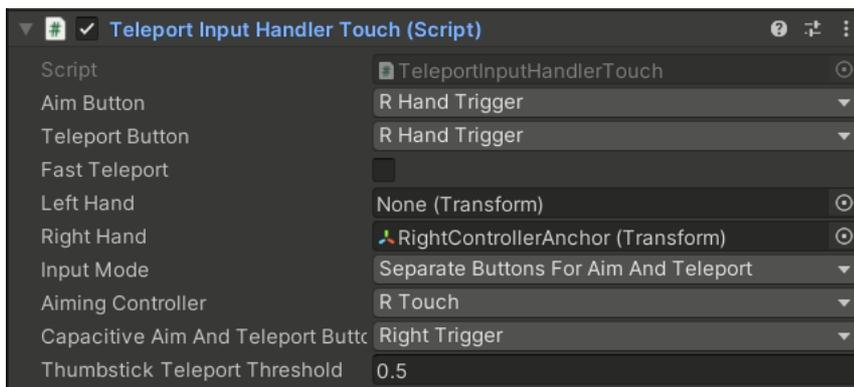
### 5.3.2.1 Locomotion Controller



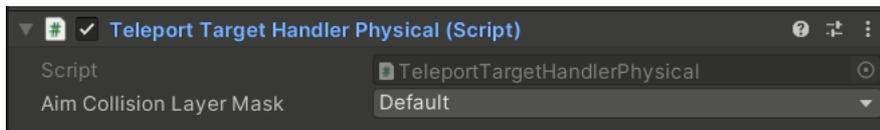
### 5.3.2.1 Locomotion Teleport



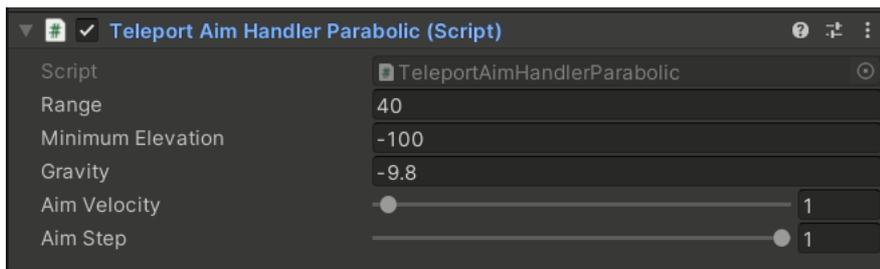
### 5.3.2.1 Teleport input Handler Touch



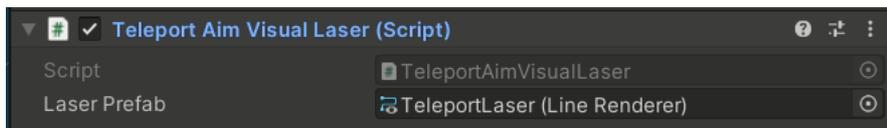
### 5.3.2.1 Teleport Target Handler Physical



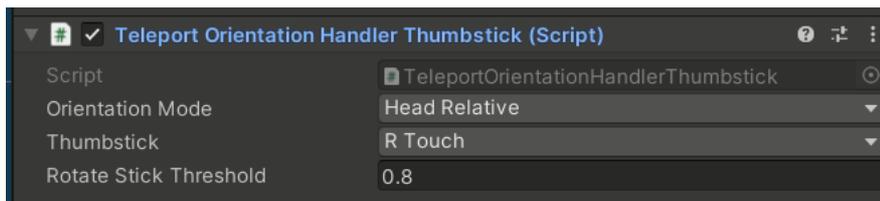
### 5.3.2.1 Teleport Aim Handler Parabolic



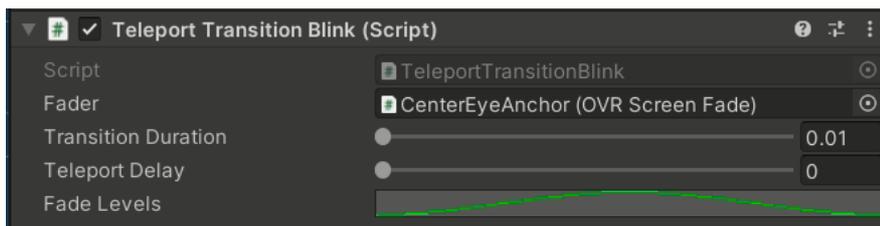
### 5.3.2.1 Teleport Aim Visual Laser



### 5.3.2.1 Teleport Orientation Handler Thumbstick



### 5.3.2.1 Teleport Transition Blink



## 5.4 Creación de menús e interfaces

Sin duda nos encontramos con uno de los elementos más importantes de toda aplicación, los menús e interfaces. Estos elementos son los que funcionan como intermediarios entre el usuario y cualquier aplicación desarrollada. En muchas ocasiones se han dado casos donde empresas descuidan este aspecto fundamental, el cual es el encargado de que el usuario se encuentre cómodo utilizando dicha aplicación y no le resulte frustrante o costoso el uso de esta.

Es por ello por lo que se ha invertido una gran cantidad de tiempo a crear una interfaz realmente atractiva y que a su vez resulte fácil y cómoda de usar para cualquier usuario.

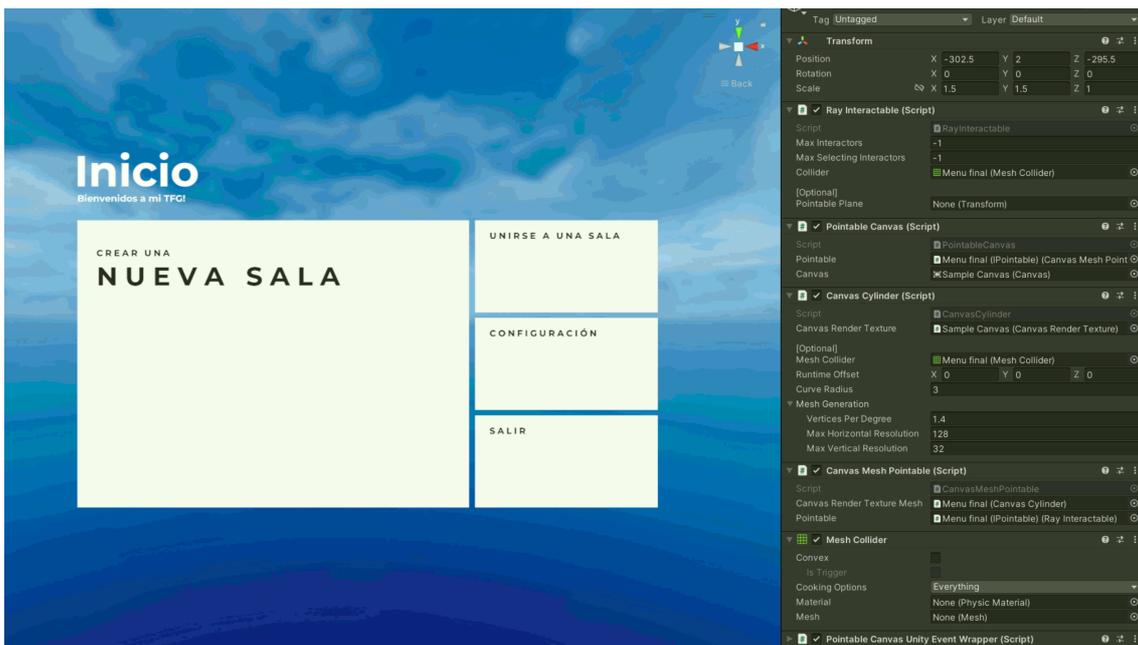
Para poner en valor el trabajo realizado, en primer lugar, se desarrollaron varios bocetos de menús y fueron siendo descartados a medida que se hacían pruebas de usabilidad con varios voluntarios de diferentes edades.

Finalmente, el resultado fue exitoso, ya que, tras hacer pruebas de la aplicación a unos sujetos de: 80 años, 56 años, 24 años y 12 años, todos los voluntarios consiguieron completar todas las tareas que se les demandaron.

Además, entendían correctamente como funcionaba la aplicación sin necesidad de ningún manual de instrucciones.

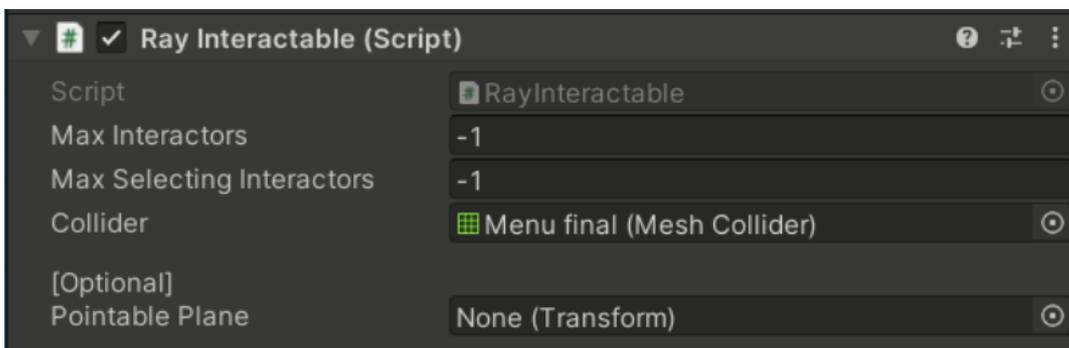
## 5.4.1 Componentes de los menús

Para que los paneles creados funcionaran correctamente en las gafas y el usuario pudiese interactuar con ellos, ya sea mediante en el uso de las manos o de los mandos de Oculus había que agregar una serie de componentes a los objetos que contenían los paneles.

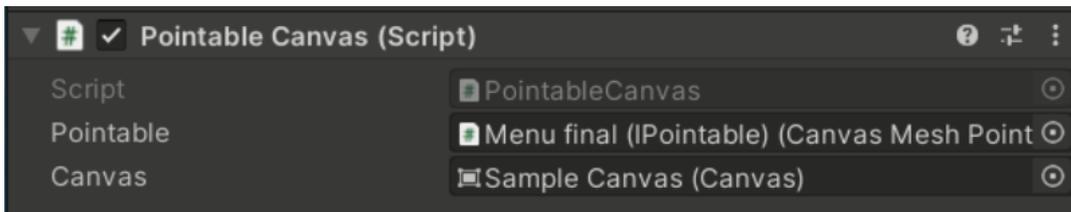


Los componentes utilizados son:

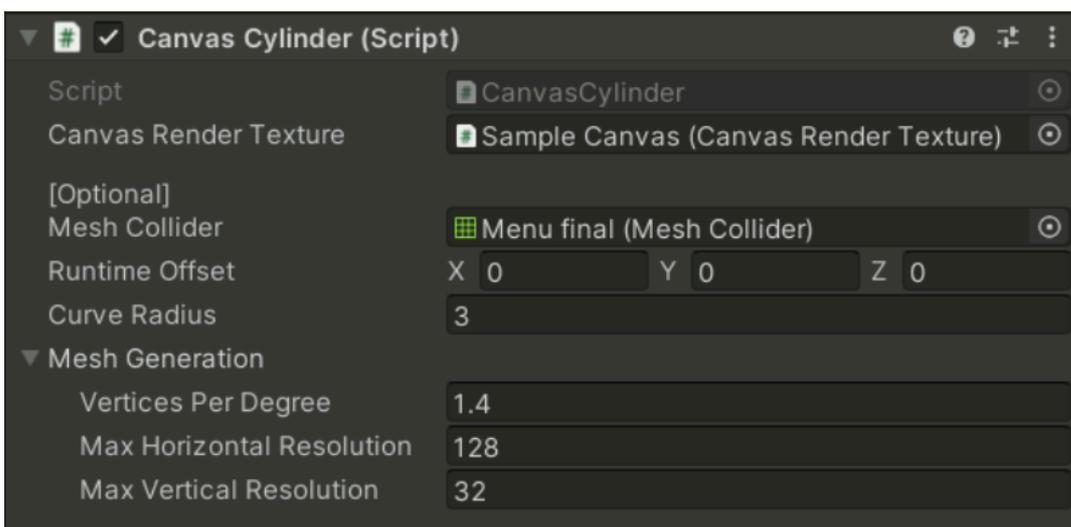
### 5.4.1.1 Ray Interactable



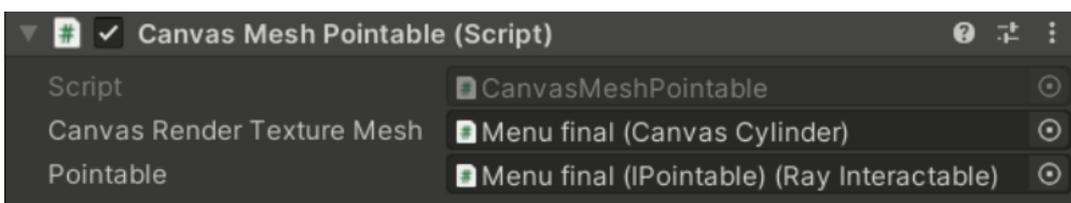
### 5.4.1.2 Pointable Canvas



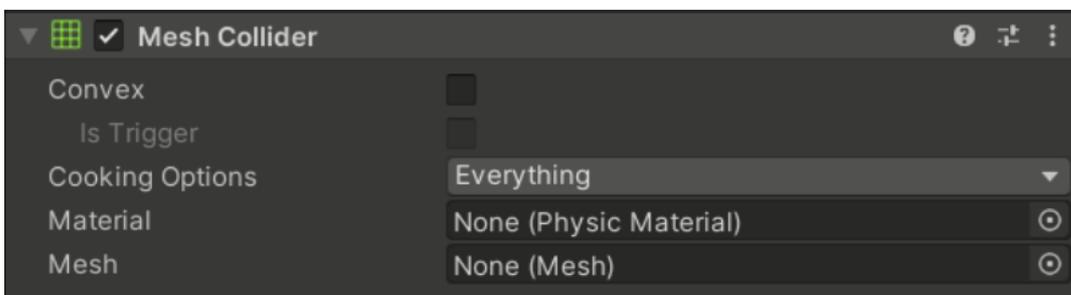
### 5.4.1.3 Canvas Cylinder



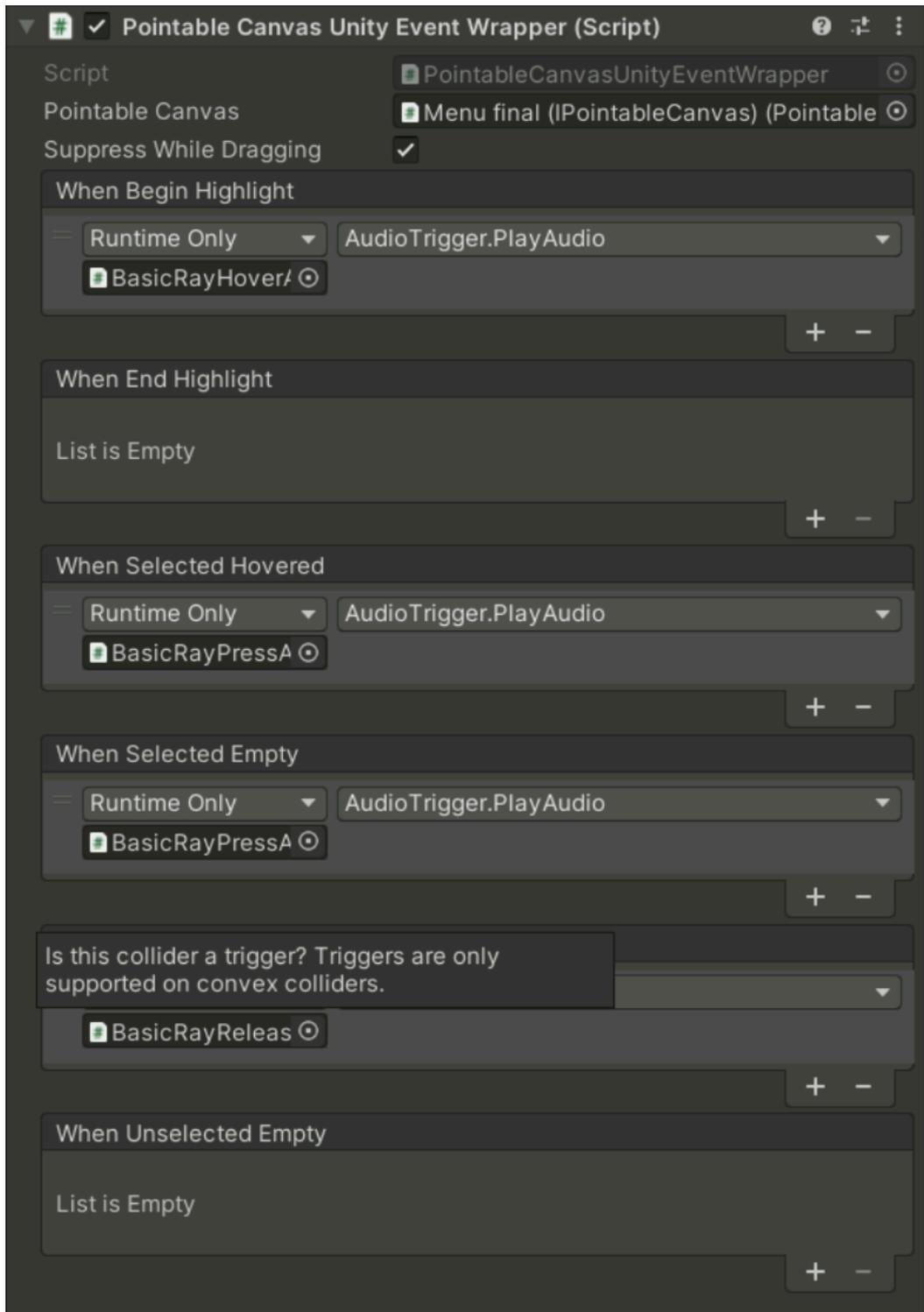
### 5.4.1.4 Canvas Mesh Pointable



### 5.4.1.5 Mesh Collider



### 5.4.1.6 Pointable Canvas Unity Event Wrapper



Todos estos componentes trabajan conjuntamente para que la interacción sea posible, desde que se muestre el raycast que es el rayo que aparece cuando el jugador apunta a cualquier panel con el que pueda interactuar, controlar el audio que se genera cuando un botón es pulsado, hasta la renderización del mismo menú.

Una de las cosas que se debe tener en cuenta a la hora de desarrollar los menús, es que, si se desea que el usuario introduzca información a la aplicación mediante un campo de texto, se debe seleccionar el objeto "Input TextMeshPro" ya que es la tecnología más novedosa y es la que no tiene ningún tipo de inconveniente con la realidad virtual.

Este error se descubrió cuando se trató de utilizar la tecnología estándar y que, a la hora de intentar introducir cualquier dato, conllevaba que el teclado virtual de las gafas no se mostrase y por consecuencia al usuario no se le permitía poder introducir la información.

## 5.4.2 Diseño final de los menús

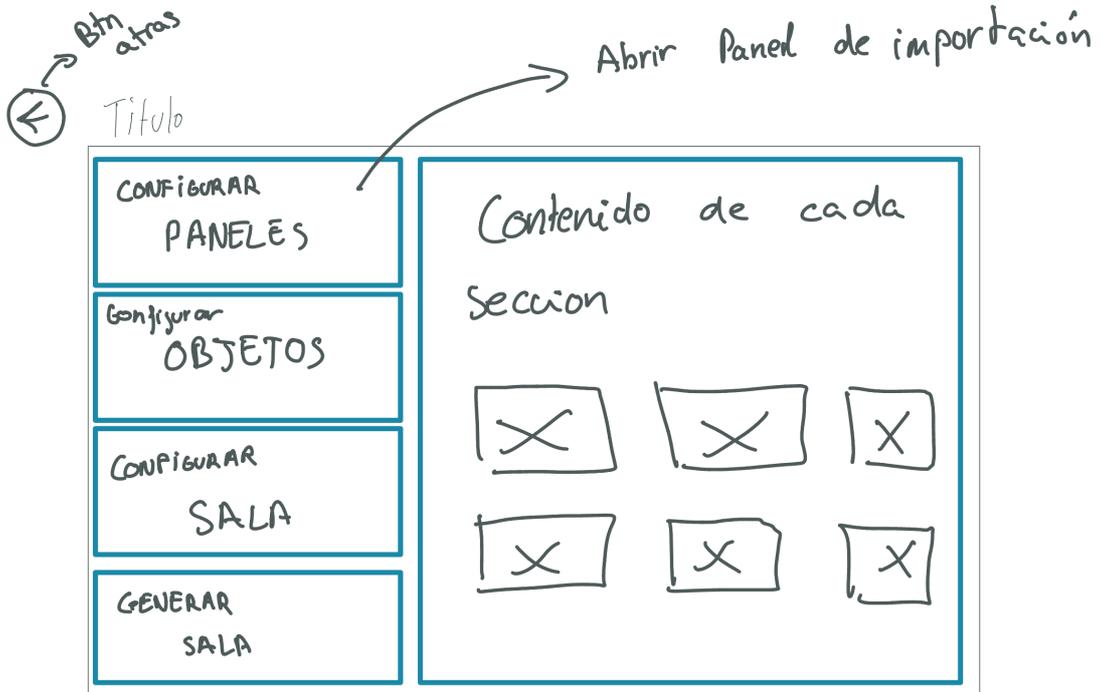
A continuación, se muestran unas capturas de los bocetos que pasaron exitosamente todas las pruebas de usabilidad y cual ha sido finalmente el resultado dentro la aplicación.

# MENÚ PRINCIPAL

Titulo Menú principal

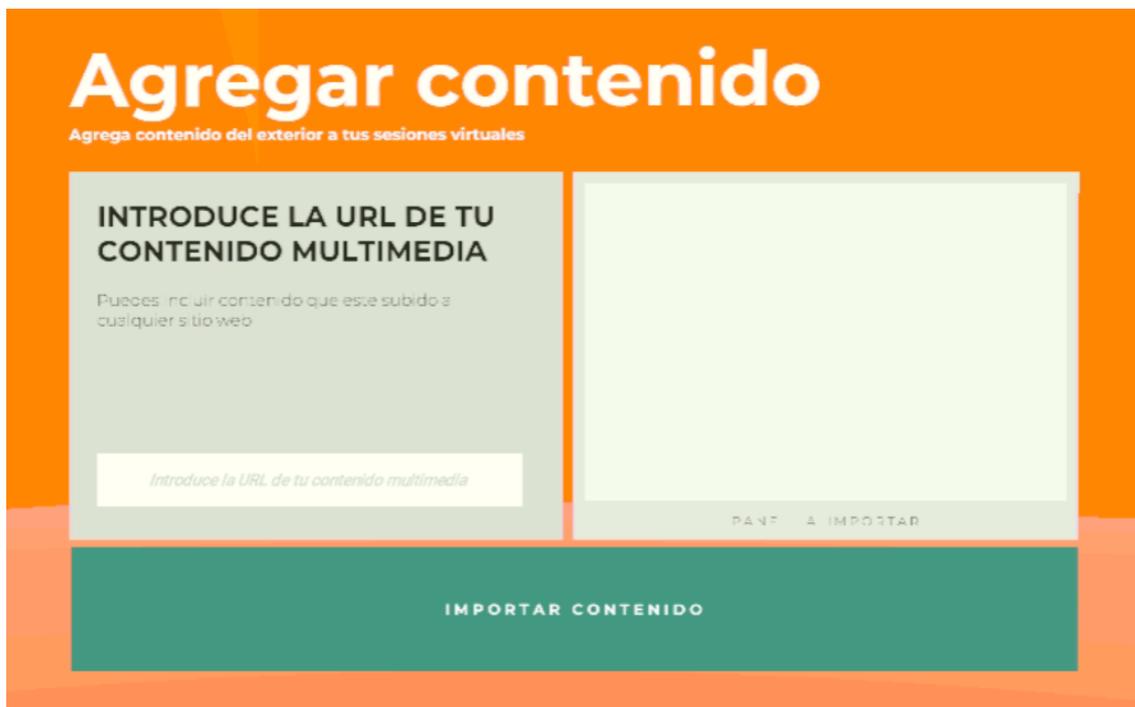
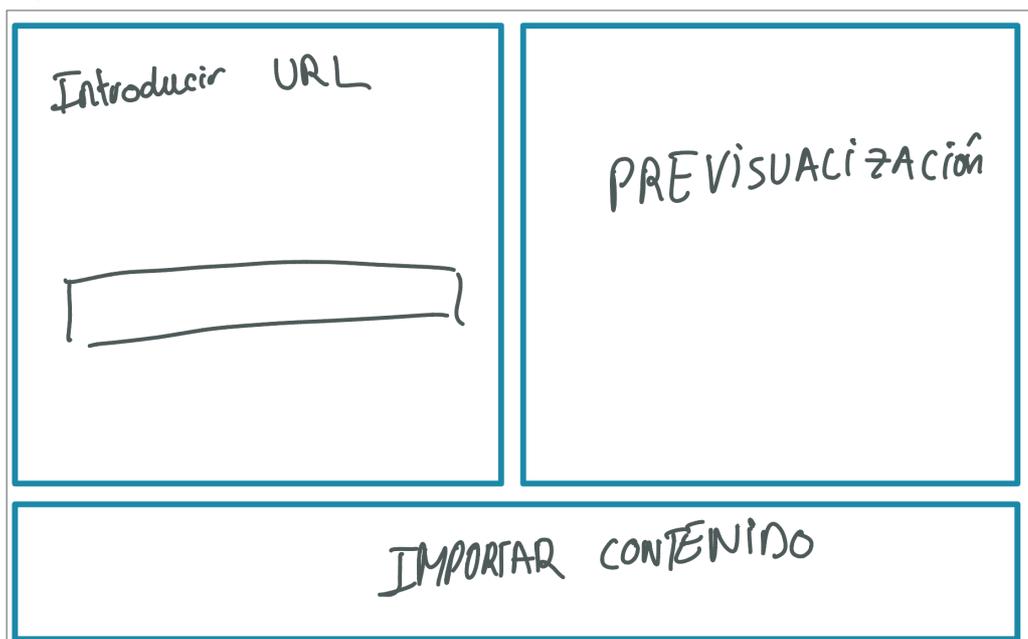


# Nueva sala



# Panel de importación

Título



# Unirse a una sala (Multijugador)

Dtn  
atras  
Titulo

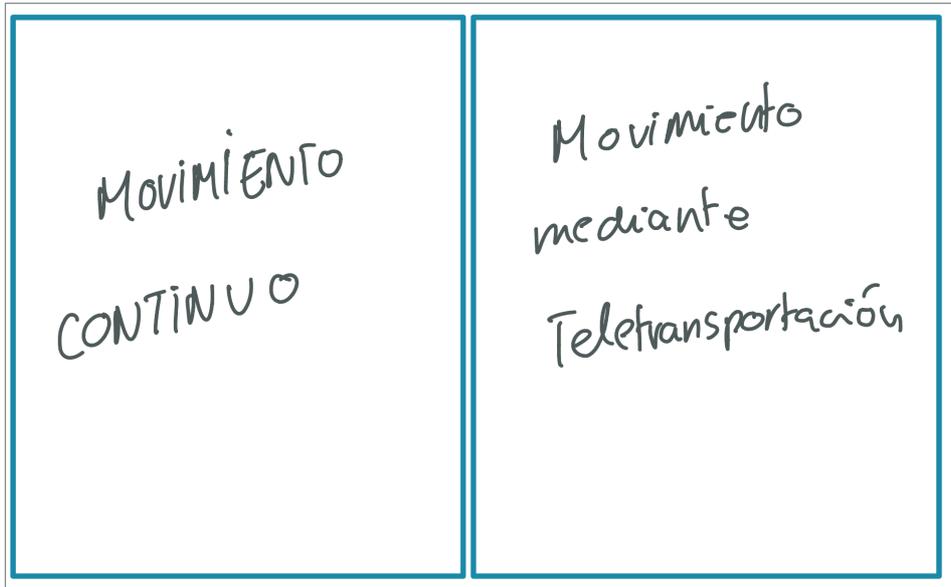
INTRODUCIA EL CÓDIGO

CONECTARSE



# Opciones de movimiento

Dtn  
atras  
Titulo



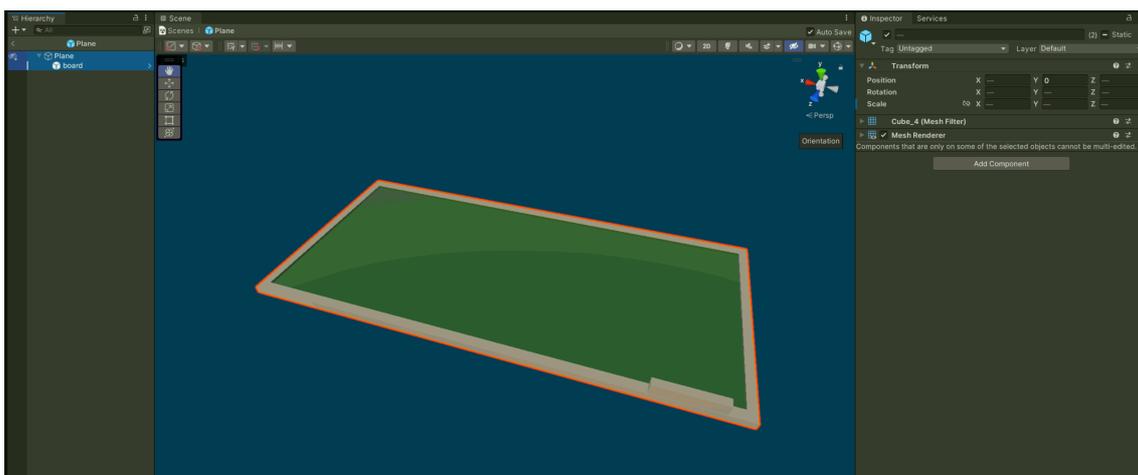
## 5.5 Creación de los paneles

La finalidad del siguiente componente reside en incorporar el material de apoyo más común que se tiene hoy en día a la hora de impartir una formación, llamadas diapositivas o transparencias.

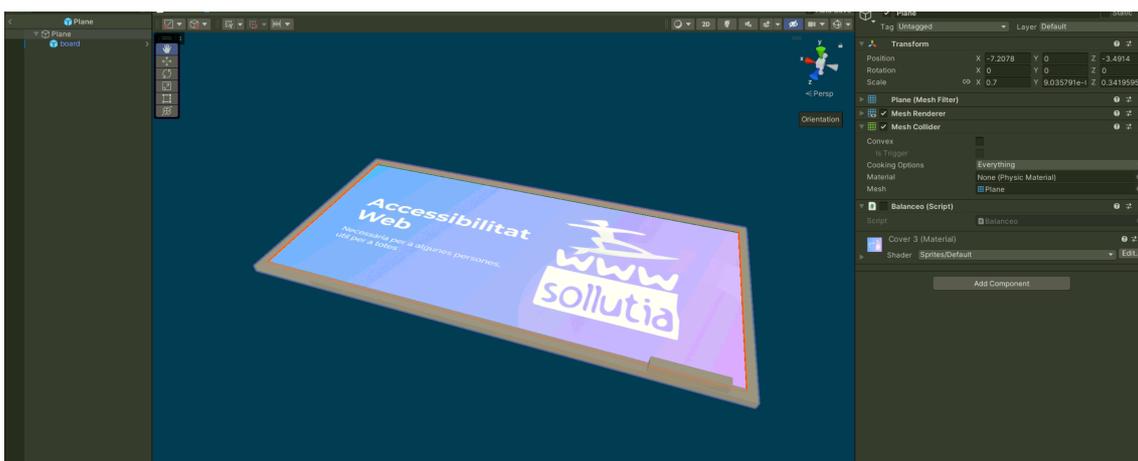
Una diferencia significativa al resto de aplicaciones ya existentes en la actualidad, ya que, tras el estudio que se realizó, se llegó a la conclusión de que dichas aplicaciones te permiten generar tus propias salas, pero con una limitación, la cual consiste en que los materiales y elementos que están a tu disposición han tenido que ser programados previamente y la posibilidad de importar tu propio material no está disponible.

Es por ello por lo que la característica de proporcionarle al usuario que pueda incorporar su propio material, con el que está familiarizado a la hora de trabajar, sería sin duda una característica muy distintiva de este proyecto.

Para desarrollar este apartado, en primer lugar, se creó un prefab cuya función es ser la base de todos los paneles que en un futuro se generarán de forma dinámica. Este prefab contiene una estructura exterior simulando un elemento muy cotidiano en toda formación, una pizarra.



Más tarde en el lugar del típico tablero verde que todos conocemos se ha posicionado un plano en el eje Y, justo encima de este tablero y con las mismas dimensiones que la pizarra. Este elemento será, con el que gracias a un script podremos manipular y conseguir editar el material con el que está hecho y será este mismo material el que contenga la imagen que el usuario ha proporcionado (su diapositiva).



## 5.6 Creación de los objetos interactivos

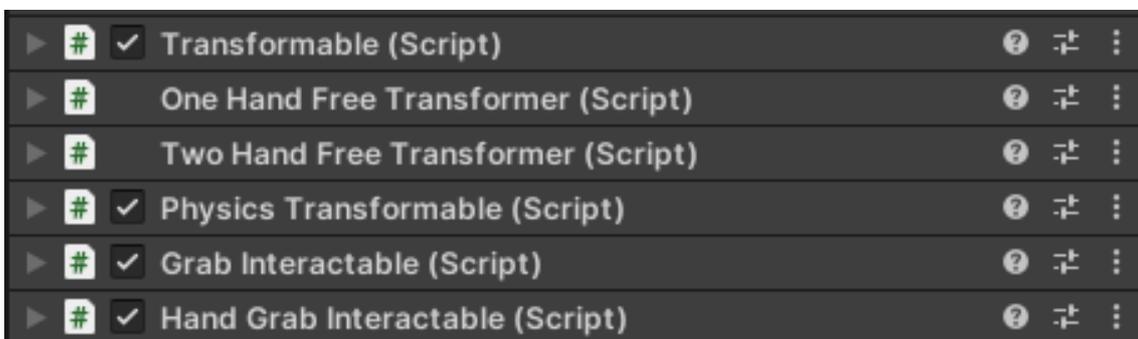
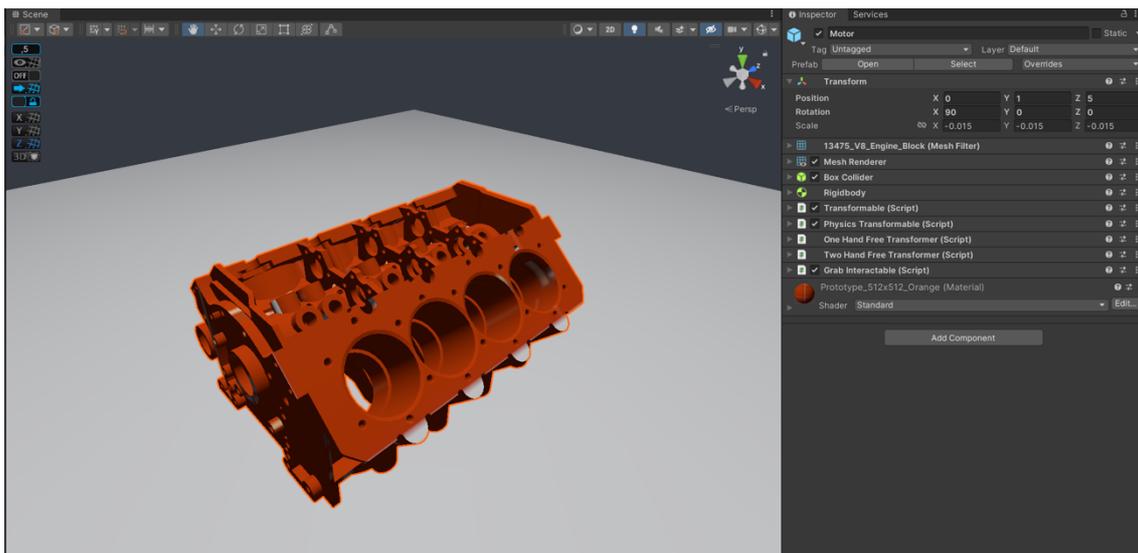
Quiero recordar que la intención de estos objetos consiste en poder manipularlos en cuanto a escala, tamaño y posición dentro de nuestro mundo virtual, de esta forma, objetos diminutos como podrían ser la cadena de huesecillos del oído interno podrían ser ampliados a un tamaño mucho mayor para poder estudiarlos con mayor detalle y precisión. También se nos podría plantear la posibilidad de mover objetos que en la vida real serían impensables como podría ser una estructura de un puente o objetos sumamente pesados como un motor de un gran vehículo que requerirían de maquinaria especializada para completar dicha tarea. Las increíbles ventajas que esto nos aporta frente al mundo real hicieron que la característica de los objetos interactivos fuera un requisito fundamental para el desarrollo de este proyecto.

A la hora de crear los objetos interactivos, primero se debía investigar cuales eran los elementos necesarios para que estos funcionasen como se esperaban.

Afortunadamente respecto a este tema podemos encontrar una gran cantidad de resultados en internet y junto a la documentación oficial de Oculus fue información más que suficiente para el desarrollo de esta característica.

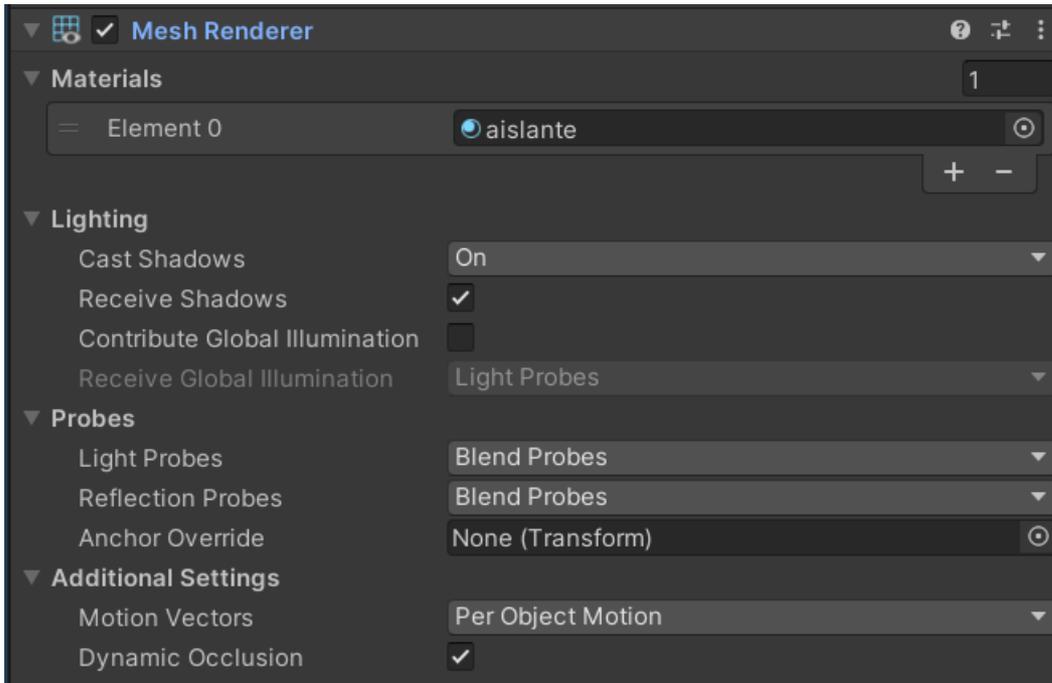
### 5.6.1 Componentes de los objetos

En la siguiente imagen se muestra un ejemplo de un objeto para observar cual será la estructura final de este. Seguidamente se explicará cada uno de los componentes que lo componen y por último un listado con todos los objetos que se han creado hasta día de hoy para este proyecto.

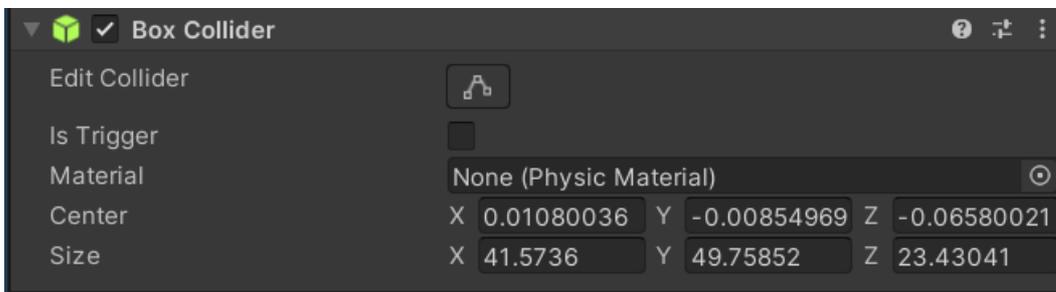


Como se puede observar los componentes que están agregados a cada objeto 3D son los siguientes:

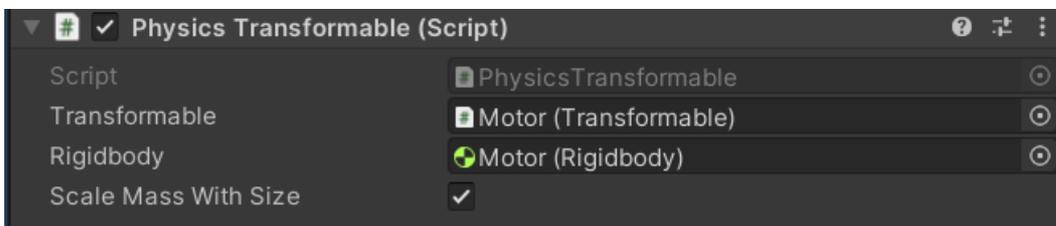
### 5.6.1.1 Mesh Renderer



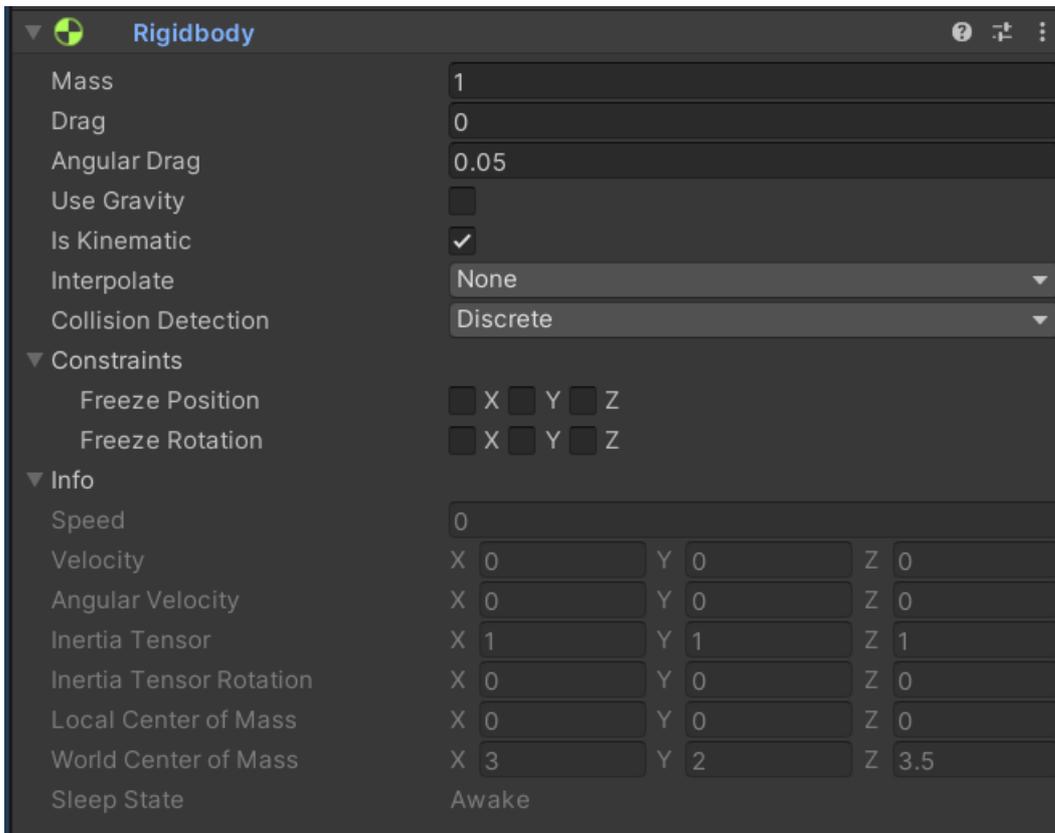
### 5.6.1.2 Box Collider



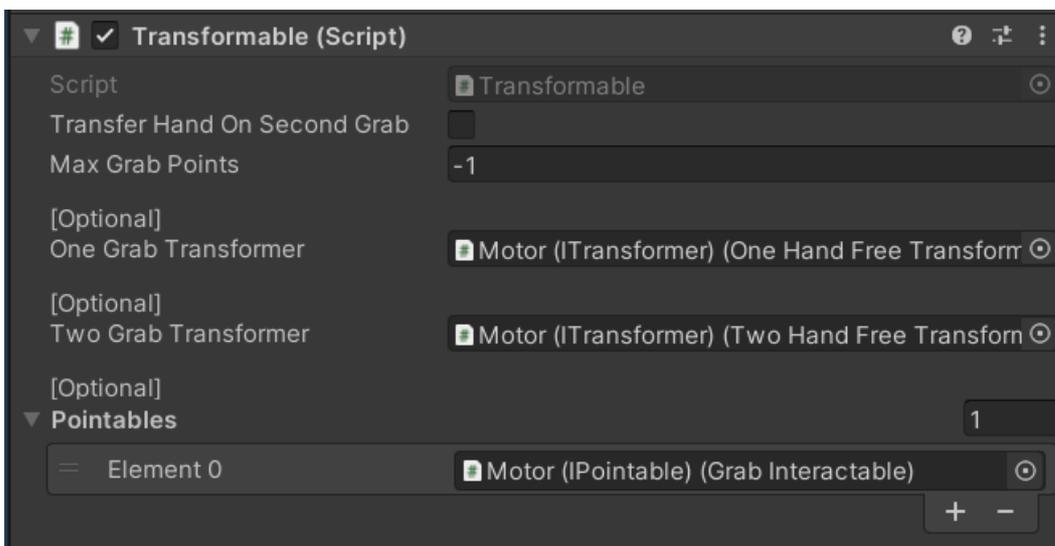
### 5.6.1.3 Physics Transformable



### 5.6.1.4 RigidBody



### 5.6.1.5 Transformable



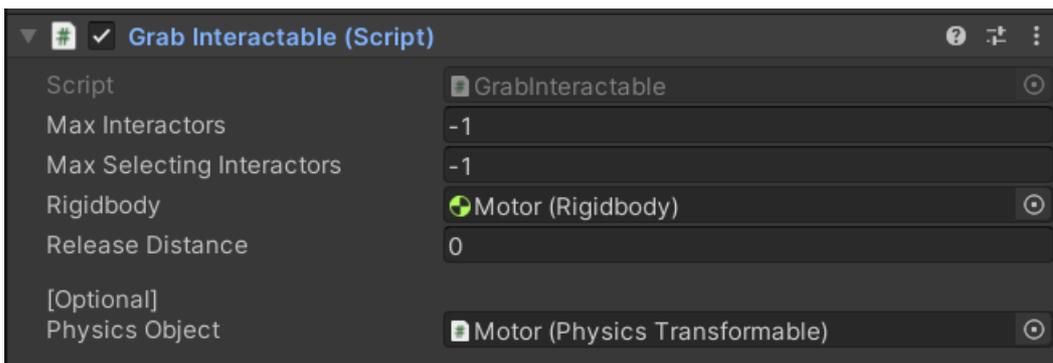
### 5.6.1.6 One hand free transformer



### 5.6.1.7 Two hand free transformer



### 5.6.1.8 Grab or Hand interactable

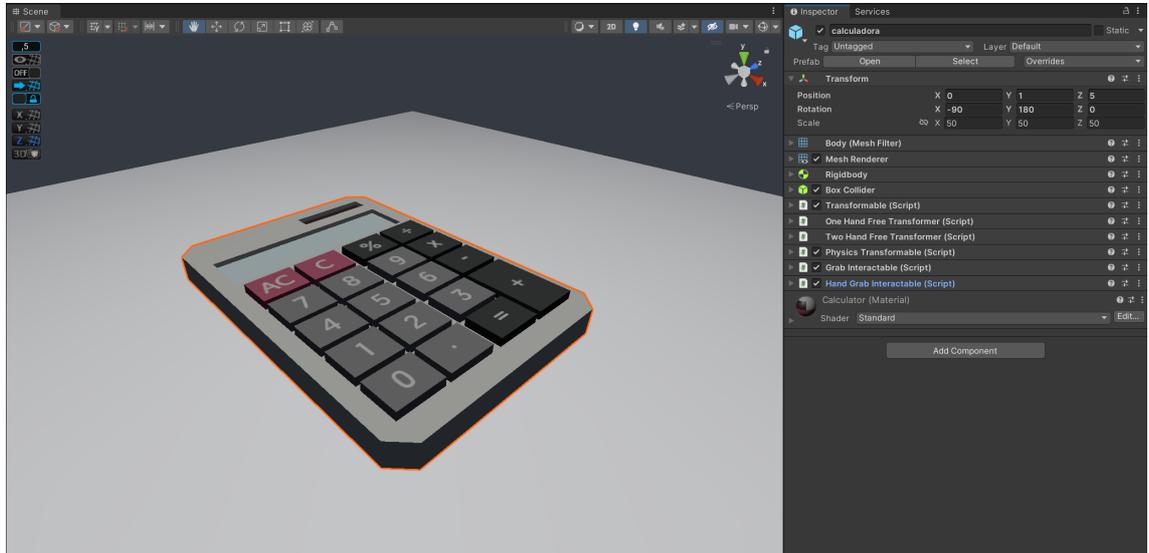


Hay que destacar que todos los objetos pueden ser manipulados tanto con las manos como con los mandos de Oculus, siempre y cuando se le introduzcan ambos componentes "Grab Interactable" y "Hand Interactable".

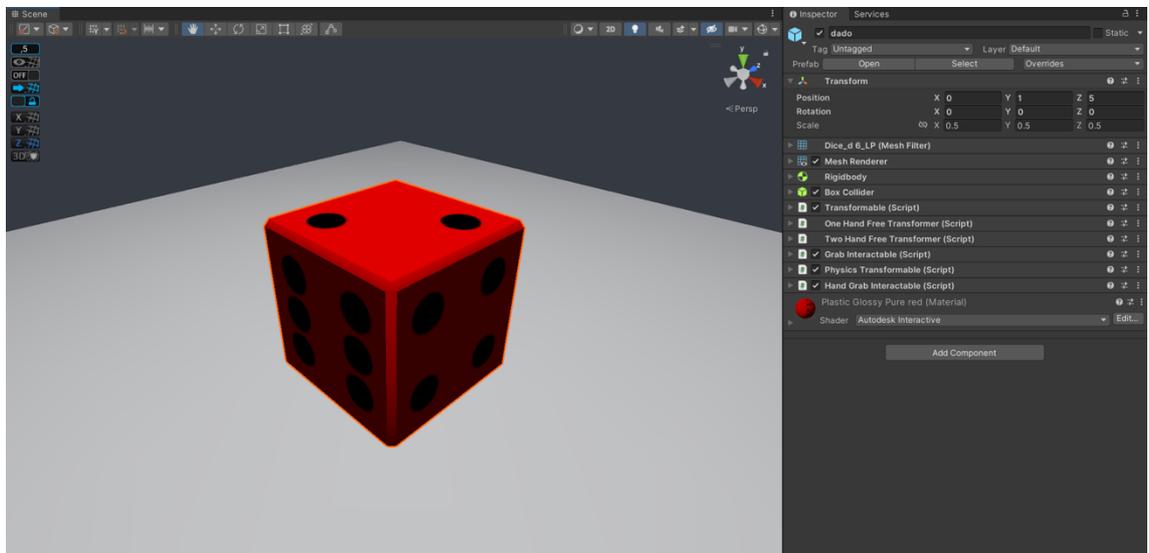
Todos estos componentes son necesarios para que el objeto sea interactivo y se puedan manipular.

## 5.6.2 Listado de los objetos

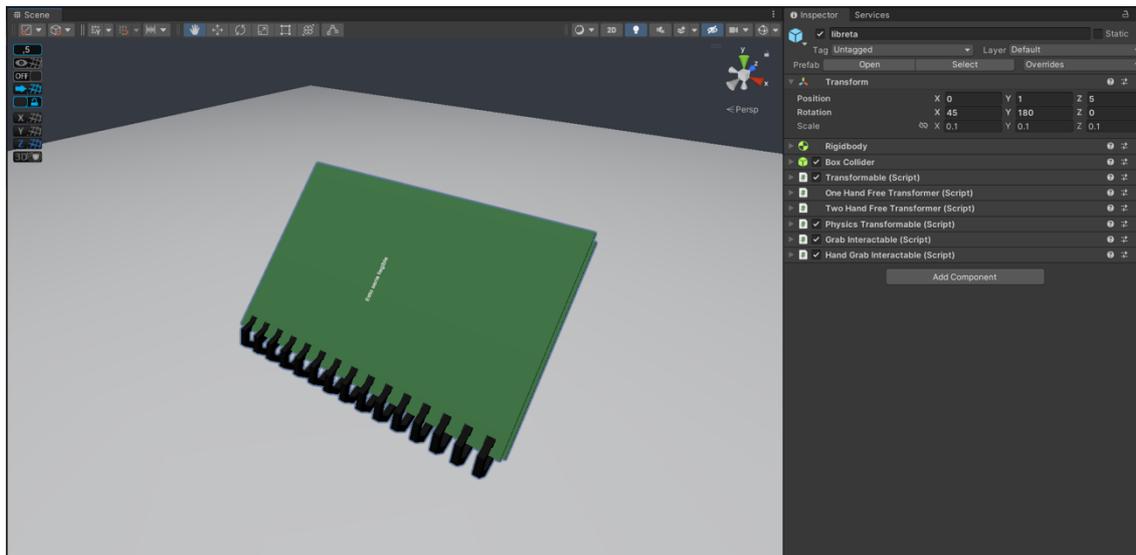
### 5.6.2.1 Calculadora



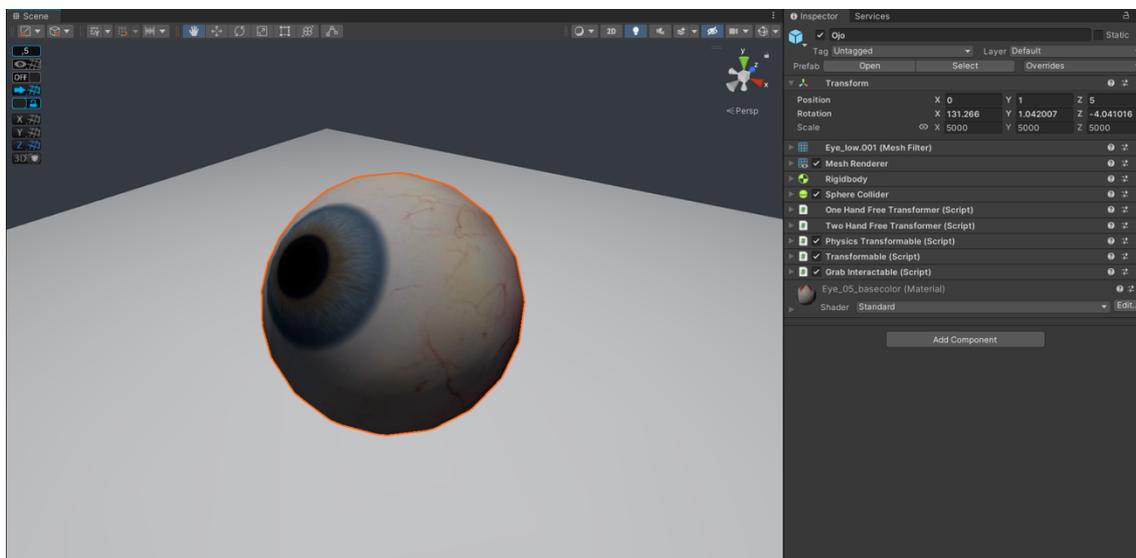
### 5.6.2.2 Dado



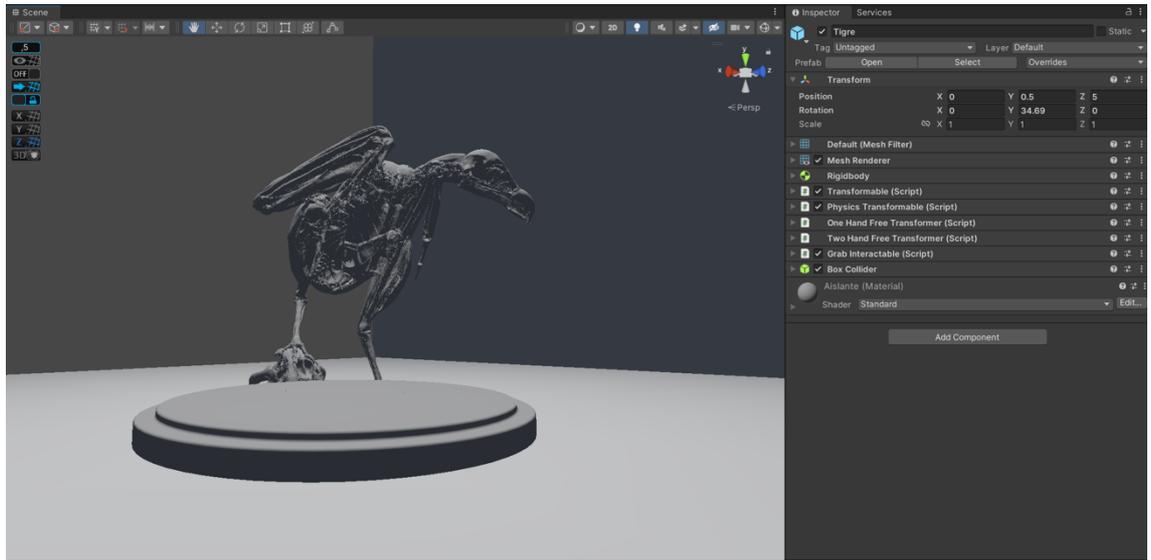
### 5.6.2.3 Libreta



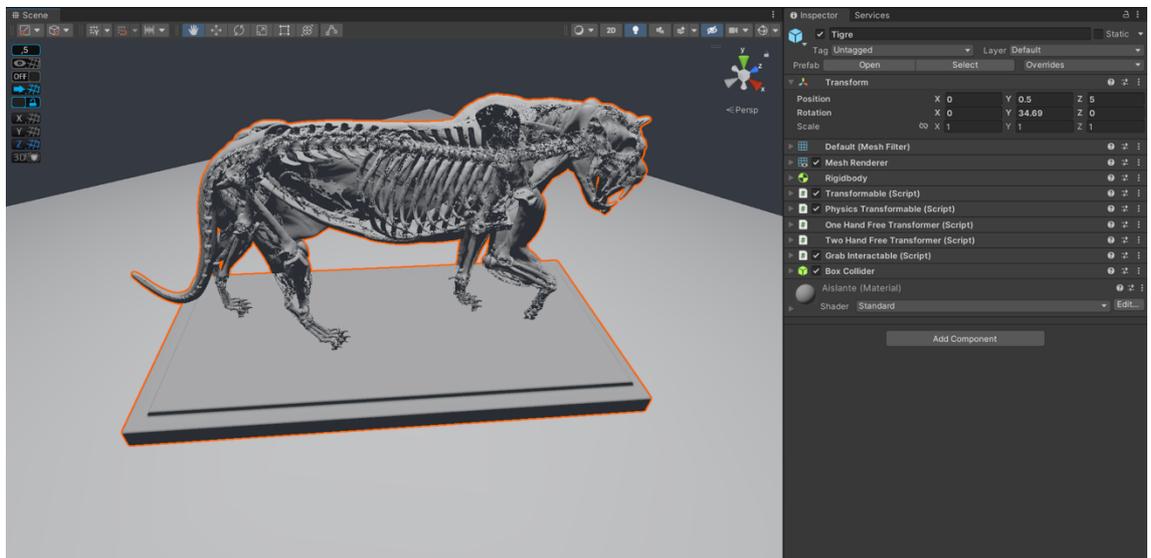
### 5.6.2.4 Ojo



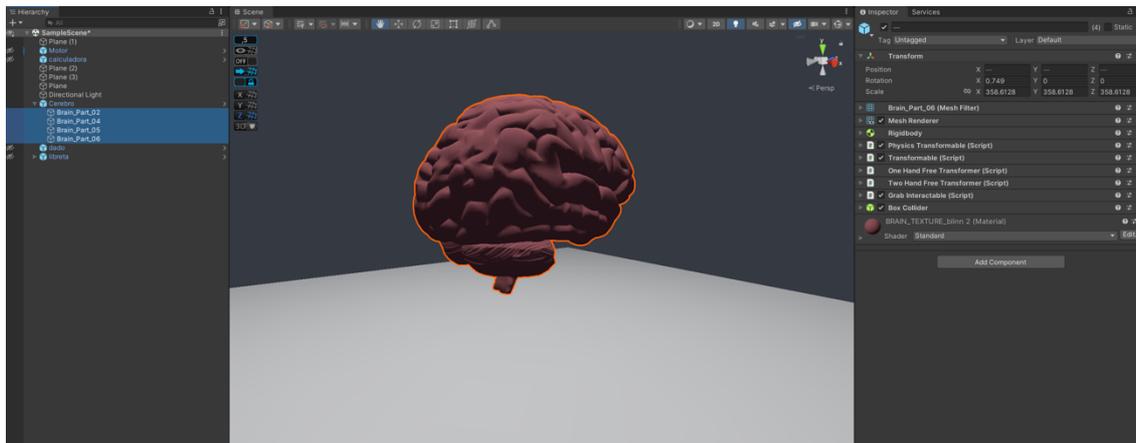
### 5.6.2.5 Pájaro



### 5.6.2.6 Puma



### 5.6.2.7 Cerebro



Este objeto en particular consta de 4 partes, se pretendía experimentar como funcionaria la idea de dividir el objeto para tener un mayor control sobre el objeto. Para conseguir esta idea, cada una de las partes que componen el objeto tienen que contener todos los componentes mencionados en el apartado anterior.

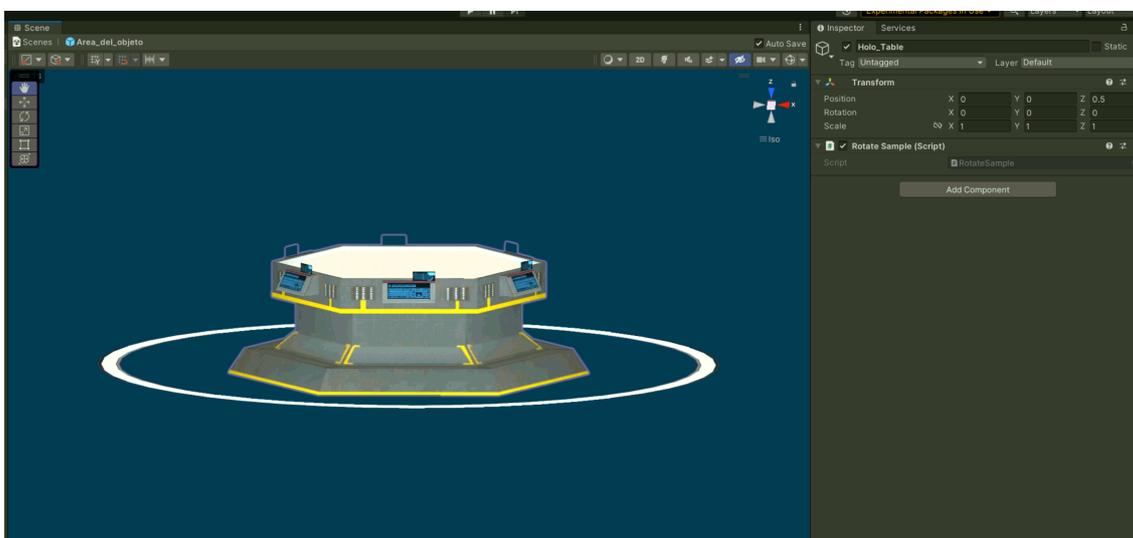
Finalmente resultó ser una idea muy práctica y funcional, pero también cabe destacar que requiere un elemento 3D más complejo.

En el desarrollo de este apartado se trató de implementar el mismo concepto que en el apartado anterior, ofrecerle la posibilidad al usuario importar sus propios elementos 3D, pero al no ser un elemento tan cotidiano ni tan estandarizado, la cantidad de problemas que esta idea conllevaba, hacían que este planteamiento fuese muy complicado e inviable con los mecanismos y tecnologías de hoy en día.

### 5.6.3 Soporte de los objetos

Este elemento es simplemente decorativo. La finalidad de este soporte para los objetos consiste en darle un poco más de coherencia al proyecto y acercar al usuario a un entorno más cotidiano para él, en donde los elementos, en lugar de estar flotando en el vacío, se consigue que aparezcan en unas mesas, lo cual lo convierte en una situación más real.

En la siguiente captura se muestra una captura del prefab creado que más tarde se posicionara mediante un script debajo de cada uno de los objetos.



Un elemento extra que se decidió aportar a este objeto, y es la razón por la que se agregó el asset Itween, consistió en agregarle un movimiento continuo a la mesa.

Remarcar que esta característica no aporta nada al funcionamiento de la aplicación, es decir, es simplemente decorativa e incluso podría ser eliminada en caso de afectar drásticamente al rendimiento de la aplicación.

Este movimiento como se puede observar en la captura se ha introducido mediante un script llamado "RotateSample", a continuación, podemos observar su contenido:

```

1  using UnityEngine;
2  using System.Collections;
3
4  public class RotateSample : MonoBehaviour
5  {
6      void Start(){
7          iTween.RotateTo(gameObject, iTween.Hash
8      ("z", 180, "easeType", "linear", "loopType", "pingP
9      ong", "time", 15, "delay", 0));
10     }
11 }

```

Como se puede observar utilizamos iTween con los parámetros correspondientes para que el movimiento de la mesa sea el esperado en el proyecto.

En esta función en concreto de iTween podemos especificarle parámetros como: eje de rotación, grados, tipo de animación, tiempo de espera, tiempo de rotación...

## 5.7 Creación de salas

### 5.7.1 Planteamiento y estructura

En este punto del proyecto, ya se tenía creado todos los prefabs de los objetos y paneles y solo quedaba darles forma dentro del entorno virtual, inicialmente la idea era distribuir todos estos objetos de forma lineal sobre un plano en el eje X como si fuese una carretera infinita con todo el material que el usuario había seleccionado. Después de valorarlo se consideró que, si estamos creando un mundo virtual, no tiene porque parecerse a la realidad y podría crear un par de salas (que en un futuro se podrían incrementar) donde el usuario pueda encontrarse más cómodo y entornos diferentes para evitar la monotonía y además quedaría un entorno virtual más realista y elaborado.

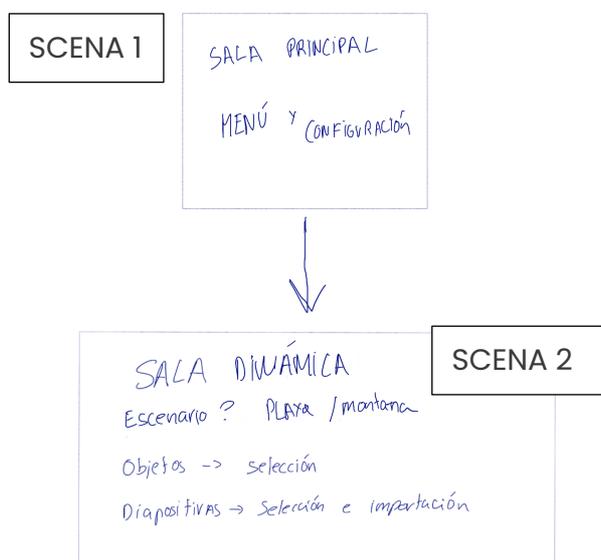
Después de decidir que se iban a crear dos entornos diferentes solo se planteó inicialmente una forma de gestionarlo, pero tras estudiar más a fondo el objetivo final del proyecto, se plantearon diversas opciones con el fin de evitar futuros problemas.

En primer lugar, consistía en crear 1 escena de forma inicial y mediante código generar una segunda escena de una sala dinámica dentro del mismo proyecto, organizadas de la siguiente forma:

- 1- **Escena del menú principal:** En esta sala es donde aparecería inicialmente el jugador una vez abriese la aplicación y todo lo que hubiese serían los menús para interactuar con el entorno y configurar la sala virtual que el usuario quisiese. Según las opciones seleccionadas por el usuario, existiría un manejador de eventos el cual

sería el encargado de recoger la información proporcionada por el jugador y generar de forma dinámica la sala escogida. De esta forma se crearía una única sala con los elementos justos y necesarios, para preparar el entorno.

- 2- **Escena dinámica final:** Esta sería la escena generada desde la escena anterior donde únicamente apareciese el escenario seleccionado y las ubicaciones de los objetos y paneles deseados.



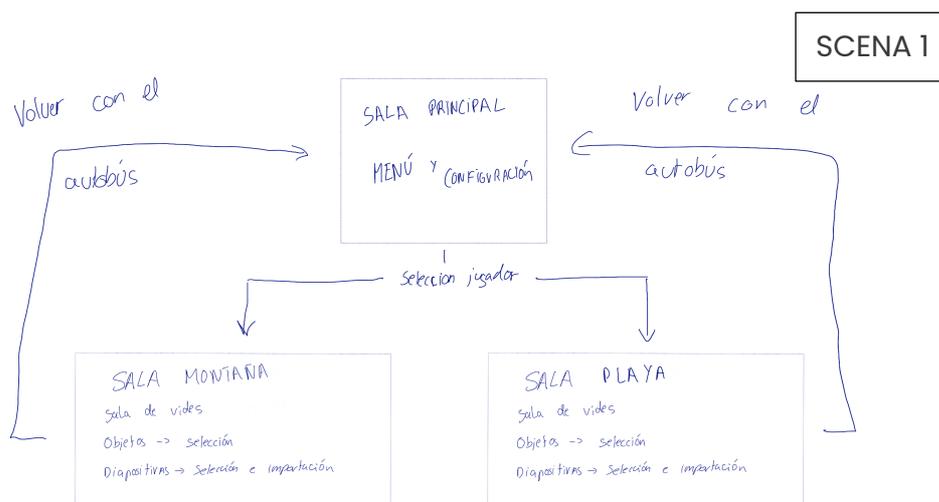
Finalmente, esta opción fue descartada, ya que, aunque se podría obtener alguna mejora de rendimiento, al tratar de sincronizar toda la información para el sistema de multijugador y utilizando el paquete de Netcode iba a conllevar muchos problemas.

La segunda opción, que es la solución más planteada por internet para lidiar los problemas de la sincronización de información, en el entorno multijugador y que además fue la opción recomendada por mi tutor (Jorge Orta), consistía en lo siguiente:

1- **Una única escena en todo el proyecto:** Donde generaría 3 habitaciones diferentes con la base de cada escenario deseado.

- a. Un menú principal.
- b. Un escenario de montaña.
- c. Un escenario de playa.

Una vez creada las tres habitaciones las debería envolver en una esfera hueca de forma que, a vista del usuario, él tuviese la sensación de estar en una escena independiente a la anterior, pero realmente en tiempo de ejecución la escena del proyecto es la misma y gracias a esto sería mucho más sencillo controlar y sincronizar toda la información generada en todas las habitaciones



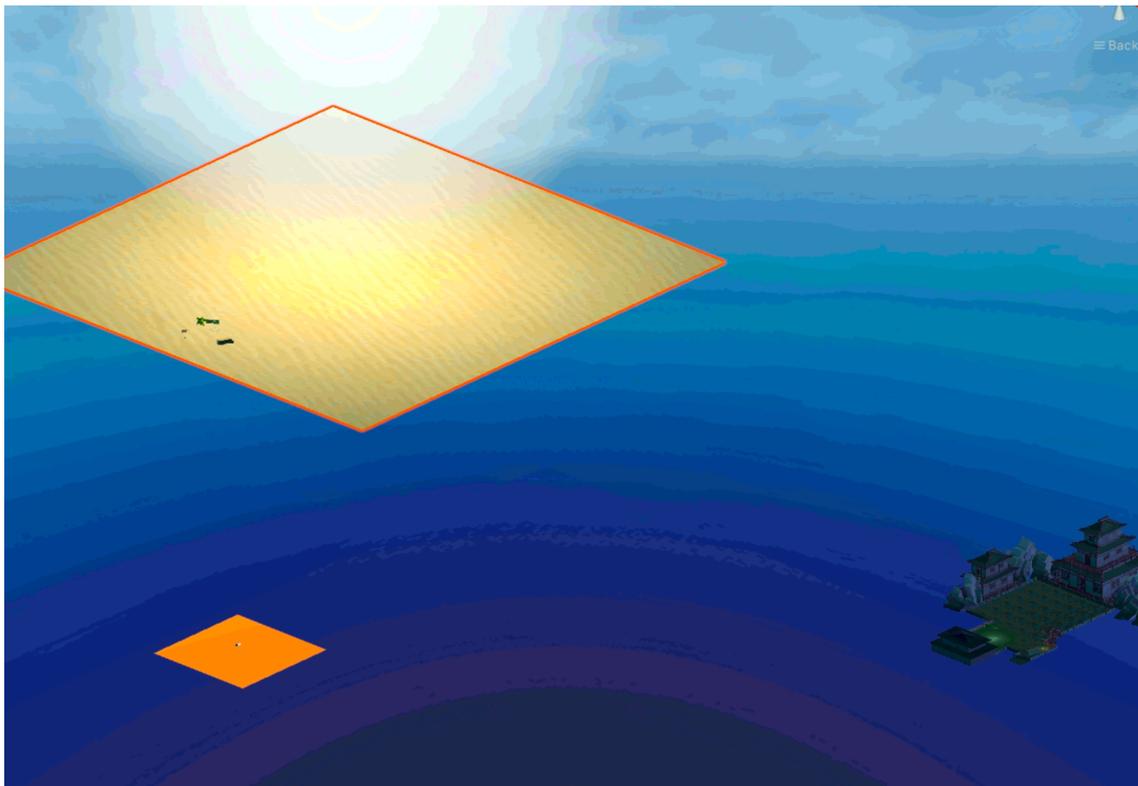
Finalmente decidí escoger esta opción por las ventajas aportaba y se reestructuró el proyecto en base a esto.

## 5.7.2 Creación de las salas.

En cuanto a la creación de las habitaciones los siguientes elementos son comunes a las tres:

- Un plano en el eje X: Es el plano que actúa como suelo de las habitaciones y es el espacio por donde el jugador se puede desplazar.
- Una zona de aparición: Es un prefab creado con el objetivo de poder recoger la información que contiene el "transform", de forma que al haber un objeto en cada habitación puedo recoger la información de forma más organizada y así conocer la posición a la que habrá que transportar al jugador una vez seleccione su nueva ubicación. Consta de un aro de luz en el suelo para que se muestre de forma visual cual ha sido el lugar de aparición y una luz focal que ilumina el terreno dando una sensación de inmersión.
- Una esfera hueca: Esta ha sido la forma de conseguir que las habitaciones parezcan independientes ya que esta esfera funciona de aislante. Para conseguir dicho efecto he necesitado crear un material con la imagen de un cielo, aunque esto, por sí solo era insuficiente, ya que, la imagen solo se aplicaba por el lado exterior de la esfera y desde dentro continuabas pudiendo ver las demás salas. Para resolver esto he utilizado el "asset" (free-double-sided-shaders) detallado anteriormente en la sección de "assets" y lo que conseguimos con él es que la imagen aplicada al material se visualice por los dos lados de la esfera y de esta forma conseguimos que sea completamente opaca y obtenemos el resultado deseado.

Finalmente, cada una de las habitaciones contienen elementos extras aportados por "assets" de objetos, para una mejor ambientación, donde una de ellas está basada un paisaje más rustico, de montañas y casas mientras que la otra habitación nos traslada a una época veraniega simulando una playa.



En esta captura se representa la vista de la única escena existente en el proyecto.

- El cuadrado naranja es la habitación la cual pertenece al menú principal.
- El plano amarillo es el escenario de la playa
- Y lo que se puede visualizar en el lado inferior derecho de la imagen, es la habitación que simula un ambiente más rustico.

Cabe destacar que las esferas que delimitan las habitaciones han sido ocultadas para tener una vista más clara de como están situadas dichas habitaciones en el escenario y poder diferenciarlas.

## 5.8 Creación del sistema multijugador

Lamentablemente esta característica finalmente no se ha podido incorporar en el proyecto definitivo, debido a que se ha intentado utilizar una tecnología muy novedosa y que está en una versión muy temprana de desarrollo.

Aunque si que es cierto que se han hecho pruebas en otros tipos de proyectos, como juegos 3D con la intención de seguir los tutoriales y documentaciones que por el momento existen, donde todo el sistema de Netcode funcionaba de forma correcta.

A la hora de instanciar jugadores en la realidad virtual con todos los componentes necesarios que conllevan, daba como resultado un comportamiento negativo haciendo que la aplicación lanzará un error y forzase la salida de esta.

Es por ello por lo que, aunque se ha invertido posiblemente la mayor cantidad de tiempo en intentar incorporar esta característica al proyecto debido a la gran utilidad que esta podría aportar, finalmente, se ha prescindido de ella ya que la versión de Netcode a fecha de hoy se encuentra en una fase muy temprana de su desarrollo y no se ha podido incorporar en este mismo proyecto de realidad virtual.

Para compensar la falta de esta característica, se decidió implementar un visualizador de videos para poder ver cualquier contenido de internet ya existente que esté en Youtube.

## 5.9 Implementación del visualizador de videos.

Para implementar esta sección, utilice un mismo panel como si de un menú se tratase y por consecuencia contiene los mismos componentes anteriormente mencionados para que la interacción con el jugador sea posible. También se ha añadido un menú complementario justo al lado del reproductor para controlar el video, poder pararlo, reproducirlo, volver a empezar el video e importar el video a través del enlace de YouTube.

Para poder incorporar esta funcionalidad al proyecto ha sido necesario incorporar el paquete “Youtube Player” el cual nos proporciona los scripts necesarios para que la importación funcione correctamente.

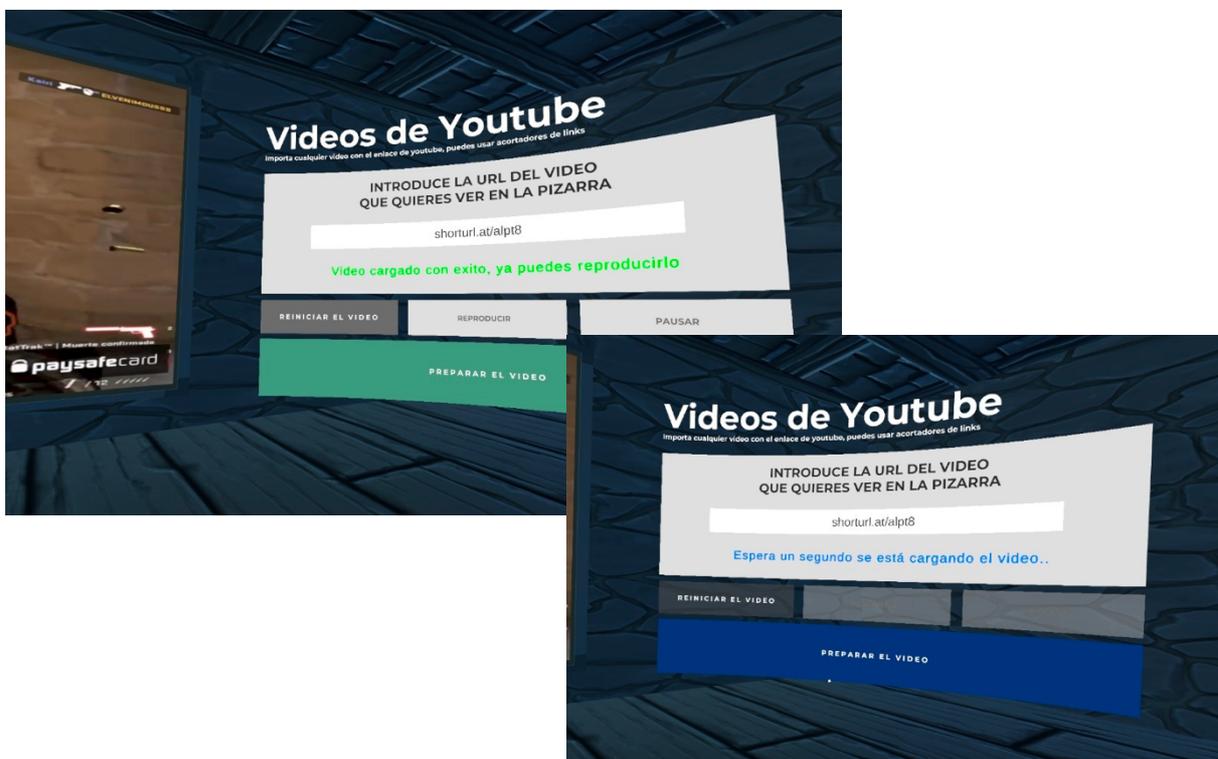
Al igual que en el caso de los paneles, donde el material es modificado mediante el script, esto funciona de la misma forma, pero salvando las diferencias, ya que, para poder visualizar el contenido dinámico, en lugar de un material sencillo de Unity es necesario crear un “Render Texture”.

Todos los videos cargan el mismo material, por lo que se consigue que todos los videos de la habitación se reproduzcan de forma simultanea, facilitando así un uso didáctico donde el profesor funcione de guía y pueda ir comentando el video visualizando por encima de el.

Finalmente, la disposición de estos paneles simula un aula real en nuestro mundo virtual. En primer lugar, tenemos un reproductor más grande en el lugar de la pizarra como si de un proyector habitual se tratase.



A su lado se encuentra el panel controlador del video, que nos permite pausar reproducir desde el principio o importar un nuevo video a través de una URL.



Y por último se agregó un reproductor delante de cada una de las sillas, simulando una pantalla digital por si los usuarios quieren ver con mayor detalle el video.



Posibles

06 actualizaciones

Dado que este proyecto se plantea para que se pueda hacer un uso público de él y mejorar la calidad de los estudios en cualquier ámbito y para todo el mundo, se proponen unas posibles actualizaciones o mejoras, que, debidas a mi poca experiencia con la realidad virtual y las limitaciones de la tecnología de hoy en día, quizás en un futuro sea muy interesantes incorporarlas a este proyecto.

## 6.1 Sistema multijugador

En primer lugar, posiblemente la característica que personalmente más frustrante ha sido para mí no poder incorporarla, debido al todo el tiempo y esfuerzo invertido en ella.

Se propone agregar un sistema multijugador donde la conexión se genere modo cliente a cliente y que sea el propio creador de la sala el que funcione como host para que el resto de los usuarios puedan unirse a él, mediante un código introducido en la pestaña del menú inicial “Unirse a una sala”

## 6.2 Avatar personalizado

La segunda propuesta y que ofrecería una inmersión muy notable, sería la de incorporar los avatares virtuales de las personas que utilicen la aplicación a dentro del mundo virtual.

En el caso de Oculus, que utiliza una cuenta Facebook obligatoriamente, se te obliga a crear un personaje virtual (un avatar) que será tu replica dentro de la realidad virtual.

Se propone que se crease la conectividad entre la cuenta Oculus del usuario y la aplicación de este proyecto para que el modelo del personaje en lugar de ser un modelo preestablecido se generase de forma automática y dinámica, con el fin de que la experiencia en un mundo virtual cuando se conectasen varios jugadores fuese mucho más reconfortante e inmersiva.

## 6.3 Importar objetos 3D personalizados

Al igual que se ha realizado con los paneles donde se permite importar el material que el usuario desee al mundo virtual, y cuando el trabajo con objetos 3D se estandarice al nivel que lo están las imágenes y existan modelos 3D compatibles en cualquier formato y plataforma independientemente de la aplicación con la que se han generado, se propone agregar esta nueva característica que sin duda los usuarios agradecerán.

## 6.4 Creación del sistema de comunicación

Cuando el sistema de multijugador esté operativo, se propone que se utilice el hardware que incorporan las propias gafas como es el micrófono, para que los propios jugadores una vez estén dentro de la sala puedan comunicarse entre ellos mediante un sistema de voz y sin tener que depender de ninguna aplicación externa a esta.

# 07 Conclusiones

Después de todo el trabajo, tiempo y esfuerzo invertido en este proyecto se ha llegado a la conclusión de que, para desarrollar una aplicación en un entorno de realidad virtual, se necesitan muchos conocimientos y tener una idea muy clara de todos los componentes que se van a necesitar desde el principio para su desarrollo, ya que, estos tipos de proyectos suelen padecer muchos problemas de incompatibilidades debido a que son tecnologías muy nuevas.

Por otro lado, me enorgullece haber completado mi idea inicial y haber superado con éxito el reto que me propuse antes de empezar este proyecto, él cual consistía en aprender una nueva tecnología y ser capaz de adaptarme a ella sin haber tenido contacto con ella nunca antes.

En cuanto al desarrollo de este proyecto ha sido un camino muy entretenido y en el que he aprendido que no hay que tener miedo a investigar temas desconocidos ya que, con esfuerzo, ganas y una buena base se puede estar preparado para afrontar cualquier tecnología.

También me gustaría agradecer a mi tutor Jorge Orta ya que me ha guiado en los momentos donde me encontraba desubicado y ha sabido aconsejarme en todo momento para poder completar este proyecto de forma exitosa.

# Bibliografía y 08 referencias

HP. (2021). *Gafas de realidad virtual HP Reverb G2*. HP® España.

<https://www.hp.com/es-es/vr/reverb-g2-vr-headset.html>

HTC Vive. (2020). *VIVE Cosmos Elite Overview*.

<https://www.vive.com/us/product/vive-cosmos-elite/overview/>

Iberdrola. (2021, 22 abril). *Metaverso, el lugar donde la realidad física y la virtual se dan la mano*. <https://www.iberdrola.com/innovacion/metaverso>

Levis, D. (2006). *¿Qué es la realidad virtual? ¿Qué es la realidad virtual?*

[https://scholar.googleusercontent.com/scholar?q=cache:ULZ4D8dlwLUJ:scholar.google.com/+realidad+virtual&hl=es&as\\_sdt=0,5&as\\_vis=1](https://scholar.googleusercontent.com/scholar?q=cache:ULZ4D8dlwLUJ:scholar.google.com/+realidad+virtual&hl=es&as_sdt=0,5&as_vis=1)

Meta. (2020). *Oculus Quest 2*. Meta Store.

<https://store.facebook.com/es/es/quest/products/quest-2/>

Rodríguez, V. (2022, 4 marzo). *PROGRAMAS PARA CREAR REALIDAD VIRTUAL. I*

AM VR. <https://i-amvr.com/programas-para-crear-realidad-virtual/>

Samsung. (2015). *Samsung Gear VR | 2015 | Samsung*.

<https://www.samsung.com/es/business/wearables/gear-vr-r323/>

Santander. (2022, 1 julio). *¿Qué es el metaverso y qué oportunidades ofrece?*

<https://www.santander.com/es/stories/metaverso-todo-lo-que-necesitas-saber-para-aprovechar-el-nuevo-mundo>

Sony. (2016). *VR | Vive el juego con PS VR*. PlayStation.

<https://www.playstation.com/es-es/ps-vr/>

Unity Technologies. (s. f.). *NetCode*. Unity. <https://unity.com/es/products/netcode>

Unity Technologies. (2016a). *Unity - Manual: GameObject*. Unity.

<https://docs.unity3d.com/es/530/Manual/class-GameObject.html>

Unity Technologies. (2016b). *Unity - Manual: Introducción a los componentes*. Unity.

<https://docs.unity3d.com/es/530/Manual/UsingComponents.html>

Unity Technologies. (2016c). *Unity - Manual: Prefabs*. Unity.

<https://docs.unity3d.com/es/530/Manual/Prefabs.html>

Valve Corporation. (2019). *Visor - Valve Index® - Mejora tu experiencia - Valve*

*Corporation*. Valve Index. <https://www.valvesoftware.com/es/index/headset>