



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 una Aplicación de Realidad Mixta para la
Estimulación Cognitiva Infantil

Trabajo Fin de Grado

Grado en Tecnología Digital y Multimedia

AUTOR/A: Fat, Denisa Marcela

Tutor/a: Rey Solaz, Beatriz

CURSO ACADÉMICO: 2023/2024

Resumen

El objetivo principal de este proyecto es proporcionar actividades interactivas y envolventes que ayuden a mejorar las habilidades cognitivas de los niños con necesidades especiales, ofreciendo una herramienta innovadora para su desarrollo y bienestar continuo. La aplicación de realidad mixta se desarrollará utilizando la plataforma Unity y estará optimizada para su uso con las gafas Meta Quest 3 exclusivamente con la tecnología de Hand Tracking. A los niños les resultará más sencillo interactuar utilizando sus manos en lugar de utilizar controles físicos adicionales. Los datos generados durante las sesiones se almacenarán para permitir una evaluación posterior por parte de los profesionales, lo que facilitará un seguimiento personalizado del progreso del niño y la adaptación continua de las actividades.

Esta aplicación está pensada para poder ser utilizada tanto en entornos clínicos bajo la supervisión de profesionales, como en la comodidad del hogar donde los padres pueden participar en las actividades de refuerzo. La realidad mixta ofrece un entorno seguro y familiar para los niños, ayudando a reducir la sensación de aislamiento y ansiedad que los niños puedan experimentar.

Resum

L'objectiu principal d'aquest projecte és proporcionar activitats interactives i envolupants que ajuden a millorar les habilitats cognitives dels xiquets amb necessitats especials, oferint una ferramenta innovadora per al seu desenvolupament i benestar continu. L'aplicació de realitat mixta es desenvoluparà utilitzant la plataforma *Unity i estarà optimitzada per al seu ús amb les ulleres Meta *Quest 3 exclusivament amb la tecnologia de *hand *tracking. Als xiquets els resultarà més senzill interactuar utilitzant les seues mans en lloc d'utilitzar controls físics addicionals. Les dades generades durant les sessions s'emmagatzemaran per a permetre una avaluació posterior per part dels professionals, la qual cosa facilitarà un seguiment personalitzat del progrés del xiquet i l'adaptació contínua de les activitats.

Esta aplicació està pensada per a poder ser utilitzada tant en entorns clínic sota la supervisió de professionals, com en la comoditat del llar on els pares poden participar en les activitats de reforç. La realitat mixta oferix un entorn segur i familiar per als xiquets, ajudant a reduir la sensació d'aïllament i ansietat que els xiquets puguen experimentar.



Abstract

The main objective of this project is to provide interactive and immersive activities that help improve the cognitive skills of children with special needs, offering an innovative tool for their continued development and well-being. The mixed reality app will be developed using the Unity platform and will be optimized for Meta Quest 3 headsets exclusively with hand tracking technology. Children will find it easier to interact using their hands rather than using additional physical controls. The data generated during the sessions will be stored to allow for subsequent evaluation by professionals, facilitating personalized monitoring of the child's progress and continuous adaptation of activities.

This application is designed to be used both in clinical settings under the supervision of professionals, and in the comfort of home where parents can participate in reinforcement activities. Mixed reality offers a safe and familiar environment for children, helping to reduce the sense of isolation and anxiety that children may experience.

RESUMEN EJECUTIVO

La memoria del TFG del Grado en Tecnología Digital y Multimedia debe desarrollar en el texto los siguientes conceptos, debidamente justificados y discutidos, centrados en el ámbito de la tecnologías digitales y multimedia

CONCEPT (ABET)	CONCEPTO (traducción)	¿Cumple? (S/N)	¿Dónde? (páginas)
1. IDENTIFY:	1. IDENTIFICAR:		
1.1. Problem statement and opportunity	1.1. Planteamiento del problema y oportunidad	S	5-6
1.2. Constraints (standards, codes, needs, requirements & specifications)	1.2. Toma en consideración de los condicionantes (normas técnicas y regulación, necesidades, requisitos y especificaciones)	S	11-14
1.3. Setting of goals	1.3. Establecimiento de objetivos	S	10
2. FORMULATE:	2. FORMULAR:		
2.1. Creative solution generation (analysis)	2.1. Generación de soluciones creativas (análisis)	S	15-20
2.2. Evaluation of multiple solutions and decision-making (synthesis)	2.2. Evaluación de múltiples soluciones y toma de decisiones (síntesis)	S	21-30
3. SOLVE:	3. RESOLVER:		
3.1. Fulfilment of goals	3.1. Evaluación del cumplimiento de objetivos	S	35
3.2. Overall impact and significance (contributions and practical recommendations)	3.2. Evaluación del impacto global y alcance (contribuciones y recomendaciones prácticas)	S	36



Índice

Capítulo 1.	Introducción	5
1.1	Contexto y Justificación	5
1.2	Realidad Aumentada y Mixta.....	6
1.3	Dispositivos.....	7
1.3.1	Meta Quest 2	8
Capítulo 2.	Objetivos	10
Capítulo 3.	Metodología	11
3.1	Hardware	11
3.2	Software	11
3.2.1	Blender	11
3.2.2	Canva.....	12
3.2.3	Meta Quest Link.....	13
3.2.4	Unity.....	14
Capítulo 4.	Desarrollo	15
4.1	Configuración previa.....	15
4.1.1	Descarga Unity y entorno RM.....	15
4.1.2	Paquetes necesarios de la Asset Store	16
4.1.3	Importación desde Blender.....	18
4.2	Escenas y su flujo.....	19
4.2.1	Organización de escenas	20
4.3	Elementos presentes en las escenas.....	21
4.3.1	Paneles.....	21
4.3.2	Cubos y tableros	24
4.3.3	Confeti.....	25
4.3.4	Plano transparente	27
4.4	Lógica del proyecto.....	27
4.4.1	Script: SnapToLocation.....	28
4.4.2	Script: SnapObject.....	29
4.4.3	Script: AllSnapped	30
4.4.4	Generación apk.....	31
4.5	Configuración por parte del profesional.....	32
4.5.1	Feedback obtenido.....	32
Capítulo 5.	Resultados obtenidos.....	35



Capítulo 6.	Conclusiones y propuesta de trabajo futuro	36
Capítulo 7.	Bibliografía.....	37
Capítulo 8.	Anexos.....	39
8.1	Anexo I: Manual de uso de Meta Quest Link	39
8.2	Anexo II: Informe del Centro Grupo Estímulo	41

Índice de figuras

Figura 1. Ikea Place.....	6
Figura 2. Primer concepto RM.....	7
Figura 3. HoloLens.....	7
Figura 4. Vision Pro.....	8
Figura 5. Meta Quest 2.....	9
Figura 6. Uso de Meta Quest 2.....	9
Figura 7. Blender.....	11
Figura 8. Canva.....	12
Figura 9. Aplicación Meta Quest Link.....	13
Figura 10. Quest Link.....	13
Figura 11. Videojuego Hollow Knight.....	14
Figura 12. Logo VS Code.....	14
Figura 13. Instalación paquete Andoid.....	15
Figura 14. Paquete Meta.....	16
Figura 15. Referencia Cámara.....	17
Figura 16. Paquete Shadow Receiver.....	17
Figura 17. Universal RP.....	18
Figura 18. FBX.....	18
Figura 19. Prefabs.....	19
Figura 20. Escenas del proyecto.....	19
Figura 21. Diagrama de flujo.....	20
Figura 22. Escenas a compilar.....	20
Figura 23. Jerarquía de los paneles.....	21
Figura 24. Función booleana.....	22
Figura 25. Script Menú.....	22
Figura 26. Panel del Menú.....	23
Figura 27. Video Player.....	24
Figura 28. Cubos de tetris.....	24
Figura 29. Cubos virtuales.....	25
Figura 30. Tag y capa de las piezas.....	25
Figura 31. Sprite mariposa.....	26
Figura 32. Colores aleatorios.....	26
Figura 33. Script Start Confeti.....	27



Figura 34. Script Restrict Position.	27
Figura 35. Método principal de SnapToLocation.	29
Figura 36. Contador de errores.....	30
Figura 37. Método booleano de AllSnapped.....	30
Figura 38. Build Settings.....	31
Figura 39. Nombre del archivo de salida.	32
Figura 40. Detección de escenario.	33
Figura 41. Información del archivo final.	33
Figura 42. Ruta de guardado.	34
Figura 43. Aplicación Meta Quest del móvil.	34
Figura 44. Configuración general.....	39
Figura 45. Dispositivo conectado.....	40

Capítulo 1. Introducción

A lo largo de estos últimos 10 años se han ido implementando cada vez más las nuevas tecnologías en el ámbito educativo, tanto entre los alumnos más jóvenes como otros más pequeños. Esta llegada de la realidad virtual y aumentada a las aulas fue seguida recientemente por la implementación de la realidad mixta. Varias son las ventajas que nos aportan las tecnologías inmersivas para el desarrollo de los alumnos, sin embargo, pocas son las tecnologías que se han llegado a utilizar en las consultas para niños con necesidades específicas. Esta revolución tecnológica llegó de forma desigual para ellos.

Este trabajo de Fin De Grado surge y se enfoca en satisfacer esta carencia tecnológica que varios profesionales del ámbito mental han detectado en sus consultas. Esta aplicación de realidad mixta ayudará a los jóvenes con necesidades especiales a mejorar sus habilidades motoras y a su desarrollo cognitivo. Para ello se utilizarán las gafas Meta Quest y Unity como software de desarrollo. La idea de realizar estos ejercicios en realidad mixta se debe a que la aplicación está enfocada en ser utilizada junto a un profesional, ya sea en la consulta o en casa. Mientras el joven hace sus ejercicios el profesional estará presente y viendo los avances en tiempo real. Una vez acabado el ejercicio, los datos se guardarán dentro de la memoria interna de las Meta Quest para un continuo seguimiento del paciente.

1.1 Contexto y Justificación

Según Billingham (2002), creador del primer libro de RA «Magic book», la RA era: «La habilidad para hacer suavemente la transición entre realidad y virtualidad». Gracias al avance de la tecnología en este campo, hoy en día podemos disfrutar de todas estas tecnologías revolucionarias a través del hardware desarrollado para ello, como son las gafas Meta Quest. Según un estudio realizado entre 2008 y 2018 en el que se analizaban alrededor de 50 estudios anteriores de expertos, se llegó a la conclusión de que la realidad aumentada ayudaba a las personas con problemas de aprendizaje a entender mejor y más rápido los conceptos [1]. Todos los sujetos presentaban una mayor motivación e interés, además de encontrar estas aplicaciones divertidas e interesantes, lo que hace que su nivel de concentración sea mucho mayor.

A partir de 2020 la realidad mixta se extendió con fuerza y las grandes empresas como Microsoft o Google empezaron a revolucionar el panorama tecnológico con los dispositivos y plataformas de realidad mixta. La prueba más reciente de esta tecnología en auge, que ha llegado para quedarse, son sin duda las Apple Vision Pro anunciadas en 2023. Unas gafas de realidad mixta que crearon furor entre los usuarios, llegando a crear colas kilométricas y listas de espera.

Aunque la realidad mixta sea una tecnología que sigue en constante desarrollo, su implementación en la educación fue un tema controversial. Es una magnífica herramienta que ayuda y, además, brinda mejores resultados en un paciente con necesidades especiales. Y es que la implementación de las TIC en cualquier ámbito de la educación presenta numerosos beneficios [2]. La Realidad mixta es un potenciador y una tecnología extremadamente útil si se enfoca en resolver esta falta de inclusión que sigue existiendo actualmente en el ámbito académico y en las consultas de los terapeutas.

1.2 Realidad Aumentada y Mixta

En 1992, Tom Caudell describió el término de realidad aumentada como un sistema visual que permitía ser más eficiente al superponer información digital sobre el mundo real [3]. Hoy en día nos referimos a la realidad aumentada como una tecnología que superpone elementos virtuales en el mundo real. Sin embargo, no interactúan profundamente con el entorno físico. En 2010 empezó a popularizarse y las grandes empresas comenzaron a apostar por ella con la llegada de los Smartphones, el desarrollo de dispositivos y plataformas. Su llegada cambió el panorama y a pesar de que al principio no fue algo muy demandado acabó siendo una tecnología al alcance de todos. A partir de ese momento comenzó a utilizarse en varios campos diferentes como el entretenimiento, turismo, industrial, videojuegos, medicina, educación, etc.

Tanto avanzó en los diferentes campos que hoy en día podemos traducir un texto en tiempo real con Google Lens, capturar personajes virtuales con la cámara de nuestro teléfono con Pokemon Go o ver como quedarían unos muebles en nuestro salón con Ikea Place (Figura 1). En el campo de la medicina aparecieron aplicaciones como AccuVein en el que se pueden ver las venas en tiempo real y en el campo de la educación muchas otras como Arloopa en el que se puede intercambiar contenidos educativos.

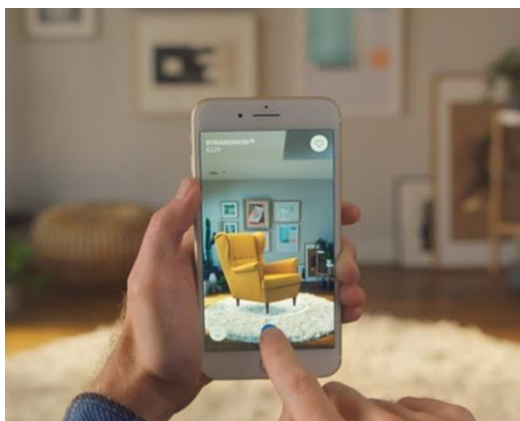


Figura 1. Ikea Place.

El término de realidad mixta surge en 1994 con la publicación del ensayo *Augmented Reality: A class of displays on the reality-virtuality continuum* [4] y define la combinación de realidad virtual y realidad aumentada (Figura 2). En muchas ocasiones, los términos de realidad mixta y realidad aumentada se utilizan de forma indistinta. Otros autores consideran que la realidad mixta va más allá de la realidad aumentada, y no solo superpone un objeto digital, sino que lo integra completamente en el mundo real haciendo posible una mayor interacción con los objetos físicos. La realidad mixta surge y evoluciona gracias a la realidad aumentada y la realidad virtual. Estas dos tecnologías se combinan para crear un entorno mucho más inmersivo brindando experiencias más envolventes y realistas, es por eso que la realidad mixta ha ganado terreno en diferentes ámbitos.

Actualmente la realidad mixta permite formar profesionales, así como realizar operaciones o tener material de apoyo durante una operación. En el campo de la educación esta tecnología enriquece y mejora la calidad de la educación, siendo capaces por ejemplo de visualizar conceptos abstractos o tener apoyo. A pesar de que la realidad aumentada y la realidad mixta comparten una base común, la realidad mixta extiende las capacidades de la realidad aumentada y es por eso que se presentan nuevas posibilidades de desarrollo en múltiples campos.

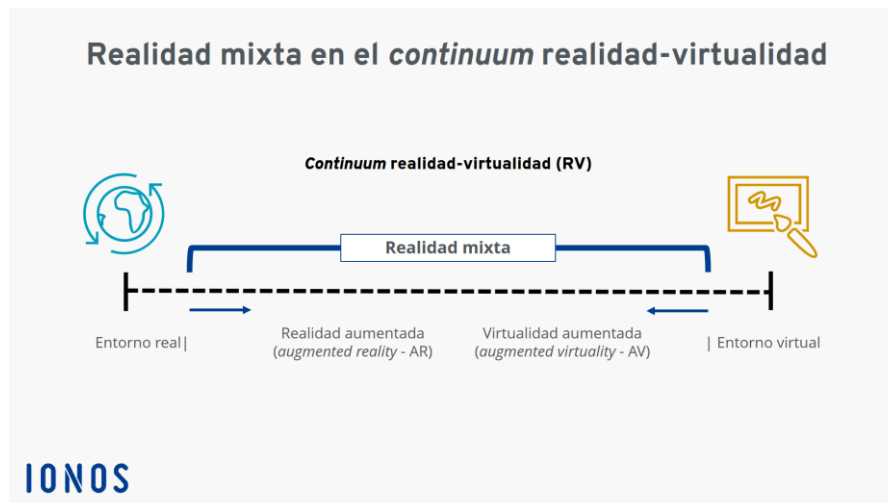


Figura 2. Primer concepto RM.

1.3 Dispositivos

Con la llegada de los Smartphones a nuestras vidas, todo el panorama tecnológico cambió. Todo pasó de forma exponencial y a gran velocidad por lo que las empresas tuvieron que desarrollar rápidamente nuevos dispositivos que pudieran seguir el ritmo de estas tecnologías emergentes. Es por eso que actualmente se brindan muchas opciones al usuario dependiendo de sus necesidades o preferencias.

Dispositivos AR

Al principio una cámara en nuestro Smartphone era suficiente para poder disfrutar de la realidad aumentada, un claro ejemplo de ello es la aplicación de Ikea Place (Figura 3). Tras el lanzamiento de Microsoft de las Google Glass, se popularizó el término de realidad aumentada. A pesar de no haber tenido éxito en este lanzamiento, Microsoft sacó las HoloLens, las primeras gafas de realidad aumentada que tuvieron un importante impacto en múltiples industrias.



Figura 3. HoloLens.

Dispositivos MR

A partir de las primeras gafas de Realidad virtual y realidad aumentada, la transformación hacia la realidad mixta se dio con posteriores mejoras. El Microsoft HoloLens está a la vanguardia de la tecnología de Realidad mixta. También son compatibles con varias aplicaciones de diversas industrias. Posee un seguimiento espacial avanzado, comandos por voz, se pueden utilizar controladores, tiene una resolución de pantalla de 2k y se pueden conectar al Wi-Fi o por Bluetooth.

El último lanzamiento de Apple, las Apple Vision Pro (Figura 4), marca un hito significativo en la tecnología. Con sus pantallas duales de 4K para cada ojo, audio espacial, y la capacidad de controlar el dispositivo usando gestos naturales y comandos de voz, sin necesidad de controladores físicos. Cuenta con un procesador M2 y el nuevo chip R1. El Vision Pro opera con visionOS, como todos los dispositivos Apple, está desarrollado para comunicarse con otros dispositivos del ecosistema. Una armonía perfecta entre el mundo virtual y el real.



Figura 4. Vision Pro.

1.3.1 Meta Quest 2

Para la realización de este proyecto de Fin de Grado se han usado las gafas de realidad virtual, las Meta Quest 2 (Figura 5). Fue desarrollado por Meta y son muy populares en el mercado. Existe una versión anterior y también otra más nueva con varias mejoras, las Meta Quest 3. Esta versión fue lanzada en 2020 y parte en un precio inicial de 250€. Es un dispositivo potente pero accesible para el bolsillo. Posee controladores físicos, los Oculus Touch, que tienen una gran precisión. Pantallas LCD con una frecuencia de actualización de 90 Hz, seguimiento de posición y controladores, audio integrado y un almacenamiento interno de 64 Gb o 256 Gb. Además, se pueden conectar las Meta Quest 2 de forma inalámbrica mediante un router WiFi 6. La autonomía ronda las 2 horas y media y los mandos funcionan con pilas. [5]

Una de las razones de su popularidad es la autonomía, no necesitan estar conectadas a un ordenador, todo el hardware necesario está integrado en las gafas. Además del rendimiento, comodidad y la relación calidad-precio, también ofrece un adaptador para la gente que usa gafas, algo muy útil para los usuarios.



Figura 5. Meta Quest 2.

Meta ofrece la posibilidad de desarrollo abierto para todos aquellos desarrolladores que quieran usar su hardware. Para este proyecto, se ha creado una cuenta de desarrollador en la página oficial de Meta y se ha descargado de la aplicación de Meta llamada Meta Quest Link. Para el proceso de desarrollo se ha utilizado el “cable link”, un cable USB-C, como método de conexión entre las Oculus 2 y el ordenador.

A pesar de haber desarrollado esta aplicación con cable link, no es necesario utilizar el cable para poder usar la aplicación final. Su finalidad es utilizarla sin cables, de forma autónoma, y sin controladores. Los niños usarán sus manos para interactuar con los objetos virtuales que se encuentren en las actividades (Figura 6). De esta manera, será mucho más sencillo para ellos y mejorarán su movilidad motora.



Figura 6. Uso de Meta Quest 2.



Capítulo 2. Objetivos

El objetivo principal de este proyecto es desarrollar una aplicación de realidad mixta que sea beneficiosa, útil y ayude al desarrollo cognitivo de niños con problemas. Aprenderán a utilizar las gafas Meta Quest 2 y realizarán los ejercicios de forma más interactiva gracias al juego programado con Unity.

Esto conlleva a otros objetivos más específicos que se esperan conseguir a lo largo del proyecto. Algunos de estos objetivos específicos son:

- La evaluación de la eficacia del proyecto. Se evaluará si ha habido una mejora de las habilidades.
- Los contenidos interactivos son atractivos. Hay un aprendizaje más activo por parte del niño.
- La aplicación se adapta a las necesidades del niño. Por ejemplo, a su edad o a su nivel.
- Comparación entre los resultados obtenidos por métodos tradicionales y la realidad mixta. Presenta una mejora u obtenemos los mismos resultados.
- Satisfacción del usuario, profesional y niño, con la aplicación.
- Es accesible y fácil de usar.
- Un análisis del impacto del uso prolongado en la enseñanza con la realidad mixta.
- Seguridad y bienestar. Aplicación segura para los niños.

A raíz de estos objetivos específicos, también podríamos establecer 2 objetivos enfocados a una investigación en este campo concreto de la educación.

1. Determinar si el proyecto es factible pedagógicamente, si será beneficioso. La integración de realidad mixta en programas para niños con dificultad para aprender.
2. Determinar si esta innovación tecnológica es posible implementarla en las consultas.

Capítulo 3. Metodología

Antes de describir en detalle cómo ha sido la realización del proyecto de forma más exhaustiva, se expondrán los pasos previos, así como todos los elementos que han hecho posible el desarrollo de esta aplicación que ayudará al desarrollo cognitivo de los niños con problemas.

3.1 Hardware

Para el desarrollo, se ha utilizado:

1. Ordenador Huawei Matebook D14. Con un procesador AMD Ryzen 5 y Microsoft Windows 11 instalado. Cuenta con una tarjeta gráfica AMD Radeon Vega 8.
2. Gafas Meta Quest 2. Con una memoria interna de 64 Gb y procesador Snapdragon XR2 que ofrece una alta calidad de gráficos. Se han elegido estas gafas por su autonomía, calidad de los gráficos y por ser compatible con varios softwares de desarrollo.
3. Un cable USB-C. Necesario para establecer una conexión entre el ordenador y las gafas Meta Quest 2 durante el desarrollo del proyecto.

3.2 Software

Para el desarrollo, se han utilizado varios softwares diferentes. La combinación de todos ellos ha hecho posible la realización del proyecto y alcanzar los objetivos propuestos para la versión final.

3.2.1 Blender

Blender (Figura 7) es un programa gratuito, de código abierto y multiplataforma enfocado al diseño y creación en 3D, lo que lo hace idóneo para los profesionales como artistas digitales y multimedia. Este software no solo se limita al diseño, también podemos crear animaciones 3D, edición de vídeos, esculturas, impresiones 3D, etc. Es una herramienta muy conocida y usada tanto en el ámbito educativo, formando a los estudiantes como los profesionales del ámbito tecnológico [6]. Blender trabaja con formatos bidimensionales y tridimensionales. También está enfocado en la creación de entornos pensados para videojuegos.



Figura 7. Blender.

Su popularidad llegó en 2011 cuando se convirtió en un Software internacional y desde ese momento no ha dejado de crecer y aportar significantes mejoras con cada actualización. Una de las mayores fue en 2019 con el motor de renderizado Eevee. Actualmente es uno de los Software de modelado 3D más reconocidos y utilizados alrededor del mundo. Además, aún sigue atrayendo tanto a profesionales como a aficionados. Junto con Autodesk Maya que se utiliza en los entornos más profesionales y de producción, son los dos softwares más populares.

Concretamente, para la realización de este proyecto, se ha utilizado Blender para crear los cubos, tableros y materiales que se utilizan a lo largo de los ejercicios. Blender posee un formato de exportación llamado .fbx con el que ha sido posible trasladar todos los elementos 3D al motor de Unity.

3.2.2 Canva

Canva (Figura 8) es una herramienta online, una plataforma gratuita, en la que es posible crear cualquier tipo de diseño gráfico. Su nombre viene dado debido a que es como un canvas en blanco en el que se puede crear cualquier diseño. La interfaz es sencilla para usuarios de diversa edad y sus aplicaciones son muchas, lo que la hace accesible [7]. Desde crear un vídeo y editarlo, hasta descargar plantillas, crear un CV, carteles, posters, portfolios, etc. Usada por aficionados y profesionales ya que se puede obtener un resultado rápido y satisfactorio. El límite es la imaginación y habilidades de uno mismo. También tiene una opción de pago con la que se desbloquean más herramientas o mejoras.

Se lanzó en 2013 y desde entonces ha seguido creciendo continuamente. Actualmente cuenta con planes para estudiantes o profesionales aparte de la opción premium para todos los usuarios. Utilizada por muchas empresas actualmente es una plataforma muy popular.



Figura 8. Canva.

Para este proyecto se ha utilizado Canva para crear los diferentes mensajes, así como los tableros y toda la tipología de los diferentes menús. Canva ofrece la opción de crear un lienzo en blanco eligiendo las medidas deseadas. De esta forma se ha creado especialmente cada panel y animación con las medidas necesarias.

3.2.3 Meta Quest Link

Meta Quest Link (Figura 9) es una aplicación especialmente diseñada para toda la gama de gafas de Meta. A través de esta aplicación se pueden conectar el modelo de gafas de Meta que estemos usando al ordenador. Este enlace se establece a través de un cable USB-C o a través del Air Link (Figura 10), de forma inalámbrica [8]. Además, tiene otras utilidades. Los usuarios gestionan el dispositivo conectado, tienen una tienda con diferentes aplicaciones y juegos, se unen a eventos, activan el modo desarrollador, se conectan con otros jugadores a través del chat, juegan con sus amigos, etc. Con sus últimas actualizaciones, hay una mejora en los enlaces para desarrolladores mejorando los flujos del trabajo y se establece una conexión automática cuando se conecta el enlace USB.

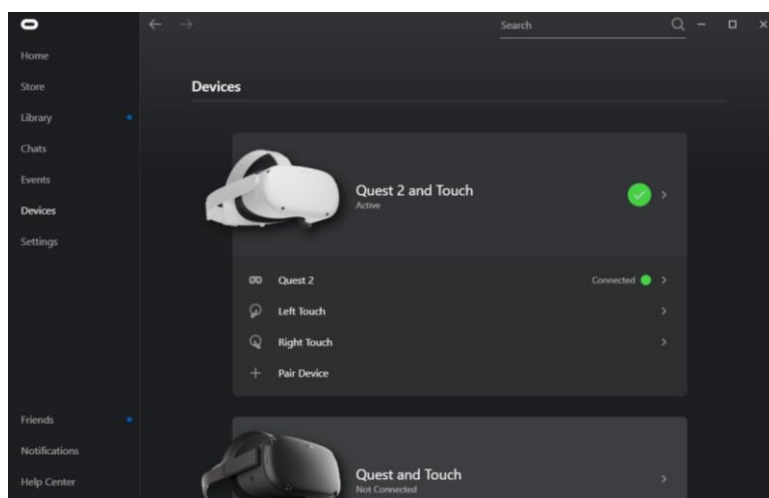


Figura 9. Aplicación Meta Quest Link.

Esta aplicación se ha utilizado para conectar las Meta Quest 2 a través del cable al ordenador para poder desarrollar con Unity este proyecto. Teniendo una cuenta de desarrollador y activando las diferentes opciones dentro de esta aplicación se ha podido desarrollar con éxito el proyecto. Una vez puestas las gafas, para que esta conexión se establezca activaremos el Quest Link anteriormente configurado en la aplicación y podremos ver los avances realizados. Este procedimiento de configuración está explicado en el Manual de la aplicación que se encuentra en el Anexo I.

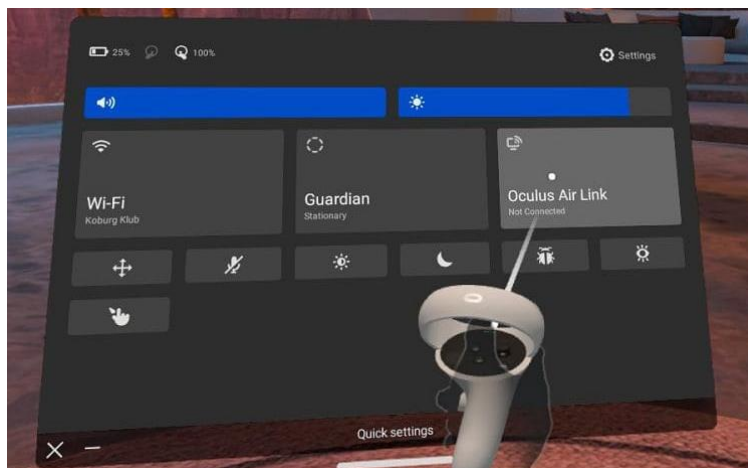


Figura 10. Quest Link.

3.2.4 Unity

Unity es una plataforma de desarrollo de contenido en 2D y 3D, principalmente videojuegos, creada por Unity Technologies. Su primera versión surgió en 2005 y a día de hoy es uno de los motores de desarrollo más populares del mundo. Esto se debe a que es un motor multiplataforma, capaz de crear un videojuego para 25 plataformas diferentes [9], algo muy atractivo para los desarrolladores. También es compatible con distintos dispositivos de realidad virtual o aumentada como las Meta Quest o HoloLens.

Muchos videojuegos populares (Figura 11) se realizaron con el motor de Unity, tanto es así que según SteamDB, más de 60% de las experiencias de realidad virtual más taquilleras de Steam en 2023 se crearon con Unity [10]. Además de los videojuegos, se desarrollan aplicaciones móviles, simulaciones, cine y aplicaciones de realidad aumentada, virtual y mixta.



Figura 11. Videojuego Hollow Knight.

El universo de Unity es extenso y cuenta con una aplicación para la gestión de versiones y proyectos, Unity Hub, y una Asset Store en la que los desarrolladores venden y compran recursos. Estos recursos agilizan el desarrollo de las aplicaciones, es beneficioso tanto para el comprador como para el vendedor. Los recursos pueden ser varios, desde un paquete completo hasta modelos en 3D. El lenguaje de programación que usa Unity es C# y es por eso que necesitamos un entorno de desarrollo, un editor de código compatible para poder desarrollar las interacciones que nos interesan.

En este proyecto se ha utilizado Unity como motor de desarrollo para la aplicación de realidad mixta y Visual Studio Code (Figura 12) como editor de código ya que es ligero, potente y con una capacidad de personalización gracias a sus numerosas extensiones.



Visual Studio Code

Figura 12. Logo VS Code.

Capítulo 4. Desarrollo

4.1 Configuración previa

En este apartado se describirán las configuraciones previas al desarrollo, necesarias para avanzar en el proyecto. Todos estos subapartados se enumerarán y explicarán según lo realizado, es decir, de forma cronológica.

4.1.1 Descarga Unity y entorno RM

En primer lugar, se ha descargado el motor de desarrollo Unity en el ordenador. De esta manera trabajaremos con la última versión disponible. Una vez descargado, debemos crear una cuenta para poder organizar todos los proyectos que tenemos a través de la aplicación de Unity Hub.

Antes de crear el proyecto tenemos que asegurarnos que la versión de Unity que utilizaremos tiene el módulo de Android instalado. Para eso en Unity Hub, en la pestaña de “Installs”, la versión que vayamos a utilizar tiene que tener un icono de Android, en este caso la 2022.3.10f1 (Figura 13). Para poder descargarla, a la derecha de la versión hay un icono en el que pincharemos y aparecerá la opción de “instalar módulos”. Dentro, seleccionaremos el módulo de Android Build Support, OpenJDK y Android SDK & NDK Tools.

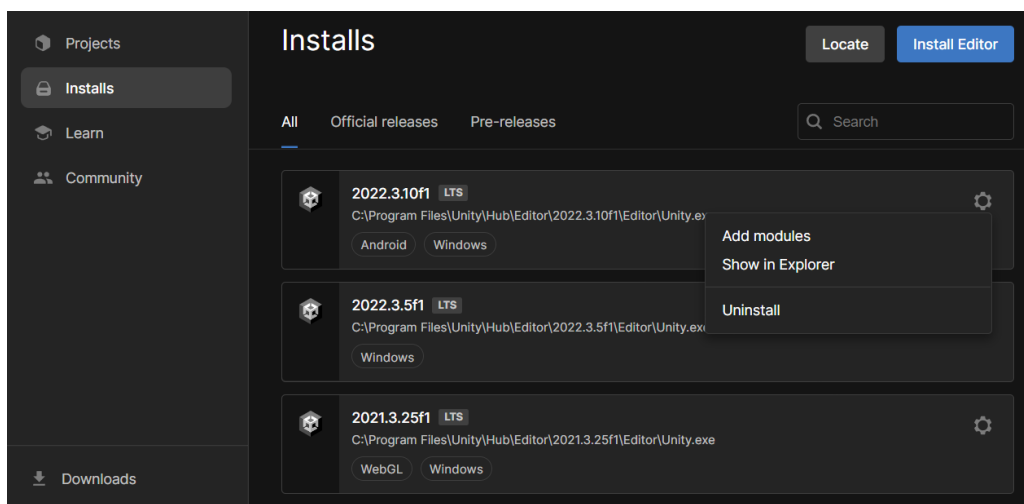


Figura 13. Instalación paquete Android.

Una vez todo instalado, podemos ir a la pestaña de proyectos y darle al botón de crear nuevo proyecto. Lo primero que haremos será seleccionar la versión de Unity que utilizaremos, después la plantilla 3D y le daremos un nombre al proyecto. Este proceso puede tardar dependiendo del hardware utilizado pero una vez se haya creado, se abrirá el entorno y podremos seguir configurándolo.

Dentro del entorno ya creado para la realización de este proyecto, entraremos en la pestaña de “File” situada arriba a la izquierda y seleccionaremos “Build Settings”. Se abrirá una pestaña en la que se cambiará la plataforma de desarrollo. Está predeterminada para Windows, Mac y Linux, pero la cambiaremos a Android ya que los dispositivos Oculus funcionan con plataformas Android.

4.1.2 Paquetes necesarios de la Asset Store

El primer paso para poder descargar cualquier paquete es tener una cuenta de Unity. Esta cuenta ya ha sido creada anteriormente para poder conectar las gafas Meta Quest 2 al ordenador y acceder a los beneficios de desarrollador. Para un desarrollo más eficiente, se han utilizado varios paquetes de la Asset Store.

Meta XR All-in-One SDK [11] es el más importante. Es un paquete gratuito desarrollado por Meta (Figura 14) que agrupa varios SDK diferentes. Es un paquete completo enfocado en el desarrollo de entornos de Realidad virtual, mixta o aumentada. La versión utilizada para este proyecto es la última versión disponible actualmente, la versión 65. Una vez importada, nos redirigirá automáticamente al editor de Unity donde nos encontraremos con la ventana de “Package Manager” ya abierta donde se encuentra el paquete. Lo instalaremos y después se reiniciará Unity. Cuando volvamos a abrir el editor de Unity, en la misma pestaña anterior de “Package Manager”, si cambiamos a “In Project”, veremos todos los módulos que contenía el paquete ya instalado en el proyecto.

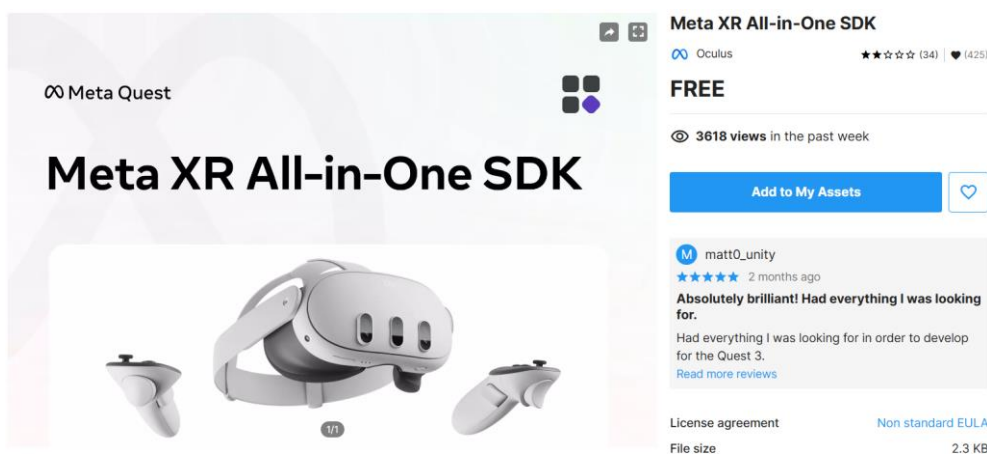


Figura 14. Paquete Meta.

El último paso para poder utilizar el paquete es realizar la configuración automática necesaria para el desarrollo. En la pestaña de Oculus entraremos en “Project Setup Tools” y verificaremos que estamos en la pestaña que corresponde a Android. En esta pestaña aparecen varios errores o sugerencias por parte del paquete y para que todo funcione correctamente aceptaremos todas. Haremos clic en el botón de “Fix all” hasta que ya no queden más y luego en la pestaña de Windows nos aseguraremos de que todo esté bien.

Una manera sencilla de utilizar los recursos del paquete es la herramienta de Building Blocks. Arrastraremos los elementos necesarios para la escena: Cámara Rig, Hand Tracking, Virtual Hands y Passthrough. Estos elementos hacen referencia a la cámara de la escena que en este caso son las gafas de Meta Quest, el seguimiento en tiempo real de nuestras manos, la creación de manos virtuales a partir de las manos reales captadas por la cámara y la opción de ver el mundo real y los elementos virtuales al mismo tiempo. Dentro de la cámara cambiaremos algunas configuraciones predeterminadas en el OVR Manager. Los dispositivos a los que está enfocado este proyecto son los Meta Quest 2 y Meta Quest 3. Además, se han cambiado algunas funcionalidades de la cámara como, por ejemplo, la interacción se realizará solo con las manos y la frecuencia del tracking será alta. Dentro de la jerarquía de la cámara debemos referenciar las manos sintéticas, para cada mano por separado, en “HandGrabVisual” (Figura 15).

Otra forma de utilizar las interacciones es añadiéndolas directamente a objetos específicos de la jerarquía. Haciendo clic derecho sobre una pieza, en el apartado de interacción SDK se encuentran las diferentes opciones que brinda el paquete. Para el desarrollo del proyecto se ha utilizado el agarre a poca distancia, “SDK Grab Interaction” y para los paneles “SDK Poke Interaction”.

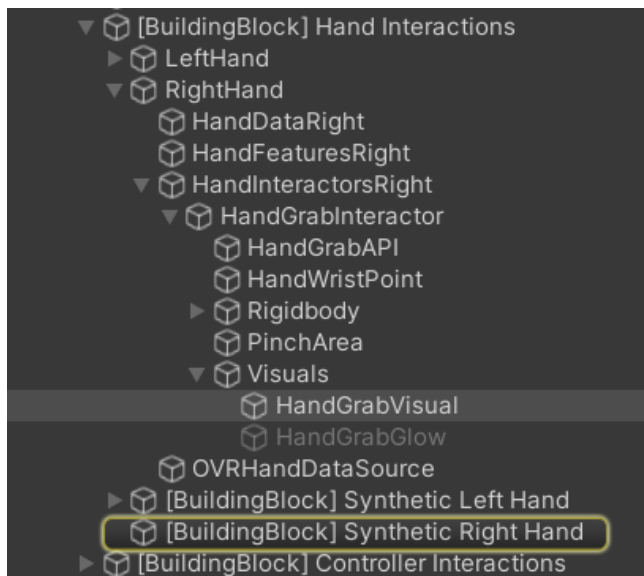


Figura 15. Referencia Cámara.

Shadow Receiver URP es un paquete gratuito disponible para todos los dispositivos y desarrollado por Dev Drunk Studio. La versión utilizada es la 3.1.0 y las sombras se pueden customizar, además es compatible con el paquete Universal RP. Proporciona sombras en tiempo real a cualquier elemento virtual 3D creados en realidad aumentada o mixta (Figura 16), lo que hace que se perciba como un entorno más realista. Dando esa sensación de profundidad, como si los objetos virtuales estuvieran realmente integrados en la realidad.

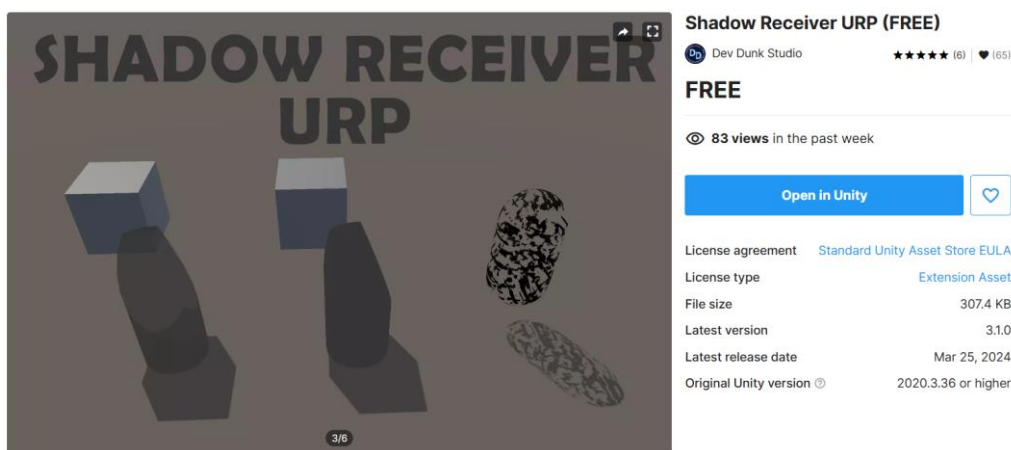


Figura 16. Paquete Shadow Receiver.

Universal RP o Universal Render Pipeline (URP) se utiliza para tener unos gráficos más avanzados ya que optimiza el renderizado en tiempo real [12]. La versión utilizada es la 14.0.8, es gratuito y útil para tener una mejor experiencia de usuario. Una vez instalado, con el botón derecho del ratón abriremos la pestaña y accederemos a Crear > Rendering > Universal Render Pipeline > Pipeline Asset. Esto nos creará dos objetos, uno de ellos llamado URP y el otro que corresponde al “rendering” del URP (Figura 17), que debemos utilizar para cambiar la configuración de los gráficos de nuestro proyecto. En la ventana de Editar abriremos “Project Settings” y entraremos en la pestaña de Gráficos.

Después de haber importado el paquete aparecerá como primera opción en esta pestaña poner un pipeline, que será el URP que hemos creado anteriormente.

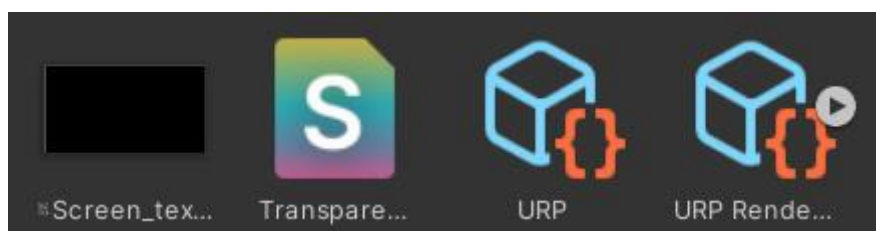


Figura 17. Universal RP.

Si tuviéramos alguna textura o material creado anteriormente, deberemos cambiar su configuración porque ya no será compatible. Para solucionar esto de manera eficiente, entraremos en la pestaña Edit > Universal Render Pipeline > Update Project Materials to UniversalRP Materials. De esta manera todo el proyecto estará configurado para soportar y brindar unos mejores gráficos.

En el URP Render habilitaremos el “Screen Space Ambient Occlusion” para tener una mejor oclusión ambiental. Además, esta oclusión se puede modificar en intensidad o calidad.

4.1.3 Importación desde Blender

Uno de los elementos principales para la realización de este proyecto son los elementos 3D. Una vez ya han sido creadas con Blender, para poder exportar las piezas, se seleccionarán y exportarán una por una. Esto se realiza abriendo la pestaña de “File”, desplazándose hasta exportar y seleccionando el formato FBX (Figura 18). Para que se exporte con todos los materiales asignados, se deberá cambiar el “Path Mode” a “Copy” y activar el icono de al lado llamado “embed textures” [13]. Al utilizar este modo de exportación, se garantiza que las texturas o materiales que van asociados al objeto 3D se exporten con él.

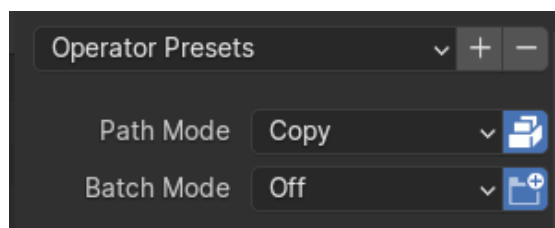


Figura 18. FBX.

La razón por la que se exporta cada pieza individualmente en lugar de exportar todos los objetos creados juntos, es porque al importarlos en Unity, se transforman en “Prefabs” (Figura 19) y la intención es que cada “Prefab” sea independiente ya que cada uno tiene objetivos y funcionalidades diferentes en su posterior desarrollo. Al tenerlos separados, se pueden ajustar de manera más eficiente y precisa dentro de Unity.

Una vez importados, cada uno se pueden añadir a la escena en la posición y orientación deseados para comenzar a programar su comportamiento. De esta forma, se mantiene una mayor organización, flexibilidad y control dentro del proyecto. Esta estrategia mejora la eficiencia del flujo de trabajo y la calidad del producto final.



Figura 19. Prefabs.

4.2 Escenas y su flujo

Cada escena es necesaria y desempeña una función específica dentro del proyecto. Este proyecto consta de un total de 6 escenas (Figura 20). Tres de ellas pertenecen a las actividades o niveles que realizarán los niños dependiendo de su edad, nivel de aprendizaje o mejora y las otras 3 escenas corresponden a los menús de libre elección que facilitan el flujo por la aplicación.

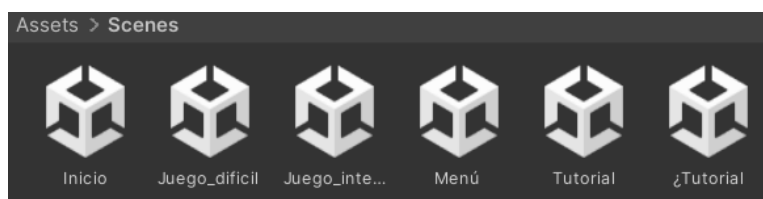


Figura 20. Escenas del proyecto.

La primera escena que nos encontramos al iniciar la aplicación es la de inicio, en ella encontramos un panel, con el que se interactúa con las manos. Se presentan dos posibilidades, comenzar el juego o salir de este. En la segunda escena, se lanza otro panel también interactivo en el que se nos da a elegir si queremos realizar un pequeño tutorial para entender mejor la dinámica de los ejercicios. Este tutorial puede ser utilizado como un nivel básico para un niño con menos nivel o una introducción para algunos niños un poco más mayores. Si aceptamos, nos llevará a la escena de dicho tutorial en el que tendremos unos paneles informativos a modo de guía. Cuando se completa el nivel, el panel cambiará para dar la enhorabuena, se activará el cañón de confeti y aparecerá el panel que nos da a entender que se está cargando automáticamente la siguiente escena.

Si por el contrario decidimos no hacer el tutorial, nos dirigirá de forma automática al nivel intermedio del proyecto que presenta una mayor dificultad en comparación con el anterior. En este, ya no se encuentran paneles informáticos y al completarlo, se activará el cañón de confeti y aparecerá un pequeño panel que nos preguntará si queremos iniciar el siguiente nivel, el nivel difícil. Si lo iniciamos nos dirige automáticamente a él. Cuando se termine aparecerá el mismo panel mencionado anteriormente que nos da a entender que se está cargando automáticamente la siguiente escena. Esta escena será el menú principal en el que se puede elegir a qué parte del proyecto queremos volver, es decir, a qué escena específica.

Todo esto se puede ver reflejado de forma más clara en el siguiente diagrama de flujo (Figura 21).

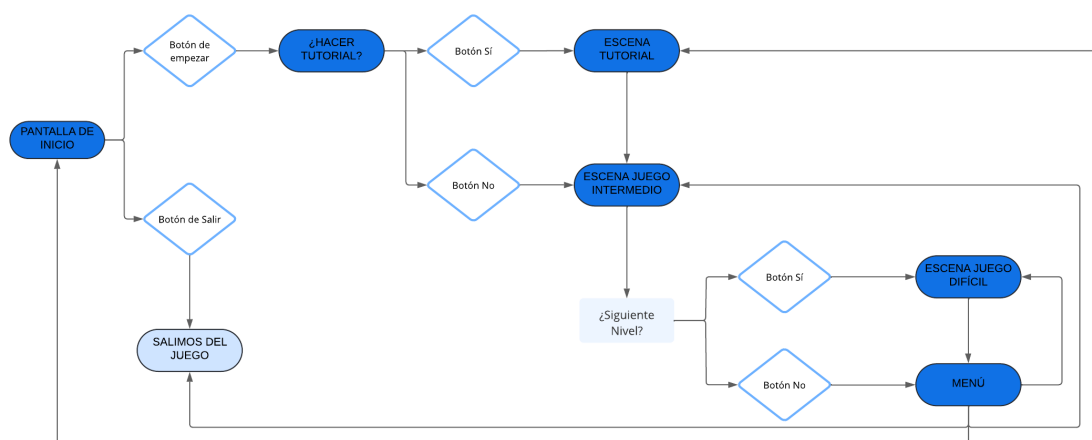


Figura 21. Diagrama de flujo.

4.2.1 Organización de escenas

Al iniciar un proyecto en Unity, se crea una carpeta llamada “Scenes” donde se almacenarán todas las escenas. Esta carpeta es fundamental para organizar y gestionar las diferentes escenas de manera coherente y estructurada. En la barra del menú de Unity, al seleccionar “File” y después “Build Settings” se abrirá la ventana que pertenece a la configuración de compilación. En la sección de “Scenes in Build” se deberán añadir todas las escenas que se deseen compilar y además se podrá ordenar (Figura 22). El orden es algo esencial, la primera escena de la lista siempre tiene que ser la misma que la del proyecto y además siempre será la 0. De esta manera cada nombre de la escena estará asociada a un número dependiendo de su posición. Esta numeración no solo nos sirve para exportar el proyecto, sino también para el desarrollo. Se podrá llamar a una determinada escena utilizando el nombre o número de la escena.

Scenes In Build	
✓ Scenes/Inicio	0
✓ Scenes/¿ Tutorial	1
✓ Scenes/Tutorial	2
✓ Scenes/Juego_intermedio	3
✓ Scenes/Juego_dificil	4
✓ Scenes/Menú	5

Figura 22. Escenas a compilar.

4.3 Elementos presentes en las escenas

El proyecto se divide en 6 escenas diferentes como se ha mencionado anteriormente y cada una de ellas tiene un propósito distinto. Todas contienen elementos esenciales para la correcta ejecución del entorno de realidad mixta: “Camara Rig” configurada anteriormente para el seguimiento de manos, “Passthrough” [14] para implementar el entorno de realidad mixta y una luz. Estos elementos están presentes en todas las escenas, sin embargo, no todas las escenas contienen los mismos elementos. A continuación, se detallará todos los tipos de elementos presentes en el proyecto, así como su configuración.

4.3.1 Paneles

A lo largo de todo el proyecto, se observa la aparición de diversos tipos de paneles. Los más relevantes son los paneles interactivos que nos conducen a otras escenas. La configuración realizada a los paneles que pertenecen a las escenas de inicio, tutorial y juego es uniforme. Todos ellos comparten características comunes. En la jerarquía de la escena (Figura 23), se ha creado un objeto vacío al que se le ha asignado un Canvas como hijo mediante el menú de la pestaña de UI. Posteriormente, se ha añadido otro objeto vacío denominado "Background", que servirá como fondo del panel. A este objeto se le ha añadido una imagen como componente a través del menú de UI, lo que permite establecer el color deseado, en este caso transparente, y modificar los bordes para darles una forma más redondeada.

Se ha establecido otro objeto vacío como hijo del Canvas, el cual formará parte del contenido del este con el modificador “Vertical LayoutGroup”. Ajustando algunos valores, como los márgenes, y configurando el modificador para que mantenga el ancho de su contenido. A continuación, dentro del contenido, se encuentran dos objetos vacíos adicionales que representan la información dividida entre la parte superior e inferior del panel. Esto significa que nuestro panel estará dividido en dos secciones. En la parte superior, hay una imagen como hijo, y en la parte inferior, mediante el menú de UI accesible con clic derecho, se ha añadido otra imagen denominada "Mask", que utilizaremos para dar color a esta sección. Dentro de esta máscara, agregaremos los botones necesarios desde la pestaña de UI, los cuales siempre serán dos en estos paneles. Estos botones pueden tener opciones como “Sí” y “No” o “Jugar” y “Salir”. Además, a cada panel interactivo se le ha asignado un “SDK Poke Interactor” que permite interactuar, es decir pulsar, el panel con las manos virtuales.

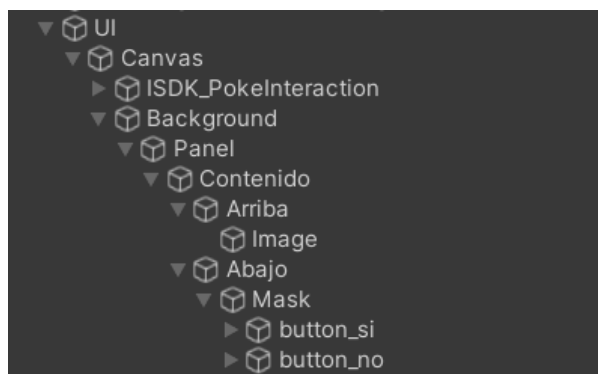


Figura 23. Jerarquía de los paneles.

Estos botones tienen un “Box Collider 2D” para detectar colisiones, es decir, para detectar si se ha pulsado y cambiar el estado de la función “On Value Changed” (Figura 24). Esta función verifica si se ha pulsado el botón, es decir, si ha pasado a verdadero. Para implementar esto, se ha utilizado un script llamado Menú previamente asociado a nuestro componente UI. En la función mencionada anteriormente, se ha enlazado este componente UI en el inspector, lo que permite navegar por el script y al cambiar a verdadero, activar esa función específica del script. Esto puede llamarse solo una vez, que será el momento en el que se detecte que se ha pulsado el botón.

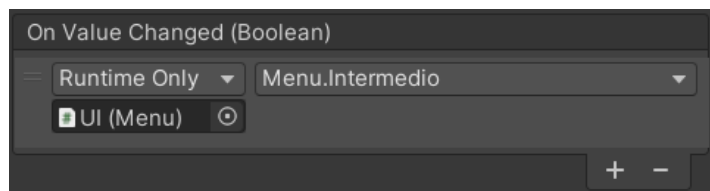


Figura 24. Función booleana.

El script asociado, llamado Menú (Figura 25), consta de varias funciones diseñadas para cumplir distintos propósitos, todas ellas pensadas para ser añadidas a los diferentes botones que se encuentran en los paneles. Dependiendo de la función invocada por cada botón, este nos redirigirá a un escenario concreto utilizando el “Scenemanager” y las escenas previamente instanciadas en el “Build Settings”.

```
public void Jugar()  
{  
    SceneManager.LoadScene(1);  
}  
  
public void Inicio(){  
    SceneManager.LoadScene(0);  
}  
  
public void Tutorial(){  
    SceneManager.LoadScene(2);  
}
```

Figura 25. Script Menú.

El panel del menú principal sigue la misma lógica que el tipo de panel anterior, con una modificación: en la parte inferior perteneciente al “Content”, se ha añadido un “Scroll View” [15]. Dentro de este, navegando hasta “Content” y se ha añadido un componente llamado “Horizontal Layout Group”, al que se le han realizado algunos ajustes estéticos. A continuación, añadiremos un segundo componente “Content Size Fitter” que mantendrá un tamaño horizontal uniforme en todo el elemento y un tercer componente llamado “Toggle Group” para manejar la lógica de los diferentes botones.

Dentro del “Content”, se añadirán tantos botones como sean necesarios, los cuales se posicionarán uno al lado del otro con su margen correspondiente. Cada botón contiene los componentes "Toggle Deselect", "Layout Element" y "Box Collider 2D". Este primero, es el encargado de la interactividad de los botones mediante la invocación de funciones, como se explicó anteriormente en este apartado. Además, se añadirá un SDK Interactor diferente, en este caso, un rayo (Figura 26).

La forma de interactuar con el panel seguirá siendo a través de las manos, pero no se podrá pulsar sobre él. Será necesario coger y arrastrar los botones con un gesto de la mano realizado en el aire, y para entrar en una de las escenas propuestas, habrá que hacer una especie de doble clic con la mano.



Figura 26. Panel del Menú.

Otro tipo de paneles son los informativos, estos los encontramos puntualmente en la escena del tutorial. Estos paneles actúan como una pequeña guía para poder comprender mejor la dinámica de las actividades. Estos paneles sirven para los niños que prueban por primera vez estas actividades. Son especialmente útiles para los niños que prueban estas actividades por primera vez, ya que una breve explicación mejora su tiempo de respuesta y reduce el estrés.

El cuarto tipo de panel presente en el proyecto es similar al típico panel de carga de cualquier aplicación. En este caso, es un cubo girando sobre sí mismo que simula el proceso de carga hacia la siguiente escena. El video seleccionado es el de un cubo para mantener la estética y temática de cubos de la actividad. Para crear este panel, se ha descargado un vídeo y, en la jerarquía, se ha creado un nuevo Canvas al que se le ha añadido una “Raw Image” como hijo. Esta imagen se escala para que coincida con el tamaño del Canvas. En la jerarquía, también se crea otro objeto vacío al que se le añade el componente de “Video Player” (Figura 27). Después, en la carpeta de materiales, se crea un “Render Texture” y se ajusta su tamaño para que coincida con el del vídeo. En la pestaña de Videoclip, se referencia en el inspector el vídeo descargado del cubo girando y en “Target Texture” se selecciona la textura creada anteriormente. Una vez configurado el componente “Video Player”, se vuelve al “Raw Image” y en la pestaña de textura se añade la textura del render. Con esta configuración podemos visualizar el vídeo del cubo girando en el proyecto de realidad mixta.

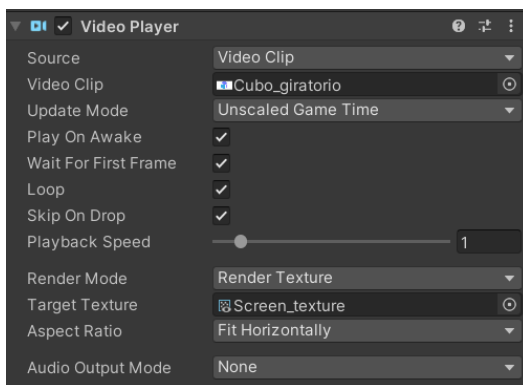


Figura 27. Video Player.

4.3.2 Cubos y tableros

Estos objetos 3D se han desarrollado utilizando la versión 4.0 de Blender, un software de diseño anteriormente nombrado para la realización de este proyecto. Una vez abierto el entorno de Blender, en la pestaña de “Layout”, se añadirá un cubo navegando hasta la pestaña de Add > Mesh. Cada pieza ha sido modificada para tener los bordes de los cubos de diferente color. Para lograr esto, se ha seleccionado uno de los cubos y se ha cambiado al modo edición. Utilizando la herramienta “Loop cut” se han logrado los cortes que crea ese efecto visual. Al acabar, se volverá al modo objeto, en el que se utiliza el modificador de “Array” para obtener la forma deseada. A cada pieza, se le ha asignado dos materiales diferentes: uno correspondiente a los bordes, previamente segmentados, y otro material para el color principal del cubo. Este material principal ha sido configurado en la pestaña de materiales dependiendo de la pieza, para crear esos colores tan vívidos y atractivos para los niños.

Estas piezas, así como los tableros, surgen de la idea inicial de que los niños puedan colocar los cubos en las zonas del tablero correctas. Para captar su atención, los tableros y los cubos se son llamativos, inspirados en el juego de Tetris, en formato físico, con sus formas y colores (Figura 28). Este diseño busca atraer visualmente a los niños y que se mantengan interesados en las actividades educativas.

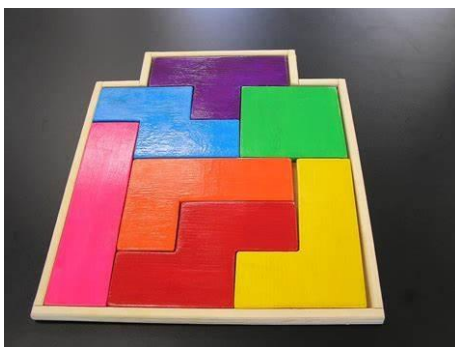


Figura 28. Cubos de tetris.

Esta idea fue extrapolada al mundo virtual creando las piezas y sus tableros correspondientes. En el entorno virtual, los niños pueden interactuar con los cubos de colores, fomentando su interés y facilitando el aprendizaje a través del juego (Figura 29).

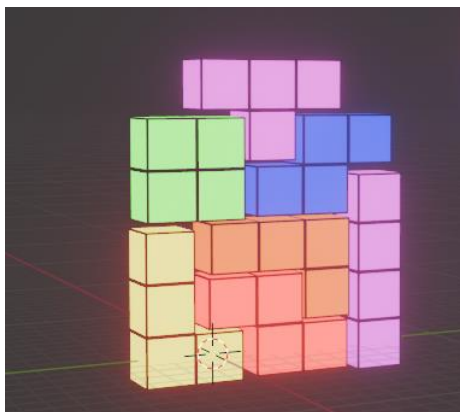


Figura 29. Cubos virtuales.

Dentro de Unity, cada cubo, así como su pieza del tablero tienen asignados un “Tag” (Figura 30) en el inspector que corresponde con el color que tiene cada pieza. Además, se ha creado una capa que se utiliza específicamente para la interacción del tablero con los cubos llamada “Interactable”. Por otra parte, los cubos de la escena tienen un “RigidBody” que estableceremos como cinemático para que no le afecte la gravedad, un “Box Collider” que identifica las colisiones y un “OVR Grabbable” que a través de la cámara determine si el objeto está siendo agarrado. Todos estos elementos son esenciales para la correcta interacción y programación.

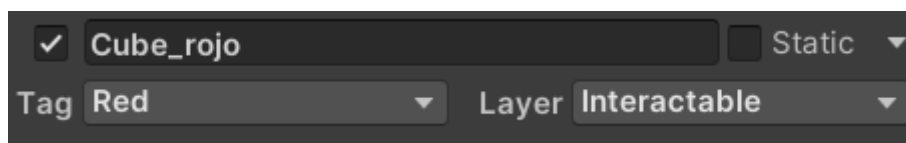


Figura 30. Tag y capa de las piezas.

Por otro lado, aunque el tablero sea un conjunto, cada pieza es independiente. Cada una tiene un “Box Collider” para detectar colisiones con la opción de “On Trigger” seleccionada. Al tener esta opción activada en la pieza del tablero y en el cubo no, se podrá detectar las colisiones realizadas entre las dos piezas virtuales. De esta manera se garantiza que obtendremos un comportamiento esperado al realizar la actividad ya que cada colisión se determina por separado.

4.3.3 Confeti

Una de las mejoras introducidas en este proyecto es un cañón de confeti que se activa al inicio del juego y cuando se completan los diferentes niveles. Está creado dentro de la plataforma de Unity a base de un sistema de partículas configurado con varias técnicas y herramientas para un movimiento realista y sincronizado con la animación para brindar ese efecto confeti de mariposa.

El primer paso para la creación de un sistema de partículas es crear un “GameObject” vacío en la escena y añadirle el componente de “Particle System” [16]. Tras establecer el sistema, se le dará forma y para conseguir esa forma de mariposa, se ha descargado un “Sprite” ya creado. Una vez importado al proyecto, debemos abrir y en la pestaña de “Texture Type” seleccionar la opción de “Sprite (2D and UI)”. Una vez aplicados los cambios aparecerá ya la forma deseada de una mariposa en el visor de Unity.

Para poder asignar este Sprite al sistema de partículas, debemos crear un nuevo material haciendo clic derecho > Create > Material y cambiar el “Shader” de este a “Mobile/Particles/Alpha Blended”.

De esta manera se podrá adjuntar el Sprite al material arrastrándolo hasta la casilla vacía (Figura 31). El siguiente paso es asignar este Material al sistema de partículas, para eso navegaremos hasta la opción de “Renderer” y asignaremos dicho material, cambiando el modo de renderizado a “Mesh” y seleccionando el “Quad” y modificando el “Render Alignment” a Local.

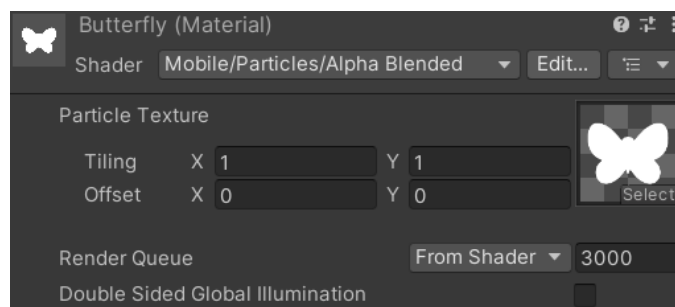


Figura 31. Sprite mariposa.

Tras tener el diseño, debemos cambiar su configuración para que actúe de la manera que deseamos. Cambiaremos el valor de la Rotación en X a -90 para que las mariposas surjan desde abajo. Habilitaremos la pestaña de “Rotation by Speed” y cambiaremos la velocidad Angular a 720. En “Shape” cambiaremos el ángulo de apertura del cañón y disminuirémos el radio a 0.1. Una de las configuraciones más importantes es el cambio de los valores en la pestaña de emisión, ya que, esto dará la impresión de que es un cañón. Después, cambiaremos la configuración de la velocidad de inicio, el tiempo inicial, el tamaño, la rotación y la gravedad haciendo que cambien de forma aleatoria entre dos valores. Como último paso, crearemos y añadiremos una paleta con 4 colores vibrantes con la misma probabilidad de aparición, es decir, cada color tiene una probabilidad del 25% (Figura 32). De esta manera aparecerán mariposas de diferentes formas, tamaños, velocidades, colores y rotaciones que caen con diferente gravedad.



Figura 32. Colores aleatorios.

Una vez diseñado se podrá colocar en todas las escenas deseadas, ajustando la posición y orientación. En la escena de inicio el confeti siempre está activo, sin embargo, en los diferentes niveles, se activa solamente cuando se detecta que todas las piezas están encajadas en el lugar correcto, por lo que ha acabado el nivel y se activa el confeti a modo de celebración. Para poder iniciar el confeti al iniciarse la aplicación se ha creado un script especialmente para ese caso y esa escena (Figura 33). Se referencia en el inspector el sistema de partículas y ese script se adjunta a un elemento presente en la escena para que al iniciarse la escena se active el cañón. Esto también es una forma de atraer la atención de los niños, motivarlos y animarlos a alcanzar el final del nivel.

```
public ParticleSystem confettiParticleSystem;

void Start()
{
    if (confettiParticleSystem != null)
    {
        confettiParticleSystem.Play();
    }
}
```

Figura 33. Script Start Confeti.

4.3.4 Plano transparente

Para que las sombras de los cubos se puedan proyectar, debemos crear un plano en cada escena en la que se desarrollen las actividades. Cada plano tendrá un “Mesh Collider” para detectar colisiones con los cubos, un material y un script asociado. En primer lugar, se ha creado un nuevo material especialmente para este plano al que en su menú de “Shader”, se ha seleccionado el “Shader” del paquete Shadow Receiver URP que se descargó anteriormente. Este, ya está creado, y lo asociaremos al material, de esta manera ahora el plano no se verá, pero sí las sombras de los elementos virtuales presentes en la escena. Además, el script asociado, restringe la posición de los elementos que interactúan con el plano en el eje Y. En otras palabras, si el cubo se desplaza por debajo del plano, automáticamente volverá a la posición 0 en el eje Y (Figura 34). Esta funcionalidad está diseñada para evitar que los niños pierdan las piezas de vista al colocarlas por debajo del nivel del suelo, que en este caso hemos definido como la mesa. De esta manera las piezas siempre estarán en un lugar visible.

```
Vector3 currentPosition = transform.position;
currentPosition.y = Mathf.Max(currentPosition.y, minY);
transform.position = currentPosition;
```

Figura 34. Script Restrict Position.

4.4 Lógica del proyecto

Una vez que todos los elementos estaban ubicados en las escenas correspondientes y configuradas las interacciones entre los paneles y las escenas, el enfoque principal se dirigió a la programación de los diferentes escenarios destinados a la realización de las actividades. Estos escenarios se dividen en: tutorial, nivel intermedio y nivel difícil. El proyecto incluye un total de siete scripts diferentes, cada uno con un propósito específico y asociado a distintas escenas. En esta sección se explicarán exclusivamente los scripts que conforman la lógica de las actividades y la generación del proyecto final.

4.4.1 Script: SnapToLocation

La lógica y comportamiento de las escenas, pertenecientes a las actividades propuestas, se desarrolla en este script. En él, se permite que los cubos se ajusten correctamente a una zona designada como correcta. Es por eso que este script está asociado a cada parte del tablero estableciendo cuales son los comportamientos esperados según las referencias añadidas en el inspector. Cuando un cubo entra en la zona de ajuste, el script determina si es el cubo correcto o incorrecto y actúa en consecuencia. Si es el cubo correcto, se ajusta a la posición y rotación que se ha referenciado a través del inspector. Si es un cubo incorrecto, volverá a la posición inicial que tenía al inicio de la escena. Este comportamiento asegura que se pueda interactuar con los objetos de manera intuitiva, garantizando que solo los cubos correctos se ajusten en las zonas designadas

Al inicio del script existen diferentes tipos de variables referenciadas. En primer lugar, existen diversas variables booleanas que controlarán los diferentes estados dentro del script. Por ejemplo, si el cubo está siendo agarrado, si ha entrado en zona de ajuste o si el objeto ha sido ajustado correctamente. Hay 3 objetos referenciados, el cubo correcto que pertenece a esa parte del tablero, la rotación que adaptará cada cubo cuando se pegue y una lista de los cubos incorrectos que podrían entrar en contacto con esa zona. Después hay otras variables que almacenan la posición, rotación, el conjunto de cubos incorrectos y la capa en la que se encuentran solo las piezas.

Existen 3 funciones clave en este Script. Dos métodos “OnTrigger” y un método “SnapObject”. Uno de los métodos “OnTrigger” se activa cuando un objeto sale del área de colisiones mientras que el segundo se activa cuando entra en el área establecida. Verificará si los dos componentes se encuentran en la misma capa y después hará las siguientes comprobaciones:

- Si un cubo es incorrecto se añadirá al conjunto de cubos incorrectos.
- Si es correcto, se establecen las variables booleanas “CorrectSnap” y “Snapped” a verdadero.
- Si se diera el caso en el que se choca con un cubo correcto e incorrecto a la vez, se priorizará el correcto. Esta casuística se debe a que cada pieza tiene su propio “Collider” por lo que muchas veces hay colisiones accidentalmente simultaneas.

Estos dos métodos se complementan y hacen exactamente lo contrario uno del otro ya que uno pertenece a la entrada y otro a la salida.

El método “SnapObject” es el que ajusta la posición y rotación de los diferentes cubos si se cumplen las condiciones. Existen 3 posibilidades reflejadas en rojo en la Figura 35. La primera de ellas es que el cubo no entre en contacto con la zona de ajustes por lo que el objeto seguirá siendo cinemático. La segunda casuística ocurre cuando se entra en la zona de ajustes y aquí se divide en dos posibilidades:

- Si en el “OnTriggerEnter” se determina que está en la zona correcta, el cubo se anclará con la rotación correcta al tablero.
- Si se determina que se han añadido cubos incorrectos al array porque han entrado en la zona incorrecta, estos cubos volverán a su posición inicial.

En el método “Update” se estará comprobando en cada fotograma a través del “OVRGrabbable” asignado al cubo si está siendo agarrado. De esta manera se asegura que los datos del movimiento detectado son en tiempo real y por tanto la lógica será más precisa.

```
void SnapObject()
{
    if (!grabbed)
    {
        if (insideSnapZone)
        {
            if (correctSnap)
            {
                SnapPart.transform.position = transform.position;
                SnapPart.transform.rotation = SnapRotationReference.transform.rotation;
            }
            else
            {
                foreach (GameObject cube in IncorrectCubes)
                {
                    if (incorrectCubesInContact.Contains(cube))
                    {
                        cube.transform.position = initialPositionsIncorrect[cube];
                        cube.transform.rotation = initialRotationsIncorrect[cube];
                    }
                }
            }
        }
        else
        {
            snapPartRigidbody.isKinematic = true;
        }
    }
}
```

Figura 35. Método principal de SnapToLocation.

4.4.2 Script: SnapObject

Este Script está relacionado con el script anterior. Es decir, se comunica con el “SnapToLocation” para solicitar datos específicos. Está asociado a cada cubo de la escena, y en el inspector se referencia la zona de anclaje del cubo al tablero. Una vez referenciado, la función “Update” del script se encarga de recibir información en tiempo real sobre si el cubo se ha adherido a su zona designada. Cuando se detecta que el cubo está anclado correctamente, se actualiza el estado de la variable booleana “isSnapped” a verdadero. Esta variable es pública ya que su valor será utilizado en el siguiente script llamado “AllSnapped”.

Además, es el encargado de incrementar un contador que pertenece al número de errores que se cometen durante la escena cada vez que un cubo vuelve a su posición y rotación inicial. Ya que el script está asociado a cada cubo, para cada uno de ellos, en el método “Start” se guardará la posición y rotación inicial. Si se detecta que un cubo abandona su posición inicial, un booleano pasará a verdadero y cuando se detecte que ha vuelto dicha posición inicial después de abandonarla, se llamará al “GameManager” para incrementar el contador (Figura 36). Este contador no será visible en las escenas ya que su valor se verá reflejado en un fichero que se guardará en la memoria interna de las gafas para su posterior consulta por parte del profesional.

Este diseño de scripts que se comunican en cascada permite que cada cubo se comporte según lo previsto y que el estado de los cambios se mantenga actualizado en todo momento. Garantizando que cada componente interactúe correctamente y de forma fluida.

```
if (!hasLeftInitialPosition && (transform.position != initialPosition || transform.rotation != initialRotation))
{
    hasLeftInitialPosition = true;
}

if (hasLeftInitialPosition && transform.position == initialPosition && transform.rotation == initialRotation)
{
    GameManager.Instance.IncrementErrorCount();
    hasLeftInitialPosition = false;
}
```

Figura 36. Contador de errores.

4.4.3 Script: AllSnapped

Este script es el último nivel de la cadena de comunicación entre los diferentes scripts. En el inspector, se referencian todos los cubos de la escena en la que se encuentre y, a partir de estos cubos referenciados en el script, se obtiene el valor booleano público de “isSnapped”.

Se ha desarrollado una función que verifica si todos los cubos están correctamente anclados (Figura 37). Itera a través de cada cubo, comprobando el estado de la propiedad “isSnapped” de cada uno. Si encuentra algún cubo que no está anclado, la función devuelve falso. Si todos los cubos están anclados, la función devuelve verdadero. En otra función distinta, si se detecta que este valor devuelto es verdadero, se activa el panel correspondiente y se dispara el cañón de confeti.

```
private bool rocketLaunch(){
    for(int i=0; i< cubos.Length; i++){
        partAdded = cubos[i].GetComponent<SnapObject>().isSnapped;
        if (partAdded == false){
            return false;
        }
    }
    return true;
}
```

Figura 37. Método booleano de AllSnapped.

Este script contempla dos escenarios diferentes dentro del mismo. El primer escenario corresponde al juego intermedio, donde se activa un panel que pregunta si se desea continuar al siguiente nivel. El segundo escenario corresponde al nivel difícil. En este caso, a través del “SceneManager” se obtiene el “index” y se verifica si nos encontramos en la escena del nivel difícil y, de ser así, en lugar de activar un panel que solicita una decisión, se carga automáticamente la siguiente escena que pertenece al menú principal de la aplicación.

Este script es esencial para determinar si el nivel actual está completado o si se debe permanecer en él. Además, al cambiar de escena, se detienen y guardan tanto el tiempo transcurrido como los errores cometidos en el archivo correspondiente. Esto garantiza un seguimiento preciso del progreso y un registro detallado de la actividad, contribuyendo a una experiencia de usuario más organizada y fluida.

4.4.4 Generación apk

Una vez que todo el proyecto esté listo para su uso, el paso final es exportarlo y generar el archivo APK de la aplicación. El archivo .apk (Android Package) es el formato utilizado para distribuir e instalar aplicaciones en dispositivos Android, como las gafas Oculus Quest 2. Dentro de estos archivos se compila el código fuente y empaqueta todos los elementos en uno solo.

Para comenzar, se abre el menú "File" y después "Build Settings". En esta ventana, se añadirán todas las escenas que se quieren incluir en el archivo .apk [17]. Es por eso que todas las escenas que estarán en el archivo deben ser seleccionadas, ordenadas y enumeradas. A continuación, se configuran las "Meta Quest 2" como el dispositivo de destino y se verifica que la plataforma de desarrollo es Android (Figura 38).

Para poder compilarlo, existen dos maneras diferentes. La primera es realizar un "Build" que generará el archivo .apk y posteriormente subirlo a las gafas Meta Quest 2 a través de una aplicación por ejemplo o realizar un "Build and Run" para que se suban automáticamente. En los dos casos, se abrirá una ventana para seleccionar la ubicación donde se guardará el archivo .apk.

Elegiremos la carpeta dentro del proyecto de Unity que pertenece al proyecto para guardarlo. Este paso asegura que el proyecto sea ejecutable en las gafas Oculus Quest 2.

El tiempo que tome este proceso puede variar dependiendo de las especificaciones del ordenador utilizado. Una vez completada la generación del archivo .apk, se podrán iniciar las gafas Oculus Quest 2. En el caso de este proyecto, se ha optado por subir el .apk directamente por lo que, al seleccionar el icono de Unity en el menú de aplicaciones de las gafas, la aplicación se ejecutará, permitiendo así experimentar el proyecto en un entorno de realidad mixta.

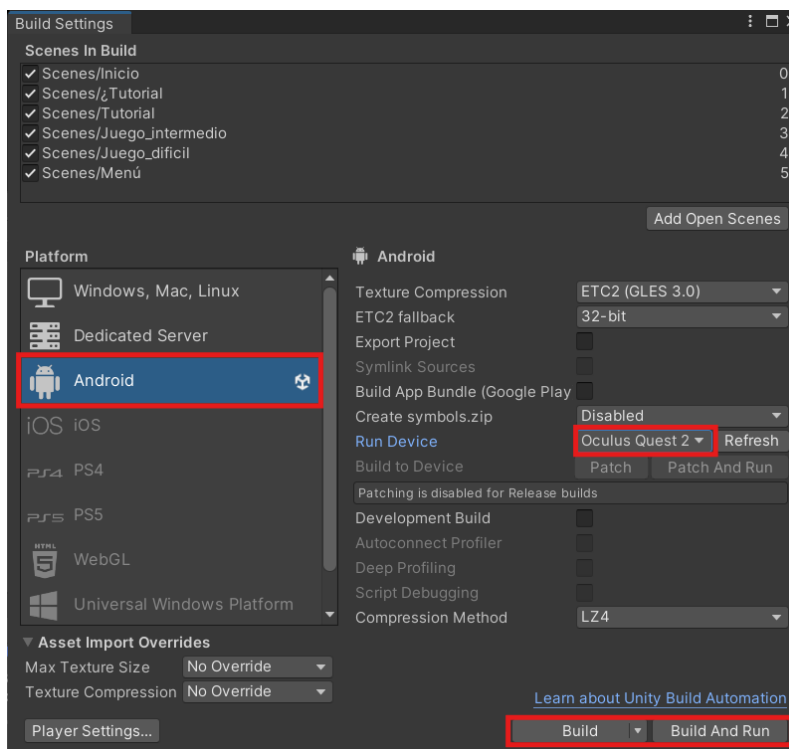


Figura 38. Build Settings.

4.5 Configuración por parte del profesional

Antes de comenzar con la sesión, el profesional debe asegurarse de que todo el material esté listo, incluyendo las gafas Meta Quest 2 completamente cargadas y con suficiente espacio de almacenamiento disponible. Además, es necesario realizar una configuración inicial de las gafas para que la aplicación funcione correctamente.

Primero, el profesional debe ponerse las gafas y sentarse en una silla dentro de la sala donde se realizarán las sesiones. Dado que Meta Quest 2 no ofrece detección automática de superficies, es necesario establecer manualmente la mesa de la sala como el suelo virtual. Para hacer esto, el se debe seguir las instrucciones proporcionadas por el dispositivo.

Otro aspecto crucial de la configuración es la creación de un límite en la sala. Meta Quest 2 ofrece dos opciones para establecer este límite: una opción automática, que crea un círculo alrededor del usuario, y una opción personalizable. Para crear un límite personalizado, el profesional usará uno de los mandos de las gafas. Presionando el botón A, puede dibujar el área que define el espacio seguro, asegurando así que el niño permanezca dentro de ese perímetro seguro durante la sesión.

Una vez que el entorno físico y virtual esté configurado, el profesional debe iniciar la aplicación. Esto se hace para que, cuando el niño se ponga las gafas, ya se encuentre directamente con el primer panel de la aplicación, listo para comenzar la actividad. Esto minimizará el tiempo de espera y asegurará que la transición al entorno virtual sea suave.

4.5.1 Feedback obtenido

Tras analizar los resultados visibles del proyecto, también debemos considerar aquellos que no son evidentes a simple vista. Es fundamental que el profesional reciba retroalimentación continua del proyecto para evaluar si se han logrado mejoras, además de las observadas bajo su supervisión directa. Por esta razón, cada vez que se inicia la aplicación, se genera automáticamente un archivo nuevo cuyo nombre corresponde a la fecha y hora del inicio (Figura 39). Este archivo contiene datos importantes como: la fecha y hora de ingreso a cada escena, el tiempo que se permanece en ella, y los errores cometidos durante la misma.

```
private void SetNameOfExitFile()
{
    fileName = DateTime.Now.ToString("dd-MM-yyyy_HH:mm:ss") + ".txt";
}
```

Figura 39. Nombre del archivo de salida.

Estos datos, aunque simples, proporcionan información valiosa para que el profesional pueda evaluar el progreso y desarrollar planes de mejora. En el archivo se registran todas las escenas visitadas, y al finalizar la sesión, se guarda el archivo definitivo. La próxima vez que se abra la aplicación, se generará un nuevo archivo. El profesional puede llevar un registro en su agenda de qué niño tiene programado en cada sesión horaria y buscar el archivo correspondiente a esa fecha y hora para su revisión.

Para lograr esto, se ha añadido un “GameObject” vacío en la escena al que se le ha asignado un script llamado “GameManager” que se inicializa en la escena del tutorial y que se mantiene activo durante toda la aplicación, guardando la información de cada escena al salir de ella.

Es decir, no se destruye. Al iniciarse el “Scenemanager” detectará en que escena nos encontramos y si cambiamos de escena, detectará que ya no es la misma por lo que se parará el tiempo y se guardarán todos los datos y al empezar una nueva escena el contador y el tiempo se resetean (Figura 40).

```
private void Start()
{
    SceneManager.activeSceneChanged += OnActiveSceneChanged;
    currentSceneName = SceneManager.GetActiveScene().name;
    startTime = Time.time;
}

private void OnActiveSceneChanged(Scene previousScene, Scene newScene)
{
    if (newScene.name != currentSceneName)
    {
        EndScene(currentSceneName);
        currentSceneName = newScene.name;
        startTime = Time.time;
        errorCount = 0;
    }
}
```

Figura 40. Detección de escenario.

Todos estos datos se irán añadiendo de manera continua al archivo existente hasta que se detenga la aplicación. De esta forma, el archivo final será un documento de texto que contendrá tanta información como escenarios haya recorrido el niño. Además, cada escena estará dividida por una línea discontinua para una mayor claridad de la información (Figura 41).

```
Escena: Tutorial
Fecha: 23-06-2024
Errores: 2
Tiempo en la escena: 18s
-----
Escena: Juego_intermedio
Fecha: 23-06-2024
Errores: 1
Tiempo en la escena: 16s
-----
Escena: Juego_dificil
Fecha: 23-06-2024
Errores: 3
Tiempo en la escena: 30s
-----
```

Figura 41. Información del archivo final.

Este sistema de guardado de la aplicación se realiza en la propia memoria interna de las Meta Quest 2. Estas gafas cuentan con su propia capacidad de almacenamiento, a la que podemos acceder para guardar los archivos generados.

Al iniciarse la aplicación, si aún no existe, se creará automáticamente una carpeta denominada “Colormatic”, que es el nombre asignado al proyecto. En esta carpeta se irán añadiendo los archivos mencionados anteriormente. El proceso de almacenamiento se logra gracias al uso del “Application.persistentDataPath” [18], un directorio público dentro de las gafas. Para visualizar estos archivos, el primer paso es conectar las gafas al ordenador mediante un cable USB-C. Luego, dentro de las Meta Quest 2, es necesario habilitar la opción de visualizar archivos. Dado que nuestro proyecto está desarrollado en Android, la carpeta “Colormatic” se encontrará almacenada en la estructura de archivos del dispositivo.

Para poder acceder a la carpeta hay que seguir un recorrido específico: Abrir la carpeta de memoria interna de las gafas > Android > data > com.DefaultCompany.TFG-Colormatic > files > Sesiones-Colormatic (Figura 42). Esta forma de guardado, asegura que toda la información relevante se registre de manera efectiva y precisa. La información guardada en este directorio no se borra a menos que alguien lo haga deliberadamente, lo que garantiza su seguridad. Esta implementación no solo facilita el acceso y revisión de los datos por parte del profesional, sino que también asegura la disponibilidad en todo momento de la información para futuras evaluaciones.

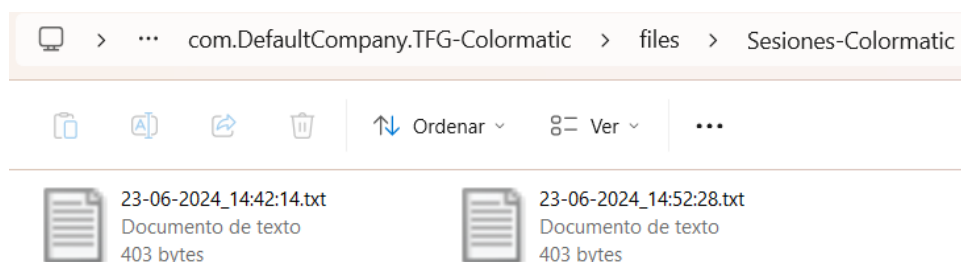


Figura 42. Ruta de guardado.

Por último, existe otro tipo de información para el profesional en tiempo real. Durante las sesiones, el profesional puede monitorear en tiempo real las acciones realizadas por el niño dentro de la actividad. Esto es posible gracias a la retransmisión en directo desde las gafas Meta Quest 2 al dispositivo móvil mediante la aplicación de Meta Quest. Es imprescindible que el profesional tenga instalada la aplicación en su dispositivo móvil y acceda con su cuenta de Meta. Para lograr esta conexión, tanto las gafas como el dispositivo móvil deben contar con acceso a internet. Además, el dispositivo móvil debe tener el Bluetooth activado para establecer la conexión con las gafas. En el dispositivo móvil, simplemente se debe presionar el botón de “transmitir” (Figura 43) para visualizar en tiempo real lo que el niño está viendo a través de las gafas. Esto permite al profesional comunicarse con el niño, proporcionándole indicaciones y consejos para facilitar su avance en las actividades.



Figura 43. Aplicación Meta Quest del móvil.

Capítulo 5. Resultados obtenidos

A lo largo de este proyecto enfocado en el desarrollo e implementación de una aplicación de realidad mixta diseñada para beneficiar y apoyar el desarrollo cognitivo de niños con problemas de aprendizaje se presentaron varias adversidades. Dado que es un proyecto enfocado a los niños, el principal problema radicaba en comprobar la efectividad y usabilidad de esta aplicación para ellos. Gracias a la colaboración con el centro Grupo Estímulo de Torrent (Valencia), se ha podido comprobar si los objetivos propuestos para este proyecto han sido alcanzados.

En primer lugar, los datos recopilados se basan en las observaciones directas por parte de los profesionales del centro y un análisis de usabilidad de los mismos. Uno de los primeros pasos fue enseñar como utilizar las gafas Meta Quest 2 y el concepto de realidad mixta. Dado que es una tecnología aún en desarrollo y debido a la innovación tecnológica desigual, es un concepto que no se suele conocer en el ámbito de la educación. Es por eso que la aplicación ha sido desarrollada y diseñada para ser accesible y fácil de usar. Así, se asegura la accesibilidad para los niños. Además, está enfocada para que funcione de manera autónoma en las gafas Meta Quest 2.

En segundo lugar, se realizó una evaluación de las características de la aplicación. Las actividades propuestas en este proyecto fueron diseñadas para ser tanto educativas como entretenidas. Es por eso que el niño tomará esta actividad como un juego más al que podrá referirse con el nombre de “Colormatic”. Todo ello para mantener el interés y la participación de los niños. La capacidad de la aplicación para adaptarse a las necesidades individuales de los niños fue otro objetivo clave. La personalización incluyó ajustes en el nivel de dificultad de los ejercicios según la edad y las habilidades de cada uno. Por otra parte, no se realiza un uso prolongado de esta tecnología en las consultas, lo que no acarrea efectos negativos. Estas consultas especializadas de los centros suelen tener una duración aproximada de 45 minutos, en los cuales se realizan diferentes actividades durante ese periodo. Por eso, la aplicación se usa como máximo 15 minutos por consulta. Lo que conlleva cumplir con los estándares de seguridad y el bienestar de los niños.

Por último, los profesionales destacaron la facilidad de integración de la tecnología en sus programas educativos. Afirmando que se trata de una herramienta viable, beneficiosa y eficaz. Todos estos elementos hacen posible su integración en las consultas y proporciona una nueva alternativa a las sesiones tradicionales. Todos estos resultados obtenidos son respaldados por un informe redactado por parte del centro Grupo Estímulo de Torrent (Valencia) que se encuentra en el Anexo II.

Capítulo 6. Conclusiones y propuesta de trabajo futuro

El desarrollo de esta aplicación de realidad mixta diseñada para el desarrollo cognitivo de niños con problemas de aprendizaje ha resultado ser un proyecto prometedor. Con este proyecto se ha comprobado que la tecnología tiene un gran potencial en la educación especializada. Las evaluaciones que han realizado los profesionales al proyecto han demostrado que los ejercicios interactivos desarrollados con Unity son atractivos y motivadores para los niños. La combinación de entretenimiento y educación ha facilitado un aprendizaje más activo y comprometido. Esta aplicación es capaz de adaptarse a las necesidades individuales de los niños y es por esa capacidad de personalización, según la edad o el nivel de cada niño, lo que abre la puerta a su implementación en las consultas. Esta adaptabilidad ha sido uno de los objetivos más destacados del proyecto ya que no se limita a un solo grupo. La posibilidad de integrar la realidad mixta como complemento, o incluso sustituto, de las consultas tradicionales es una alternativa innovadora y efectiva.

Cabe destacar que esta tecnología sigue en desarrollo y este proyecto es un prototipo. Con una inversión en la investigación de esta área podría llegar a ser una realidad del día a día en las consultas especializadas. La alta satisfacción y utilidad que han manifestado los profesionales del proyecto, respalda la posibilidad de continuar desarrollando y utilizando esta tecnología en contextos educativos. De esta manera, en un futuro, la tecnología y la educación se unirán para crear experiencias de aprendizaje más inclusivas para todos los niños.

Aunque los resultados obtenidos son prometedores, este proyecto solo es el comienzo. A partir de estos resultados, se podrían presentar varias propuestas de futuro para continuar avanzando en esta área.

La primera de ellas sería ampliar el campo de estudio con niños que presenten diferentes dificultades de aprendizaje, edades o problemas diversos. Por otro lado, se podría seguir desarrollando nuevos contenidos dentro de la aplicación. Crear nuevos ejercicios y juegos interactivos que aborden una mayor variedad de habilidades cognitivas y educativas. Algunos ejercicios nuevos propuestos por los profesionales han sido crear escenarios en los que se pueda trabajar la frustración y añadir, dentro de las escenas existentes, elementos virtuales para poder trabajar la distracción.

La segunda, está relacionada con la integración de las tecnologías en el ámbito de la educación. Explorar esta integración de la realidad mixta junto con otras tecnologías educativas, y fomentar la colaboración entre los desarrolladores de tecnología, educadores, terapeutas y especialistas en desarrollo infantil. Además, una formación de los profesionales con estas nuevas tecnologías es importante para que puedan utilizarla eficazmente en sus sesiones diarias y sean capaces de saber explicárselo a los pacientes. Otro punto importante respecto a esta propuesta es comprobar el coste-beneficio de implementar estas tecnologías a gran escala en instituciones educativas y terapéuticas.

Capítulo 7. Bibliografía

- [1] Quintero, J., Baldiris, S., Rubira, R., Cerón, J., & Velez, G. (2019). Augmented Reality in Educational Inclusion. A Systematic Review on the Last Decade. *Frontiers In Psychology*, 10. <https://doi.org/10.3389/fpsyg.2019.01835>
- [2] Gómez, O. y. A. (2018). Las TIC como herramientas cognitivas. <https://www.redalyc.org/journal/5610/561059324005/html/>
- [3] Emprendiendo, T. (2024b, mayo 16). Descubre la evolución de la realidad mixta frente a la realidad aumentada. Triunfa Emprendiendo. <https://triunfaemprendiendo.com/como-se-expande-la-realidad-mixta-sobre-la-realidad-aumentada/>
- [4] Equipo editorial de IONOS. (2020, 29 septiembre). ¿Qué es la realidad mixta? IONOS Digital Guide. <https://www.ionos.mx/digitalguide/online-marketing/vender-en-internet/realidad-mixta/>
- [5] García, J. (2020b, octubre 17). Oculus Quest 2, análisis: una de las mejores (y asequibles) opciones para iniciarse en la realidad. . . Xataka. <https://www.xataka.com/analisis/oculus-quest-2-analisis-caracteristicas-precio-especificaciones>
- [6] Acerca de Blender — Blender manual. (s. f.). https://docs.blender.org/manual/es/4.0/getting_started/about/index.html
- [7] Fernández, Y. (2023b, junio 9). Qué es Canva, cómo funciona y cómo usarlo para crear un diseño. Xataka. <https://www.xataka.com/basics/que-canva-como-funciona-como-usarlo-para-crear-diseno>
- [8] Redaccion. (2022, 5 febrero). Conecta tus Meta Quest 2 al PC con Link en 2022. Alehandoro VR. <https://alehandorovr.com/conecta-tus-meta-quest-2-al-pc-con-link-en-2022/>
- [9] Líderes en formación tecnológica, reskilling y upskilling | OpenWebinars. (s. f.). OpenWebinars.net. <https://openwebinars.net/blog/que-es-unity/>
- [10] ¿Quieres saber qué es Unity? Conoce nuestra historia y nuestros planes para el futuro | [site:name]. (s. f.). Unity. <https://unity.com/es/our-company>
- [11] Meta XR All-in-One SDK | Integration | Unity Asset Store. (2023b, noviembre 15). Unity Asset Store. <https://assetstore.unity.com/packages/tools/integration/meta-xr-all-in-one-sdk-269657>



[12] Shadows in the Universal Render Pipeline | Universal RP | 12.0.0. (s. f.-b).
<https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@12.0/manual/Shadows-in-URP.html>

[13] Technologies, U. (s. f.-a). Unity - Manual: Guía para exportar FBX.
<https://docs.unity3d.com/es/530/Manual/HOWTO-exportFBX.html>

[14] Get Started with Passthrough | Oculus Developers. (2022). Oculus.com.
<https://developer.oculus.com/documentation/unity/unity-passthrough-gs/>

[15] Scroll View. (s. f.). <https://www.foundations.unity.com/components/scroll-view>

[16] Technologies, U. (s. f.). Utilizando Sistemas de Partículas en Unity - Unity Manual.
<https://docs.unity3d.com/es/2018.4/Manual/ParticleSystemUsage.html>

[17] Technologies, U. (s. f.-a). Unity - Manual: Build your application for Android.
<https://docs.unity3d.com/Manual/android-BuildProcess.html>

[18] Technologies, U. (s. f.-a). Unity - Scripting API: Application.PersistentDataPath.
<https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html>

Capítulo 8. Anexos

8.1 Anexo I: Manual de uso de Meta Quest Link

Para poder establecer la conexión entre el ordenador y las gafas Meta Quest 2 y comenzar a desarrollar un proyecto con Unity, es necesario tener la aplicación de Meta Quest Link instalada en el ordenador. Esta aplicación permite conectar dispositivos de Meta a una red Wi-Fi y cuenta de Meta.

El primer paso a realizar es descargar esta aplicación desde la página oficial de Meta: [Install the Oculus PC app for Meta Quest Link | Meta Store](#). Después:

1. La aplicación comenzará a descargarse. Abre la carpeta de descargas y haz clic en el ejecutable.
2. Comenzará la instalación. Selecciona la carpeta dentro del ordenador donde quieras guardarlo y pulsa instalar.
3. Una vez se haya instalado, se podrá abrir la aplicación. Para poder acceder a ella es necesario tener una cuenta creada en Meta.

Dentro de la aplicación, para poder usar Unity es necesario activar las configuraciones de desarrollador, en la pestaña de Configuración > General deberemos activar algunas funcionalidades (Figura 44).

1. Orígenes desconocidos para permitir ejecutarse dentro de las gafas el programa de Unity, ya que es una aplicación que no está revisada por Meta (desconocido).
2. Entorno de ejecución OpenXR.



Figura 44. Configuración general.

Además, en la pestaña de Beta activaremos también la pestaña que corresponde a las funciones para desarrolladores. Para establecer una conexión entre el ordenador y las gafas, existen dos maneras. Utilizar un “Cable Link” o utilizar el “Air Link”. El “Air Link” establece una conexión a través del Wi-Fi mientras que el “Cable Link” establece la conexión entre los dispositivos mediante un cable USB-C.

En este caso, usaremos el cable USB-C para establecer la conexión. En el apartado de “Dispositivos” aparecerá el tipo de dispositivo conectado. Si hubiera algún problema de conexión se vería reflejado en este apartado. El icono verde al lado del dispositivo nos informa de que la conexión se ha realizado con éxito (Figura 45).

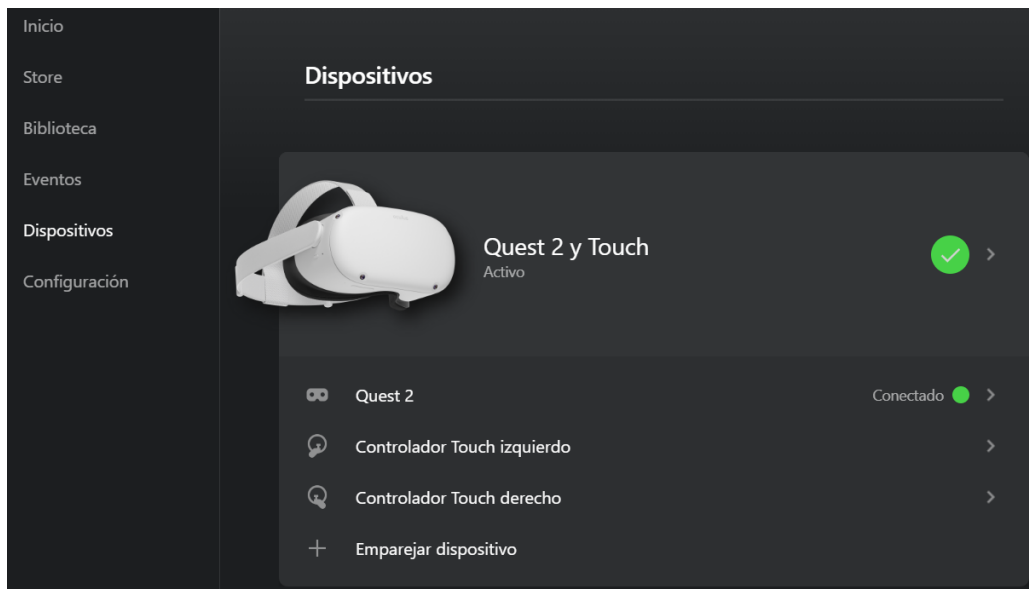


Figura 45. Dispositivo conectado.



8.2 Anexo II: Informe del Centro Grupo Estímulo

Doña MARIA AMELIA CARRATALÁ MORENO, con DNI 53363483-X y número de colegiada CV 10805, actuando como profesional y como Directora del Centro **GRUPO ESTÍMULO TORRENT**, situado en Avenida al Vedat, número 101B, puerta 2 de Torrent (Valencia), declara:

Que tras probar las gafas del proyecto “Desarrollo de una aplicación de realidad mixta para la estimulación cognitiva infantil” de Denisa Marcela Fat consideramos muy interesante y útil la propuesta de actividad, ya que con ese juego se puede trabajar la atención selectiva y sostenida, la planificación y organización, así como la orientación espacial. Además, el hecho que hayan niveles ayuda a poder ajustar la dificultad en función de las limitaciones de los/as niños/as y a diferentes edades.

En las dinámicas de intervención del Centro priorizamos la motivación de los/as niños/as para lograr aprendizajes significativos y consideramos una herramienta muy interesante en el trabajo diario el Centro.

Y para que así conste, firmo el siguiente documento, en Torrent a 17 de junio de 2024.



GRUPO ESTIMULO
TORRENT S.L.U.
C.I.F. B-16371428
AVD. AL VEDAT Nº 101B. PTA. 2
46900 TORRENT (Valencia)

Fdo: María Amelia Carratalá Moreno
Directora del Centro