



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

– **TELECOM** ESCUELA
TÉCNICA **VLC** SUPERIOR
DE INGENIERÍA DE
TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de
Telecomunicación

Desarrollo de un reproductor de vídeos estereoscópicos
para el sector turístico

Trabajo Fin de Grado

Grado en Tecnología Digital y Multimedia

AUTOR/A: Torrijos Saiz, Carmen

Tutor/a: Rey Solaz, Beatriz

Cotutor/a externo: MONTESA ANDRES, JESUS-JAVIER

CURSO ACADÉMICO: 2022/2023

Resumen

Las tecnologías inmersivas digitales, como la realidad virtual y la realidad aumentada, se encuentran en constante evolución y desarrollo.

Este trabajo fin de grado se enmarca en el proyecto PromotionVR, en el cual colabora el equipo de I+D de la empresa Brainstorm Multimedia. Este proyecto busca la aplicación de la realidad virtual en el sector turístico, para promover espacios turísticos y patrimoniales.

El objetivo principal de este trabajo es desarrollar un reproductor de vídeos estereoscópicos. Para ello, el contenido de los vídeos será capturado en distintos lugares turísticos de la ciudad de Valencia con una cámara panorámica estereoscópica y será reproducido en unos cascos de realidad virtual. Además, se incorporará a ellos la grabación de un guía turístico describiendo las características del lugar. El reproductor también permitirá incluir elementos tridimensionales en la escena como contenido adicional. Por último, se creará una interfaz de selección de escenas a través de la cual el usuario podrá acceder a cada una de las experiencias.

En este documento se detalla el proceso de investigación, diseño y desarrollo del reproductor y sus contenidos, utilizando la herramienta de software Unity y los cascos de realidad virtual "Meta Quest 2", así como las propuestas de trabajo futuro.

Resum

Les tecnologies immersives digitals, com la realitat virtual i la realitat augmentada, es troben en constant evolució y desenvolupament.

Aquest treball de fi de grau s'emmarca en el projecte PromotionVR, en el qual col·labora l'equip d'I+D de l'empresa Brainstorm Multimedia. Aquest projecte busca l'aplicació de la realitat virtual en el sector turístic per promoure espais turístics i patrimonials.

L'objectiu principal d'aquest treball és desenvolupar un reproductor de vídeos estereoscòpics. Per a això, el contingut dels vídeos serà capturat en diferents llocs turístics de la ciutat de València amb una càmera panoràmica estereoscòpica i es reproduirà en uns auriculars de realitat virtual. A més, s'incorporarà la gravació d'un guia turístic que descriu les característiques de cada lloc. El reproductor també permetrà incloure elements tridimensionals a l'escena com a contingut addicional. Finalment, es crearà una interfície de selecció d'escenes a través de la qual l'usuari podrà accedir a cada una de les experiències.

En aquest document es detalla el procés d'investigació, disseny i desenvolupament del reproductor i els seus continguts, utilitzant l'eina de programari Unity i els auriculars de realitat virtual "Meta Quest 2", així com les propostes de treball futur.



Abstract

Digital immersive technologies, such as virtual reality and augmented reality, are constantly evolving and developing.

This bachelor's thesis is part of the PromotionVR project, in which the R&D team of Brainstorm Multimedia company collaborates. This project aims to apply virtual reality in the tourism sector to promote tourist and heritage spaces. The main objective of this work is to develop a stereoscopic video player. To achieve this, the content of the videos will be captured in different tourist locations in the city of Valencia using a stereoscopic panoramic camera and, it will be played on virtual reality glasses. Additionally, recordings of a tour guide describing the features of each location will be incorporated. The player will also allow the inclusion of three-dimensional elements as additional content in the scene. Finally, a scene selection interface will be created, allowing users to Access each of the experiences.

This document details the research, design, and development process of the player and its content, using the Unity software tool and the “Meta Quest 2” virtual reality headsets, as well as future work proposals.



Índice

Capítulo 1.	Introducción	1
1.1	Estereoscopia.....	1
1.2	Dispositivos para la visualización y creación de contenido estereoscópico.....	3
1.2.1	Creación de contenido estereoscópico	3
1.2.2	Visualización de contenido estereoscópico	4
Capítulo 2.	Objetivos	8
Capítulo 3.	Metodología	9
3.1	Planificación.....	9
3.1.1	Diagrama de Gantt	10
3.2	Materiales necesarios y requerimientos técnicos del proyecto.....	10
3.2.1	Hardware	10
3.2.2	Software	11
Capítulo 4.	Desarrollo	16
4.1	Configuración de Unity como entorno VR	16
4.1.1	Descargar la aplicación Unity	16
4.1.2	Instalación de los paquetes necesarios	17
4.1.3	Debug de la aplicación	19
4.2	Creación de la jerarquía necesaria para visualizar vídeo estereoscópico	20
4.2.1	Sistema de seguimiento de tracking	20
4.2.2	Componente Head y Cámaras	22
4.2.3	Componente VideoPlayer	23
4.2.4	Render Texture.....	24
4.2.5	Esferas	25
4.3	Análisis del eje óptico de la cámara estereoscópica.....	26
4.3.1	Toma de medidas y capturas	27
4.3.2	Obtención de los datos	27
4.3.3	Análisis de los datos con Microsoft Excel	28
4.3.4	Obtención de resultados	29
4.4	Conversión de vídeo RGB a vídeo con capa Alpha	30
4.4.1	Importar la secuencia de vídeo.....	30
4.4.2	Aplicar el efecto Ultrakey	31
4.4.3	Exportar el vídeo resultante.....	32
4.5	Visualización Vídeo Alpha en Unity.....	33



4.6	Integración del presentador en la escena final.....	34
4.7	Introducción de un elemento 3D animado y temporizado en la escena	35
4.8	Creación de las diferentes escenas para el visualizador	36
4.9	Desarrollo interfaz de selección de escenas	37
4.9.1	Implementación del sistema de seguimiento e interacción de los controladores	38
Capítulo 5.	Resultados obtenidos.....	41
Capítulo 6.	Conclusiones y propuesta de trabajo futuro	42
Capítulo 7.	Bibliografía.....	43
Capítulo 8.	ANEXOS.....	46
8.1	ANEXO I: Manual de usuario de la aplicación.....	46
8.1.1	Generación del archivo APK que contendrá la aplicación final.....	46
8.1.2	Instalación de la aplicación Oculus	47
8.1.3	Instalación de la aplicación SideQuest	48
8.1.4	Subida de un fichero .apk a los cascos de RV a través de SideQuest	49
8.1.5	Ejecución de la aplicación desde las gafas de RV.....	49

Índice de figuras

Figura 1. Esquema de estereoscopia [6].....	2
Figura 2. Ejemplo de cámara binocular [7].....	2
Figura 3. Ejemplos de cámaras con dos lentes [8].	3
Figura 4. Imagen Estereoscópica [9].....	3
Figura 5. Ejemplos de cámaras estereoscópicas con una lente y un espejo [10, 11].....	3
Figura 6. Visor estereoscópico Holmes [7].....	4
Figura 7. Gafas de RV [13] y Hologramas [14].....	4
Figura 8. Head Mounted Display [16].	4
Figura 9. Meta Quest 2 [17].	5
Figura 10. Interfaz Meta Quest 2 [18].....	5
Figura 11. Modo autónomo [1].	6
Figura 12. Modo conectado [17].....	6
Figura 13. Modo Inalámbrico WiFi [1].....	7
Figura 14. Interfaz de Unity [19].	11
Figura 15. Diagrama comparativo del uso de OpenXR [23].....	12
Figura 16. Ejemplo de aplicación realizada con XR Interaction Toolkit.	12
Figura 17. Interfaz de Adobe Premiere Pro [26].....	13
Figura 18. Ejemplo de ChromaKey [23].....	13
Figura 19. Interfaz de Adobe Photoshop [28].....	14
Figura 20. Interfaz de la aplicación Oculus.....	15
Figura 21. Interfaz Side Quest.....	15
Figura 22. Instalación de Unity desde Unity Hub.....	16
Figura 23. Módulos adicionales Unity.	16
Figura 24. Instalación del plug-in OpenXR.	17
Figura 25. Carpeta OpenXR instalada en la Carpeta Assets.	18
Figura 26. Instalación plug-in XR Interaction Toolkit.....	18
Figura 27. Interfaz Aplicación Oculus Dispositivos Móviles.	19
Figura 28. Scripts predeterminados del plug-in OpenXR.	21
Figura 29. Estructura Componente PlayerRig.....	21
Figura 30. Componente Gaze.....	21
Figura 31. Cámara Z CAM K1 Pro.	22
Figura 32. Vídeo Estereoscópico.	22
Figura 33. Elemento Head y sus componentes.....	23
Figura 34. Componente VideoPlayer [31].	23
Figura 35. Características VideoPlayer.....	24

Figura 36. Creación de una Render Texture.....	24
Figura 37. Shader Backculling Off	25
Figura 38. Estructura para visualizar vídeo estereoscópico	26
Figura 39. Resultado de la representación de un vídeo estereoscópico	26
Figura 40. Imágenes de prueba a 20 cm.....	27
Figura 41. Extracción de los datos con Photoshop.....	28
Figura 42. Centros de la imagen analizada.....	28
Figura 43. Resultados obtenidos tras el análisis de los datos	29
Figura 44. Gráfica Final Obtenida.....	29
Figura 45. Captura de guías turísticos sobre fondo verde	30
Figura 46. Importar el vídeo a Adobe Premiere.....	30
Figura 47. Listado efectos disponibles Premiere.....	31
Figura 48. Pestaña Controles de Efectos.....	31
Figura 49. Previsualización del canal Alfa.....	32
Figura 50. Exportar el vídeo resultante	32
Figura 51. Representación de un vídeo con capa Alfa en Unity	33
Figura 52. Representación final de un vídeo con capa Alfa en Unity	34
Figura 53. Integración del guía turístico en la escena	34
Figura 54. Script Timer	35
Figura 55. Script Animation.....	35
Figura 56. Integración Objeto 3D animado en la escena	36
Figura 57. Escenas desarrolladas	37
Figura 58. Creación Elemento Canvas.....	37
Figura 59. Interfaz del Visualizador de escenas turísticas	38
Figura 60. Interaction Profile	38
Figura 61. Ejemplo LeftHand Controller	39
Figura 62. Elemento XR Origin	39
Figura 63. Modelos de manos virtuales	40
Figura 64. Elemento Ray Tracing	40
Figura 65. Estructura Sistema de Interacción.....	40
Figura 66. Generar el fichero .apk.....	46
Figura 67. Instalación aplicación Oculus	47
Figura 68. Orígenes Desconocidos.....	48
Figura 69. Conexión de las gafas RV al PC.....	48
Figura 70. Interfaz de SideQuest.....	49

Capítulo 1. Introducción

Este trabajo de final de grado parte de la idea de crear un visualizador de vídeos estereoscópicos para visualizar experiencias inmersivas a través de unos cascos de realidad virtual de manera autónoma.

Este proyecto está enmarcado en el proyecto “Promoción de espacios turísticos basado en tecnologías inmersivas digitales”, al que se nombra como “ProMotionVR”.

ProMotionVR consiste en la promoción turística de diferentes lugares turísticos de la ciudad de Valencia a través de la creación de experiencias inmersivas. Dichas experiencias están formadas por un vídeo estereoscópico, en el cual se inserta un guía turístico y un elemento 3D relacionado con la experiencia [1].

Por otro lado, en la empresa Brainstorm Multimedia, se ha llevado a cabo un desarrollo similar para el proyecto. No obstante, el objetivo de ese visualizador era que funcionara sobre una versión de las gafas conectada al PC. Dicha versión no presenta las capacidades necesarias para que fuera ejecutable de manera autónoma sobre los cascos de realidad virtual. Esto se debe a la alta capacidad necesaria para reproducir los vídeos estereoscópicos de alta calidad y a su vez realizar una extracción de silueta sobre fondo verde en tiempo real.

Es por ello, que el proyecto presentado a lo largo de esta memoria pretende implementar una solución prototipo que presente capacidades similares de visualización, pero que no requiera del uso externo de una tarjeta gráfica. Para conseguir limitar la potencia necesaria, se explorará la realización de la extracción de la silueta del presentador de manera externa a la aplicación del visualizador. Además, se realizará un estudio del eje óptico de la cámara estereoscópica para asegurar su correcto funcionamiento.

Este trabajo contribuye al objetivo de desarrollo sostenible (ODS) N°9 - Industria, Innovación e Infraestructura. Este ODS se centra en impulsar y obtener el máximo potencial de la innovación, la infraestructura y la industrialización. Estos elementos desempeñan un papel esencial a la hora de implementar y fomentar innovaciones tecnológicas, facilitar el comercio global y garantizar la utilización eficiente de los recursos disponibles. Una forma de promover la innovación es a través de los proyectos I+D+i desarrollados por empresas, especialmente empresas de base tecnología que desean promover y explorar nuevas tecnologías [2].

Además, este proyecto de fin grado se relaciona directamente con la meta 12b del ODS N°12 - Producción y Consumo Responsables. El objetivo principal de esta meta es establecer y emplear herramientas que permitan monitorear los impactos en el desarrollo sostenible, con el fin de alcanzar un turismo sostenible que genere empleo y fomente la cultura y los productos autóctonos [3].

A continuación, se explican algunos conceptos teóricos necesarios para comprender el desarrollo del proyecto en su totalidad.

1.1 Estereoscopia

Como se ha mencionado anteriormente, el desarrollo principal del proyecto utiliza vídeos estereoscópicos. A continuación, se explica en profundidad el significado de la estereoscopia y sus principales aplicaciones en la actualidad.

La estereoscopia es una técnica que utiliza la percepción visual humana para crear la ilusión de profundidad en una imagen bidimensional. Es decir, se trata de “engañar a la mente” para que, al ver una imagen de dos dimensiones, el efecto visual sea de tres dimensiones y, por lo tanto, exista una sensación de profundidad en la imagen mostrada. Esto se logra mediante la presentación de dos imágenes ligeramente diferentes, una para cada ojo, que el cerebro fusiona en una sola imagen tridimensional [4, 5].

La Figura 1 muestra de manera esquemática el proceso de visualización de una imagen estereoscópica.

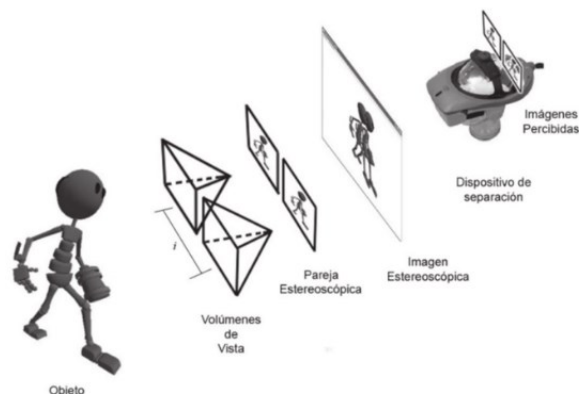


Figura 1. Esquema de estereoscopia [6].

La estereoscopia se utiliza actualmente en diversas aplicaciones, como la fotografía, el cine y los sistemas de visualización en 3D, como las gafas de realidad virtual y aumentada. Sin embargo, las primeras ideas sobre la estereoscopia ya fueron descritas por Euclides en el año 280 antes de Cristo.

A veces, el término estereoscopia también se utiliza para referirse a la técnica de producir imágenes estereoscópicas o para referirse a la imagen resultante en sí misma. Los videos y las imágenes estereoscópicas se utilizan para crear una sensación de inmersión más realista y mejorar la experiencia del espectador o del usuario al proporcionar una percepción de profundidad y volumen en la escena a visualizar.

Una de las aplicaciones más relevantes de la estereoscopia fue en el ámbito de la fotografía y tuvo su auge en el siglo XIX. Tradicionalmente, la fotografía estereoscópica se realizaba con una sola cámara que se movía ligeramente entre la primera y la segunda toma una distancia aproximada a la distancia existente entre los ojos de una persona. Más adelante, surgieron nuevas técnicas menos aparatosas como colocar dos cámaras iguales, una junto a la otra y destapar sus objetivos al mismo tiempo.

A principio de la segunda mitad del siglo XIX se introdujeron las cámaras con dos objetivos o cámaras binoculares (Figura 2) y, desde ese momento en adelante, todas las cámaras estereoscópicas posteriores se basan en esta configuración. En el siguiente apartado, se explica con más detalle el funcionamiento de estas cámaras [7].



Figura 2. Ejemplo de cámara binocular [7].

No obstante, para poder visualizar correctamente el efecto producido al aplicar la técnica de la estereoscopia, es necesario disponer de dispositivos especializados que sean capaces de mostrar imágenes ligeramente diferentes para cada ojo de manera individual. A lo largo de las siguientes secciones, se detallan los dispositivos más usados para la visualización y creación de imágenes o vídeo estereoscópico.

1.2 Dispositivos para la visualización y creación de contenido estereoscópico

1.2.1 Creación de contenido estereoscópico

En la actualidad, existen diferentes tipos de cámaras que se pueden utilizar para la creación de imágenes estereoscópicas. En general, se pueden clasificar en dos grandes categorías: las cámaras con dos lentes y las cámaras con una lente y un espejo o prisma.

Las cámaras con dos lentes son las más comunes y se utilizan para capturar dos imágenes en paralelo, una para cada ojo, que luego se combinan para crear la sensación de profundidad en una única imagen estereoscópica a través de software especializado. Algunos ejemplos de cámaras con dos lentes incluyen la GoPro Max, la Insta360 Pro y la Z CAM K1 Pro (Figura 3) [8].



Figura 3. Ejemplos de cámaras con dos lentes [8].

Un ejemplo de una imagen obtenida con este tipo de cámaras es el mostrado en la Figura 4. Como se puede observar, se obtiene una imagen compuesta por dos imágenes muy similares entre sí. Sin embargo, estas dos imágenes no son exactamente iguales, sino que están tomadas desde dos puntos de vista ligeramente separados. De esta forma, al mostrar una imagen a cada ojo, se obtiene el efecto de profundidad deseado [9].



Figura 4. Imagen Estereoscópica [9].

Las cámaras con una lente y un espejo o prisma funcionan de manera similar. En lugar de tener dos lentes separadas, utilizan un elemento reflectante para dividir la imagen en dos y capturarlas simultáneamente. Algunos ejemplos de estas cámaras incluyen la Z CAM V1 Pro y la Nikon KeyMission 360 (Figura 5).



Figura 5. Ejemplos de cámaras estereoscópicas con una lente y un espejo [10, 11].

1.2.2 Visualización de contenido estereoscópico

Para visualizar contenido estereoscópico, es necesario el uso de dispositivos externos especializados. Uno de los primeros visores especialmente diseñados para visualizar contenidos estereoscópicos fue el visor diseñador por Wheatstone en 1832. Sin embargo, este visor fue rápidamente sustituido por el famoso visor Holmes (Figura 6), gracias a su manejo mucho más sencillo [7].



Figura 6. Visor estereoscópico Holmes [7].

Este visor fue uno de los más utilizados por los victorianos y gracias a él, la fotografía estereoscópica tuvo un gran auge en Inglaterra consiguiendo vender unos diez millones de copias estereoscópicas [7].

En la actualidad, estos dispositivos han evolucionado significativamente y se pueden encontrar en forma de gafas o pantallas. Sin embargo, los avances continúan y ya se está trabajando con tecnología láser para la creación de imágenes tridimensionales que simulan flotar, también conocidas como imágenes holográficas (Figura 7) [12].



Figura 7. Gafas de RV [13] y Hologramas [14].

Hay varios modelos de gafas especializadas que permiten visualizar este tipo de contenido: gafas polarizadas, gafas de obturación activa o lentes de diferente color para imágenes de anaglifo. Sin embargo, las más utilizadas debido a su relación calidad-precio son los sistemas conocidos como Head Mounted Displays o HMD (Figura 8).

Cuando hablamos de HMDs nos referimos a cascos de realidad virtual o aumentada. Estos cascos permiten a los usuarios visualizar entornos virtuales completos y gráficos tridimensionales de una forma completamente inmersiva [15].



Figura 8. Head Mounted Display [16].

Se trata de un concepto relativamente antiguo, pero gracias a la tecnología de pantalla OLED, la resolución y la calidad de las imágenes que ofrecen ha mejorado muchísimo, así como el campo de visión ofrecido al usuario. Otra de las tecnologías cuya evolución ha permitido el gran avance ocurrido en estos dispositivos, es el seguimiento óptico de usuario y la adición de mandos.

En la actualidad existen distintos sistemas HMD domésticos que ofrecen una elevada sensación de inmersión a costes muy reducidos en comparación con generaciones antiguas. Además, es muy probable que la gran evolución que se está produciendo continúe su camino y alcance cotas de calidad superiores a las disponibles actualmente [15].

El dispositivo de realidad virtual de referencia en el proyecto son los cascos de realidad virtual Meta Quest 2 (Figura 9). Son uno de los HMD más populares y mejor valorados del mercado. Fue desarrollado por la empresa Facebook como sucesor del Oculus Quest Original. Su lanzamiento en el mercado fue en el año 2020 y presenta mejoras en varios aspectos en comparación con su predecesor.



Figura 9. Meta Quest 2 [17].

Presenta una resolución de 1832x1920 por ojo lo que permite que las imágenes se vean con mucho más detalle y de manera más nítida (Figura 10). Su tasa de refresco es de 90 Hz y cuenta con un procesador Qualcomm Snapdragon XR2, el cual es más rápido y potente que el procesador Snapdragon 835 del Quest original. Esto se traduce en una experiencia más suave y sin problemas [17].

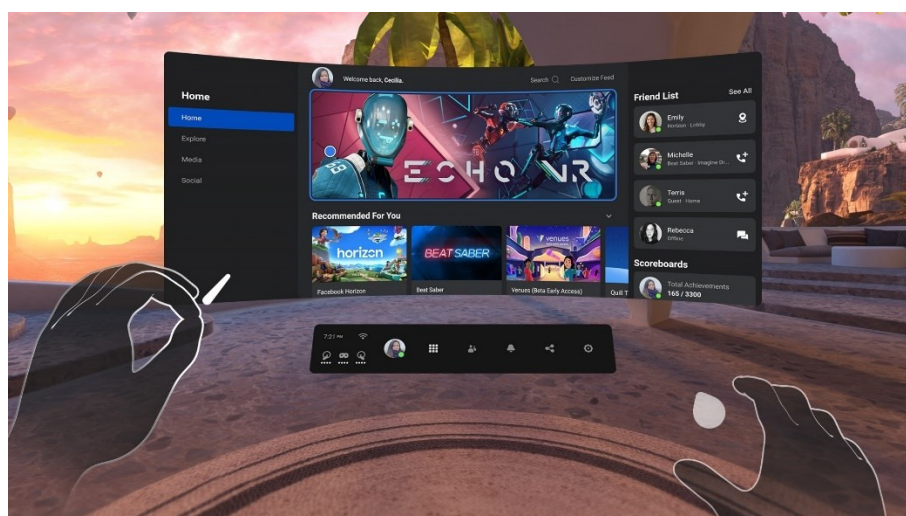


Figura 10. Interfaz Meta Quest 2 [18].

En cuanto al peso, es un HMDs bastante ligero con un peso alrededor de 503 gramos (1.11 libras) y cómodo de llevar gracias a la correa ajustable para la cabeza y el acolchado facial intercambiable que presenta.

Actualmente, el Meta Quest 2 presenta dos versiones disponibles en el mercado, la primera cuenta con un almacenamiento interno de 128 GB y su precio ronda los 350€ y la segunda versión con un almacenamiento de 256 GB cuesta alrededor de 399€.

Todas estas características lo hacen un casco de realidad virtual muy potente y accesible. Sin embargo, lo que lo ha convertido en uno de los mejores es su capacidad de funcionar de manera completamente autónoma (Figura 11). Además, continúa permitiendo la conexión a un PC para ejecutar aplicaciones más complejas que requieran de una tarjeta gráfica y un procesador con mayor potencia [17].

A priori, el modo autónomo es mucho más práctico dado que no son precisos cables ni equipos adicionales, pero por otro lado la potencia gráfica de unas Quest 2 dista mucho de la disponible en una estación gráfica actual.



Figura 11. Modo autónomo [1].

Este proyecto trata de visualizar vídeos estereoscópicos de alta calidad, por tanto, la potencia de cálculo requerida para su decodificación se dobla dada la necesidad de decodificar una imagen por ojo. Además, al tratarse de vídeos panorámicos, en función del ángulo subtendido por la proyección, 180° o 360°, la superficie se multiplica de nuevo y es fácil alcanzar el límite del dispositivo a resoluciones relativamente bajas, y sobre todo insuficientes para conseguir experiencias de usuario realmente positivas.

Por contra, con el dispositivo conectado a una estación gráfica (Figura 12), la potencia se multiplica, y la resolución de los vídeos estereoscópicos puede ser mucho más elevada [18].



Figura 12. Modo conectado [17].

Existe, además, la posibilidad de conectar las Meta Quest 2 a la estación gráfica de forma inalámbrica mediante un router WiFi 6 (Figura 13). Esta configuración requiere de una estación gráfica y permite disponer de toda su potencia. Además, al no requerir una conexión cableada permite una movilidad de uso similar al modo autónomo. Sin embargo, se hace uso de la batería interna del dispositivo lo cual puede limitar la duración de las sesiones [18].

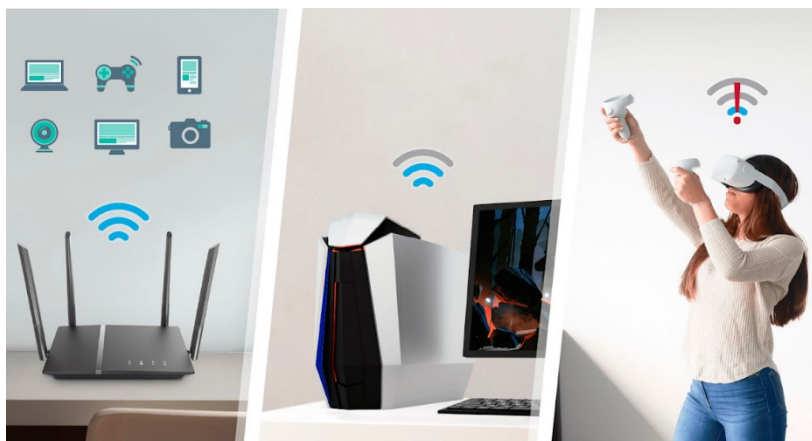


Figura 13. Modo Inalámbrico WiFi [1].

A continuación, se presenta una tabla comparativa (Tabla 1) de los tres modos de funcionamiento que presentan las gafas de realidad virtual Meta Quest 2.

	Autónomo	Cable	WiFi 6
Potencia Gráfica	Limitada	Alta	Alta
Movimiento	Libre	Limitado	Libre
Autonomía	Limitada	Ilimitada	Limitada

Tabla 1. Modos de funcionamiento de las gafas Meta Quest 2.

La intención en este trabajo de final de grado es explorar una nueva solución que permita mantener la resolución 1080p de los vídeos estereoscópicos sin la necesidad de trabajar con las gafas conectadas a una estación gráfica.



Capítulo 2. Objetivos

Este apartado enumera los objetivos generales y específicos que se plantean resolver a lo largo de este trabajo. El objetivo principal en el cual está enmarcado este proyecto es desarrollar un visualizador de contenidos estereoscópicos para los cascos de RV Meta Quest 2, haciendo uso de la herramienta Unity.

A continuación, se listan los diferentes objetivos específicos que se cumplimentarán a lo largo de la realización completa del proyecto:

- Comprender el funcionamiento del vídeo estereoscópico
- Analizar el eje de visualización de la cámara estereoscópica panorámica Z CAM K1 Pro
- Entender el manejo de la herramienta Unity para contenidos VR
- Analizar las librerías disponibles para la visualización de contenido estereoscópico
- Visualizar vídeo estereoscópico con Unity
- Realizar la extracción del fondo verde de un vídeo en Adobe Premiere Pro
- Visualizar vídeos con capa Alpha en Unity
- Integrar un vídeo de un guía turístico en la escena estereoscópica
- Integrar un elemento 3D en la escena estereoscópica
- Animar y temporizar un elemento 3D en Unity
- Crear una interfaz de contenidos para el visualizador
- Implementar la interacción del usuario con la interfaz a través de los controladores
- Crear los diferentes contenidos que presenta el visualizador
- Demostrar las experiencias inmersivas a través del visualizador

Capítulo 3. Metodología

En este capítulo se presenta la planificación realizada para llevar a cabo el desarrollo completo de este trabajo de final de grado. En primer lugar, se exponen los paquetes de trabajo escogidos junto con las tareas que los componen. Más adelante, se muestra el diagrama de Gantt del trabajo. Por último, se especifica el listado de materiales usados para la realización del proyecto, tanto hardware como software.

3.1 Planificación

El proyecto ha comenzado su desarrollo en el mes de febrero de 2023 y su fecha de finalización es junio de 2023.

Para asegurar un correcto desarrollo del proyecto, se ha definido una planificación compuesta por paquetes de trabajo enfocados en un objetivo concreto. A continuación, se explican los diferentes paquetes de trabajo definidos y las tareas que los componen.

PT1 – Gestión del Proyecto

Este paquete de trabajo está compuesto por todas las actividades de planificación, como son la definición de los objetivos, la planificación de reuniones semanales de revisión del proyecto y el reparto de la carga de trabajo diaria.

PT2 – Investigación Necesaria

Este paquete de trabajo comienza en febrero de 2023 y presenta una duración de dos meses. Se centra en el estudio de las diferentes técnicas y herramientas actuales para desarrollar aplicaciones de realidad virtual. Gracias a la realización de estos estudios, se definirán los materiales a usar en el proyecto y sus requerimientos técnicos. Con el cumplimiento de esta tarea, se pretenden obtener todos los conocimientos necesarios para llevar a cabo este proyecto.

Este paquete de trabajo está compuesto por las siguientes tareas:

- *T2.1. Estudio Códecs de Codificación de Video:* Se analizan las diferentes librerías compatibles con Unity y VR para escoger la más adecuada para el proyecto.
- *T2.2. Estudio de la Cámara Estereoscópica:* Se realiza un análisis completo del eje óptico de la cámara estereoscópica para evitar efectos indeseados en el funcionamiento del visualizador.
- *T2.3. Identificación de materiales y requerimientos técnicos:* Se escogen los diferentes materiales, tanto hardware como software, necesarios para el desarrollo del proyecto.

PT3 – Desarrollo del Proyecto

Este paquete de trabajo engloba todas aquellas tareas necesarias para realizar el desarrollo completo del sistema y su integración.

Está compuesto por las siguientes tareas, ordenadas según su realización durante el proyecto:

- *T3.1. Configuración Unity:* Configurar Unity como entorno de trabajo de VR y descarga de los módulos necesarios.
- *T3.2. Reproducir vídeo en Unity:* Estudio del componente VideoPlayer de Unity y su funcionamiento.
- *T3.3. Reproducir vídeo estereoscópico en Unity:* Creación de la estructura necesaria para reproducir vídeo estereoscópico en Unity.
- *T3.4. Eliminación del fondo verde de un vídeo de un presentador:* Edición de un vídeo sobre fondo verde con la herramienta Adobe Premiere Pro para realizar la extracción de la silueta del presentador.
- *T3.5. Visualización de un vídeo con fondo transparente:* Estudiar el método para visualizar vídeo con capa Alpha incluida en Unity.

- *T3.6. Inserción de elementos 3D en la escena:* Descarga de modelos 3D y su inserción en las escenas turísticas de Unity correspondientes.
- *T3.7. Temporización y animación de los elementos 3D:* Desarrollo de los scripts necesarios para temporizar la aparición de los objetos en las escenas y sus animaciones.
- *T3.8. Creación de diferentes escenas turísticas:* Descarga de todos los elementos necesarios y desarrollo de cuatro escenas turísticas distintas.
- *T3.9. Interfaz de selección de escenas:* Diseño y desarrollo de un elemento Canvas para la selección de escenas. Desarrollo del sistema de seguimiento e interacción de los controladores.
- *T3.10. Ajustes y refinamiento del sistema:* Realización de pruebas del sistema para obtener sus puntos débiles y mejorarlos en los próximos desarrollos.

3.1.1 Diagrama de Gantt

El siguiente diagrama de Gantt (Tabla 2) muestra la planificación mostrada en la sección anterior de una manera más visual y ordenada.

PAQUETES DE TRABAJO	feb2023	mar2023	abr2023	may2023	jun2023
PT1- Gestión Proyecto					
PT2 - Investigación Necesaria					
T2.1. Estudio Códecs de Codificación de Vídeo					
T2.2. Estudio de la Cámara Estereoscópica					
T2.3. Identificación de materiales y requerimientos técnicos					
PT3 - Desarrollo Proyecto					
T3.1. Configuración Unity como entorno VR					
T3.2. Reproducir vídeo en Unity					
T3.3. Reproducir vídeo estereoscópico en Unity					
T3.4. Eliminación del fondo verde de un vídeo de un presentador					
T3.5. Visualización de un vídeo con fondo transparente Unity					
T3.6. Inserción de Elementos 3D en la escena					
T3.7. Temporización y animación de los elementos 3D					
T3.8. Creación de diferentes escenas turísticas					
T3.9. Interfaz de selección de escenas					
T3.10. Ajustes y refinamiento del sistema					

Tabla 2. Diagrama de Gantt.

3.2 Materiales necesarios y requerimientos técnicos del proyecto

En esta sección del documento, se listan los elementos hardware y software necesarios para el desarrollo completo del visualizador de vídeos estereoscópicos.

3.2.1 Hardware

Los componentes hardware utilizados durante el desarrollo de este proyecto han sido:

- Ordenador portátil MSI GE75 Raider 9SG: Presenta un procesador Intel® Core™ i9-9880H CPU @ 2.30 Ghz, su sistema operativo es Microsoft Windows 11 Pro y utiliza la tarjeta gráfica NVIDIA GeForce RTX 2080.
- Cable de conexión: Cable de conexión de tipo USB-A a USB-C utilizado para conectar las gafas de RV al ordenador.
- Gafas Oculus Quest 2: De entre todos los modelos actuales de HMDs que hay en el mercado se ha escogido este dispositivo como visualizador del proyecto por los siguientes motivos. En primer lugar, es un modelo asequible y con una buena ergonomía. Además, presenta una gran capacidad de procesamiento, lo cual es un requisito indispensable en este proyecto ya que se necesita visualizar vídeos estereoscópicos de muy alta calidad.

3.2.2 Software

A continuación, se listan las herramientas software utilizadas durante el desarrollo de este proyecto acompañadas de una explicación de sus características más importantes y el motivo por el cual han sido escogidas.

Unity

Unity (Figura 14) es un motor gráfico de tiempo real comúnmente utilizado para el desarrollo de videojuegos multiplataforma, abarcando desde potentes aplicaciones para PC hasta algunas más sencillas para dispositivos móviles. También incluye la posibilidad de desarrollo de aplicaciones web o para cascos de RV.

En los últimos años, Unity se ha convertido en una herramienta comercial muy extendida con gran soporte y una amplia comunidad de usuarios y desarrolladores. Gracias a esta comunidad, Unity continúa ampliando y añadiendo nuevas funcionalidades a su motor a través de contenidos y plug-ins (contenido adicional que aporta más valor a la herramienta principal) aportados de manera continua por su comunidad para que nunca se quede obsoleto [19].



Figura 14. Interfaz de Unity [19].

Por todos estos motivos, se ha decidido trabajar con Unity como herramienta principal gracias a su versatilidad y las múltiples opciones que presenta para trabajar sobre realidad virtual. El trabajo ha sido realizado con la versión de Unity 2021.2.8f1 y se ha hecho uso de las diferentes librerías gratuitas y plug-ins que presenta la herramienta [20].

OpenXR

OpenXR es un estándar de código abierto y libre de regalías que facilita el acceso a plataformas y dispositivos de realidad virtual y realidad aumentada. Ofrece una capa de abstracción, permitiendo así que un desarrollo en RV o RA sea interoperable entre diferentes plataformas y dispositivos. Gracias a su aparición, ya no es necesario tener una aplicación con diferentes versiones para cada plataforma, sino que al utilizar OpenXR se abstrae el funcionamiento de estas independientemente del hardware del que se haga uso.

Cada vez más, las diferentes empresas que trabajan con aplicaciones para RA y RV están implementando la posibilidad de utilizar este estándar para sus desarrollos. Un ejemplo de esto es Oculus, la cual ya no obliga a usar su API Oculus Mobile para realizar los desarrollos de las gafas de RV Meta Quest 2, sino que pone a disposición de los desarrolladores de sus motores de juegos la posibilidad de usar OpenXR [21, 22].

En la Figura 15, se puede observar cómo al usar este estándar, el desarrollo de aplicaciones se simplifica de manera significativa y se unifican todas las posibilidades que antes había disponibles para realizar aplicaciones en un solo estándar.

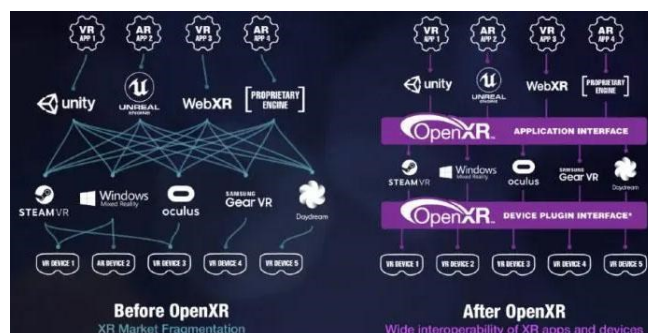


Figura 15. Diagrama comparativo del uso de OpenXR [23].

Debido a la gran flexibilidad e interoperabilidad que presenta este plugin, se ha decidido realizar el reproductor utilizando el plug-in OpenXR que implementa la herramienta Unity. De esta manera, la aplicación desarrollada será accesible para cualquier hardware compatible con este estándar.

En cuanto a las funcionalidades que permite OpenXR como librería de trabajo respecto a las que ya había en el mercado, como puede ser SteamVR y la librería de Oculus, éstas son muy similares. El principal motivo por el cual es recomendable hacer el cambio a este estándar es la mayor compatibilidad que ofrece con todos los dispositivos RV.

XR Interaction Toolkit

El paquete XR Interaction Toolkit (XRI) (Figura 16) es un sistema de interacción de alto nivel desarrollado por Unity para facilitar la creación de experiencias de realidad virtual y realidad aumentada.

Presenta un marco común de herramientas de interacción para los desarrolladores, de forma que permite la creación más sencilla y eficiente de experiencias interactivas a través de controladores en entornos virtuales y agiliza la creación multiplataforma.

Este paquete incluye una gran variedad de tipos de interacciones desde agarrar, soltar, tocar o pulsar objetos virtuales dentro de la experiencia hasta la definición de zonas interactivas y la asignación de acciones a los botones del controlador [24].

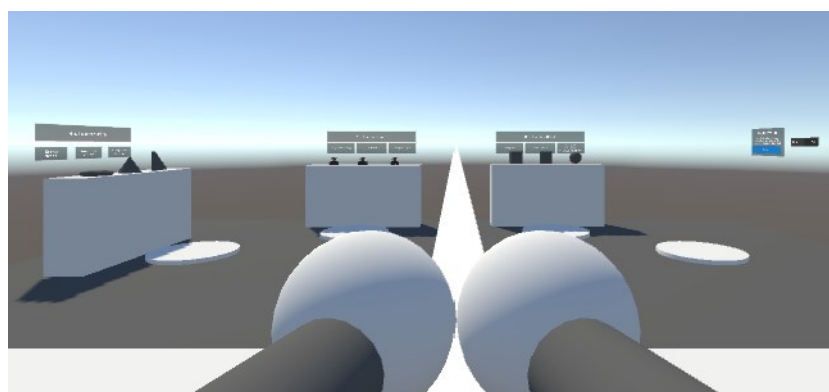


Figura 16. Ejemplo de aplicación realizada con XR Interaction Toolkit.

Es una herramienta muy completa e intuitiva, a través de la cual los desarrolladores pueden configurar interacciones físicas y lógicas para usuarios y objetos en entornos de realidad extendida.

Este proyecto de fin de grado cuenta con una interacción sencilla de los controladores. Para llevarla a cabo, se han utilizado los diferentes recursos que proporciona este plug-in de Unity. En el capítulo 4, se dedica una sección para explicar cómo realizar la instalación de este plug-in dentro del proyecto de Unity.

Adobe Premiere Pro

Adobe Premiere Pro (Figura 17) es un software de edición de video utilizado principalmente por profesionales de la producción de cine, televisión y video. Está incluido en la suite de programas de Adobe Creative Cloud y permite a los usuarios editar videos de alta calidad con funciones avanzadas como edición multicámara, ajustes de color, efectos visuales y de audio, animación de títulos y más.

Adobe Premiere Pro es compatible con una amplia gama de formatos de video y audio, lo que lo convierte en una herramienta versátil para la edición de videos en cualquier tipo de proyecto, desde producciones de alta gama como películas de cine hasta la edición de videos para redes sociales [25].

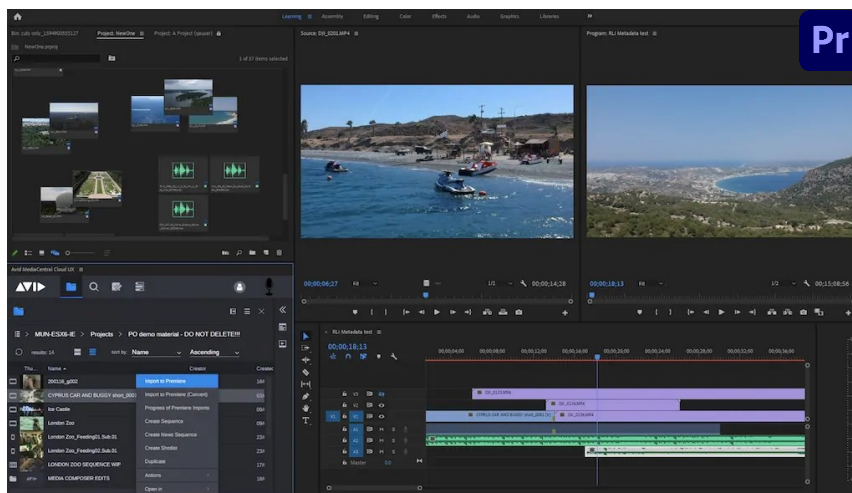


Figura 17. Interfaz de Adobe Premiere Pro [26].

El principal uso de esta herramienta en el proyecto es la eliminación del fondo verde de los vídeos de los guías turísticos (Figura 18), para integrarlos en la escena de los lugares turísticos. Se ha escogido la herramienta Adobe Premiere Pro ya que presenta una gran variedad de herramientas avanzadas para la corrección de color, lo que permite ajustar los niveles de color, saturación y tono de las imágenes. Además, cuenta con herramientas de máscaras que permiten seleccionar áreas específicas de la imagen para aplicar efectos o ajustes concretos. Al combinar estas herramientas correctamente, se permite aislar y eliminar el fondo verde de un video para reemplazarlo con cualquier imagen o video. También es posible sustituirlo por una capa transparente o Alpha.



Figura 18. Ejemplo de ChromaKey [23].

Adobe Photoshop

Adobe Photoshop (Figura 19) es un software de edición de imágenes utilizado por profesionales y aficionados para manipular y mejorar imágenes digitales de muchas maneras distintas. Permite modificar y retocar fotografías y otros tipos de imágenes digitales mediante una gran variedad de herramientas y técnicas, como ajustes de color, filtros, selecciones y capas.

Además, Photoshop proporciona herramientas de zoom y navegación precisas que permiten acercarse y examinar los detalles más pequeños de una imagen. Otro tipo de herramientas de medición que proporciona son las reglas que facilitan la toma de medidas precisas de los elementos de la imagen, las cuales serán un elemento fundamental para el análisis de la cámara [27].



Figura 19. Interfaz de Adobe Photoshop [28].

Se ha decidido utilizar Adobe Photoshop en este proyecto gracias a su facilidad de uso y su gran cantidad de opciones que permite realizar ajustes y análisis de fotografías de una manera sencilla y con muy buenos resultados. Además, permite trabajar con una gran cantidad de formatos de imágenes distintas y adecuarlas a las necesidades del proyecto que se necesitan.

Oculus

La aplicación de Oculus (Figura 20) es una herramienta diseñada especialmente para todos los usuarios de cualquier dispositivo Oculus, como por ejemplo el Meta Quest 2. Desde esta herramienta, los usuarios pueden administrar y gestionar sus dispositivos, explorar cientos de aplicaciones en la tienda, descubrir eventos de realidad virtual y mucho más.

Es una herramienta muy útil ya que proporciona una experiencia centralizada para administrar y aprovechar al máximo la realidad virtual. Al configurar y sincronizar el dispositivo Oculus, los usuarios podrán acceder a una biblioteca personalizada de juegos y aplicaciones. Además, facilita la comunicación y la conexión con amigos y jugadores de todo el mundo, ya que permite chatear, enviar mensajes y unirse a eventos sociales en realidad virtual.

En este trabajo, ha sido necesaria la utilización de la aplicación Oculus para sincronizar y configurar el dispositivo Meta Quest 2 así como para realizar las diferentes pruebas de desarrollo durante proyecto e instalar la versión final en las gafas [29].

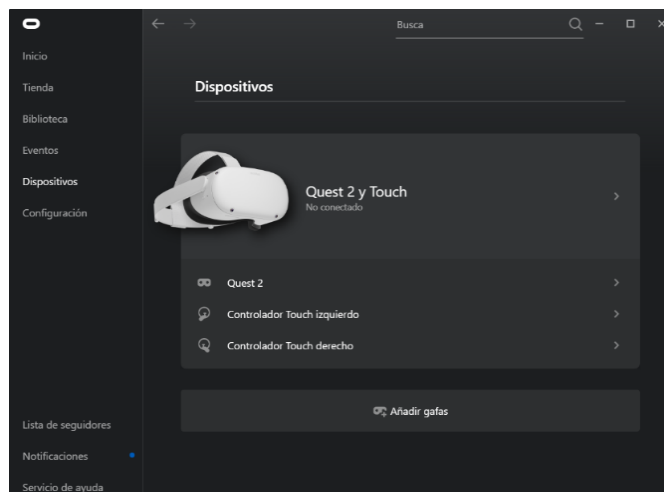


Figura 20. Interfaz de la aplicación Oculus.

SideQuest

SideQuest (Figura 21) es una plataforma y aplicación independiente en la que los usuarios pueden obtener una gran variedad de contenidos para dispositivos Oculus. En gran medida, se utiliza para obtener contenido adicional, aplicaciones y juegos no oficiales de Oculus. Además, también funciona como distribuidor de proyectos de realidad virtual creados por desarrolladores y creadores.

Gracias a esta funcionalidad, cualquier aplicación puede ser instalada y usada en el dispositivo Oculus a pesar de no estar disponible en la Oculus Store. Esta aplicación presenta algunos requisitos como la necesidad de trabajar con el sistema operativo Windows 8 o superior y la necesidad de realizar la transferencia de aplicaciones a través de un cable de conexión USB tipo C.

En este proyecto, se hace uso de la aplicación para instalar el desarrollo final obtenido con Unity en las gafas de realidad virtual Meta Quest 2. En el anexo I, se explica con más detalle la realización de este proceso [30].

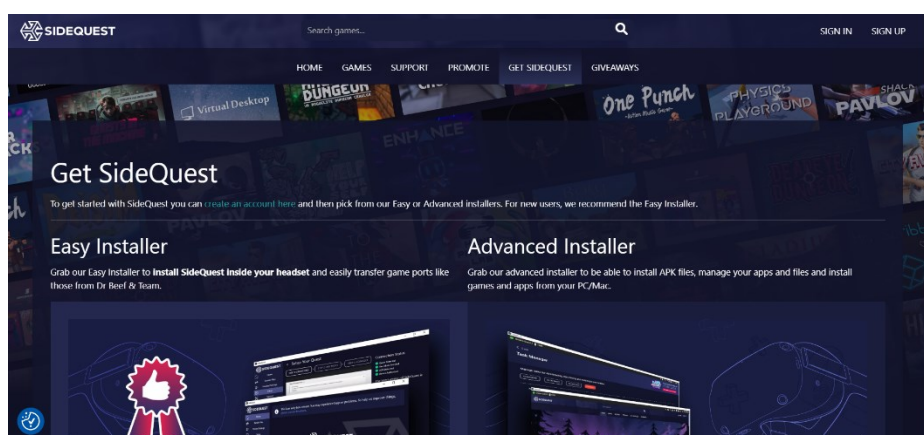


Figura 21. Interfaz Side Quest.

Capítulo 4. Desarrollo

4.1 Configuración de Unity como entorno VR

4.1.1 Descargar la aplicación Unity

La forma más sencilla de instalar la aplicación de Unity es a través de la aplicación Unity Hub. Esta aplicación es gratuita y permite el acceso a todo el ecosistema que forma el entorno de Unity.

Unity Hub permite administrar los diferentes proyectos creados, instalar diferentes versiones del propio editor, activar o desactivar las licencias e instalar componentes complementarios y plug-ins para el desarrollo de los proyectos que se necesite.

Una vez instalada la aplicación Unity Hub, desde el apartado “Installs” en la parte izquierda de la interfaz, se puede instalar Unity. Para ello, hay que seleccionar el botón situado en la parte superior derecha “Install Editor”. Unity Hub permite a los usuarios seleccionar la versión de Unity que más se acople a las necesidades de sus desarrollos, así como los elementos adicionales que necesitan que incluyan sus proyectos. La Figura 22 muestra el resultado tras instalar la versión de Unity 2021.3.8f1.

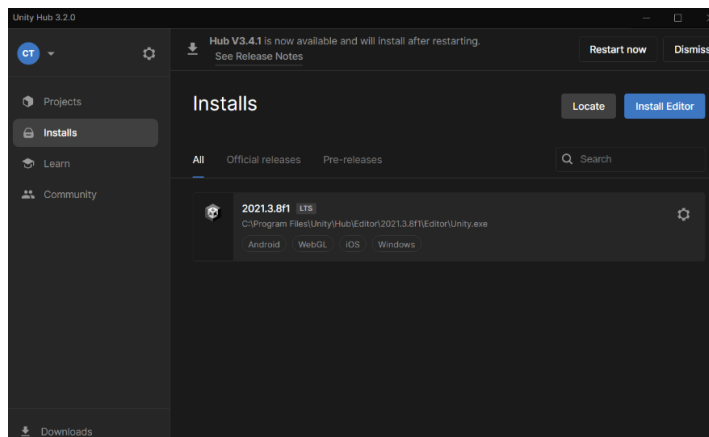


Figura 22. Instalación de Unity desde Unity Hub.

En el caso de este proyecto, es necesario realizar la instalación de varios módulos adicionales para el funcionamiento del proyecto. Los módulos necesarios son Android Build Support, Android SDK & NDK Tools y OpenJDK, debido a que la plataforma de ejecución de este proyecto son las gafas de realidad virtual Oculus Quest 2 y todos los dispositivos Oculus funcionan a través de la plataforma Android (Figura 23).

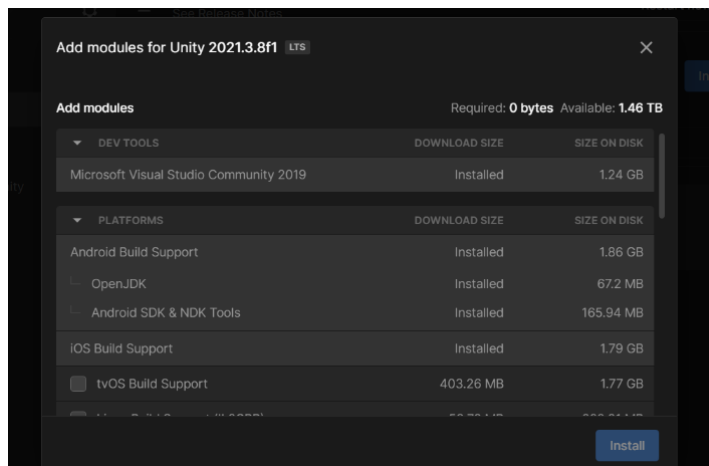


Figura 23. Módulos adicionales Unity.

4.1.2 Instalación de los paquetes necesarios

En esta sección del documento se explica el proceso de instalación de los diferentes paquetes y plug-ins que se van a utilizar durante el desarrollo del proyecto.

Plug-in Open XR

Tal y como se ha mencionado en el apartado “3.2. Materiales”, el desarrollo de este proyecto de realidad virtual hace uso del plug-in Open XR.

Este plug-in presenta todas las características necesarias para realizar un proyecto VR en Unity, como puede ser el seguimiento de tracking de las gafas, los controladores y las cámaras predefinidas.

A continuación, se detallan los pasos necesarios a seguir para realizar la instalación completa de este plug-in.

1. Accede a la pestaña *Window* de Unity y selecciona el apartado *Package Manager*.
2. Al seleccionar ese apartado, se abre un desplegable del mismo nombre. En la esquina superior izquierda, comprueba que está seleccionada la opción *Packages: Unity Registry* ya que si aparece la opción *Packages: In Project* sólo se mostrarán aquellos paquetes que ya estén implementados en el proyecto.
3. En la esquina superior derecha, escribe en la barra de buscador el nombre del plug-in que desees buscar. En este caso, el nombre completo es *OpenXR Plugin*.
4. Con el plug-in seleccionado, pulsa el botón *Instalar* que se muestra en la parte inferior de la pantalla. De esta manera, se completa la instalación del plug-in dentro del proyecto.
5. Este apartado es opcional, solo han de realizarlo aquellas personas que deseen hacer uso de los componentes que incluye el plug-in OpenXR. Para descargarlos y que aparezcan en la carpeta *Assets* del proyecto hay que realizar las siguientes acciones:
 - a. Seleccionar y desplegar el apartado de nombre *Samples*
 - b. Importar los assets que sean necesarios para el proyecto. El plug-in incluye assets de tipo *Controller*, *Meshing Subsystem* y *Intercept Feature*.

En la siguiente imagen (Figura 24) se puede observar el resultado de instalar correctamente el plug-in OpenXR y sus assets en el proyecto.

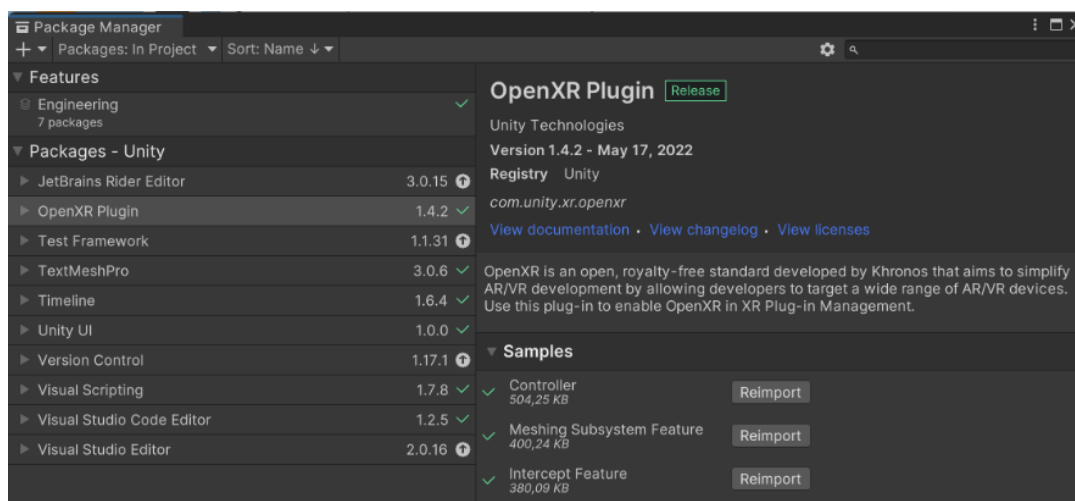


Figura 24. Instalación del plug-in OpenXR.

Otra forma de comprobar que se ha realizado la instalación del plug-in correctamente es observar que ha aparecido una carpeta de nombre *OpenXR Plugin* dentro de la carpeta *Assets/Samples* del proyecto (Figura 25).

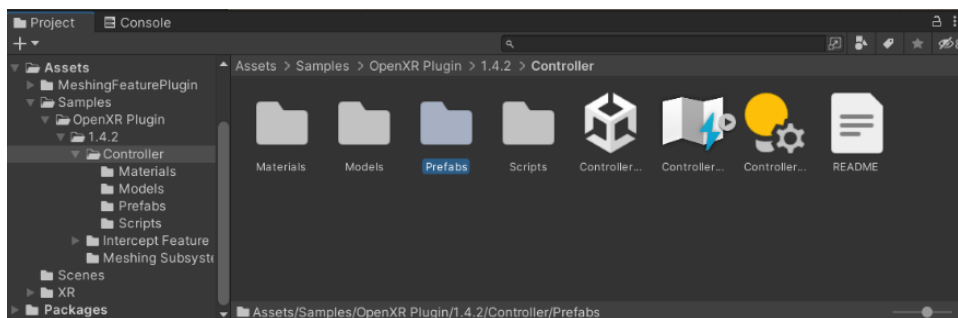


Figura 25. Carpeta OpenXR instalada en la Carpeta Assets.

Plug-in XR Interaction Toolkit

El plug-in *XR Interaction Toolkit* es necesario para realizar los diferentes tipos de interacción en escenas de RV y RA.

Para realizar su instalación, hay que seguir los siguientes pasos los cuales son muy similares a los mencionados anteriormente.

1. Accede a la pestaña *Window* de Unity y selecciona el apartado *Package Manager*.
2. Al seleccionar ese apartado, se abre un desplegable del mismo nombre. En la esquina superior izquierda, comprueba que está seleccionada la opción *Packages: Unity Registry* ya que si aparece la opción *Packages: In Project* sólo se mostrarán aquellos paquetes que ya estén incluidos en el proyecto.
3. En la esquina superior derecha, escribe en la barra de buscador el nombre del plug-in que desees buscar. En este caso, el nombre completo es *XR Interaction Toolkit*.
4. Con el plug-in seleccionado, pulsa el botón *Install* que se muestra en la parte inferior de la pantalla. De esta manera, se completa la instalación del plug-in dentro del proyecto.
5. Este apartado es opcional para aquellas personas que deseen hacer uso de las muestras que incluye el plug-in. Para descargarlas y que aparezcan en la carpeta *Assets* del proyecto hay que realizar las siguientes acciones:
 - a. Seleccionar y desplegar el apartado de nombre *Samples*
 - b. Importar los assets que sean necesarios para el proyecto. El plug-in incluye assets de tipo *Starter Assets*, *XR Device Simulator*, *Tunneling Vignette*, *Meta Gaze Adapter*, *Hands Interaction Demo*.

La Figura 26 muestra el resultado de instalar correctamente el plug-in OpenXR y sus assets en el proyecto. En este caso, sólo es necesario el paquete *Starter Assets*.

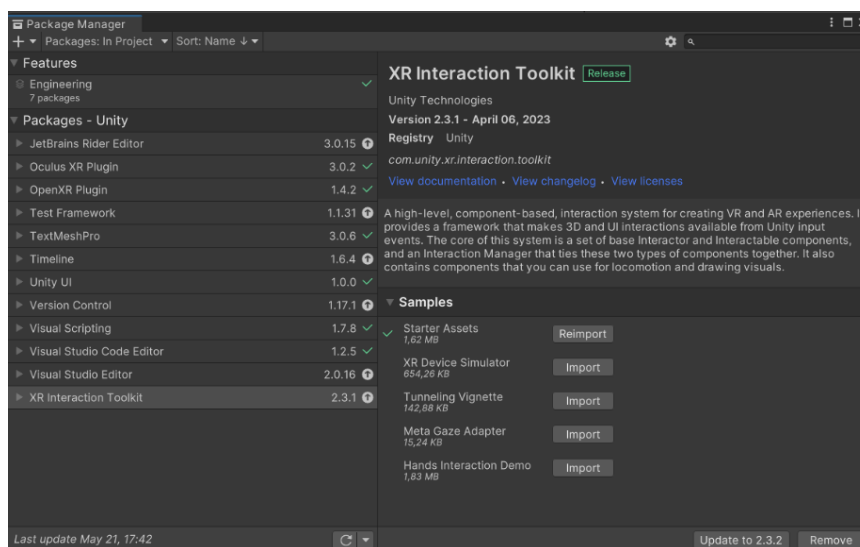


Figura 26. Instalación plug-in XR Interaction Toolkit.

4.1.3 Debug de la aplicación

Para trabajar con el software de Oculus y sus dispositivos es necesario llevar a cabo una serie de pasos. En primer lugar, se debe instalar la aplicación Oculus. Es recomendable instalar esta aplicación tanto en el PC como en el teléfono móvil ya que uno de los próximos pasos sólo puede ser realizado a través del teléfono. En el Anexo I se describen los pasos a seguir para instalar la aplicación.

El siguiente paso es la creación de una cuenta de desarrollador de Meta. Esta cuenta permite a los desarrolladores acceder a las herramientas y recursos de desarrollo para la creación, testeo y distribución de aplicaciones o juegos para los dispositivos Oculus.

Para crear la cuenta de desarrollador hay que acceder al siguiente enlace:

<https://developer.oculus.com/>

Dentro de la página web se debe realizar el inicio de sesión o el registro en caso de que sea la primera vez que se accede. Para completar el registro es necesario aportar algunos datos de información personal como el nombre, el nombre de usuario y un correo electrónico.

Tras confirmar y verificar la cuenta, es necesario hacer un paso adicional para aquellos usuarios que deseen trabajar con sus dispositivos en modo desarrollador. Para este paso, es necesario tener instalada en un dispositivo móvil la aplicación de Oculus. Desde el apartado Menú de la aplicación, seleccione Dispositivos y Configuración de las gafas (Figura 27). Una vez llegado a ese punto, debe activar el apartado que indica modo desarrollador. De esta manera, se podrá trabajar con las gafas para probar y ejecutar aplicaciones externas.



Figura 27. Interfaz Aplicación Oculus Dispositivos Móviles.

Para conectar las gafas al ordenador simplemente hay que conectarlas a través de un cable de conexión de tipo USB-A a USB-C. Tras realizar la conexión, aparecen unas instrucciones muy sencillas en la aplicación de Oculus que muestran si la sincronización es correcta.

Para poder reproducir aplicaciones de ordenador en las gafas, es necesario acceder al modo Oculus Link que presentan las gafas Oculus. Oculus Link es una función de software desarrollada por Oculus para sus dispositivos que proporciona a los usuarios la capacidad de conectar sus gafas al ordenador. Como resultado se obtiene una biblioteca más amplia de aplicaciones y experiencias de realidad virtual de más alta calidad con gráficos mejorados y rendimiento más fluido.

Para acceder a Oculus Link desde las gafas de realidad virtual hay que seguir una serie de pasos:

1. Asegurarse de que las gafas están correctamente conectadas desde la aplicación Oculus
2. Acceder al menú Configuración desde el menú flotante de las gafas.
3. Seleccionar la ventana que indica Oculus Link
4. Dentro de la ventana aparecerá el nombre del ordenador al que están conectadas las gafas, clicar en él y pulsar el botón Activar
5. En la aplicación Oculus del PC aparecerá una notificación indicando que el dispositivo ha sido conectado. Pulsar en el botón Continuar.
6. Tras este punto, dentro de las gafas, aparecerá el nuevo entorno Oculus Link.

En este proyecto, Oculus Link ha tenido un papel fundamental durante todo el desarrollo. Su principal función ha sido realizar las diferentes comprobaciones y pruebas de los desarrollos conforme avanzaba el proyecto. Gracias a esto, se han podido ajustar los vídeos y las escenas a la visualización en gafas de realidad virtual en tiempo real, sin necesidad de estar creando una aplicación para cada prueba.

4.2 Creación de la jerarquía necesaria para visualizar vídeo estereoscópico

El proyecto sigue una estructura lo más compacta y simplificada posible, de manera que el tiempo necesario de ejecución y renderizado sea el mínimo posible. Esto se busca con la finalidad de simplificar la potencia necesaria que requerirá la aplicación a la hora de visualizarla de manera autónoma en las gafas de RV.

Esta sección detalla la estructura necesaria para visualizar vídeos estereoscópicos en Unity. En cada apartado, se explica cada uno de los pasos que se han seguido y sus elementos principales. Antes de realizar la estructura para visualizar vídeo estereoscópico, hay que establecer el sistema de seguimiento del casco de realidad virtual. Además, se realizó un estudio sobre las diferentes formas de visualizar un vídeo convencional en Unity, de tal manera que fuera más sencillo comprender cuáles serán los componentes necesarios para trabajar con vídeos estereoscópicos.

4.2.1 Sistema de seguimiento de tracking

Para el sistema de seguimiento de tracking de los cascos de realidad virtual, se va a hacer uso de las muestras y ejemplos predeterminados que presenta el plugin de OpenXR. En los apartados anteriores, se detalla cómo realizar la instalación completa de este plugin y sus ejemplos.

El primer paso consiste en crear un elemento vacío de nombre “PlayerRig”. Este componente se utiliza comúnmente en el desarrollo de experiencias de realidad virtual y realidad aumentada para proporcionar un punto de referencia para el jugador.

Sirve como objeto que contiene todos los componentes necesarios para posicionar y rotar la cámara y los controladores del usuario dentro del espacio virtual. Además, se encarga de proporcionar la información de seguimiento necesaria para la interacción de los dispositivos de entrada y el mantenimiento de la escala y perspectiva correctas durante toda la experiencia.

En este caso, se han utilizado tres scripts predeterminados del plugin OpenXR (Figura 28):

- “*Tracking Mode Origin*”: Script encargado de determinar el punto de origen del sistema de seguimiento del dispositivo de realidad virtual. Este punto sirve como referencia para la posición y la orientación del usuario dentro del espacio virtual.
- “*Action To Visibility*”: Script utilizado para mostrar u ocultar objetos en la escena en función de las acciones de los controladores.
- “*Action Asset Enabler*”: Este script presenta funcionalidades similares al anterior, permite habilitar o deshabilitar activos de Unity en respuesta a las acciones de los controladores.

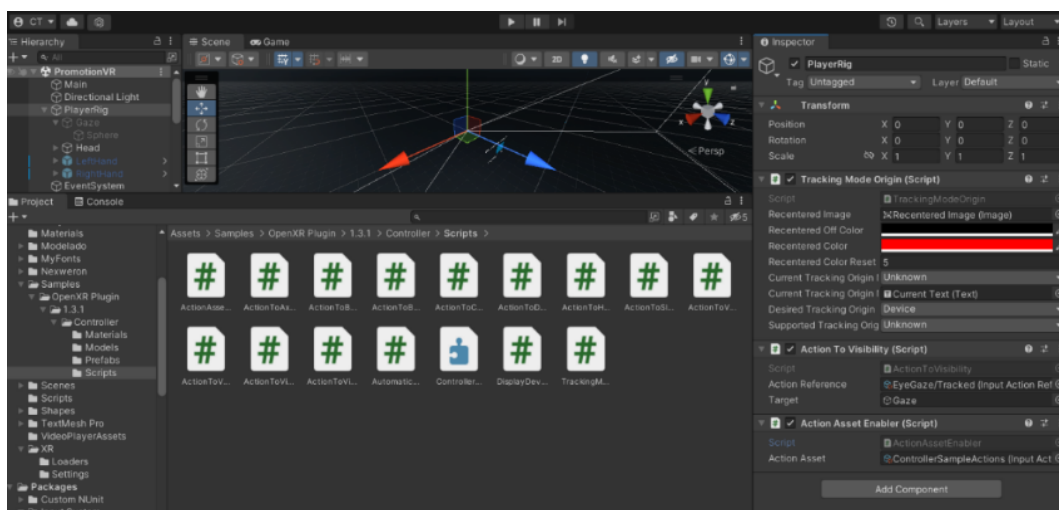


Figura 28. Scripts predeterminados del plug-in OpenXR.

Dentro del componente “PlayerRig” se encuentran tres elementos básicos: el elemento de nombre Gaze, el elemento Head el cual contiene la estructura de cámaras que se explicará en el siguiente apartado y, por último, se pueden añadir opcionalmente y si la escena los necesita los elementos LeftHand y RightHand, que son los prefabs que presenta el plugin de OpenXR para utilizar controladores en la escena. Por tanto, la estructura final que presenta el componente PlayerRig se puede observar en la Figura 29.

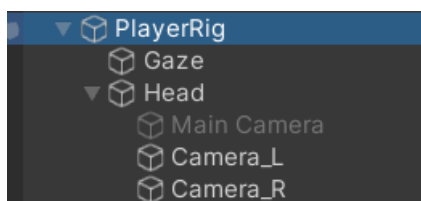


Figura 29. Estructura Componente PlayerRig.

El elemento Gaze sirve como identificador de la posición y rotación de la cabeza del usuario gracias a la implementación del script “Tracked Pose Driver” (Figura 30). Este script se utiliza para ajustar la posición y orientación del objeto en función del movimiento del dispositivo y gracias a ello, se consiguen experiencias más inmersivas para los usuarios.

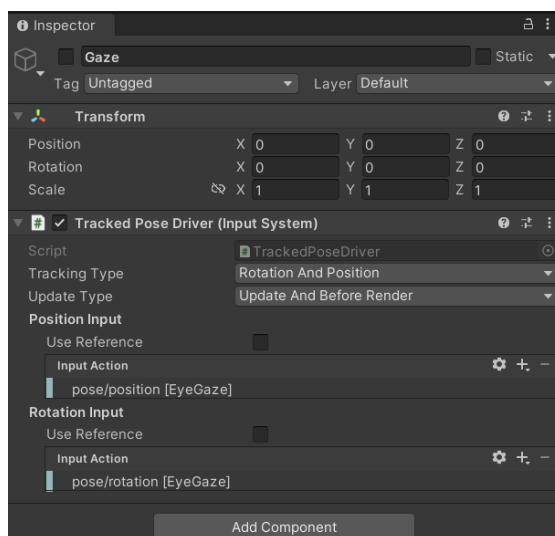


Figura 30. Componente Gaze.

4.2.2 Componente Head y Cámaras

El reproductor desarrollado en este proyecto debe ser capaz de visualizar vídeos estereoscópicos. Todos los vídeos estereoscópicos utilizados en el desarrollo final han sido capturados con la cámara Z CAM K1 Pro (Figura 31). Esta cámara ha sido fabricada por la empresa Z CAM y presenta dos lentes que imitan la visión humana binocular lo cual permite capturar imágenes tridimensionales.



Figura 31. Cámara Z CAM K1 Pro.

Entre las características más importantes que presenta podemos destacar la capacidad de capturar tanto imágenes como vídeo, la inclusión de un puerto Ethernet y el soporte del formato VR180. Cada lente de la cámara va asociada a una tarjeta SD donde se almacenan las imágenes o vídeos capturados para cada ojo. Por tanto, para generar la imagen estereoscópica es necesaria la edición posterior de estos elementos para convertirlos en un solo fichero Side by Side, donde la parte izquierda de la imagen almacena la imagen del ojo izquierdo y la parte derecha de la imagen almacena la imagen del ojo derecho.

Al juntar ambas imágenes, se obtiene lo que se conoce como imagen estereoscópica. A continuación, se muestra un ejemplo de uno de los vídeos resultantes tras su edición (Figura 32).



Figura 32. Vídeo Estereoscópico.

Para representar este tipo de vídeos a través de la herramienta Unity de manera estereoscópica hay que visualizar la parte derecha del vídeo con el ojo derecho y la parte izquierda con el ojo izquierdo respectivamente.

Unity presenta varias maneras para representar imágenes distintas para cada lente de las gafas de realidad virtual. En este proyecto, se va a trabajar con dos cámaras convencionales de Unity, de manera que cada una de ellas enfoque hacia la posición de la imagen correspondiente. Al crear esta jerarquía no es necesario usar cámaras de realidad virtual más complejas que requieren de conocimientos más específicos para su uso.

Para que las cámaras mantengan el mismo sistema de seguimiento, se ha creado un elemento padre de nombre *Head* al cual se le asignado el script "*Tracked Pose Driver*". Como hijos de este elemento se han colocado dos cámaras de Unity de nombre "*Camera_R*" y "*Camera_L*" cada una representando a la imagen que verán, imagen derecha (*right*) e imagen izquierda (*left*), respectivamente. La Figura 33 muestra la jerarquía del elemento *Head* y sus componentes.

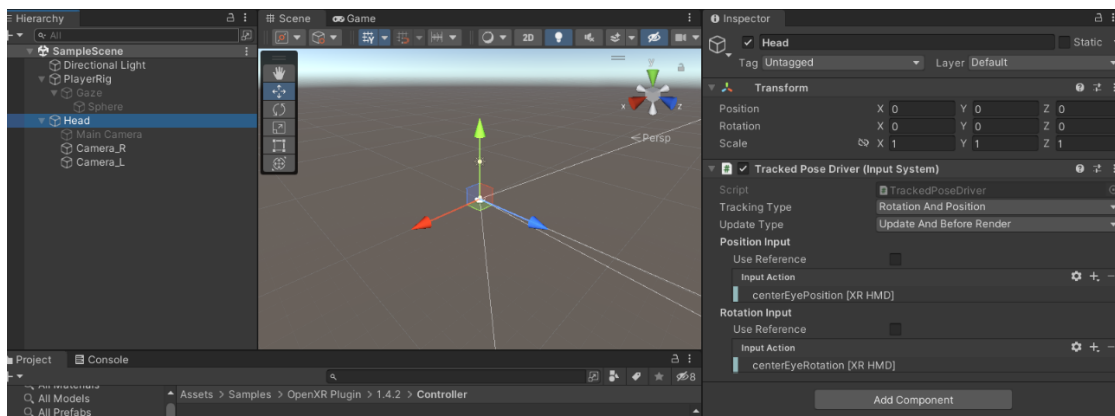


Figura 33. Elemento Head y sus componentes.

Es importante remarcar la propiedad *Target Eye* de las cámaras. Esta propiedad se utiliza para especificar en qué ojo del usuario se renderizará la imagen capturada por la cámara. Las opciones disponibles para la propiedad son “*Both*”, “*Left*” y “*Right*” y como su nombre indica renderizan para ambos ojos, para el ojo derecho o para el ojo izquierdo.

Cada proyecto es único y diferente, por eso es necesario escoger la opción adecuada en función del objetivo de la cámara y del entorno con el que se está trabajando. Para este proyecto, la configuración ha quedado de la siguiente manera:

- Camera_R con propiedad Target Eye → Right
- Camera_L con propiedad Target Eye → Left

4.2.3 Componente VideoPlayer

Unity dispone de un componente de nombre *VideoPlayer* el cual se utiliza para adjuntar archivos de vídeo a objetos y reproducirlos sobre la textura de ese objeto en tiempo real. La Figura 34 muestra un objeto *GameObject* esférico al cual se le ha aplicado el componente *VideoPlayer*.

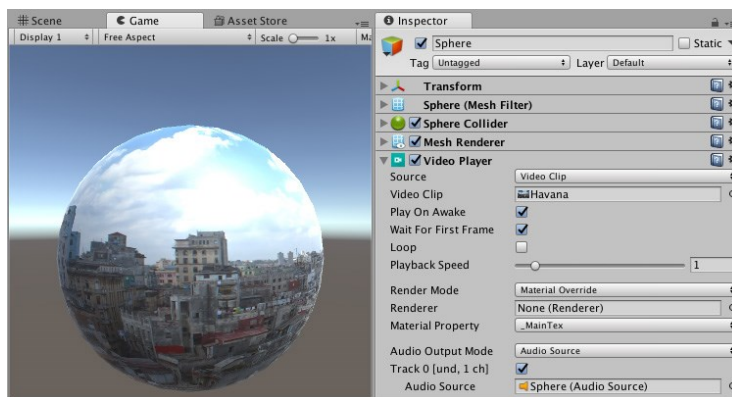


Figura 34. Componente VideoPlayer [31].

Por defecto, el modo de renderizado viene establecido como Material Override, lo que significa que el vídeo se asigna automáticamente a la textura del objeto en su malla. En la imagen anterior, la esfera tiene un componente *Mesh Renderer*, por lo que el clip del vídeo se reproduce sobre la textura del *Mesh Renderer* de la esfera.

Sin embargo, hay otros modos de renderizado de vídeo con este componente. Los targets específicos de renderizado (Render mode) diferentes que proporciona el componente son: *Camera Far/Near Plane*, *Render Texture* o *API Only*. En el siguiente apartado, se explica el motivo por el cual se ha escogido el modo *Render Texture* y sus características.

Antes de explicar el modo de renderizado escogido, se explican otras características del componente *VideoPlayer* (Figura 35). Desde el apartado *Source* se escoge el vídeo a reproducir, el cual puede ser un vídeo que esté implementado dentro del proyecto o una URL de un vídeo externo.

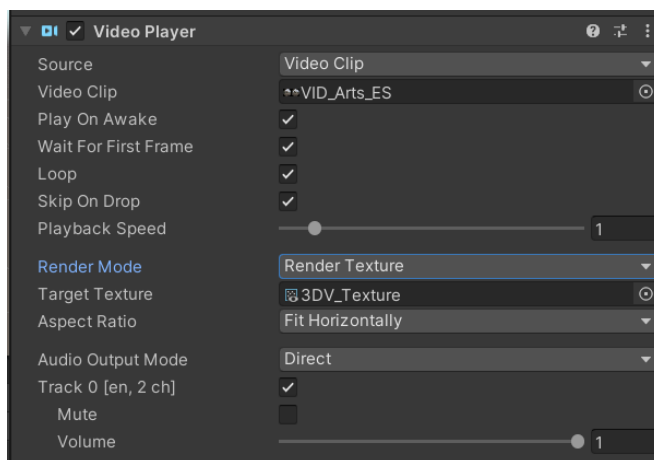


Figura 35. Características VideoPlayer.

En este proyecto se ha trabajado con los vídeos importados dentro del proyecto de Unity de manera que, al generar la aplicación final para instalarla en las gafas, todos los vídeos estuvieran descargados y no fuera necesario dedicar recursos a la descarga de los mismos desde una dirección URL. Además, el posterior desarrollo realizado sobre los vídeos de los guías turísticos requiere de un paso obligatorio que debe ser realizado dentro del propio proyecto de Unity.

4.2.4 *Render Texture*

Este proyecto utiliza el modo *Render Texture* del componente *VideoPlayer* para visualizar los vídeos estereoscópicos. Las *Render Texture* son un tipo específico de texturas que se crean y actualizan en tiempo de ejecución [32].

Este tipo de texturas se crean de una manera muy sencilla: desde el apartado Assets se pulsa el botón izquierdo del ratón y dentro del desplegable pulsar Create → Render Texture. Al crear la textura, aparece una pestaña la cual muestra las características más importantes de la misma como se puede apreciar en la Figura 36.

De todas las características se remarca el apartado Size. Este apartado establece el tamaño de la textura en píxeles y es necesario que coincida con el tamaño del vídeo a reproducir para que no posteriormente no haya problemas de visualización en la calidad del vídeo. En este proyecto, los vídeos se han grabado con una resolución 2K o Quad HD para proporcionar una imagen de calidad y nítida al visualizarse sobre el visor de realidad virtual.

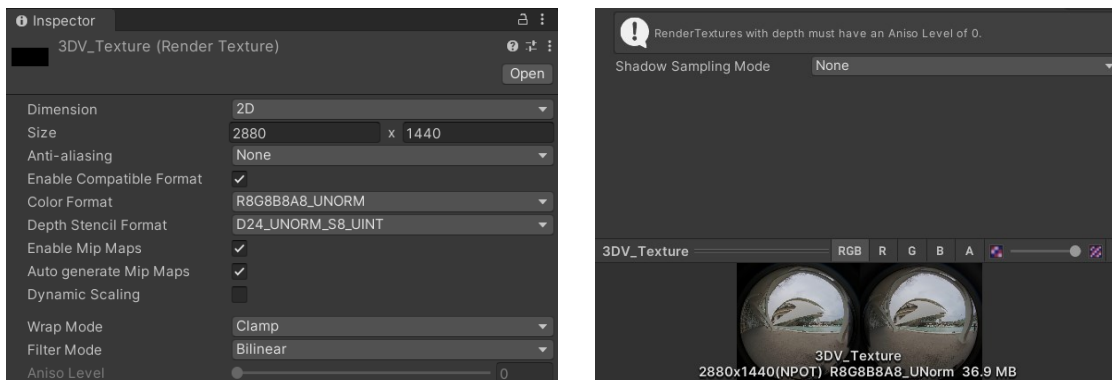


Figura 36. Creación de una Render Texture

4.2.5 Esferas

El vídeo estereoscópico ha sido grabado de manera panorámica, lo que significa que tiene una relación de aspecto más ancha que un vídeo en formato estándar. Comúnmente, el formato de los vídeos panorámicos es 16:9 y es necesario respetarlo para garantizar que la imagen se muestra correctamente en todos los dispositivos y pantallas sin ser distorsionada. Gracias a este formato, los vídeos presentan una experiencia de visualización más inmersiva ya que es similar al campo de visión humano y permiten mostrar una gran cantidad de detalles en las imágenes.

El vídeo será proyectado sobre el interior de dos esferas (una para cada ojo), colocadas de manera superpuesta una sobre otra en la posición central de la escena. A pesar de estar colocadas en la misma posición, cada una estará situada en un *layer* diferente de manera que cada cámara vea sólo una. Ambas esferas proyectan el mismo vídeo por lo que es necesario girar una de las esferas 180° sobre el eje Y de manera que la imagen que visualiza la cámara derecha sea la contraria a la que visualiza la cámara izquierda. Ambas cámaras se situarán en el centro interior de estas esferas, de manera que el usuario sentirá que está dentro de la propia escena turística.

Al colocar la cámara en el interior de las esferas surge un problema, ya que Unity sólo renderiza las texturas de un objeto por el lado exterior del mismo de manera predeterminada. Sin embargo, para que se pueda visualizar el vídeo debe ser proyectado en el interior de las esferas. Para conseguir que así sea, es necesaria la utilización de un *Shader BackCullingOff*. En Unity, los *Shader* se utilizan para definir cómo se renderizan los objetos en la escena, es decir definen la apariencia de los objetos desde su textura, color y brillo hasta si presentan sombras o efectos especiales.

En concreto, el *Shader BackCullingOff* utilizado en este proyecto es un tipo de *shader* que permite desactivar la propiedad *culling* en la parte posterior de los objetos (Figura 37). La propiedad *culling* es un proceso de Unity que elimina los objetos que no son visibles en la pantalla. Al desactivarla, los objetos se dibujarán por ambos lados, aunque el espectador sólo visualice uno de ellos. Es un proceso muy interesante y suele ser usado en objetos planos como árboles, sin embargo, puede limitar el rendimiento de la aplicación ya que se están dibujando más polígonos de los necesarios [33, 34].

Para evitar problemas posteriores, durante el desarrollo del proyecto se han realizado pruebas para comprobar que este proceso era posible realizarlo de manera autónoma en los cascos de realidad virtual Oculus Quest 2.

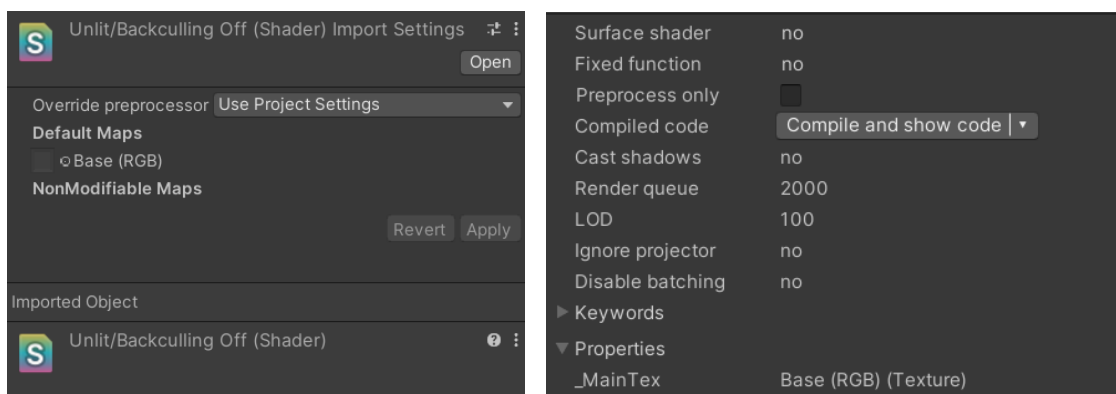


Figura 37. Shader Backculling Off

Dentro de cualquier proyecto de realidad virtual realizado con Unity, el usuario tiene libertad de movimiento dentro de la propia escena por lo que, si desplaza la cabeza, la escena también se desplaza.

En este proyecto, el rango de visualización adecuado es de aproximadamente 180° en vertical y en horizontal. Para que al usuario le sea más fácil mantenerse enfocado en la escena, se han colocado dos planos, uno para cada ojo los cuales simulan un efecto de *clipping*, de ahí su nombre.

El *clipping* es un proceso utilizado en gráficos 3D para eliminar objetos que se encuentran fuera del rango de visión de la cámara. En otras palabras, se trata de recortar objetos que están demasiado lejos o demasiado cerca de la cámara para que no aparezcan en la escena.

En el caso de este proyecto, se va a simular este efecto a partir de la creación de unos planos de color negro colocados en mitad de las esferas. De esta manera, si el usuario comienza a girar la cabeza más de 180° verá por pantalla un borde negro que indica que no está colocado en la posición correcta.

Para mantener una estructura ordenada y jerarquizada dentro del proyecto de Unity, se ha hecho uso de diferentes objetos vacíos los cuales sirven de referencia para la posición y orientación de los elementos hijos. Toda la estructura que corresponde a representar un vídeo estereoscópico de un lugar turístico se ha agrupado dentro del elemento de nombre Video360 (Figura 38).

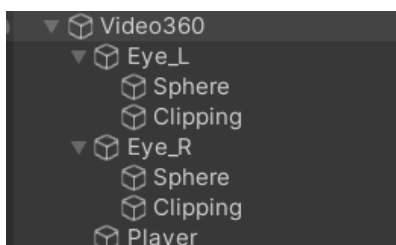


Figura 38. Estructura para visualizar vídeo estereoscópico

El resultado de las esferas en la escena de Unity debe quedar similar a lo mostrado en la Figura 39.

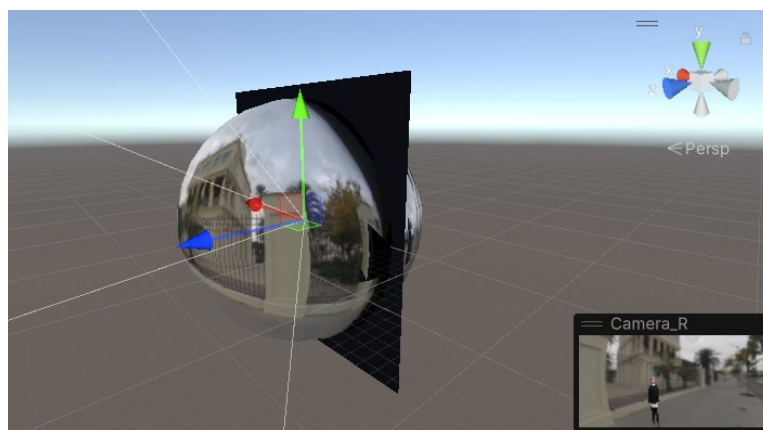


Figura 39. Resultado de la representación de un vídeo estereoscópico

4.3 Análisis del eje óptico de la cámara estereoscópica

Esta sección del documento detalla el estudio realizado sobre el eje óptico de la cámara estereoscópica utilizada para las grabaciones del proyecto. El propósito de este análisis es comprobar qué tipo de eje presenta el eje óptico de la cámara Z CAM K1 Pro para poder realizar el desarrollo teniendo en cuenta los resultados de este estudio.

Se espera que la cámara siga un seguimiento lineal ya que suelen estar diseñadas de esta manera para evitar provocar mareos en los usuarios a la hora de visualizar los vídeos que graban.

Estas sensaciones de mareo o náuseas se conocen como *Virtual Sickness* y es bastante común que se produzca con el mal uso o el uso prolongado de dispositivos de RV como los HMD. Esta sensación es causada por la discrepancia entre lo que el cerebro espera ver y lo que realmente está sucediendo en el entorno virtual o aumentado. En otras palabras, si en la experiencia el cuerpo

del avatar se está moviendo y en la realidad el usuario no se está moviendo, se produce esta discrepancia que produce sensación de malestar. Algunas personas son más propensas a experimentar esta sensación que otras, y puede ser mitigada en cierta medida por medio de ajustes en la configuración de la experiencia de realidad virtual o aumentada [35].

Por este motivo, se ha realizado el siguiente análisis para comprobar cuál es el funcionamiento de la cámara y realizar los ajustes necesarios en la configuración en caso de que fuera necesario. A continuación, se describen los diferentes pasos que se siguieron para la realización de este análisis.

4.3.1 Toma de medidas y capturas

El primer paso para realizar el análisis del eje óptico de la cámara estereoscópica es obtener una base de la cual poder comparar resultados. Para ello, se han realizado un conjunto de capturas similares a diferentes distancias. Cada captura enfoca a un cinta métrica colocada en la misma posición, de manera que posteriormente se pueda comparar la distancia en centímetros y su distancia correspondiente en píxeles obteniendo así el grado de distorsión de la cámara.

En primer lugar, se realizaron unas fotografías de prueba para observar el funcionamiento de la cámara y encontrar la distancia y lugar correctos en los que posicionar la cinta métrica (Figura 40). Se han realizado capturas a diferentes distancias: a 5 cm respecto a la pared, a 15 y 20 cm.

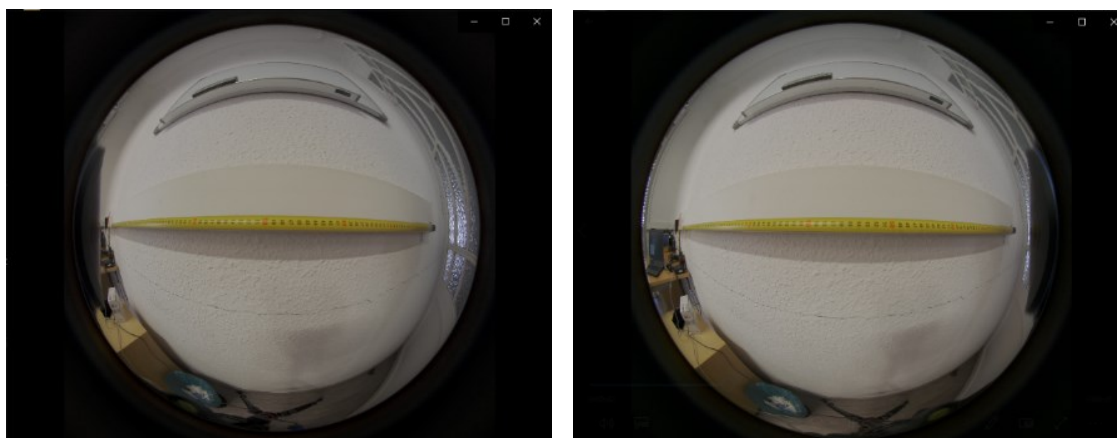


Figura 40. Imágenes de prueba a 20 cm

4.3.2 Obtención de los datos

Tras realizar la captura de las diferentes imágenes, se escogió la imagen más nítida para poder trabajar sobre ella con la herramienta Adobe Photoshop y obtener los datos necesarios para el posterior análisis.

El primer paso consiste en marcar el eje central de la cámara, tanto de manera horizontal como de manera vertical (Figura 41). Tras marcar los ejes, se tomaron las medidas correspondientes de los centímetros visualizados en la imagen y su medida correspondiente en el sistema de medición en píxeles que marca Photoshop.



Figura 41. Extracción de los datos con Photoshop

4.3.3 Análisis de los datos con Microsoft Excel

Los datos obtenidos en Photoshop se anotaron en una tabla de Microsoft Excel para poder analizarlos posteriormente. Se debe apuntar tanto la medida en centímetros en el eje horizontal y vertical (Xcm,Ycm). Además, se tomaron las medidas del centro óptico de la imagen (X0, Y0) tanto en centímetros como en píxeles (Figura 42).

X0 (cm)	48,5
Y0 (cm)	0,28
Centro óptico en píxeles	
X0 (px)	1435
Y0 (px)	1455

Figura 42. Centros de la imagen analizada

Esta medida es importante obtenerla para poder realizar el segundo paso. Este consiste en afinar el valor obtenido de Photoshop al valor real respecto al centro de la imagen. Para realizar este ajuste, se ha seguido la siguiente fórmula:

$$\begin{aligned}
 x'_{cm} &= x_{cm} - x_{0cm} \\
 y'_{cm} &= y_{cm} - y_{0cm} \\
 x'_{pi} &= x_{pi} - x_{0pi} \\
 y'_{pi} &= y_{pi} - y_{0pi}
 \end{aligned}
 \tag{1.1}$$

A continuación, se va a obtener la distancia (d), tanto en centímetros como en píxeles. Esta distancia es necesaria para realizar el cálculo del ángulo de distorsión (Alpha) que provoca la cámara.

$$\begin{aligned}
 d &= \sqrt{x'^2 + y'^2} \\
 \alpha &= \text{tg}^{-1}\left(\frac{d}{z_{cm}}\right)
 \end{aligned}
 \tag{1.2}$$

La Figura 43 muestra la tabla final obtenida con todos los datos resultantes.

X (cm)	Yc(cm)	Z (cm)	X' (cm)	Y' (cm)	d (cm)	alpha (º)	xpi (px)	ypi (px)	X' (px)	y' (px)	d (px)
20	0	14,5	-28,5	-0,28	28,501	63,04	2333	1468	898	13	898,094
21	0	14,5	-27,5	-0,28	27,501	62,20	2321	1468	886	13	886,095
22	0	14,5	-26,5	-0,28	26,501	61,32	2308	1468	873	13	873,097
23	0	14,5	-25,5	-0,28	25,502	60,38	2295	1468	860	13	860,098
24	0	14,5	-24,5	-0,28	24,502	59,38	2286	1468	851	13	851,099
25	0	14,5	-23,5	-0,28	23,502	58,33	2268	1468	833	13	833,101
26	0	14,5	-22,5	-0,28	22,502	57,20	2256	1468	821	13	821,103
27	0	14,5	-21,5	-0,28	21,502	56,01	2235	1468	800	13	800,106
28	0	14,5	-20,5	-0,28	20,502	54,73	2218	1468	783	13	783,108
29	0	14,5	-19,5	-0,28	19,502	53,37	2199	1468	764	13	764,111
30	0	14,5	-18,5	-0,28	18,502	51,91	2181	1468	746	13	746,113
31	0	14,5	-17,5	-0,28	17,502	50,36	2158	1468	723	13	723,117
32	0	14,5	-16,5	-0,28	16,502	48,70	2135	1468	700	13	700,121
33	0	14,5	-15,5	-0,28	15,503	46,91	2107	1468	672	13	672,126
34	0	14,5	-14,5	-0,28	14,503	45,01	2083	1468	648	13	648,130
35	0	14,5	-13,5	-0,28	13,503	42,96	2054	1468	619	13	619,136
36	0,02	14,5	-12,5	-0,26	12,503	40,77	2022	1467	587	12	587,123
37	0,02	14,5	-11,5	-0,26	11,503	38,43	1986	1467	551	12	551,131
38	0,04	14,5	-10,5	-0,24	10,503	35,92	1954	1466	519	11	519,117
39	0,04	14,5	-9,5	-0,24	9,503	33,24	1910	1466	475	11	475,127
40	0,04	14,5	-8,5	-0,24	8,503	30,39	1867	1466	432	11	432,140
41	0,06	14,5	-7,5	-0,22	7,503	27,36	1823	1465	388	10	388,129
42	0,06	14,5	-6,5	-0,22	6,504	24,16	1778	1465	343	10	343,146
43	0,08	14,5	-5,5	-0,2	5,504	20,78	1728	1464	293	9	293,138
44	0,08	14,5	-4,5	-0,2	4,504	17,26	1678	1464	243	9	243,167
45	0,12	14,5	-3,5	-0,16	3,504	13,58	1623	1463	188	8	188,170
46	0,13	14,5	-2,5	-0,15	2,504	9,80	1570	1462	135	7	135,181
47	0,14	14,5	-1,5	-0,14	1,507	5,93	1517	1462	82	7	82,298

Figura 43. Resultados obtenidos tras el análisis de los datos

4.3.4 Obtención de resultados

Para observar el comportamiento de la cámara, es necesario comparar la distancia en píxeles y el ángulo de distorsión. Para ello, se ha realizado una gráfica de puntos en la que se representan los datos obtenidos en el anterior apartado (Figura 44).

La gráfica representa la distancia obtenida en píxeles respecto al Alpha calculado a partir de los datos en centímetros. Como se puede ver, los valores siguen un aumento prácticamente lineal, lo que demuestra que la cámara sigue un comportamiento lineal y no es necesario realizar ningún ajuste en la misma.

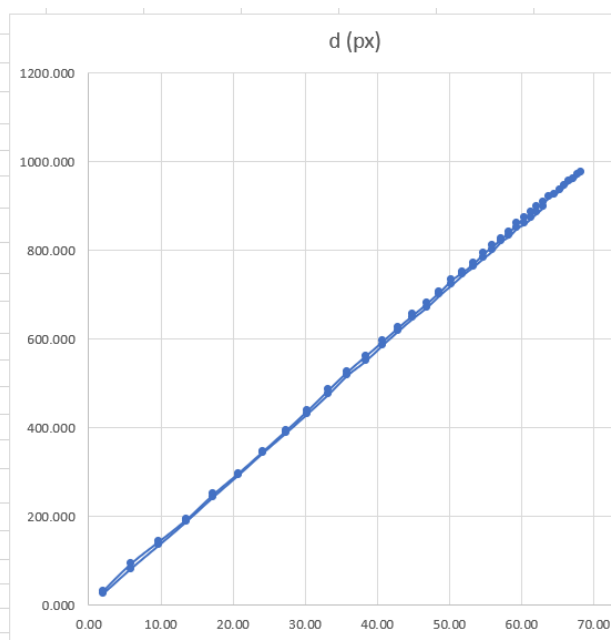


Figura 44. Gráfica Final Obtenida

4.4 Conversión de vídeo RGB a vídeo con capa Alpha

En esta sección del documento se explican los pasos necesarios para convertir la capa RGB de un vídeo a una capa *Alpha*. El canal *Alpha* es un canal añadido al RGB que, en vez de controlar la cantidad de rojos, verdes o azules que hay en un píxel, se encarga de medir el grado de opacidad de los píxeles. De esta manera, permite crear opacidad, semitransparencia o transparencia completa en la capa que se aplique [36].

4.4.1 Importar la secuencia de vídeo

Este paso se lleva a cabo tras la captura de varios guías turísticos en un estudio con *chroma walls*. Para realizar la captura, se recomienda utilizar una cámara profesional, un trípode para mantener la imagen fija y una iluminación uniforme alrededor de toda la escena para intentar evitar tonos muy diferentes de verde. El resultado obtenido de estas capturas son un conjunto de vídeos formados por un fondo verde, un guía turístico y la grabación de audio de la explicación de una localización turística (Figura 45).

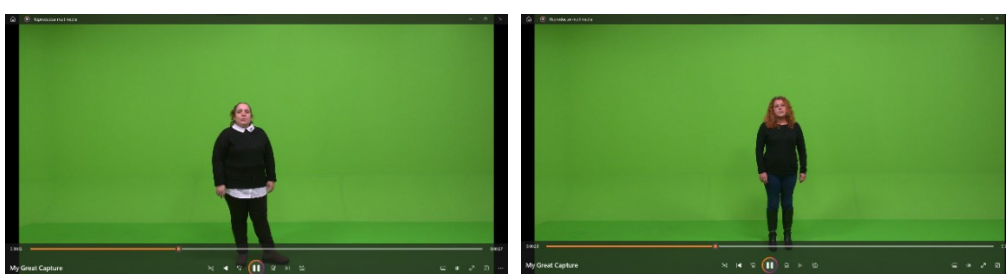


Figura 45. Captura de guías turísticos sobre fondo verde

Para el proyecto, debemos extraer el guía turístico y el audio para insertarlos en la escena con el fondo correspondiente. Para lograrlo, hay que eliminar el fondo verde del vídeo. Esta técnica se conoce como *croma* o *chroma key* y es un procedimiento muy utilizado en cine y vídeo que sirve para eliminar todos los píxeles de un color determinado que hay en una imagen.

Hay muchas aplicaciones en la actualidad capaces de realizar este efecto como pueden ser After Effects, OBS Studio o Final Cut Pro, entre otras. Sin embargo, para este proyecto se ha utilizado la aplicación Adobe Premiere Pro debido a que incorpora una de las herramientas más potentes para eliminar fondos verdes.

El primer paso que hay que realizar es importar la secuencia de vídeo que se quiere editar a la interfaz. Para ello, se selecciona la misma desde la pestaña “Navegador de medios” situada en la parte inferior izquierda de la pantalla y se arrastra hasta la línea de tiempos.

Al colocarla en la línea de tiempos, se puede observar cómo se ha colocado el vídeo y el audio de manera separada, así como la duración completa de la secuencia y la previsualización de esta en la pestaña “Programa” (Figura 46).

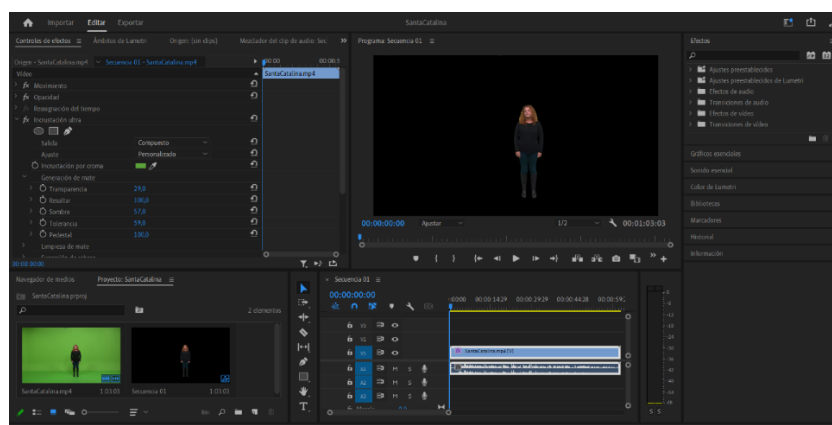


Figura 46. Importar el vídeo a Adobe Premiere

4.4.2 Aplicar el efecto Ultrakey

Con la secuencia de vídeo y audio insertada, el siguiente paso es aplicar el efecto correspondiente. Adobe Premiere Pro cuenta con una gran cantidad de efectos predefinidos en su biblioteca. Para acceder a ellos, sólo es necesario colocarse en la ventana de efectos y buscar a través de la barra de buscador o en la jerarquía de proyectos el que más convenga para tu proyecto.

Premiere presenta diferentes efectos con capacidades similares de extracción de fondo verde. Para este proyecto, tal y como se ve en la siguiente imagen, se ha escogido realizar la eliminación de fondo con el efecto “Incrustación Ultra” localizado en la carpeta “Efectos de vídeo” – “Clave”, que es una de las muchas opciones que presenta Premiere para realizar *chroma key* (Figura 47).

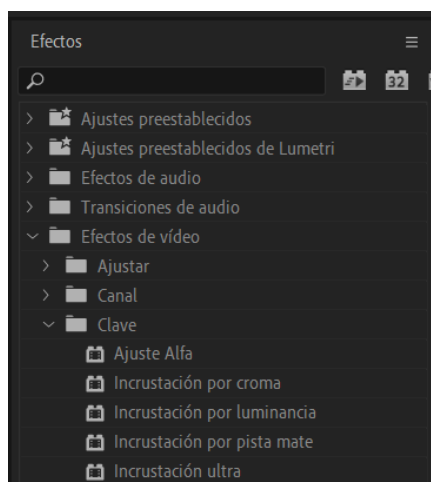


Figura 47. Listado efectos disponibles Premiere

Para aplicar el efecto al vídeo, es tan sencillo como arrastrarlo desde la carpeta hasta el clip situado en la línea de tiempo. Una de las formas de comprobar que se ha aplicado es mirando la pestaña Controles de efecto. Esta pestaña muestra todos los efectos y las características editables que presenta la secuencia de vídeo seleccionada (Figura 48).

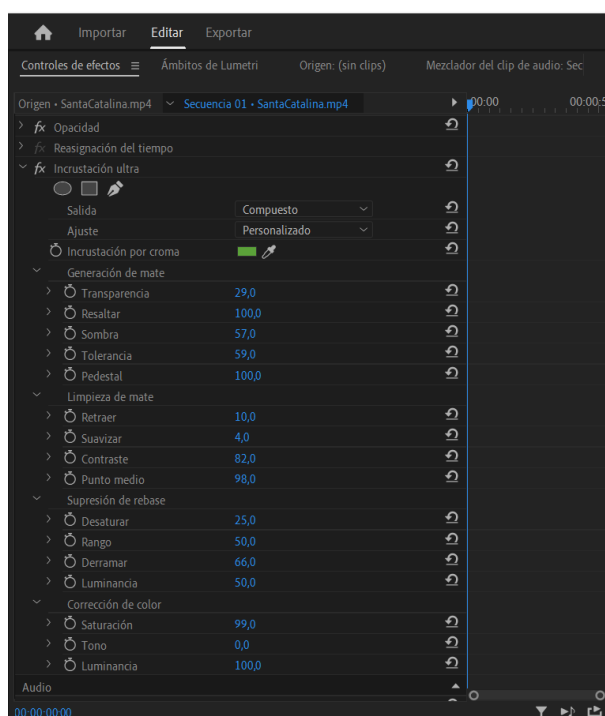


Figura 48. Pestaña Controles de Efectos

El efecto Incrustación Ultra presenta varios apartados con una gran variedad de propiedades. En primer lugar, hay que escoger la salida y el ajuste, es recomendable escoger un ajuste personalizado ya que no todos los cromas son iguales y cualquier diferencia en la sombra o iluminación del vídeo puede no ser detectable y producir una visualización no tan limpia.

El siguiente paso es seleccionar el color del vídeo con la herramienta de Muestra de color. Lo ideal es seleccionar el tono de verde que ocupe la mayoría del vídeo. Posteriormente, Premiere ya muestra una primera previsualización del chroma, pero para apreciarlo de una manera más precisa y exacta, es recomendable seleccionar la salida como Canal Alfa (Figura 49). Esta salida muestra todos los píxeles que se van a mantener de color blanco y todos los píxeles que hay que eliminar de color negro. El siguiente paso es ajustar los valores Generación de Matte y Limpieza de Matte hasta encontrar aquellos que mejor se ajusten a las características del vídeo.

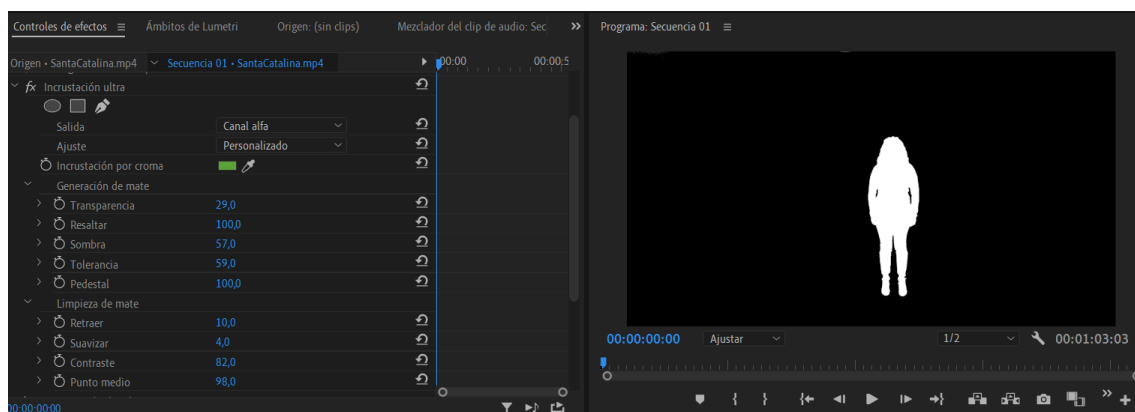


Figura 49. Previsualización del canal Alfa

4.4.3 Exportar el vídeo resultante

El último paso es exportar el proyecto generado con Premiere (Figura 50). Este paso es muy intuitivo, sin embargo, hay que realizar algunas modificaciones importantes para este proyecto.

El formato de salida del vídeo debe ser Webm ya que Unity sólo admite este tipo de formato para visualizar vídeo transparente. Para poder seleccionar esta salida, hay que instalar el códec previamente a través de un plug-in [37].

También, es importante codificar el audio del vídeo con Vorbis ya que Unity trabaja con este códec. Por último, hay que comprobar que el tamaño del vídeo es el adecuado, en este caso se mantiene el mismo que el vídeo de origen. Por último, se selecciona la casilla “Incluir Canal Alpha” para exportar la capa transparente del vídeo.

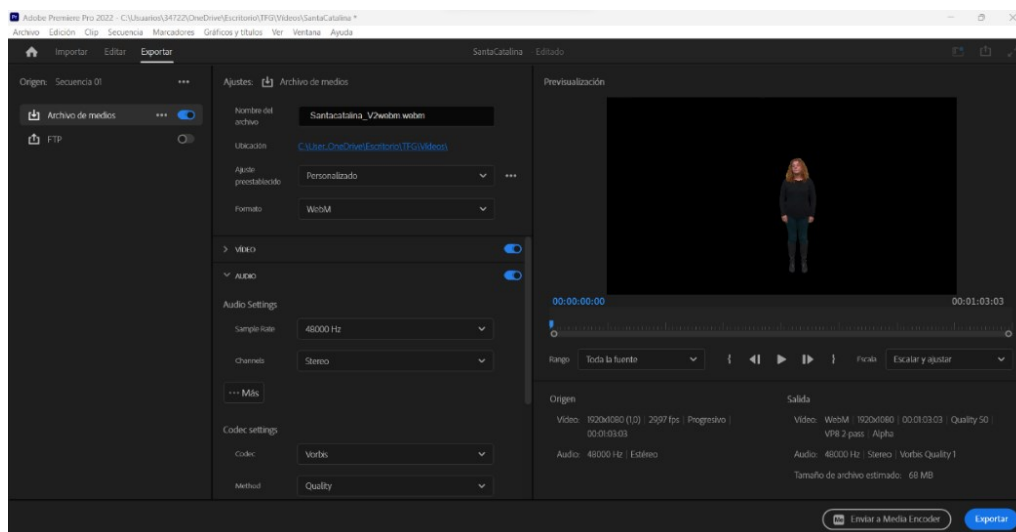


Figura 50. Exportar el vídeo resultante

4.5 Visualización Vídeo Alpha en Unity

Una vez realizada la conversión del formato RGB a *Alpha* de los vídeos, el siguiente paso es visualizarlos dentro de un entorno de Unity.

Estos vídeos van a visualizarse integrados en la misma escena que visualiza los vídeos de fondo de las diferentes localizaciones turísticas. Sin embargo, se ha decidido llevar a cabo una primera prueba individual en la cual se comprueba la correcta visualización de los vídeos para poder realizar posteriormente su integración.

La estructura de elementos necesaria es muy similar a la utilizada para representar el vídeo estereoscópico de fondo. En primer lugar, hay que percatarse de que ahora estamos trabajando con vídeos convencionales, no con vídeos panorámicos estereoscópicos. Por tanto, será necesario proyectar el vídeo sobre un *GameObject* de tipo plano y no sobre un *GameObject* esférico.

Para proyectar el vídeo, se utiliza de nuevo el componente *VideoPlayer*. Este componente utiliza una *Render Texture* y el vídeo. Al importar el vídeo, Unity ofrece la posibilidad de codificarlo. Es importante realizar este paso, de nombre “*Transcode*”, marcando la opción “*Keep Alpha*” para que Unity almacene la capa transparente de la imagen y codificando con el códec VP8 [38].

La Figura 51 muestra un ejemplo de una guía representada sobre un plano. Como se puede observar, la transparencia es correcta ya que se aprecian todos los cubos de colores colocados tras el vídeo de la guía turística. Sin embargo, el presentador se ve pixelado.

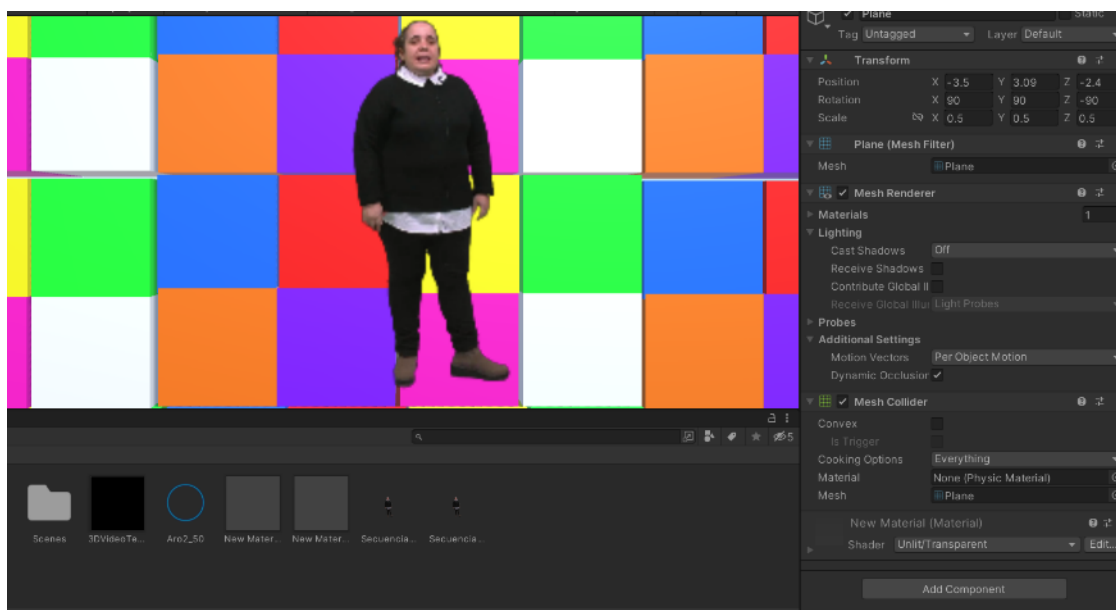


Figura 51. Representación de un vídeo con capa Alfa en Unity

Este efecto se produce debido a que la escala del vídeo y la escala de la textura no coinciden. El vídeo del presentador ha sido capturado con calidad Full HD – 1920 x 1080 por lo que para solucionar el problema hay que reescalar el tamaño de la textura “3DVideoTexture” multiplicando por 16 y dividiendo por un valor de 9 sus valores de tamaño. La Figura 52 muestra el resultado final en el cual se observa a la guía turística nítida, con buena calidad y sin rastro del color verde del fondo original. El funcionamiento de este proyecto de prueba ha sido verificado a través de los cascos de realidad virtual tanto de manera autónoma como conectados al ordenador.

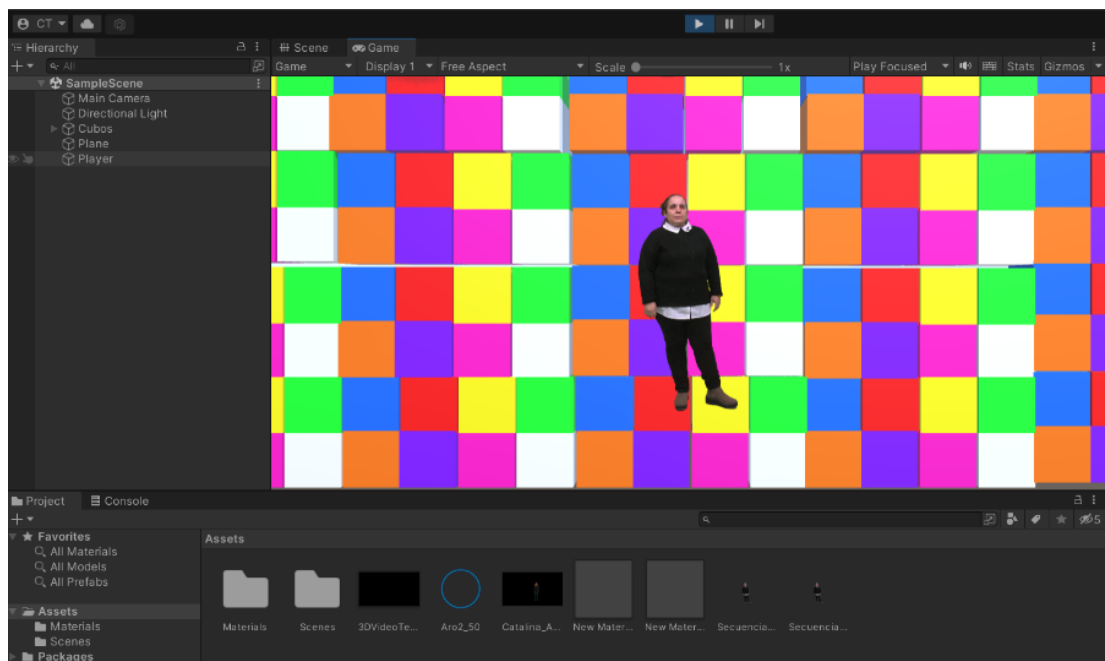


Figura 52. Representación final de un vídeo con capa Alfa en Unity

4.6 Integración del presentador en la escena final

El proyecto de prueba explicado en la sección anterior verifica que Unity es capaz de visualizar vídeos con capa *Alpha* codificados con VP8. El siguiente paso es integrar el presentador en la escena turística desarrollada con Unity anteriormente.

En primer lugar, hay que visualizar el vídeo sobre un plano, siguiendo la misma estructura que en el proyecto de prueba. Esta estructura es muy sencilla, está compuesta por dos elementos: un componente *VideoPlayer* y un elemento 3D de tipo plano. Es importante mantener la relación 16:9 que presentan los vídeos de los guías turísticos para evitar distorsiones en la proyección.

Por último, hay que colocar el presentador dentro de la escena en una posición y escala realista, de manera que parezca que forma parte de la propia escena. Las experiencias turísticas desarrolladas están pensadas para visualizarse a través de los cascos de realidad virtual, por tanto, es complicado escoger la posición visualizando la escena en pantalla plana. Por ese motivo, se recomienda ajustar la escala y posición del plano visualizando simultáneamente el resultado a través de los cascos de RV.

Las Figura 53 muestra el resultado de la integración del guía y el vídeo del lugar turístico.

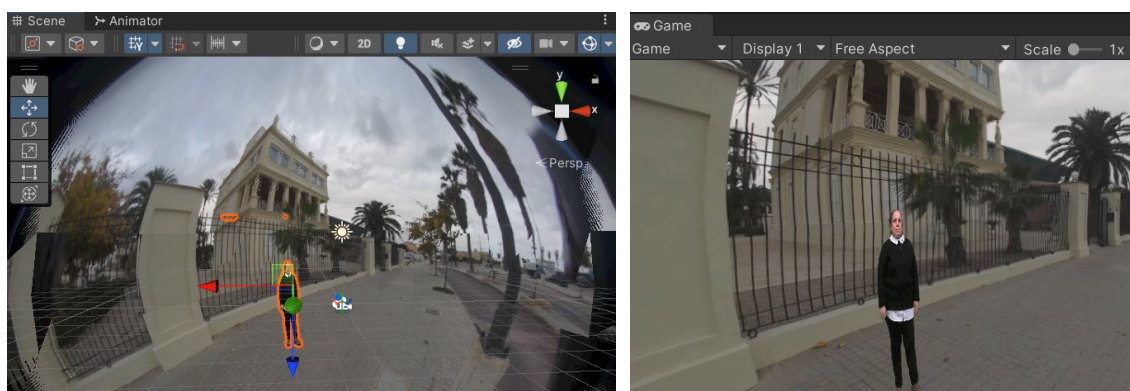


Figura 53. Integración del guía turístico en la escena

4.7 Introducción de un elemento 3D animado y temporizado en la escena

Este proyecto de fin de grado busca la creación de experiencias turísticas inmersivas para los usuarios. Para crear una experiencia novedosa y atractiva, los vídeos se acompañan de un elemento 3D que representa algo característico del lugar de interés.

Estos elementos han sido desarrollados por artistas y modeladores externos a través de diferentes herramientas de creación cómo son Blender o Maya. El visualizador desarrollado acepta cualquier tipo de elemento cuyo formato sea estándar, en este caso se ha trabajado con elementos FBX.

Para mantener la estructura del proyecto organizada, la integración del elemento 3D se realiza en un elemento vacío de nombre *3D_Scene*. Tras cargar el objeto en Unity, se arrastra a la jerarquía como hijo del objeto *3D_Scene*. El elemento *3D_Scene* incorpora un script *Timer* que indica cuándo se debe activar el objeto en la escena (Figura 54) [39].

```
public class Timer : MonoBehaviour
{
    public GameObject Objeto;
    public float second = 22f;
    private float starttime;

    void Start()
    {
        Debug.Log(Time.time);
        starttime = Time.time;
    }

    void Update()
    {
        if (Time.time - starttime > second)
        {
            Objeto.SetActive(true);
        }
    }
}
```

Figura 54. Script Timer

Además, el objeto 3D cuenta otro script de nombre *Animation* (Figura 55). En este script se describen las instrucciones necesarias para animar el objeto. Los objetos cuentan con dos tipos de animaciones. La primera animación consiste en un desplazamiento lateral desde un punto de inicio “*Start*” a un punto final “*End*”. Esta animación comienza en el instante de tiempo en el que el objeto se activa marcado por el script *Timer*. En este script se describen todas las instrucciones necesarias para la animación de desplazamiento lateral del objeto.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Animation : MonoBehaviour
{
    public Transform Start;
    public Transform End;
    public float speedMove = 1f;
    private float Track = 0f;

    Vector3 result = new Vector3(0.0f, 0.0f, 0.0f);
    public Vector3 range = new Vector3(0.0f, 5.0f, 0.0f);

    public float speed = 90.0f;
    float value = 0.0f;
    float factor = 1.0f;

    public float second = 22f;

    // Update is called once per frame
    void Update()
    {
        if (Time.time > second)
        {
            Track += speedMove * Time.deltaTime;
            transform.position = Vector3.Lerp(Start.position, End.position, Track);
        }

        if (Track >= 1)
        {
            value = value + Time.deltaTime * speed * factor;
            Vector3 result = Vector3.Lerp(-range, range, value);
            gameObject.transform.localEulerAngles = result;

            if (value >= 1.0f)
                factor = -1.0f;

            if (value <= 0.0f)
                factor = 1.0f;
        }
    }
}
```

Figura 55. Script Animation

La segunda animación modifica cada cierto tiempo los valores de los ejes del objeto de manera que aumenten y disminuyan en bucle. Para cada objeto, se puede seleccionar el tiempo de modificación de los valores y que eje modificar. El resultado es una animación sencilla y muy visual.

La figura 56 muestra el resultado de la implementación del elemento 3D en la escena de Santa catalina.

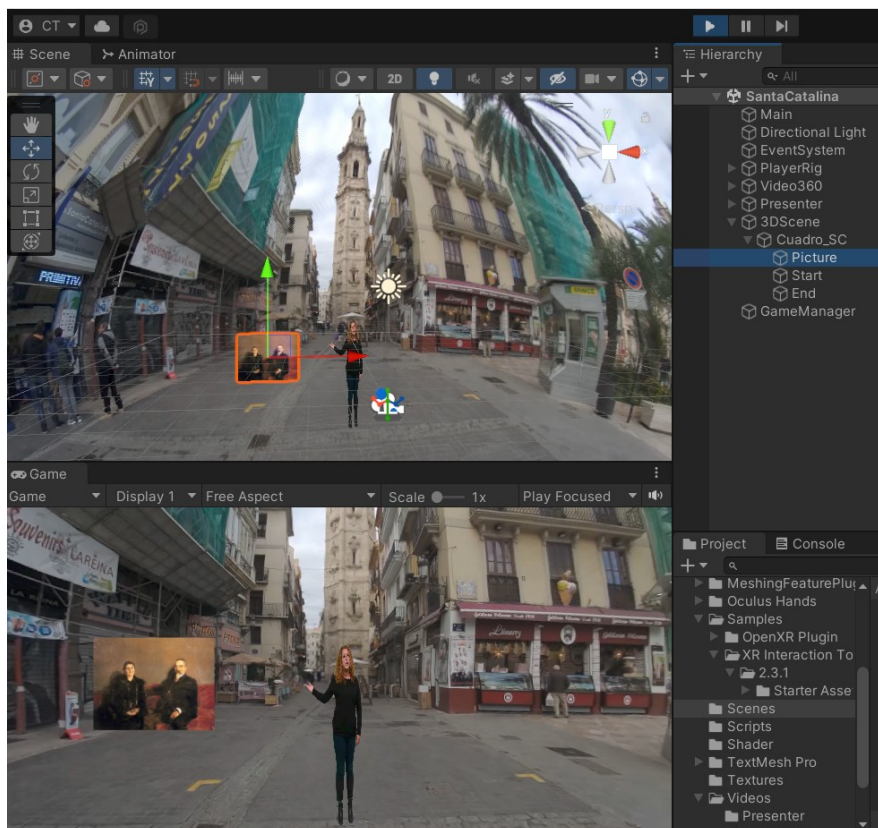


Figura 56. Integración Objeto 3D animado en la escena

4.8 Creación de las diferentes escenas para el visualizador

Las anteriores secciones (4.2, 4.4, 4.5, 4.6 y 4.7) detallan el procedimiento a seguir para la realización de una escena turística completa, formada por un vídeo de fondo, una explicación de un guía y un elemento 3D animado.

Este proyecto consiste en la creación de un visualizador estereoscópico de escenas turísticas con Unity. El visualizador desarrollado cuenta con cuatro escenas turísticas, dos enmarcadas dentro de lo que se conoce como la Ruta Sorolla de Valencia y dos representando lugares importantes de la vida de Blasco Ibáñez (Figura 57). Para la creación del resto de escenas, se ha repetido el procedimiento descrito anteriormente tantas veces como escenas tiene el proyecto.

El resultado obtenido es el siguiente:

- Escena 1 – Santa Catalina: Formada por el vídeo de fondo de la calle Santa Catalina, el vídeo de la explicación de la guía turística sobre la iglesia y como objeto 3D se muestra el cuadro de los padres de Sorolla localizado en el interior de la iglesia Santa Catalina.
- Escena 2 – Plaza Redonda: Formada por el vídeo de fondo grabado en la plaza redonda, la explicación de la guía turística y la fuente modelada en 3D.
- Escena 3 – San Martín: Formada por el vídeo de fondo de la fachada de San Martín, el vídeo de la explicación de la guía turística sobre la iglesia y como objeto 3D se muestra el cuadro de Clotilde, la mujer de Sorolla.
- Escena 4 – Casa Museo: Formada por el vídeo de fondo, la explicación de la guía turística y un barco representando la vida de la albufera.



Figura 57. Escenas desarrolladas

4.9 Desarrollo interfaz de selección de escenas

El último paso para completar el desarrollo del visualizador consiste en la creación de una interfaz de selección de escenas. Esta interfaz ha sido desarrollada con el elemento Canvas de Unity.

El elemento Canvas de Unity representa un área rectangular donde se colocan y muestran todos los elementos UI del proyecto. Además, los elementos Canvas son visibles en el modo de edición de escena de Unity, por lo que es relativamente sencillo colocar los elementos en su posición y tamaño adecuados sin necesidad de visualizar la vista Game.

Hay dos formas de crear un elemento Canvas en una escena de Unity: Desde el menú GameObject – Canvas o crear un elemento UI cualquiera ya que se creará automáticamente como objeto padre. Dentro de los elementos Canvas se pueden colocar diferentes objetos como imágenes, texto, botones, paneles, etc (Figura 58) [40].

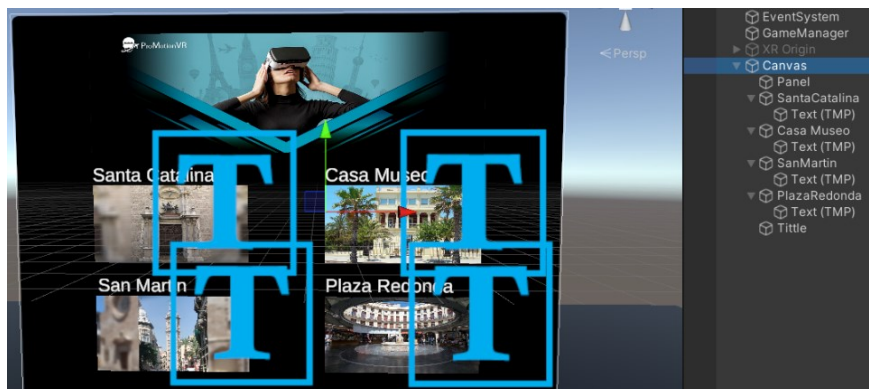


Figura 58. Creación Elemento Canvas

Este proyecto utiliza un elemento de tipo *Image* en el cual se presenta el nombre y título del proyecto y cuatro elementos de tipo *Button* a través de los cuales se accederá a las diferentes experiencias turísticas (Figura 59).



Figura 59. Interfaz del Visualizador de escenas turísticas

A continuación, se explica la creación del sistema de interacción de los mandos y el acceso a las diferentes escenas acompañado de imágenes demostrativas.

4.9.1 Implementación del sistema de seguimiento e interacción de los controladores

Para el sistema de interacción se ha utilizado el plug-in XR Interaction Toolkit ya que presenta una serie de características y opciones predeterminadas que simplifican el proceso de interacción.

En primer lugar, se ha creado un plano de color negro que simula un suelo para el usuario. A continuación, es necesario realizar algunos ajustes desde la pestaña *File – Build Settings – Player Settings*. Si se desea trabajar con varios dispositivos VR es recomendable marcar *Open XR* tras instalar el plug-in, sin embargo, también es compatible marcar la casilla de *Oculus*.

El siguiente paso es seleccionar el *Interaction Profile*. Esto permite añadir con qué mandos se va a realizar la interacción en el proyecto. En el caso de este trabajo, al trabajar con las gafas y los controladores Meta Quest 2, hay que seleccionar *Oculus Touch Controller Profile* para modo *Windows* y *Android* (Figura 60) [41].

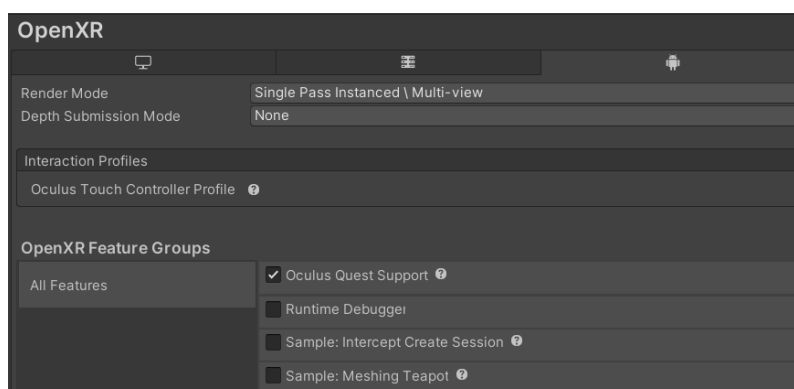


Figura 60. Interaction Profile

A continuación, se importa la estructura XR Origin que ofrece el plug-in. Esta estructura presenta la jerarquía necesaria y sus componentes para que la cámara se convierta a una cámara de realidad virtual.

Esta jerarquía presenta los elementos necesarios referentes a la cámara, por tanto, es necesario añadir los elementos para la interacción con los mandos. Para ello, se crean dos objetos vacíos, de nombre LeftHandController y RightHandController, a los cuales se les asigna el script “XR Controller (Action-based)” (Figura 61). Este script permite seleccionar cada una de las interacciones (rotar, seleccionar, mover, pulsar) que se necesiten para el controlador. Aunque parece un paso muy costoso, el plug-in XR Interaction Toolkit presenta unas acciones predeterminadas por defecto para asignar a los controladores [41].

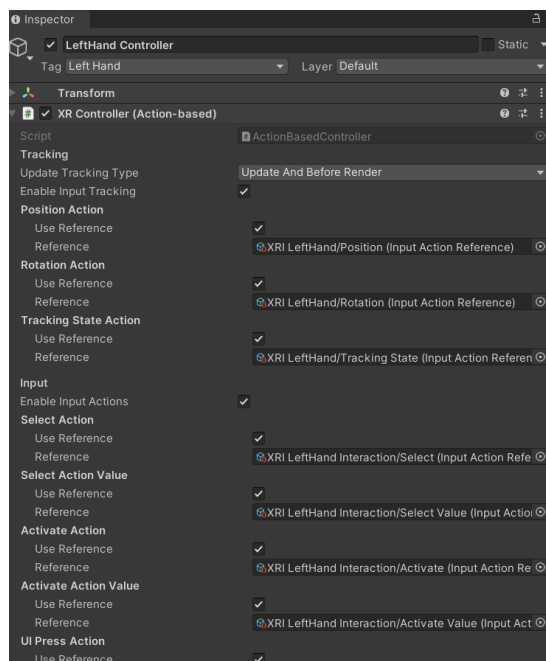


Figura 61. Ejemplo LeftHand Controller

Por último, pero no menos importante, es necesario añadir el script “Input Action Manager” y las acciones “XRI Default Input Actions” al elemento XR Origin para permitir que los controladores sigan el movimiento de la cámara (Figura 62) [41].

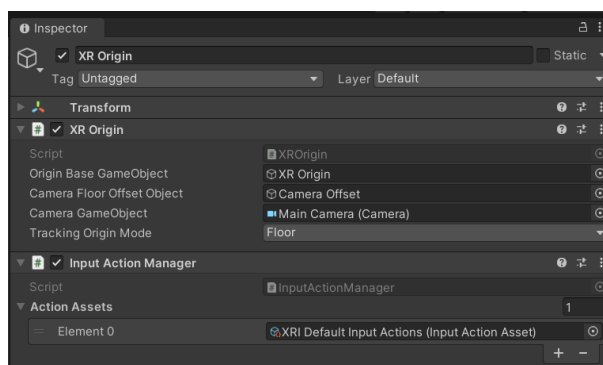


Figura 62. Elemento XR Origin

Tras realizar estos pasos, se obtiene un sistema de interacción sencillo con una cámara VR y dos controladores. Para que el sistema sea más atractivo y visual para los usuarios, se han añadido dos modelos de manos que ofrece el paquete *Oculus Hand Package* (Figura 63). Cada una de las manos representa la posición y rotación del controlador del usuario. Además, se han establecido unas animaciones sencillas al pulsar los botones para que sea más fácil para el usuario detectar si ha pulsado correctamente el botón.

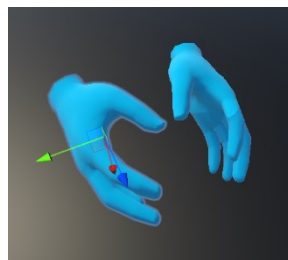


Figura 63. Modelos de manos virtuales

Por último, se ha utilizado el elemento Ray Tracing del plug-in (Figura 64). Este elemento permite asignar un rayo láser a cada uno de los controladores. En este proyecto, se ha activado el rayo láser para ayudar a los usuarios a apuntar a la imagen de la escena que quieren seleccionar. Estos rayos sólo aparecen en el rango de enfoque del elemento Canvas, de manera que no distraigan a los usuarios. Para ello, se ha modificado la propiedad Invalid Color Gradient del elemento XR Interaction Visual a color transparente [41].

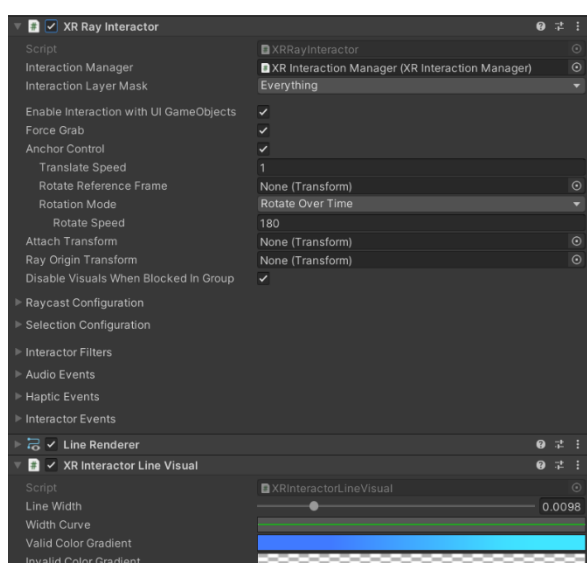


Figura 64. Elemento Ray Tracing

En resumen, se ha creado un sistema de interacción completo que permite a los usuarios visualizar las diferentes escenas turísticas al apuntar a ellas y pulsar el botón Trigger del controlador. La estructura final de este sistema de interacción se muestra en la Figura 65.

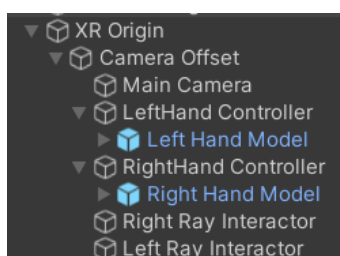


Figura 65. Estructura Sistema de Interacción

Para que el usuario se transporte a la escena correspondiente, se ha establecido un script GameManager. Este script presenta todas las funciones necesarias para el control de las escenas, el cual se ha conseguido con la instrucción de Unity “Scene.Load” la cual permite cargar la escena de la experiencia correspondiente [42].

Tras acabar de visualizar la escena turística, el visualizador vuelve a mostrar de nuevo la interfaz de Menú para que los usuarios puedan escoger una nueva escena. Este es el último paso para completar el desarrollo de un visualizador estereoscópico de escenas turísticas para cascos de RV.



Capítulo 5. Resultados obtenidos

A lo largo de este proyecto de fin de grado se llevan a cabo una serie de tareas para el desarrollo completo de un visualizador de escenas turísticas estereoscópicas. El resultado principal obtenido es la creación de una aplicación prototipo a través de la herramienta Unity que permite la visualización de contenidos estereoscópicos de manera autónoma a través de los cascos de realidad virtual Meta Quest 2. Gracias a este desarrollo, se demuestra la potencia y la capacidad de visualización que presenta este casco de RV y, además, se ofrece una nueva aplicación de esta tecnología innovadora, como es la realidad virtual, aplicada en el sector del turismo.

Además, a lo largo del desarrollo, se han realizado pruebas y análisis que han permitido obtener otro tipo de resultados. En primer lugar, se ha obtenido un análisis completo sobre el comportamiento del eje óptico de la cámara panorámica estereoscópica Z CAM K1 Pro. Este análisis permite que los próximos desarrollos que hagan uso de esta cámara puedan basarse en él para ajustar las imágenes o vídeos al comportamiento lineal que presenta la cámara.

Por otra parte, se ha desarrollado la aplicación para que funcione de manera autónoma en las gafas Meta Quest 2. En el Anexo I se presenta un manual de usuario sobre la aplicación, en el que se detalla cómo generarla a partir del proyecto desarrollado con Unity hasta cómo instalarla y ejecutarla desde las propias gafas de realidad virtual Meta Quest 2.

La aplicación desarrollada presenta una estructura innovadora para representar vídeos estereoscópicos con la herramienta de Unity. Así mismo, permite la inclusión de elementos tridimensionales animados en la escena para crear experiencias lo más enriquecedoras posibles para los usuarios.

Por último, gracias al desarrollo de este proyecto se ha podido explorar un nuevo método para extraer fondos verdes de los vídeos y representarlos dentro de un proyecto de Unity. A pesar de ser un método más laborioso que extraer la silueta en tiempo real al necesitar preprocesado de los vídeos, se ha demostrado que el resultado obtenido es de muy buena calidad y no requiere de un gran gasto de potencia por lo que las gafas Meta Quest 2 pueden visualizarlo de forma autónoma.

Capítulo 6. Conclusiones y propuesta de trabajo futuro

A lo largo de este documento se describe el desarrollo completo de un visualizador prototipo de escenas turísticas estereoscópicas. Se presenta la metodología seguida para su desarrollo, así como los objetivos planteados y los materiales utilizados.

Más adelante, se detallan cada de uno de los pasos llevados a cabo para el desarrollo completo de la aplicación. Para cada paso, se explican los elementos utilizados y el resultado final obtenido.

El resultado obtenido es una solución prototipo de un visualizador de escenas turísticas para cascos de realidad virtual autónomos. Este desarrollo ha permitido explorar diferentes herramientas para escoger la solución óptima para representar los vídeos estereoscópicos a través de los cascos de realidad virtual Meta Quest 2. Además, ha permitido detectar algunos comportamientos irregulares de las herramientas y materiales utilizados para futuros desarrollos.

La cámara panorámica estereoscópica Z CAM K1 Pro presenta dos lentes y ambas graban los contenidos con un campo de visión de 180°. Esto provoca que parte del objetivo de una lente se cuele en la grabación de la otra y viceversa. No es un efecto muy problemático, sin embargo, sí que afecta levemente a la experiencia de usuario al visualizar las escenas turísticas. Por ello, se plantea el desarrollo de un sistema que simule el comportamiento de los párpados humanos para esconder la parte de la grabación dónde se visualiza la lente.

El segundo comportamiento irregular viene provocado por las tapas de las lentes de la cámara. Este comportamiento se produce al introducir y extraer las tapas de las lentes y produce que la posición de las lentes se desajuste por muy pocos píxeles prácticamente imperceptibles al ojo humano pero perceptibles al crear la imagen estereoscópica y reproducirla. Este efecto ha sido localizado tras desarrollar la escena turística de Casa Museo, ya que al visualizar el vídeo grabado con los cascos de realidad virtual se ve borroso. Por ese motivo, se ha realizado un análisis sobre la imagen obtenida comparándola con otras imágenes capturadas de la misma cámara, llegando a la conclusión de que las lentes se desplazan varios píxeles ligeramente al colocar las tapas. Para desarrollos próximos con esta cámara se recomienda quitar una vez las tapas, realizar todas las grabaciones y cuando se termine volver a colocar las tapas para que todos los resultados tengan la misma referencia.

Cabe mencionar que la solución desarrollada es un prototipo cuyo objetivo es demostrar la funcionalidad del visualizador para cascos autónomos. Por ese motivo, no presenta una gran variedad de escenas turísticas. Como trabajo futuro se explorará la opción de ampliar la interfaz de contenidos y desarrollar nuevas escenas turísticas de diferentes puntos de interés de la ciudad de Valencia como el Museo de las Artes y las Ciencias o la Plaza del Ayuntamiento.

Otra de las opciones que se desea explorar en futuros desarrollos es la implementación de un servidor de contenidos. Este servidor almacenará todos los recursos de la aplicación, tanto los vídeos de las escenas turísticas como los vídeos de los guías turísticos, y la aplicación descargará los vídeos de la aplicación de manera automática.

En conclusión, este trabajo de final de grado ha sido una experiencia de aprendizaje muy prometedora, la cual ha cumplido todos los objetivos propuestos y ha demostrado potencial para su uso futuro.

Capítulo 7. Bibliografía

- [1] Brainstorm Multimedia, “ProMotionVR”, <https://promotionvr.brainstorm3d.com/> [Online].
- [2] Naciones Unidas, “Objetivos de Desarrollo Sostenible. Objetivo 9: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación”, <https://www.un.org/sustainabledevelopment/es/infrastructure/> [Online].
- [3] Naciones Unidas, “Objetivos de Desarrollo Sostenible. Objetivo 12: Garantizar modalidades de consumo y producción sostenibles”, <https://www.un.org/sustainabledevelopment/es/sustainable-consumption-production/> [Online].
- [4] Franco Castro, Proyecto IDIS, “Estereoscopia”, <https://proyectoidis.org/estereoscopia/> [Online].
- [5] Museo Virtual Filmoteca Unam, “Visión Estereoscópica”, <https://museovirtual.filmoteca.unam.mx/temas-cine/vision-estereoscopica/> [Online].
- [6] Wilson J. Sarmiento, Research Gate, “Esquema de Generación y Percepción de una Imagen Estereoscópica”, https://www.researchgate.net/figure/ESQUEMA-DE-GENERACION-Y-PERCEPCION-DE-UNA-IMAGEN-ESTEREOSCOPICA_fig3_287411849 [Online].
- [7] 120Lomo, “La fotografía estereoscópica. 3D desde el siglo XIX”, Junio 10, 2019., <https://www.120lomo.com/historia-fotografia/la-fotografia-estereoscopica-3d-desde-el-siglo-xix/>
- [8] Phototools, “Insta360 Pro 2”, <https://phototoolsweb.com/inicio/2903-insta360-pro-2.html> [Online].
- [9] Artes Visuales Animara, “Fotografía Estereoscópica”, <http://artesvisualessemimara.blogspot.com/p/fotografia-estereoscopica.html> [Online].
- [10] Yossy Mendelovich, YmCinema, “The Largest Production Ever Filmed in Space Shot on Z CAM V1 Pro”, Enero 19, 2021., https://ymcinema.com/2021/01/19/the-largest-production-ever-filmed-in-space-shot-on-z-cam-v1-pro/#google_vignette?utm_content=cmp-true [Online].
- [11] Amazon, “Nikon Keymission 360 Action Camera”, <https://www.amazon.sa/-/en/Nikon-Keymission-360-Action-Camera/dp/B07XYGFHKY> [Online].
- [12] Enrique Pérez, Xataka, “Cómo funcionan las gafas de realidad virtual por dentro”, Julio 20, 2020., <https://www.xataka.com/realidad-virtual-aumentada/como-funcionan-gafas-realidad-virtual-dentro> [Online].
- [13] Monsuton, “Gafas de realidad virtual ¿qué son y cuáles son las mejores?”, <https://www.monsuton.com/gafas-realidad-virtual/> [Online].
- [14] Hologram 3D Display, “Transparent Holographic Showcase Holographic Projector”, <https://www.hologram3ddisplay.com/sale-8592176-transparent-3d-holographic-showcase-holographic-projector-50hz-60hz.html> [Online].
- [15] Jaume Ferrer, Universidad Oberta de Catalunya, “HMD (head-mounted display)”, <http://multimedia.uoc.edu/blogs/rx/es/2018/03/11/hmd-head-up-display/> [Online].
- [16] Promethean World, “La realidad virtual en el aula”, <https://www.prometheanworld.com/es/recursos/blogs/la-realidad-virtual-en-el-aula-novedad/> [Online].
- [17] Meta, “Especificaciones Técnicas Meta Quest 2”, <https://www.meta.com/es/quest/products/quest-2/tech-specs/#tech-specs> [Online].

- [18] Nfortec, “Análisis Oculus Quest 2: unas gafas de realidad virtual para todo el mundo”, <https://www.nfortec.com/blog/conoce-mas-a/analisis-oculus-quest-2-unas-gafas-de-realidad-virtual-para-todo-el-mundo> [Online].
- [19] Dani Candil, Vida Extra, “Unity, el motor de desarrollo capaz de partir la historia de los videojuegos en dos”, Febrero 21, 2014., <https://www.vidaextra.com/industria/unity-el-motor-de-desarrollo-capaz-de-partir-la-historia-de-los-videojuegos-en-dos> [Online].
- [20] Máster Móviles UA, “El motor Unity”, <https://mastermoviles.gitbook.io/graficos-y-multimedia/el-motor-unity> [Online].
- [21] Khronos, “OpenXR”, <https://www.khronos.org/openxr/> [Online].
- [22] Unity Documentation, “OpenXR Plugin”, <https://docs.unity3d.com/es/2021.1/Manual/com.unity.xr.openxr.html> [Online].
- [23] Quasar Dynamics, “OpenXR: Unificando la Realidad Extendida”, <https://quasardynamics.com/openxr-unificando-la-realidad/> [Online].
- [24] Unity Documentation, “XR Interaction Toolkit” Plugin”, <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.3/manual/index.html> [Online].
- [25] Adobe, “Adobe Premiere Pro”, <https://www.adobe.com/es/products/premiere.html> [Online].
- [26] Panorama Audiovisual, “Avid abre nuevos horizontes al abrir la colaboración con Adobe Premiere Pro”, <https://www.panoramaaudiovisual.com/2020/09/22/avid-mediacentralpanel-aodbe-premiere/> [Online].
- [27] Adobe, “Adobe Photoshop”, <https://www.adobe.com/es/products/photoshop.html> [Online].
- [28] Molly Bohannon, Forbes, “Adobe anunció Firefly, una nueva herramienta de Inteligencia Artificial que se integrará a Photoshop y está dando que hablar”, Mayo 23, 23., <https://www.forbesargentina.com/innovacion/adobe-anuncio-firefly-una-nueva-herramienta-inteligencia-artificial-integrara-photoshop-esta-dando-hablar-n34264> [Online].
- [29] Meta, “Cómo instalar la aplicación de Oculus en el ordenador”, <https://www.meta.com/es-es/help/quest/articles/getting-started/getting-started-with-rift-s/install-oculus-pc-app/> [Online].
- [30] SideQuest, “SideQuest”, <https://sidequestvr.com/> [Online].
- [31] Unity Documentation, “Video Player component”, <https://docs.unity.cn/es/2020.1/Manual/class-VideoPlayer.html> [Online].
- [32] Unity Documentation, “Render Texture”, <https://docs.unity3d.com/Manual/class-RenderTexture.html> [Online].
- [33] Unity Documentation, “Occlusion Culling”, [https://docs.unity3d.com/560/Documentation/Manual/OcclusionCulling.html#:~:text=Occlusion%20Culling%20is%20a%20feature,\(occluded\)%20by%20other%20objects](https://docs.unity3d.com/560/Documentation/Manual/OcclusionCulling.html#:~:text=Occlusion%20Culling%20is%20a%20feature,(occluded)%20by%20other%20objects) [Online].
- [34] CG Aura, Youtube, “Two-Sided Shader in Unity”, <https://www.youtube.com/watch?v=kbBDZYi2IMA> [Online].
- [35] Kate Brush, Techtargget, “Virtual reality sickness (VR motion sickness)”, <https://www.techtargget.com/iotagenda/definicion/virtual-reality-sickness-VR-motion-sickness> [Online].
- [36] Bruno Capuano, Campus MVP, “Cómo funcionan las transparencias en las imágenes: el canal “Alpha”, <https://www.campusmvp.es/recursos/post/Como-funcionan-las-transparencias-en-las-imagenes-el-canal-Alpha.aspx> [Online].
- [37] Fnord, “WebM Open Movie Plug-In for Premiere”, <https://www.fnordware.com/WebM/> [Online].



- [38] Unity Documentation, “Video transparency support”, <https://docs.unity3d.com/Manual/VideoTransparency.html> [Online].
- [39] Unity Documentation, “Time.time”, <https://docs.unity3d.com/ScriptReference/Time-time.html> [Online].
- [40] Unity Documentation, “Canvas”, <https://docs.unity.cn/es/2020.1/Manual/UIColor.html> [Online].
- [41] Valem Tutorials, Youtube, “How to make a VR Game in Unity”, <https://www.youtube.com/watch?v=HhtTtvBF5bI&t=750s> [Online].
- [42] Unity Documentation, “SceneManager.LoadScene”, <https://docs.unity3d.com/ScriptReference/SceneManager.LoadScene.html> [Online].
- [43] SideQuest VR, Youtube, “How to Use SideQuest – PC Advanced Installer Tutorial”, <https://www.youtube.com/watch?v=wNku19H19Dw&t=229s> [Online].

Capítulo 8. ANEXOS

8.1 ANEXO I: Manual de usuario de la aplicación

Este anexo detalla el manual de usuario completo de la aplicación PromotionVR. Se describen todos los pasos necesarios a seguir para poder generar la aplicación, instalarla en las gafas Oculus Quest 2 y visualizarla a través de ellas. Para realizar estos pasos, es necesario el uso de aplicaciones externas, las cuáles también están indicadas a lo largo del manual.

8.1.1 Generación del archivo APK que contendrá la aplicación final

Tras realizar el desarrollo completo de la aplicación en la herramienta Unity, es necesario generar un archivo .apk que servirá para ejecutar la aplicación de manera autónoma sobre los cascos de RV, ya que los dispositivos de la empresa Oculus trabajan sobre el sistema operativo Android.

Los archivos .apk (*Android Package Kit*) son los formatos de archivo utilizados por el sistema operativo Android para distribuir e instalar aplicaciones. Estos archivos contienen todos los elementos necesarios para instalar y ejecutar una aplicación, desde el código fuente y los recursos (imágenes, archivos de sonido, videos, etc.) hasta las bibliotecas y el manifiesto de aplicación.

Los archivos .apk son creados al compilar el código fuente de una aplicación y empaquetar todos los archivos necesarios en un solo archivo. Para realizar la compilación del código de nuestra aplicación en Unity, hay que seguir una serie de pasos:

- Abrir la pestaña *Build Settings* desde el apartado *File – Build Settings*.
- Una vez abierta, comprobar que el proyecto está desarrollado en la plataforma Android. Si no es así, seleccionar la pestaña correspondiente y pulsar el botón *Switch Platform* localizado en la esquina inferior derecha.
- Como nuestro proyecto está compuesto por cinco escenas, hay que comprobar que todas aparecen en la pestaña *Scenes In Build*. En el caso de que alguna no apareciera, se puede añadir desde el botón *Add Open Scenes*.
- Finalmente, asegurarse de que todos los apartados son correctos y darle al botón *Build* que abrirá una pestaña para seleccionar la localización en la que se guardará el archivo APK en el ordenador.

Tras realizar todos los pasos mencionados, la pestaña *Buid Settings* presenta la estructura mostrada en la Figura 66.

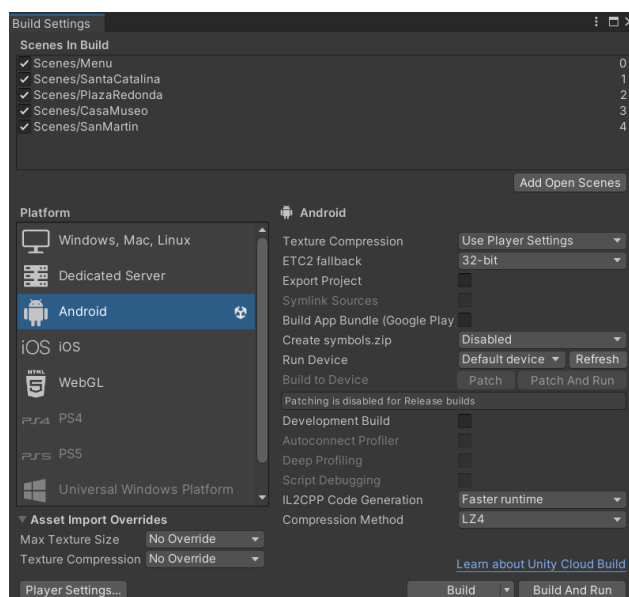


Figura 66. Generar el fichero .apk

8.1.2 Instalación de la aplicación Oculus

Este proyecto está desarrollado para ser ejecutado de manera autónoma a través de los cascos de realidad virtual Meta Quest 2.

A pesar de realizar una aplicación que funcione de manera autónoma, es necesario realizar la conexión de los cascos al ordenador para instalar el archivo APK dentro del sistema. Para comprobar que los cascos están correctamente conectados, la empresa Oculus proporciona una aplicación de ordenador a través de la cual se pueden controlar los dispositivos conectados.

La aplicación, de nombre Oculus, se utiliza para configurar y conectar dispositivos de Oculus VR, como el Meta Quest 2 a una red Wi-Fi y a una cuenta de Oculus. Además, permite descargar y actualizar aplicaciones y juegos de la tienda de Oculus, ajustar la configuración del dispositivo, acceder a tutoriales y recibir notificaciones importantes.

Para descargar e instalar la aplicación de Oculus en el ordenador, hay que seguir una serie de pasos:

1. Ve a la página de descarga de la aplicación:
https://www.meta.com/es/quest/setup/?utm_source=www.meta.com&utm_medium=ollyredirect
2. Encuentra el sistema con el que se vaya a trabajar (Meta Quest 2) y haz clic en Descargar software (Figura 67).



Figura 67. Instalación aplicación Oculus

3. La aplicación Oculus comenzará a descargarse en el ordenador.
4. Abre el programa de instalación de Oculus en la carpeta de descargas del ordenador y haz clic en Empezar.
5. Lee y acepta los términos y condiciones de uso para continuar.
6. Selecciona la unidad donde quieras instalar la aplicación de Oculus haciendo clic en el desplegable debajo de Ubicación.
7. Haz clic en Instalar.
8. Cuando la instalación esté completa, inicia sesión o crea una cuenta de Meta.

Tras seguir estos pasos, ya se ha finalizado la instalación de la aplicación Oculus. A continuación, es necesario activar el apartado de Orígenes desconocidos de la pestaña “Configuración – General” para permitir ejecutar en las gafas aplicaciones no revisadas por Oculus (Figura 68).

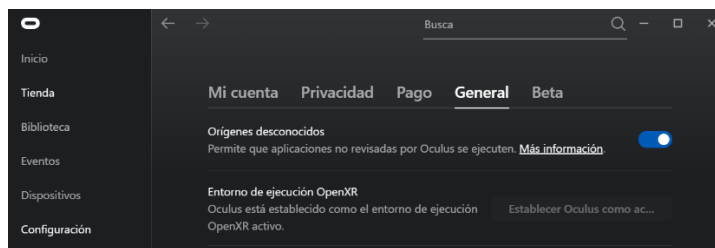


Figura 68. Orígenes Desconocidos

Para conectar las gafas de realidad virtual, es necesaria la utilización de un “Cable Link”. Un cable Link es un cable USB-C a USB-3.0 que permite conectar los dispositivos Oculus Quest y Meta Quest 2 a un ordenador para obtener experiencias de juegos más inmersivas y con mejor rendimiento que la conexión inalámbrica, ya que permite una transferencia de datos más rápida y estable.

Tras conectar las gafas al ordenador a través del cable Link, se puede comprobar su correcto funcionamiento desde el apartado “Dispositivos” de la aplicación Oculus como muestra la figura 69.

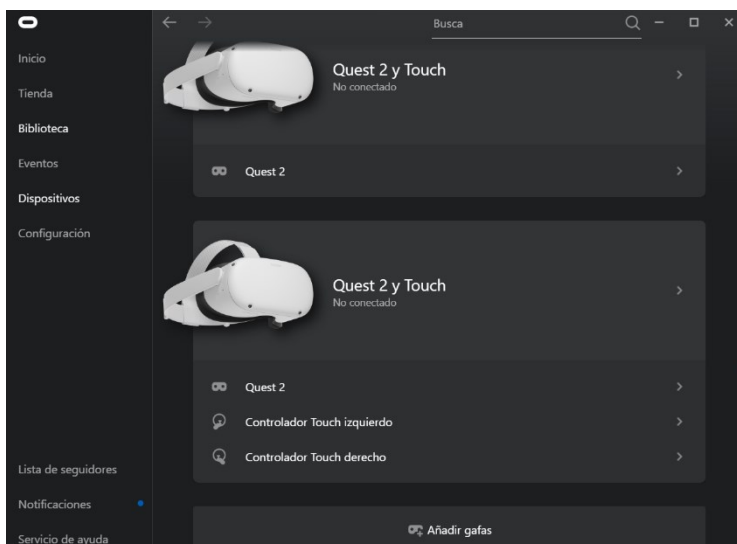


Figura 69. Conexión de las gafas RV al PC

8.1.3 Instalación de la aplicación SideQuest

La aplicación SideQuest es una plataforma de distribución que permite compartir software y aplicaciones entre los usuarios de los dispositivos de realidad virtual. En este proyecto, se ha utilizado para instalar un archivo .apk dentro del sistema de las gafas Meta Quest 2.

A continuación, se describen los pasos para instalar esta aplicación en el ordenador:

1. Accede a la página de descarga de la aplicación a través del siguiente enlace: <https://sidequestvr.com/setup-howto>
2. Para trabajar con SideQuest es necesario tener una cuenta. Se puede crear desde el siguiente enlace: <https://sidequestvr.com/sign-up>
3. Se presentan dos instaladores: Easy Installer y Advanced Installer. Este proyecto requiere el uso de Advanced Installer para la gestión de archivos y la instalación de apk [43].

8.1.4 Subida de un fichero .apk a los cascos de RV a través de SideQuest

El primer paso que hay que realizar es abrir la aplicación SideQuest en el ordenador. A continuación, es necesario conectar los cascos de RV al ordenador. Para ello, se utiliza un cable de conexión de tipo USB-C.

Si la conexión es correcta, la aplicación SideQuest muestra en la esquina superior izquierda un círculo verde. En caso de que la conexión no sea correcta o no se hayan dado los permisos necesarios a los cascos, la aplicación muestra el círculo de color rojo o naranja, respectivamente (Figura 70).

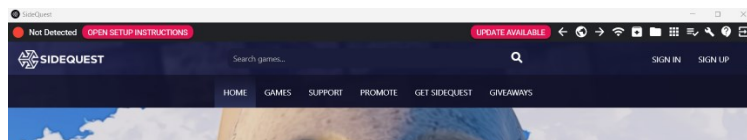


Figura 70. Interfaz de SideQuest

El siguiente paso es clicar en el icono de la esquina superior derecha para importar el fichero .apk. Al abrir el fichero, la aplicación comenzará la descarga del fichero en las gafas. Al terminar, se muestra un mensaje y el proceso queda finalizado.

8.1.5 Ejecución de la aplicación desde las gafas de RV

Para ejecutar la aplicación instalada en las gafas, hay que seleccionar el apartado Aplicaciones desde el menú que muestra la interfaz de las gafas Meta Quest 2. Este apartado muestra todas las aplicaciones que hay instaladas dentro del casco de RV.

Sin embargo, las gafas Meta Quest 2 presentan un apartado exclusivo de nombre “Orígenes Desconocidos” para aquellas aplicaciones que no son oficiales de Oculus, como es el caso de la aplicación desarrollada.

Si la instalación y la subida del fichero .apk se ha realizado de forma correcta, la aplicación debe aparecer en la lista de aplicaciones con el nombre que se haya escogido. Al clicar en ella, las gafas comienzan la ejecución del programa de manera automática.