



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

DSIC
DEPARTAMENT DE SISTEMES
INFORMÀTICS I COMPUTACIÓ

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Sistemas Informáticos y Computación

Aplicación de realidad mixta en Quest 3 para el desarrollo y mejora de la inteligencia espacial en la rehabilitación de personas con daño cerebral adquirido.

Trabajo Fin de Máster

Máster Universitario en Ingeniería y Tecnología de Sistemas
Software

AUTOR/A: Sarabia Chamero, Jaime

Tutor/a: Jaén Martínez, Francisco Javier

CURSO ACADÉMICO: 2023/2024

Este Trabajo Fin de Máster se ha depositado en el Departamento de Sistemas Informàticos y Computaci3n de la Universitat Politècnica de València para su defensa.

Trabajo Fin de Máster

Máster Universitario en Ingeniería y Tecnología de Sistemas Software

Título: Aplicaci3n de realidad mixta en Quest 3 para el desarrollo y mejora de la inteligencia espacial en la rehabilitaci3n de personas con daño cerebral adquirido

«09-2024»

Autor(a): Jaime Sarabia Chamero

Director(a): Francisco Javier Jaén Martínez

Departamento de Sistemas Informàticos y Computaci3n
Universitat Politècnica de València

Resum

El projecte està enfocat en el desenvolupament d'una aplicació de Realitat Mixta (MR) amb la finalitat d'ajudar a la rehabilitació de persones amb dany cerebral adquirit. L'aplicació serà desenvolupada per al nou dispositiu de Meta: Meta Quest 3. Amb aquest dispositiu i la seua tecnologia Depth API, serem capaços de desenvolupar aplicacions que integren amb una precisió molt detallada elements virtuals dins del món físic, fent així més amigable l'ús per a aquest tipus de pacients. La implementació consistirà en el desenvolupament d'una aplicació en Unity, al costat de l'entorn de desenvolupador atorgat per Meta i la Depth API de les noves Meta Quest 3.

Resumen

El proyecto está enfocado en el desarrollo de una aplicación de Realidad Mixta (MR) con la finalidad de ayudar a la rehabilitación de personas con daño cerebral adquirido. La aplicación será desarrollada para el nuevo dispositivo de Meta: Meta Quest 3. Con este dispositivo y su tecnología Depth API, seremos capaces de desarrollar aplicaciones que integren con una precisión muy detallada elementos virtuales dentro del mundo físico, haciendo así más amigable el uso para este tipo de pacientes. La implementación consistirá en el desarrollo de una aplicación en Unity, junto al entorno de desarrollador otorgado por Meta y la Depth API de las nuevas Meta Quest 3.

Abstract

The project is focused on the development of a Mixed Reality (MR) application aimed at assisting in the rehabilitation of individuals with acquired brain injury. The application will be developed for Meta's new device: Meta Quest 3. With this device and its Depth API technology, we will be able to develop applications that integrate virtual elements into the physical world with very detailed accuracy, thus making it more user-friendly for this type of patients. The implementation will consist of developing an application in Unity, alongside the developer environment provided by Meta and the Depth API of the new Meta Quest 3.

Glosario

Acrónimos

- **API** Application Programming Interface: conjunto de reglas o protocolos que permite a las aplicaciones informáticas comunicarse entre sí para intercambiar datos, características y funcionalidades.
- **APK** Android Application Package: archivo ejecutable que contiene todos los datos que se necesitan para instalar y hacer funcionar una aplicación Android.
- **AR** Augmented Reality: es el término que se usa para describir al conjunto de tecnologías que permiten que un usuario visualice parte del mundo real a través de un dispositivo tecnológico con información gráfica añadida por este.
- **MR** Mixed Reality: es la combinación de realidad virtual y realidad aumentada. Esta combinación permite crear nuevos espacios en los que interactúan tanto objetos y/o personas reales como virtuales.
- **VR** Virtual Reality: es un entorno generado mediante tecnología informática, que crea en el usuario la sensación de estar inmerso en él.
- **XR** Extended Reality: un término general que abarca todas las tecnologías inmersivas, incluidas la realidad virtual (VR), la realidad aumentada (AR) y la realidad mixta (MR), que combinan los mundos físico y digital.
- **MRUK** Mixed Reality Utility Kit: es una herramienta transformadora diseñada para agilizar el desarrollo de aplicaciones de realidad mixta (MR) para Meta Quest.
- **SUS** System Usability Scale: es una herramienta de evaluación de usabilidad que se utiliza para medir la facilidad de uso de un sistema, producto o servicio.
- **TLX** Task Load Index: es una herramienta de evaluación diseñada por la NASA que califica la carga de trabajo percibida con el fin de evaluar la efectividad de una tarea, sistema o equipo, u otros aspectos del rendimiento.

Tabla de contenidos

1. Introducción	7
1.1. Motivación	7
1.2. Objetivos	8
1.3. Estructura	8
1.4. Relación con otros estudios	9
2. Contexto del trabajo	10
2.1. Herramientas de Realidad Extendida	11
2.2. La Realidad Extendida en la Medicina	12
2.3. Realidad Virtual, Realidad Aumentada y Realidad Mixta	13
2.4. Meta, Apple y actualidad	14
3. Diseño e implementación	16
3.1. Método de trabajo	16
3.2. Estudio de viabilidad	19
3.2.1. Estudio de las herramientas a utilizar	19
3.2.2. Desarrollo de la aplicación piloto	20
3.2.3. Pruebas	20
3.3. Desarrollo de la aplicación final	21
3.3.1. Modelado Conceptual	21
3.3.2. Diseño	23
3.3.3. Interfaz de usuario	26
3.4. Entorno de trabajo	28
3.4.1. Herramientas	28
3.4.1.1. Ordenador y Windows 11	28
3.4.1.2. Meta Quest 3	28
3.4.2. Software	28
3.4.2.1. Unity y Oculus MRUK	28
3.4.3. Meta Horizon	30
3.4.4. Visual Studio	31
3.4.5. NASA TLX	31
4. Resultados	33
4.1. System Usability Scale	33
4.1.0.1. Resultados SUS	34
4.1.1. NASA TLX	35

4.1.1.1. Resultados NASA TLX	36
4.2. Comparación de resultados	37
5. Conclusiones y trabajos futuros	38
5.1. Futuras mejoras	39
Bibliografía	41
Anexo I: Código C#	42
Anexo II: Cuestionarios usabilidad y carga de trabajo	46

Capítulo 1

Introducción

En los últimos años, gracias al avance tanto de la medicina por un lado, como de las tecnologías de realidad extendida, se han podido desarrollar numerosas técnicas que permiten mejorar la rehabilitación de personas que han sufrido cualquier tipo de problema, accidente o enfermedad, ya sea crónica o repentina. Estas técnicas siempre deben unificar el conocimiento de médicos, en cuanto a qué debe hacer el paciente, con el de los ingenieros de desarrollo, que deben poner en práctica esas peticiones del médico.

En este trabajo, se tratará en concreto el problema que se les genera a los pacientes que sufren algún tipo de daño cerebral y deben realizar un tratamiento de rehabilitación para recuperar ciertas funciones motoras o psíquicas que su cuerpo o bien ha olvidado, o bien ha sufrido un deterioro por el cual deben volver a aprenderlas.

1.1. Motivación

El avance de las tecnologías de realidad extendida ha supuesto una nueva ventana para todo tipo de avances en diferentes sectores. Hasta ahora, los usuarios necesitados de una rehabilitación de cualquier tipo, comúnmente debían acudir a espacios diseñados para tratar esta clase de problemas. En cambio, ahora, de la mano de estas tecnologías, es posible diseñar actividades enfocadas a la rehabilitación de estos usuarios para que pueda ser realizada en la comodidad del hogar y midiendo con precisión los posibles avances realizados por el paciente con métricas precisas.

Es por esto que este proyecto plantea una aplicación multidisciplinar que ayude tanto a rehabilitadores como a pacientes a crear y realizar, respectivamente, unas tareas de rehabilitación óptimas, con la capacidad de ser realizadas sin la supervisión de un técnico una vez explicadas al paciente, todo esto utilizando la última de las tecnologías en este sector para asegurar una fidelidad con la realidad máxima, haciendo que las actividades realizadas y progresos obtenidos dentro de la aplicación se ajusten lo máximo posible a la



Figura 1.1: Las tareas de rehabilitación neuropsicológicas suelen estar relacionadas con el movimiento de objetos pequeños y tareas de precisión

realidad para que estos avances se trasladen a la mejoría en las capacidades del paciente.

1.2. Objetivos

Por tanto, los objetivos de este Trabajo de Fin de Máster son:

- Desarrollar un aplicación de realidad mixta en Quest 3.
- Generar dos tipos de perfiles de trabajo en la aplicación: rehabilitador y paciente.
- Permitir el uso de la aplicación sin necesidad de los mandos propios de las gafas Quest 3, solo con las manos para asemejarse en mayor medida a la realidad en el perfil de paciente.
- Demostrar la aplicabilidad del trabajo en la vida real.
- Evaluar la usabilidad de la aplicación desarrollada así como la carga de trabajo de los usuarios que la utilicen.

1.3. Estructura

Este proyecto consta de 5 secciones. La primera de ellas presenta la introducción al proyecto, junto a las motivaciones que han llevado a comenzararlo y los objetivos que pretende cumplir, así como su relación con estudios similares.

La segunda sección trata el contexto actual de las tecnologías utilizadas en este estudio en el pasado y la actualidad, y su posibles aplicaciones en el ámbito de la medicina.

La tercera sección tiene el objetivo de documentar y explicar todas las he-

Introducción

ramientas utilizadas y decisiones tomadas durante el desarrollo de la aplicación, así como un repaso acerca de los elementos que tendrá la aplicación desarrollada, sus casos de uso e interfaces. También contendrá detalles sobre las pruebas realizadas a los sujetos utilizando la aplicación y como evaluarán esta.

La cuarta sección constará de los resultados de estas pruebas y un estudio sobre estos, comparando los resultados siguiendo diversos criterios para la posterior conclusión.

La quinta sección tratará las conclusiones acerca de los resultados de las pruebas así como diversas propuestas de mejora de la aplicación para hacer que esta sea lo más completa posible en el futuro.

1.4. Relación con otros estudios

Este proyecto esta fuertemente inspirado en las tecnologías aprendidas y utilizadas en la asignatura *Desarrollo de Interfaces Multimodales Avanzadas*, asignatura en la cuál aprendí a programar aplicaciones que utilizan diversas tecnologías novedosas como Kinect, SpeechAPI de Azure, programación para Android o, lo que más me ha servido en este caso, el uso de tecnologías de realidad extendida.

Por otra parte, gracias a otras asignaturas como *Ingeniería del Software* adquirí los conocimientos necesarios para poder programar en el lenguaje de programación que acepta Unity y la biblioteca MRUK: C#.

Por último, y aparte de los conocimientos adquiridos en las asignaturas ya mencionadas, he de destacar diversos tutoriales utilizados para aprender a utilizar tanto la plataforma Unity como la biblioteca de utilidades de Meta MRUK [1], gracias a los cuáles he conseguido terminar este proyecto con un resultado más que satisfactorio.

Capítulo 2

Contexto del trabajo

Como ya se ha mencionado en la introducción, las tecnologías de realidad extendida forman cada vez más parte del día a día del ser humano. Hoy en día, la realidad aumentada es una característica incluida en prácticamente todos los teléfonos móviles actuales. La mayoría de plataformas de vídeo disponen de la posibilidad de ver el contenido en realidad virtual mediante tecnologías con un precio irrisorio. Es por esto que podemos afirmar que las tecnologías de realidad extendida son ya una realidad y su uso debe de ser considerado en todo aquel sector que pueda obtener un beneficio de estas.

Sin embargo, no ha sido hasta los últimos tiempos que esto es así. En los últimos 3 años, el sector de la realidad extendida ha aumentado su tamaño de mercado un 187,8%, y se prevé que aumente de nuevo un 84% para 2026. Es por esto, que se debe echar un vistazo atrás para ver como ha ido evolucionando este sector.

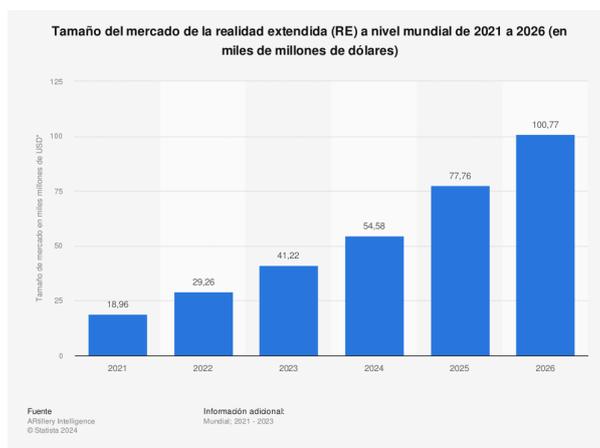


Figura 2.1: Gráfica del aumento del tamaño de mercado de la realidad extendida junto a sus previsiones por *Artillery Intelligence*

2.1. Herramientas de Realidad Extendida

Aunque a la mayoría de la población le parezca increíble, la realidad virtual no es un desarrollo tan moderno como se cree, de hecho, hay que remontarse a los años 60 para encontrar las primeras creaciones de dispositivos de realidad virtual. En este apartado, se realizará un paso por la historia de esta tecnología para saber como hemos llegado a la situación actual que nos permite el desarrollo de la aplicación que tiene como objeto este trabajo.

En 1968, Ivan Sutherland y su estudiante Bob Sproull desarrollaron el primer dispositivo de realidad virtual conocido como "La espada de Damocles". Este dispositivo permitía la visualización de gráficos simples en un entorno 3D mediante un aparato montado en la cabeza del usuario.

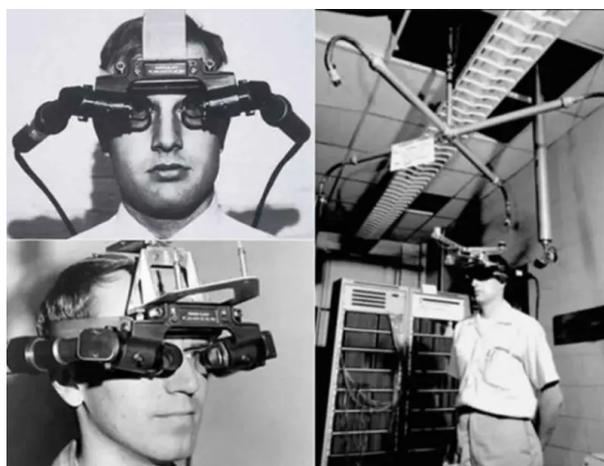


Figura 2.2: La espada de Damocles, el primer dispositivo de RV

Durante las décadas de 1970 y 1980, la NASA desarrolló el sistema "Virtual Interface Environment Workstation"(VIEW), que utilizaba guantes y gafas para interactuar con entornos simulados y en 1985, Jaron Lanier fundó VPL Research, una de las primeras empresas en comercializar tecnologías de RV, como gafas o guantes de datos.

En los años 90, el auge de estas tecnologías pasó a centrarse en su uso para videojuegos y por ello, empresas como SEGA o Nintendo crearon sus propios dispositivos VR, sin embargo, esta tecnología no llegó al gran público y por tanto, no disfrutó del éxito que tuvieron el resto de sus consolas.

Sin embargo, no fue hasta el año 2012, cuando las tecnologías VR volvieron a tener un espacio en el mercado gracias a Palmer Luckey, quien lanzó una campaña para financiar sus Oculus Rift. En 2014, este dispositivo, antepasado directo de la tecnología usada para el desarrollo de este trabajo, fue adquirida por Facebook (actual Meta), lo que convenció a muchos de que esta tecnología había llegado para quedarse.

Para terminar, en los últimos años la realidad virtual (así como la realidad aumentada y la realidad mixta) ha avanzado a pasos agigantados. El rebranding



Figura 2.3: Oculus Rift, el primer antepasado de las Meta Quest 3

de Facebook a Meta buscando potenciar el metaverso y la realidad virtual, ha hecho que los dispositivos de este tipo cobren una importancia vital en el avance de la tecnología. Los lanzamientos de Meta Quest 3 y Apple Vision Pro han supuesto un antes y un después en el ámbito de la Realidad Extendida, siendo estos, dispositivos con características y prestaciones nunca antes vistas en este campo.



Figura 2.4: Quest 3, el último lanzamiento de Meta y el dispositivo usado para este proyecto

2.2. La Realidad Extendida en la Medicina

Por otra parte, también debe ser objeto de este apartado un breve análisis del marco de uso actual de la realidad virtual en el ámbito de la medicina.

En 2022, el mercado de la realidad virtual aplicado a la medicina fue valorado en 2.400 millones de dólares, con un crecimiento previsto de un 42,5% anual, llegando en 2028 a los 18.400 millones de dólares. Esto es solo un reflejo más de la fuerza con la que la realidad virtual se está colando en todos los ámbitos de la vida cotidiana.

La aplicación de esta tecnología a la medicina permite tanto una aceleración de la recuperación de los pacientes mediante técnicas de rehabilitación como una mejoría en el entrenamiento de los profesionales al poder practicar las técnicas a realizar previamente en un entorno controlado y con un parecido casi total a la realidad [2].

Estudios recientes, han probado la efectividad de este tipo de métodos en estas situaciones, presentando la gran mayoría de pacientes una mejoría notable tras el uso de estas tecnologías [3]. Otros estudios, además, aseguran que el uso de estas tecnologías en la rehabilitación, ayuda a los pacientes a mantener la motivación y el interés [4]. De esta forma, y pese a no utilizar sujetos que necesiten rehabilitación en las pruebas de este proyecto, podemos afirmar que las pruebas realizadas en pacientes serían efectivas, puesto que son muy similares a las ya realizadas en otros estudios pero con una tecnología mas avanzada y cercana a la realidad.

Por otro lado, estos estudios muestran que entre las limitaciones principales, se encuentra la escasez de estudios sobre resultados no motores, como la calidad de vida o la relación costo-efectividad. También se observa que la mayoría de los estudios se centran en la función del miembro superior, lo que sugiere la necesidad de explorar otros resultados como el equilibrio o la movilidad en futuras investigaciones [5].

De esta forma, y pese a no utilizar sujetos que necesiten rehabilitación en las pruebas de este proyecto, podemos afirmar que las pruebas realizadas en pacientes serían efectivas, puesto que son muy similares a las ya realizadas en otros estudios pero con una tecnología mas avanzada y cercana a la realidad.

2.3. Realidad Virtual, Realidad Aumentada y Realidad Mixta

Debido al crecimiento repentino de tecnologías de realidad extendida (XR) hay mucha desinformación e ignorancia a la hora de nombrar a los diferentes tipos de tecnologías y se suele englobar a todos bajo el nombre de Realidad Virtual (VR) pues esta es la primera que se creó y hasta hace poco prácticamente la única que se utilizaba, pero esto es un error. Existen tres tipos de realidad extendida diferentes:

- **Realidad Virtual (VR):** La realidad virtual consiste en sumergir a los usuarios en entornos puramente virtuales. Esta tecnología es la más común y antigua de todas, puesto que funciona como una consola y es prácticamente independiente del mundo real.
- **Realidad Aumentada (AR):** En este caso, la realidad aumentada superpone objetos virtuales sobre el mundo real. Esto hace que el propio mundo real, generalmente visto a través de una cámara, sea el marco perfecto para los elementos virtuales.
- **Realidad Mixta (MR):** Por último, la realidad mixta es la tecnología que no solo superpone objetos virtuales en el mundo real, sino que los ancla y permite interactuar con ellos como si de objetos reales se tratasen. Esta tecnología solo la poseen los dispositivos mas modernos como las Quest 3 y las Apple Vision Pro, y sus capacidades van aumentando a medida



Figura 2.5: A la izquierda, Invizimals (2009) utilizaba una carta física para localizar los elementos virtuales en el mundo real. A la derecha, Pokemon GO (2016), localizaba a los elementos virtuales sin necesidad de objetos físicos

que actualizan el software de estas. Este es el tipo de tecnología utilizada para este proyecto, y se trata de la tecnología más avanzada en el mundo de la realidad extendida.

2.4. Meta, Apple y actualidad

En los últimos meses, ha comenzado una guerra tecnológica en el mundo de la realidad mixta. El 28 de octubre de 2023, Facebook anuncia un giro radical en la compañía, apostando por el mundo de la realidad virtual y cambiando su nombre a Meta, prometiendo enfocarse principalmente en el sector de la realidad mixta. Este cambio supuso un antes y un después en este ámbito, pues una de las compañías tecnológicas más potentes del mundo estaba cambiando por completo su modelo de negocio e introduciéndose de lleno en el mundo de la realidad virtual, realidad aumentada y realidad mixta. De hecho, ese mismo octubre, aún bajo el nombre de Facebook, lanzaron el dispositivo Meta Quest 3, el dispositivo de realidad aumentada más potente hasta la fecha.

Por otra parte, Apple, la empresa tecnológica mas grande del mundo, lanzó también un dispositivo de realidad aumentada solo 4 meses después de que lo hiciera Meta: las Apple Vision Pro. Estas presentaban una serie de novedades nunca antes vistas en este sector, y además, reafirmaba la importancia de las tecnologías de realidad mixta. De esta forma, dos de las empresas que han dominado el mercado en los últimos años apostaban por las tecnologías de realidad aumentada y virtual, lo que supondría un antes y un después en la historia de este tipo de dispositivos. Desde entonces, ambas empresas han acelerado el desarrollo de nuevas actualizaciones, intentando competir unas con otras y otorgando así una riqueza enorme para el sector.

Todavía es pronto para determinar donde llegará todo esto, pero no es descabellado afirmar que esta tecnología es el futuro, y poder utilizarla para el desarrollo de un proyecto como este, es algo que hay que tener altamente en

Contexto del trabajo

cuenta y explotar al máximo sus posibilidades para ver donde es capaz de llegar.



Figura 2.6: Apple Vision Pro y su principal novedad, *EyeSight*

Capítulo 3

Diseño e implementación

En este capítulo, se explicará cómo se ha procedido en el desarrollo de la aplicación de realidad mixta. Para ello, se explicará el entorno en el que se ha generado la aplicación junto con todas las herramientas necesarias para ello. Se hará énfasis en los estudios previos al proyecto y su análisis de viabilidad, la funcionalidad de la aplicación y las herramientas utilizadas junto al por qué de estas.

3.1. Método de trabajo

Con el objetivo de completar este estudio, se siguieron ciertas metodologías y tomaron ciertas decisiones que tendrían gran relevancia en el resultado final, cómo qué herramientas usar, el alcance de la aplicación o las pruebas que se realizarían.

Para la realización de este proyecto, se han seguido una serie de fases en un cierto orden, con el fin de conseguir que el resultado del estudio sea de la mayor calidad posible:

1. **Evaluación de la viabilidad:** La primera de las fases fue la evaluación de la viabilidad del proyecto. Previo a comenzar con el desarrollo, se debía asegurar que la aplicación era posible, tenía fundamentos y las herramientas eran lo suficientemente avanzadas para conseguir los objetivos. Podemos agrupar esta evaluación de viabilidad en tres partes:
 - **Investigación de estudios previos:** Previo al comienzo de este estudio, era necesario investigar otros estudios similares, que también utilizaran tecnologías de realidad extendida para la rehabilitación de pacientes con problemas neuropsicológicos. Utilizando algunos de estos artículos como base, se ideó en que consistiría la aplicación final y hacia que tipo de paciente estaría destinada, al menos esta versión inicial, puesto que el objetivo de este prototipo es ser multidisciplinar. También, se investigó acerca de métodos de evaluación

de aplicaciones, tanto de usabilidad como de carga de trabajo, puesto que estas dos características son fundamentales para asegurar una buena calidad de los resultados reales en las rehabilitaciones de los pacientes, y debido a que no haría pruebas con pacientes reales, esta es la única forma de evaluar el funcionamiento de la aplicación.

- **Aprendizaje y elección de las herramientas:** Posterior a esto, fue necesario investigar sobre las herramientas que se podrían utilizar para el desarrollo y se tomaron las decisiones correspondientes respecto a esto, como se profundizará más en el siguiente punto. Una vez tomadas estas decisiones, se investigó acerca del funcionamiento de estas herramientas (MRUK de Meta y Unity) mediante el uso de tutoriales y documentación. Se hizo uso de la documentación de esta biblioteca para aprender así sus funciones clave, indagando más en aquellas que serían interesantes para la aplicación que se iba a desarrollar y, mediante el visionado de vídeos de creadores de contenido de este ámbito, se aumentaron los conocimientos con respecto a estas tecnologías, aunque la mayoría de estos estaban desfasados debido a que la versión utilizada para el desarrollo de la aplicación era la última actualización, que incorporaba grandes cambios con respecto a las versiones anteriores, por lo que se tuvo que centrar la gran mayoría de esta investigación únicamente en la documentación y el ensayo y error.
- **Desarrollo de la aplicación piloto:** Por último en este paso de evaluación de viabilidad, se comenzó con el desarrollo de una aplicación de prueba para las funcionalidades que posteriormente serían utilizadas en la aplicación final, para confirmar que esta podría llegar a tener todas las características que se plantearon en un inicio (reconocimiento del entorno, interacción con los controladores y uso de “handtracking”, etc).

De esta forma, la evaluación de viabilidad se dio por concluida, y se confirmó que las herramientas escogidas eran capaces de asegurar el funcionamiento deseado de la aplicación final, por lo que se prosiguió con el desarrollo de esta.

2. **Desarrollo de la aplicación final:** Una vez interiorizados los conocimientos necesarios para la utilización de las herramientas de desarrollo, se comenzó con el desarrollo de la aplicación final, implementando todas las funcionalidades que fueron planteadas en la investigación de estudios previos, siendo algunas de estas descartadas tanto por falta de tiempo como por escasez de características de la biblioteca MRUK (Van añadiendo funciones poco a poco, pues se trata de una biblioteca muy reciente). Estas funcionalidades descartadas serán explicadas en el capítulo 5 “Conclusiones y trabajos futuros”.
3. **Pruebas:** Como ya se ha mencionado previamente, las pruebas no serían

3.1. Método de trabajo

realizadas con pacientes reales, sino con sujetos sin ningún tipo de problema neuropsicológico debido a la dificultad de encontrar pacientes de este tipo y su condición. Por ello, posterior al desarrollo de la aplicación, se realizaron las pruebas de usabilidad y carga de trabajo pertinentes, las cuales serán explicadas más adelante en el capítulo 4 “Resultados”. La elección de qué cuestionarios utilizar para que los sujetos evalúen las pruebas fue en parte elección del tutor del proyecto, dada su experiencia en el uso de este tipo de cuestionarios. Se entrará más en profundidad en estas pruebas en el punto 3.2.3 “Pruebas”.

4. **Redacción de la memoria:** Finalmente, posterior al desarrollo de la aplicación y una vez realizadas las pruebas y obtenidos los resultados, ya estaba toda la información lista para comenzar con la redacción de la memoria.



Figura 3.1: Diagrama de Gantt del proyecto

3.2. Estudio de viabilidad

Una vez explicada la cronología y la metodología de trabajo que se siguió para desarrollar el proyecto, pasaremos a indagar en cada una de estas fases, especificando detalles concretos sobre cada una de estas y plasmando los detalles del desarrollo del proyecto.

3.2.1. Estudio de las herramientas a utilizar

Previo al comienzo del desarrollo de la aplicación, fue necesario estudiar todas las herramientas disponibles para este tipo de tecnologías. Como el dispositivo que se iba a utilizar eran las gafas de realidad aumentada Meta Quest 3, la elección de la biblioteca MRUK era esencial, puesto que esta está diseñada específicamente para este dispositivo, y contiene la implementación de todas las funcionalidades y características de las gafas en las aplicaciones. Por otra parte, el motor con el que se desarrollaría la aplicación sería Unity ya que esta biblioteca solamente está creada para este motor gráfico.

Esta biblioteca consiste en un conjunto de herramientas enfocadas en el desarrollo de aplicaciones de realidad mixta. El MRUK sustituye al All in One SDK, que era un SDK anterior de Meta utilizado principalmente para desarrollar aplicaciones de realidad virtual y aumentada. El All in One SDK proporcionaba una solución más general, pero a medida que las capacidades de los dispositivos de Meta crecieron, se necesitaba una versión más avanzada y enfocada específicamente en las nuevas tecnologías de la compañía. El MRUK ofrece mejoras en términos de rendimiento, facilidad de uso y compatibilidad con los dispositivos más modernos (Meta Quest 3), además de un enfoque más robusto para integrar realidad mixta de manera eficiente en Unity. En resumen, este fue creado para aprovechar las últimas innovaciones de hardware y software de Meta.

Por esto, podemos concluir que la elección de las herramientas viene totalmente condicionada por el dispositivo utilizado, y este viene condicionado por voluntad de utilizar el mejor dispositivo del mercado con las funcionalidades más modernas y novedosas.

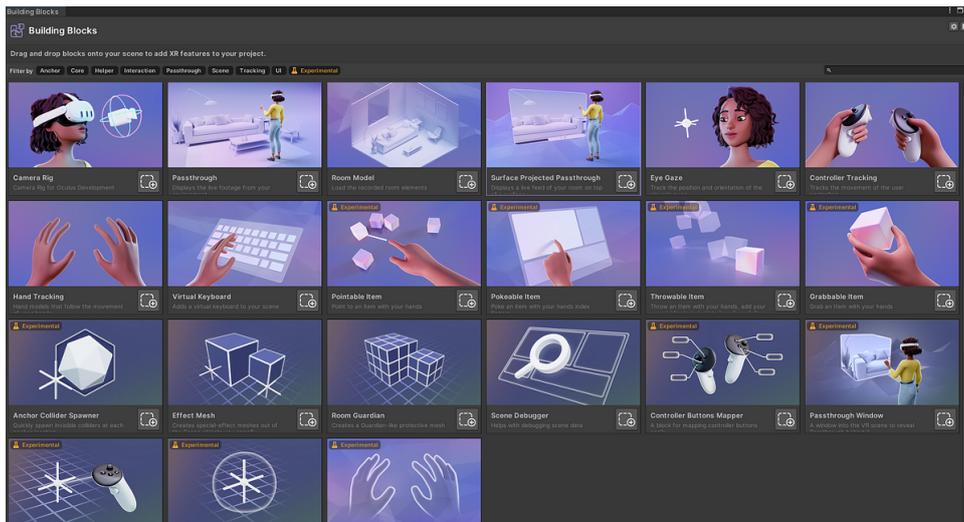


Figura 3.2: Ejemplo de las funcionalidades que otorga la biblioteca MRUK, para Unity, con sus “Building Blocks”

3.2.2. Desarrollo de la aplicación piloto

Como ya se ha mencionado anteriormente, previo al desarrollo de la aplicación final, se desarrolló una aplicación piloto la cual sirvió para probar todas las posibilidades que otorgaba el nuevo dispositivo Meta Quest 3, como el “passthrough” o la interacción con los controladores, así como también una prueba de lo que es desarrollar en Unity una aplicación de realidad aumentada.

Esta aplicación no utilizaba la biblioteca MRUK, debido a que no existía ningún tutorial actualizado en internet. Por ello, se utilizó la biblioteca de Unity “XR Interaction Toolkit”, con la que ya se tenía una experiencia previa. La aplicación era muy sencilla: consistía en reconocer la habitación en la que estaba el usuario, y generar las denominadas “bounding boxes” alrededor de los muebles reconocidos por las gafas. También era posible en esta aplicación generar objetos o “prefabs” concretos mediante la interacción de los rayos de los controladores con estos muebles.

Una vez conseguidos estos objetivos, se dispuso a comenzar la aplicación final, esta vez sí utilizando la biblioteca MRUK para poder implementar todas las funcionalidades que no eran posibles utilizando el “XR Interaction Toolkit” de Unity. Esta aplicación será explicada con detalle en la sección 3.3 “Desarrollo de la aplicación final”.

3.2.3. Pruebas

A la hora de plantear que pruebas deberían realizar los usuarios se llegó a la conclusión de que estas deberían ser sencillas y rápidas, porque a la hora de utilizar esta aplicación con un sujeto en rehabilitación, las tareas que

propondría el rehabilitador no serían muy extensas ni pesadas. Por ello, las tareas que se propusieron para las pruebas fueron tanto utilizando el modo rehabilitador como el modo paciente, y fueron las siguientes:

- Usando los controladores, coloca en la superficie que tienes frente a ti 2 áreas de colocación y 2 objetos a tu elección.
- Usando solo las manos, agarra el primer objeto y muévelo hasta el área de colocación roja. Después, agarra el segundo objeto y muévelo hasta el área de colocación azul.

Después de estas pruebas, a los usuarios se les hicieron 2 cuestionarios, tanto de usabilidad como de carga de trabajo, SUS y NASA TLX respectivamente, que serán abordados en el capítulo 4 “Resultados”.

3.3. Desarrollo de la aplicación final

Ahora comenzaremos a explicar el paso a paso de como se desarrolló la aplicación, tratando temas como casos de uso, métodos y funcionalidades, interfaces y características a destacar.

Primero, analizaremos en que se basará esta aplicación. Esta, constará, como ya se ha mencionado, de dos partes, la del rehabilitador y la del paciente. La primera consistirá en, utilizando los controladores y la interfaz de la aplicación, seleccionar lugares en la superficie de la habitación donde colocar tanto los objetos proporcionados por la interfaz como las áreas de colocación, que se tratan de 2 círculos de colores rojo y azul. Por otra parte, el modo paciente consistirá en, usando las manos y el “handtracking” que proporciona la biblioteca MRUK, agarrar y soltar los objetos en las áreas de colocación previamente situadas por el rehabilitador.

3.3.1. Modelado Conceptual

Esta aplicación se trata de un prototipo bastante simple, con funciones muy sencillas pero que abren un mar de posibilidades a los usuarios que la utilizan.

Los casos de uso de esta aplicación son solo dos, y muy bien marcados. Esta aplicación puede ser utilizada tanto por rehabilitadores, que la utilizarán para crear los escenarios con los elementos y las áreas correspondientes que usarán posteriormente los pacientes, y los pacientes, que utilizarán los escenarios creados por los rehabilitadores para realizar las tareas correspondientes, también determinadas por el rehabilitador.

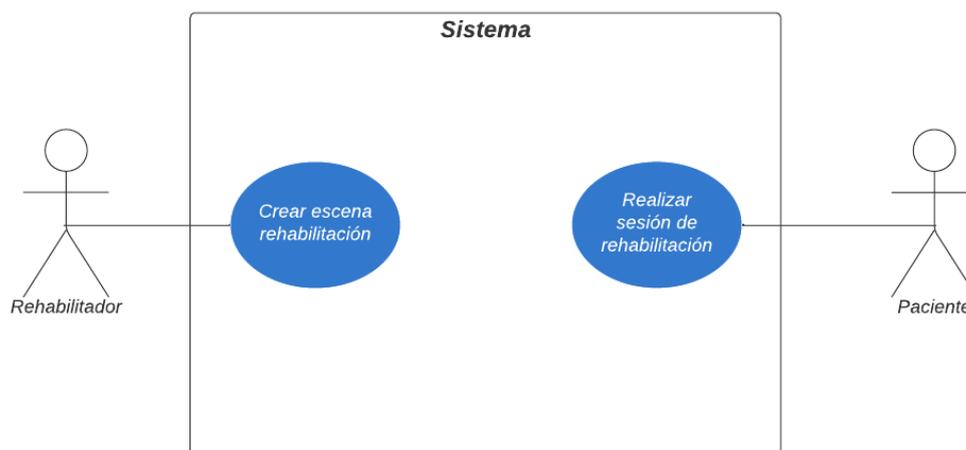


Figura 3.3: Diagrama de casos de uso de la aplicación

Las funciones, en el caso del rehabilitador, serían las siguientes:

- **Creación de área de colocación:** El rehabilitador debe ser capaz de crear unas áreas de colocación en las cuales el paciente está destinado a colocar los objetos correspondientes al ejercicio. Para ello, apuntará al lugar deseado y presionará el botón "A" del controlador derecho. Estas áreas de colocación se representan en la aplicación como círculos de colores: rojo y azul. Estos fueron creados a partir de cilindros 3D truncando su altura para que esta sea imperceptible a la vista y parezca un objeto 2D sobre la superficie.

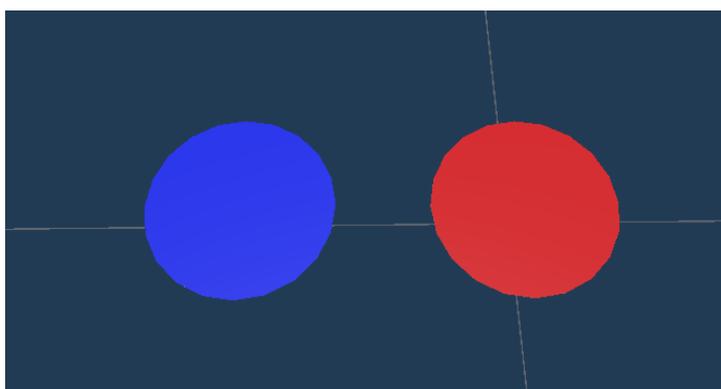


Figura 3.4: Áreas de colocación en el visor de prefabs de Unity

- **Selección de objeto:** El rehabilitador ha de poder seleccionar el objeto que desea posteriormente colocar, de entre los 5 ofrecidos por la interfaz, utilizando la interfaz previamente mencionada. Para ello, apuntará con el controlador derecho al objeto que desea colocar y presionará el gatillo derecho (ver figura 3.9).

- **Generación de objeto:** El rehabilitador podrá, previamente habiendo seleccionado el objeto en cuestión, colocar este objeto en cualquier superficie de la habitación apuntando sobre ella y presionando el gatillo del controlador izquierdo tantas veces como objetos de ese tipo desee colocar.



Figura 3.5: Ejemplos de objetos virtuales en la aplicación

- **Ocultación de interfaz:** Una vez finalizado el posicionamiento tanto de objetos como de áreas de colocación, el rehabilitador será capaz de, presionando el botón “X” del controlador izquierdo, ocultar la interfaz de usuario para que el trabajo del paciente no se vea influido por agentes externos, de forma que este se centre únicamente en los objetos y las áreas de colocación.

Por parte del paciente, las funciones que podrá realizar será únicamente la manipulación de objetos virtuales con sus propias manos, utilizando el hand-tracking de las Meta Quest 3.

3.3.2. Diseño

En cuanto al diseño de la aplicación podemos distinguir dos partes en esta. Principalmente, el esqueleto creado por la propia biblioteca MRUK: Controladores, Hand Trackers, Visión desde el OVRCameraRIG (el visor de las gafas) y la habitación escaneada previamente por el dispositivo. Por otra parte, estaría la lógica y la interfaz gráfica desarrollada durante el transcurso del proyecto, así como también la selección de los objetos que serían utilizables en la aplicación.

Después, se implementaron los “Building Blocks” de MRUK. Estos consisten en, como su propio nombre indica, bloques de código prefabricado para su funcionamiento inmediato. Estos Building Blocks serán representados en el diagrama de clases con un color naranja, dejando ver que toda su lógica ha sido desarrollada por los ingenieros de Meta. Algunos de los building blocks pueden observarse en la figura 3.2.

Los building blocks que fueron utilizados en esta aplicación fueron:

3.3. Desarrollo de la aplicación final

- **Camera Rig:** Este bloque permite que la cámara de la escena sean las gafas Quest 3, haciendo así que el movimiento del usuario suponga un movimiento en la cámara en la aplicación. En el diagrama de clases constituye a la clase `OVRCameraRig`.
- **Passthrough:** Este bloque permite la visión a través de las cámaras del dispositivo Meta Quest 3, haciendo que al iniciarse la aplicación sea el mundo real lo que veamos. Esta función se observa en el diagrama de clases bajo el nombre de `OVRPassthroughLayer`.
- **Room Model:** Este bloque permite utilizar la habitación escaneada previamente por las gafas en la que estás situado como un elemento más para la aplicación, pudiendo editar diversos factores como solidez de los muebles para objetos virtuales, efectos concretos en ciertos tipos de muebles/superficies o diversas características más. En el caso de esta aplicación, este bloque se utilizó para reconocer el entorno y poder situar los objetos virtuales y áreas de colocación en este tipo de superficies, así como para generar la oclusión necesaria entre los elementos de la habitación y los objetos virtuales, dándole a estos un mayor realismo. Es representado en el diagrama como `MRUKRoom`



Figura 3.6: Ejemplo de oclusión entre objetos virtuales y elementos de la habitación

- **Controller Tracking:** Se trata de un modelo de los controladores sobrepuestos a los controladores reales, haciendo que estos se conviertan también en objetos virtuales y estén mejor integrados en la aplicación. También, abre la posibilidad de decidir si quieres que aparezca un rayo desde tu controlador con el que poder apuntar, o de otorgar funciones únicas a los botones de estos. Hay dos en el diagrama de clases, tanto el izquierdo (a la izquierda) como el derecho (a la derecha) y cada uno de ellos está relacionado con sus funciones propias.
- **Hand Tracking:** Este bloque consiste en un seguimiento de las manos del usuario en tiempo real, generando así una malla constante que hace posible la interacción manual, sin controladores de por medio, con objetos virtuales en la aplicación, haciendo que los usuarios puedan empujar y contactar con estos objetos con sus propias manos, así como de nuevo, permite la oclusión de estos objetos a través de la mano, cosa que parecería impensable en cualquier otro dispositivo de realidad extendida. Se pueden observar en el diagrama de clases relacionados con los controladores.

- **Grabbable Item:** Este bloque permite al prefab que lo contenga, la capacidad de ser agarrado mediante las manos del usuario, ya sea mediante “pinch grab” (agarrar un objeto pellizcándolo, como un lápiz, el asa de una taza o un cepillo de dientes) o “palm grab” (agarrar un objeto con toda la mano, como puede ser un libro, un mando o un teléfono móvil). Este building block es esencial para esta aplicación, pues es la base de la interacción del paciente con los objetos virtuales, y una buena implementación es necesaria para que la experiencia de usuario sea satisfactoria. Para ello fue necesario modificar algunos parámetros como la caja de colisiones de los “prefabs”, la distancia que debía de haber entre los dedos para que el sistema entendiese si se estaba agarrando el objeto o si se estaba soltando, o el tipo de agarre que debía permitirse según el objeto. Puede observarse en el diagrama con una relación con los GameObjects, como ya ha sido explicado.

En cuanto al resto de funciones implementadas en la aplicación, estas han sido explicadas en la sección 3.3.1 “Modelado Conceptual”. Constituyen las clases “SpawnCircle”, “PrefabSelector”, “LeftControllerRaycast” y “HideCanvas”, respectivamente como aparecen en la sección previa. En cuanto a la clase “PointableCanvas”, se trata de una interfaz gráfica que será explicada en la sección 3.3.3 “Interfaz de usuario”.

3.3. Desarrollo de la aplicación final

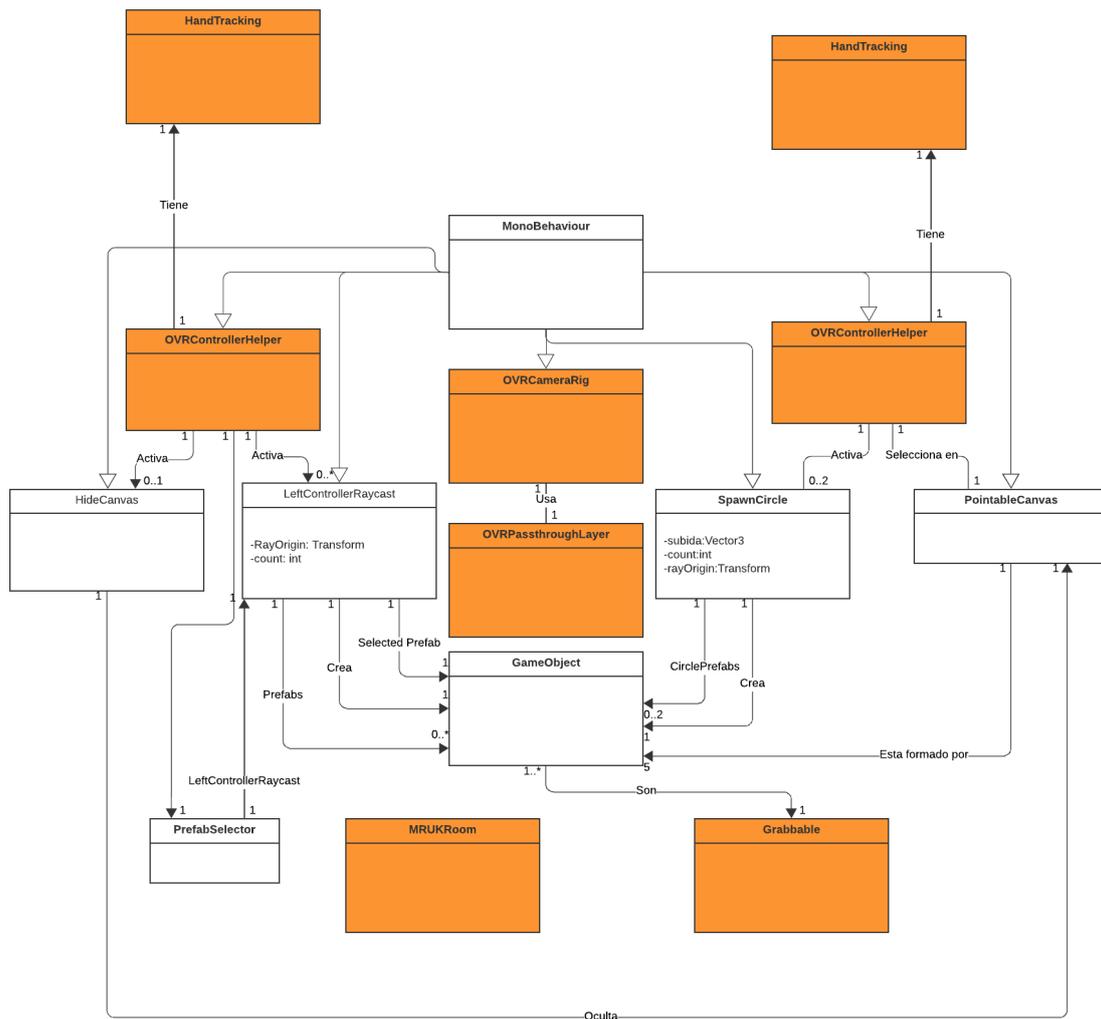


Figura 3.7: Diagrama de clases de la aplicación

3.3.3. Interfaz de usuario

Ahora pasaremos a analizar la interfaz de usuario de la aplicación. El objetivo de esta interfaz era explicar a los usuarios, tanto rehabilitadores como pacientes, el funcionamiento de la aplicación sin necesidad de explicaciones externas ni ayuda de técnicos, algo fundamental a la hora de que los pacientes vayan a utilizar esta aplicación personalmente. A su vez, la interfaz debía constituir una parte de los propios controles de la aplicación, en este caso como una forma de seleccionar el objeto a colocar por parte del rehabilitador. Por ello, se introdujo junto a las instrucciones de la aplicación un tablero con todos los posibles objetos a colocar, para poder ser seleccionados y colocados a gusto del terapeuta. En las imágenes inferiores podemos observar como se vería esta interfaz mientras se utiliza la aplicación.

Diseño e implementación

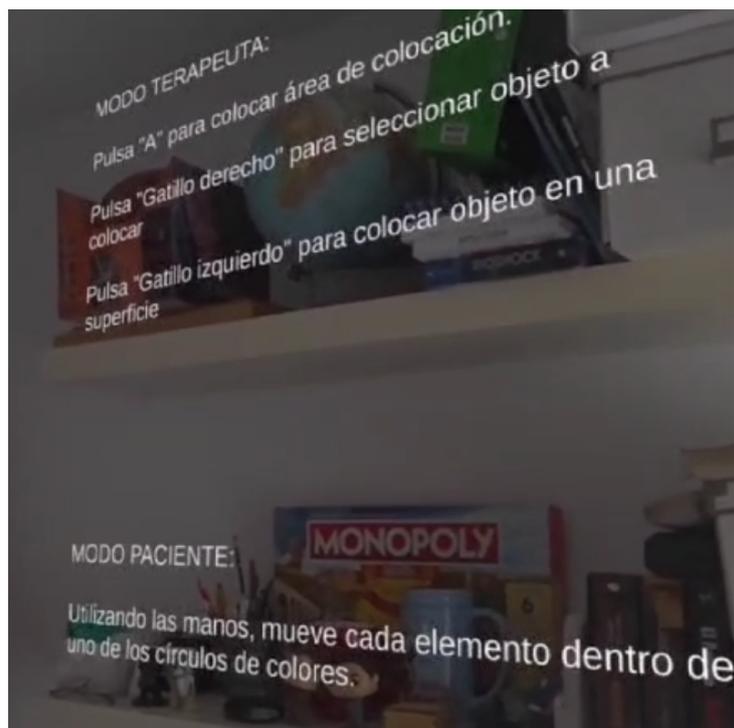


Figura 3.8: Captura de las instrucciones de la aplicación durante una ejecución de esta



Figura 3.9: Captura del selector de "prefabs" dura una ejecución de la aplicación

3.4. Entorno de trabajo

3.4.1. Herramientas

Para la realización de este proyecto, han sido utilizadas diversas herramientas, tanto para su desarrollo como para su posterior utilización.

3.4.1.1. Ordenador y Windows 11

Para el desarrollo de la aplicación, se utilizó un ordenador de potencia media (Intel Core i7 9700F 3GHz y NVIDIA GeForce GTX1650) con Windows 11 instalado junto con otros programas de desarrollo de aplicaciones de este tipo que se mencionará en el siguiente apartado 3.4.2. Este ordenador era necesario para generar la APK que contiene la aplicación ejecutada en el dispositivo Meta Quest 3, utilizando el modo desarrollador para poder depurar correctamente la aplicación a medida que el desarrollo avanzaba.

3.4.1.2. Meta Quest 3

El dispositivo de realidad mixta Meta Quest 3 es uno de los más avanzados cuando se habla de esta tecnología. Este dispositivo fue escogido para ser el empleado en este proyecto, puesto que tiene varias características y funciones únicas en este tipo de dispositivos.

Estas gafas poseen la característica de poder mostrar el mundo real a través de ellas de forma detallada, gracias a sus cámaras de alta calidad, proporcionando así el denominado “passthrough” es decir, poder ver el mundo real mientras se utiliza el dispositivo junto a la interfaz de este. Además, este dispositivo es capaz de realizar un escaneo detallado de las áreas de trabajo físicas para posteriormente ser utilizadas en cualquier aplicación. A su vez, también es capaz de mostrar oclusión de objetos virtuales, haciendo que la profundidad de estos en la escena sea mucho mas realista y creíble, generando así una sensación de que el objeto realmente está compartiendo el mismo espacio físico que el usuario.

Todo esto junto a la gran calidad de procesamiento del dispositivo, hace que este sea idóneo para la tarea propuesta en este proyecto, y podemos asegurarnos de que se está utilizando uno de los mejores y mas avanzados sistemas del mercado.

3.4.2. Software

3.4.2.1. Unity y Oculus MRUK

Unity es una plataforma de desarrollo integral y un motor de videojuegos utilizado para crear aplicaciones interactivas y experiencias en 2D, 3D, realidad aumentada (AR) y realidad virtual (VR).



Scene without occlusions

The virtual character is out of *place* in this example. The scene is not believable and the experience is severely impacted.

Scene with an occlusion

The virtual character appears hidden behind the furniture, improving the experience.

Figura 3.10: Ejemplo de como funciona la oclusión con Meta Quest 3

Una de las grandes ventajas del uso de Unity es su soporte de múltiples plataformas de VR, incluyendo Oculus, HTC Vive, PlayStation VR y Meta Quest, lo que facilita el desarrollo de aplicaciones que funcionen en diversos dispositivos. Además, cuenta con herramientas y bibliotecas específicas para RV. Un ejemplo es la biblioteca de utilidades y herramientas utilizada en este proyecto: Mixed Reality Utility Kit (MRUK) [6]. MRUK proporciona las herramientas necesarias para desarrollar aplicaciones “spatially-aware” en Unity para Meta Quest 3, es decir, aplicaciones que utilicen el entorno físico como una parte más de la aplicación. Esta biblioteca aporta utilidades como trazado de rayos para seleccionar o apuntar a un espacio concreto, escaneo de la habitación y posterior mallado de los muebles y paredes para que tengan colisiones, o escenas de ejemplo para aprender a utilizar estas herramientas.

Por otra parte, Unity permite a desarrolladores de diferentes niveles de experiencia crear aplicaciones complejas, con una interfaz de usuario intuitiva y un flujo de trabajo visual. Además, para el desarrollo del código, permite un link directo con aplicaciones como Visual Studio utilizando C como lenguaje.

En el caso de esta aplicación, es necesario el uso de objetos de la vida cotidiana para que la aplicación sirva al paciente para rehabilitarse antes de pasar a las pruebas en un entorno real. Por eso, Unity es una muy buena opción ya que ofrece una gran biblioteca de assets [7] y modelos 3D para usar y así acelerar el desarrollo de sus aplicaciones.



Figura 3.11: Logo de Unity

3.4.3. Meta Horizon

Meta Horizon es el nombre de la aplicación actual desarrollada por Meta para el control móvil de las gafas Meta Quest 3 (y otros modelos de la marca) mediante un dispositivo móvil. Para este trabajo, esta aplicación se ha utilizado con el fin de visualizar que está viendo el usuario portador de las gafas en tiempo real, así como realizar grabaciones de las acciones que este realiza. Esta aplicación ha cambiado de nombre y sido actualizada varias veces desde el lanzamiento de las primeras Meta Quest hasta su interfaz actual.

Por otra parte, esta aplicación fue desarrollada también para acceder al “Metaverso” creado por Meta y desde ella permite conectar con amigos y acceder a una tienda para este metaverso. Además, permite la retransmisión inmersiva de eventos como conciertos, eventos deportivos u otros espectáculos.

A todo esto, se le añade la función de creación de avatares, lo que permitiría la interacción con otros usuarios mostrando una figura “imitación” de nuestra forma en el mundo real. Con este tipo de funciones, Meta pretendía que el Metaverso fuese la nueva revolución tecnológica y que en los próximos años, la mayoría de las interacciones sociales se realizarasen a través de este metaverso, sin embargo, este proyecto se ha ralentizado bastante debido a la falta de aceptación entre la masa social.

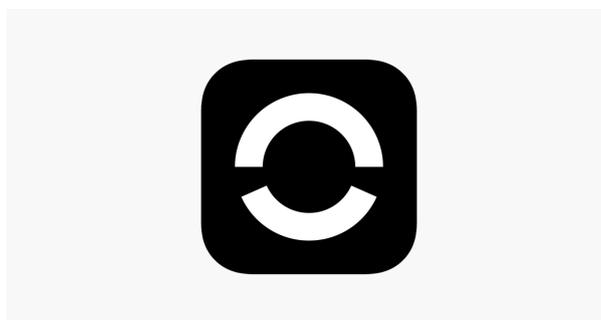


Figura 3.12: Logo de Meta Horizon

3.4.4. Visual Studio

Visual Studio es un entorno de desarrollo integrado creado por Microsoft y diseñado para facilitar el desarrollo de software. Esta plataforma está principalmente centrada en el desarrollo de aplicaciones extensas en C, C++, C# o .NET, y permite algunas funciones como facilidades para implementar Git y conexión sencilla con Unity.

En este trabajo, el programa ha sido utilizado con el lenguaje de programación C# y la aplicación conectada directamente con Unity ha permitido el uso simultáneo de ambas, probando los cambios en el código rápidamente en la aplicación de Unity.

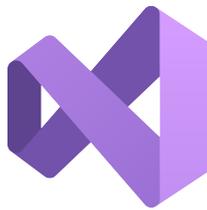


Figura 3.13: Logo de Visual Studio

3.4.5. NASA TLX

NASA TLX (Task Load Index) es una herramienta desarrollada por la NASA en la década de los 80 que permite realizar una evaluación subjetiva de cualquier aplicación en la que se mide la carga de trabajo mental y física percibida por una persona mientras realiza una tarea. Este es uno de los dos métodos de evaluación que utilizaremos para evaluar la aplicación final y que será explicada más adelante.

En este caso, la herramienta se ha utilizado a través de su aplicación disponible en Apple Store para realizar un cuestionario a los sujetos sometidos al test. La plataforma utilizada ha sido un iPad, de forma que a los sujetos les resultase fácil e intuitivo el desarrollo del cuestionario.

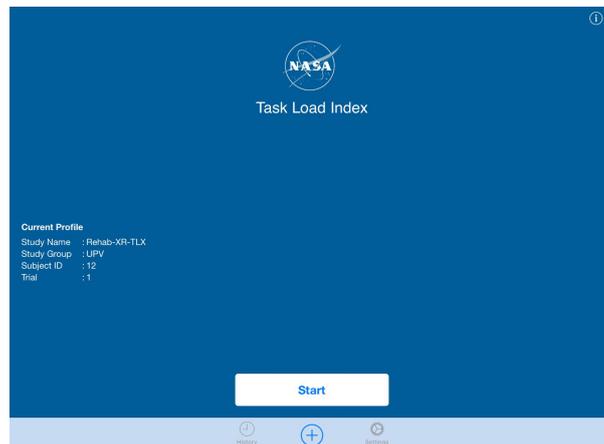


Figura 3.14: Pantalla principal de la aplicación de NASA TLX para Ipad

Capítulo 4

Resultados

Los sujetos que realizaron las pruebas en este proyecto, como ya se ha mencionado anteriormente, son usuarios con ningún tipo de condición neuropsicológica adversa, sino que son usuarios comunes a los cuáles se les hacen dos cuestionarios para evaluar cuestiones de usabilidad y carga de trabajo. Los usuarios sometidos a estas pruebas fueron un total de 12, con unos rangos de edad desde los 17 hasta los 82 años. Las circunstancias de las pruebas eran variables según la persona, puesto que estas se llegaron a hacer en tres localizaciones diferentes, pero todas ellas tenían entornos de trabajo muy similares y las pruebas fueron idénticas.

Los resultados de las pruebas, en cuanto a sensaciones generales, fueron muy buenos. Todos los usuarios consiguieron el objetivo principal de la prueba, conseguir colocar y mover los objetos con éxito, ya sea empleando más o menos tiempo, dependiendo del usuario. Ningún usuario presentó problemas a la hora de seguir las instrucciones, pero algunos cometían fallos a la hora de colocar los objetos debido a la falta de familiaridad con los botones de los controladores. Por todo esto, podemos concluir que las pruebas fueron un éxito, a falta de observar los resultados de los cuestionarios realizados a los usuarios.

4.1. System Usability Scale

El “System Usability Scale” (Escala de Usabilidad del Sistema), comúnmente conocido como SUS, es un cuestionario de 10 afirmaciones para las cuales el usuario debe expresar su grado de acuerdo o desacuerdo en una escala de 1 a 5, con la premisa de ponerse en la piel de un terapeuta que esté planteándose el uso de la aplicación, siendo 1 completamente en desacuerdo y 5 completamente de acuerdo. Este tipo de escala es conocida como escala de Likert. De las 10 afirmaciones, las 5 de número impar son afirmaciones positivas y las 5 pares son negativas. Para calcular el índice de usabilidad, en primer lugar se deben sumar los resultados de las afirmaciones positivas y restarle 5. A continuación, se deben sumar los resultados de las afirmaciones negativas y

4.1. System Usability Scale

restar este número a 25. Por último, se deben sumar los dos resultados obtenidos y multiplicar el resultado por 2,5. Este será el índice de usabilidad del sistema en una escala de 0 a 100.

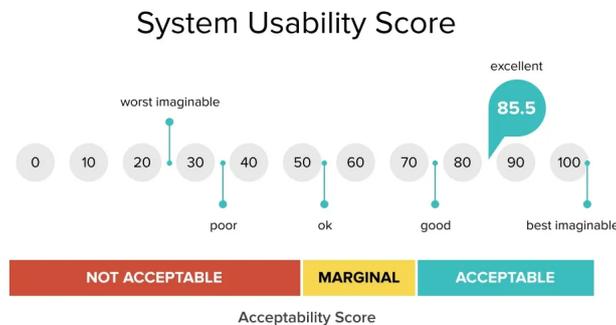


Figura 4.1: Escala de Usabilidad del Sistema según la puntuación en el cuestionario SUS

4.1.0.1. Resultados SUS

Con respecto a los resultados del cuestionario SUS, ninguno de los usuarios otorgó una puntuación menor a 75, por lo que todos los usuarios opinaban que, siguiendo la escala de la imagen anterior, la aplicación era buena en términos de usabilidad. La media de puntuación entre todos los usuarios es de 87.91, es decir, una usabilidad excelente. La disparidad entre estas puntuaciones viene altamente sesgada por la edad del usuario, siendo los usuarios más familiarizados con las tecnologías aquellos que mejores puntuaciones otorgaban a la aplicación, y siendo los más mayores los que otorgaban las puntuaciones más bajas. Estos fueron los resultados de cada una de las preguntas, que vienen adjuntas en el Anexo 1. Hay que tener en cuenta que para las preguntas pares se espera una respuesta positiva, mientras que para las impares una negativa, sobre un máximo de 5 por pregunta:

1. Media: 4,5 Varianza: 0,272727273
2. Media: 1,25 Varianza: 0,204545455
3. Media: 4,666666667 Varianza: 0,242424242
4. Media: 2,083333333 Varianza: 1,71969697
5. Media: 4,416666667 Varianza: 0,628787879
6. Media: 1,25 Varianza: 0,386363636
7. Media: 4,5 Varianza: 0,272727273
8. Media: 1,666666667 Varianza: 1,515151515
9. Media: 4,666666667 Varianza: 0,424242424
10. Media: 1,333333333 Varianza: 0,424242424

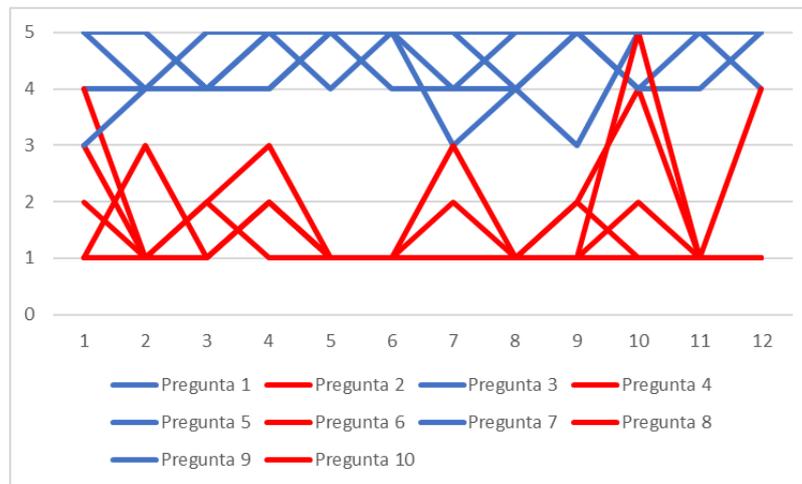


Figura 4.2: Gráfica de respuestas del cuestionario SUS. En azul, preguntas pares, en rojo, las preguntas impares.

Por todo esto, podemos afirmar que la aplicación es muy buena en términos de usabilidad, y que no supone ningún problema aprender a utilizarla, uno de los objetivos principales de este proyecto.

4.1.1. NASA TLX

Según el Instituto Nacional de Seguridad y Salud en el Trabajo (INSST) [8] el NASA TLX es un procedimiento de valoración multidimensional que da una puntuación global de carga de trabajo, basada en una media ponderada de las puntuaciones en seis subescalas, cuyo contenido es el resultado de la investigación dirigida a aislar de forma empírica y a definir los factores que son de relevancia en la experiencia subjetiva de carga de trabajo.

Este cuestionario pregunta a los usuarios cuál de entre 6 factores, preguntado por pares, le ha parecido en su experiencia el más relevante para completar la tarea asignada, a la vez que pide evaluar cada uno de estos 6 factores según su experiencia (baja/alta o bueno/malo).

Todo esto, y gracias en gran medida a su aplicación, permite obtener una serie de estadísticas sobre la carga de trabajo que supone realizar la tarea asignada, dejando ver en nuestro caso cuales son los factores que una persona sana debe emplear para completar las tareas previamente explicadas.

4.1.1.1. Resultados NASA TLX

En las siguientes gráficas se puede observar los resultados de los cuestionarios NASA TLX de todos los usuarios. Podemos observar factores como cuál es el factor más elegido, cuál el menos y ciertas tendencias.

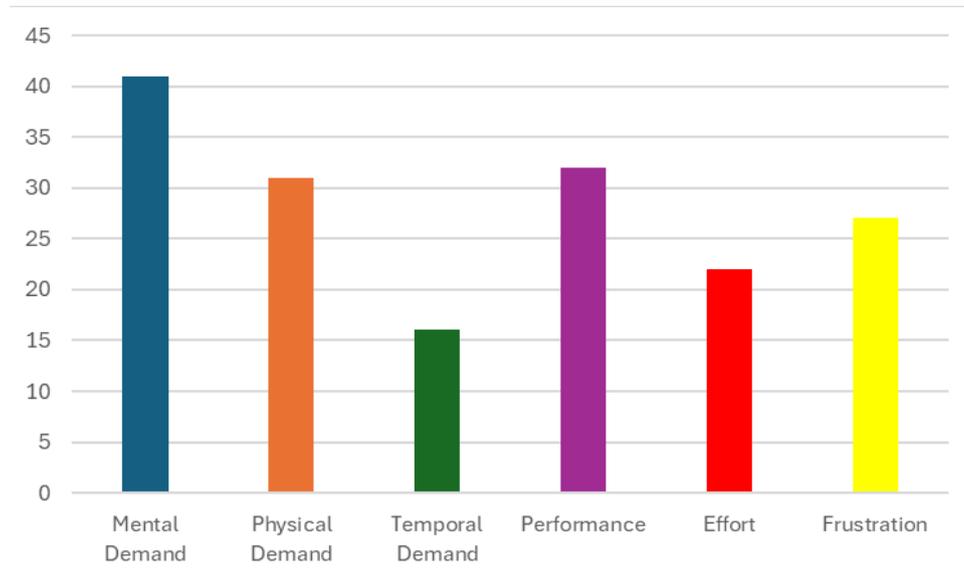


Figura 4.3: Gráfica de elección por factor

Como se puede observar en el gráfico, el factor que más demanda esta tarea es la demanda mental, con una diferencia significativa. Por otra parte, el factor que menos importancia se le da es la demanda temporal. Estas dos afirmaciones tienen sentido, debido a que, al no tener problemas físicos estos usuarios, el principal reto de esta tarea es a la hora de comprender los controles y aprender a utilizar la aplicación, es decir, la parte del rehabilitador. En cuanto a la parte del paciente, se apreció una gran facilidad en el manejo de la aplicación por parte de todos los usuarios que realizaron las pruebas, por lo que podemos confirmar que a los pacientes reales no les supondrá un problema comprender el funcionamiento de la parte de la aplicación que ellos utilizarán, y serán simplemente los rehabilitadores los que deberán familiarizarse con los controles de la aplicación para crear escenarios útiles.

Con respecto a la evaluación de cada factor, los usuarios también determinaron que el factor con más carga fue la demanda mental, pero en la gráfica observamos que de un total de 1200, los usuarios no puntuaron ninguno de los factores por encima del 30%, es decir, ningún factor les pareció especialmente exigente. Por esto, podemos determinar que la carga de trabajo necesaria para realizar las pruebas, y por ende para utilizar la aplicación, es bastante baja, algo bueno puesto que no se busca que las tareas, ni para el rehabilitador ni para el paciente, sean complejas, ya que el objetivo es tratar de conseguir que estos puedan completarlas con facilidad si no poseen problemas neuropsicológicos. Podemos observar en el siguiente gráfico como

Resultados

ninguno de los factores supera el 30% y como de nuevo los factores de máxima y mínima demanda son los mismos que en el caso anterior.

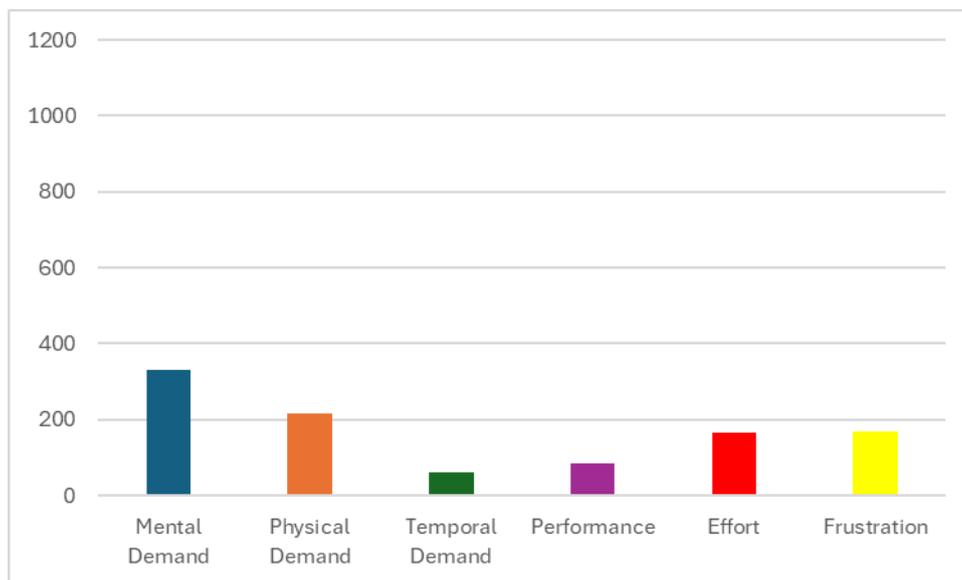


Figura 4.4: Gráfica de evaluación por factor

4.2. Comparación de resultados

Una vez obtenidos todos los resultados, podemos observar que la aplicación tiene una usabilidad adecuada, y que la carga de trabajo al utilizarla no es considerable. Por ello, podemos afirmar que la aplicación es sencilla de utilizar, uno de los objetivos principales de esta, pues no está destinada a ser una aplicación compleja en su uso, sino que cumpla su papel de ayudar a los pacientes a realizar tareas muy similares a las tareas cotidianas, como lo son el movimiento de objetos ligeros. Por parte de la creación de estos escenarios de mano del rehabilitador, también observamos que es sencillo su uso, y que todos los usuarios que la utilizaron completaron las tareas sin dificultad, por lo que podemos asegurar que no supondría ningún reto para ningún terapeuta que quiera hacer uso de esta.

Capítulo 5

Conclusiones y trabajos futuros

Una vez observado tanto el desarrollo como los resultados de las pruebas, podemos obtener una serie de conclusiones con respecto a los objetivos iniciales y la motivación de este proyecto.

Una de las motivaciones del proyecto era desarrollar una aplicación multidisciplinar para la ayuda de personas con daño cerebral adquirido. Esta aplicación, sin embargo, al ser un prototipo, no tiene por ahora varias disciplinas con las que trabajar, pero sienta un precedente y unas bases para futuras mejoras, como veremos en el siguiente punto, y proporciona una gran cantidad de herramientas con las que muy fácilmente sería posible crearlas.

Otro de los objetivos era crear una aplicación totalmente innovadora en este ámbito, que funcionase para ayudar a personas con problemas neuropsicológicos utilizando la tecnología, pero tratando de ser lo menos intrusivos posible y haciendo que esta tecnología se asemeje a la realidad lo máximo posible. Este objetivo podemos afirmar que se ha cumplido con creces, pues hemos conseguido crear una aplicación de realidad mixta, en la que el paciente observa durante todo momento el mundo real a su alrededor, pero interactúa con objetos hechos a su medida y de forma que pueda ser infinitamente replicable. Por otra parte, hemos conseguido desarrollar una aplicación altamente usable tanto para rehabilitadores como para pacientes, haciendo sencilla la tarea de creación de escenarios, y más sencilla aún la tarea de manipulación de objetos virtuales.

Uno de los principales problemas que han surgido a lo largo del proyecto, ha sido la falta de algunas características esenciales para la definición total del proyecto. Esto es debido a la utilización de tecnologías tan prematuras, y el uso de herramientas todavía en desarrollo. Como se mencionará en el siguiente apartado, muchas de las características que se buscaron introducir en la aplicación, no fue posible incorporarlas, en gran medida debido a este problema. Es por esto que a medida que avance esta tecnología, y más en concreto esta biblioteca de herramientas, la aplicación podrá ser mejorada hasta

el punto de conseguir todas las funcionalidades planteadas inicialmente.

Pese a esto, podemos afirmar que la aplicación ha sido todo un éxito y podría ser continuada para añadir varias funcionalidades que le otorguen una mayor completitud, como veremos en el siguiente punto.

5.1. Futuras mejoras

Como se ha ido mencionando durante esta memoria, este proyecto se trata de un prototipo de aplicación multidisciplinar, que sirva de ayuda en el campo de la rehabilitación de personas con problemas neuropsicológicos. Por ello, al ser un prototipo, falta que esta pueda ser continuada, añadiendo todas las características que le hagan falta para hacerla cada vez más y más completa. Algunas de las futuras mejoras que fueron planteadas durante el desarrollo de la aplicación, pero que fueron descartadas por falta de tiempo, fueron las siguientes:

- **Evaluación en tiempo real de los resultados del paciente:** Uno de las mejoras que se propuso inicialmente en el proyecto era un medidor tanto de precisión como de tiempo total que tardaba el usuario en realizar la tarea. Esto serviría para analizar la mejoría del usuario a medida que utiliza la aplicación, haciendo que el terapeuta pueda observar si este tipo de actividades están siendo útiles para el paciente o si existen otras formas de hacer que el paciente avance más rápido en su rehabilitación.
- **Guardado de escenarios para uso independiente:** Otra de las funcionalidades que se plantearon inicialmente fue el guardado de escenarios en la memoria de las gafas, de manera que el rehabilitador pudiese guardar estos con un solo botón, y el usuario presionando otro botón tuviese ese escenario cargado en su sesión. Esto sería una de las mejoras más significativas de la aplicación, puesto que significaría que el usuario puede utilizar la aplicación independientemente de si el terapeuta está con el o no, ya que este habría dejado las tareas preparadas para cuando el paciente tuviese que realizarlas.
- **Diversos tipos de actividad:** Como ya se ha mencionado previamente, otra de las mejoras que podrían realizarse sería la creación de diversos tipos de actividad. Por ahora, solo existe la tarea de mover objetos virtuales a las áreas de colocación correspondientes, pero otras tareas como ejercicios de profundidad, reflejos o precisión también podrían ser diseñadas para ser utilizadas en el futuro.
- **Ampliación de biblioteca de objetos virtuales:** Otra mejora posible sería la ampliación de la biblioteca de objetos virtuales a utilizar. Actualmente existen cinco, escogidos cuidadosamente para que se ajusten a las necesidades y consistencia de la aplicación, pero podrían añadirse más si fuera necesario.

- **Creación de escenarios remota:** Por último, la creación de escenarios de forma remota, sería la última de las mejoras que supondría un gran avance en la aplicación, pues significaría que el rehabilitador sería capaz de diseñar los escenarios para los pacientes sin necesidad de tener una consulta presencial con estos, lo que facilitaría el avance en la rehabilitación sustancialmente.

A pesar de esto, podemos asegurar que la aplicación está lista para el uso en pacientes reales, y tiene un gran potencial para cambiar el mundo de la rehabilitación de pacientes con estas características. Esperemos que la tecnología continúe avanzando y podamos ver más aplicaciones enfocadas en este sector para ayudar a aquellos que puedan verse beneficiados por ella.

Bibliografía

- [1] [Online]. How to make a Mixed Reality app for the Quest 3 *Available at: <https://www.youtube.com/watch?v=ibBzydrumc0&list=PLWcLPdrF6kOnoLNP846xfzeWY15h6S5I&abchannel=LudicWorlds>*
- [2] angela.tuduri (2024, Jun 3). [Online]. Realidad Virtual en medicina: Actualidad y aplicaciones *Available at: <https://blogs.salleurl.edu/es/realidad-virtual-en-medicina-actualidad-y-aplicaciones>*
- [3] David R. et al (2023, November). “Immersive virtual reality gameplay detects visuospatial atypicality, including unilateral spatial neglect, following brain injury: a pilot study” in Journal of NeuroEngineering and Rehabilitation, doi: 10.1186/s12984-023-01283-9
- [4] Maria Grazia Maggio et al (2019, April) “Virtual Reality and Cognitive Rehabilitation in People With Stroke: An Overview” in Journal of Neuroscience Nursing, doi:10.1097/JNN.0000000000000423
- [5] Hatem Lazem et al (2023, August), “The Extent of Evidence Supporting the Effectiveness of Extended Reality Telerehabilitation on Different Qualitative and Quantitative Outcomes in Stroke Survivors: A Systematic Review” in International Journal of Environmental Research and Public Health, doi: 10.3390/ijerph20176630
- [6] (2024, Ago 24). [Online]. Mixed Reality Utility Kit Overview. *Available at: <https://developer.oculus.com/documentation/unity/unity-mr-utility-kit-overview/>*
- [7] [Online]. Unity Asset Store. *Available at: https://assetstore.unity.com/?srsltid=AfmBOoo8B4OjmFwxvOLqj1PRoZba0m0tsYmioG-azfNlwSWH4YsKp_A*
- [8] [Online]. NTP 544: Estimación de la carga mental de trabajo: el método NASA TLX *Available at: <https://www.insst.es/documents/94886/524376/Carga+Mental+en+el+trabajo/5a3492ae-9ef0-41fd-b538-385c682ba42f>*

Anexo I: Código C#

```
1 using Meta.XR.BuildingBlocks;
2 using System;
3 using System.Collections;
4 using System.Collections.Generic;
5
6 using UnityEngine;
7 using UnityEngine.XR.Interaction.Toolkit;
8
9 Script de Unity (2 referencias de recurso) | 1 referencia
10 public class LeftControllerRaycast : MonoBehaviour
11 {
12
13     public List<GameObject> Prefabs;
14     public Transform rayOrigin;
15     public int count;
16     public SpatialAnchorCoreBuildingBlock spatialAnchorspawner;
17     public GameObject selectedPrefab;
18
19
20 Mensaje de Unity | 0 referencias
21 void Start()
22 {
23     if (Prefabs.Count > 0)
24     {
25         selectedPrefab = Prefabs[0]; // Prefab por defecto
26     }
27     // Update is called once per frame
28 Mensaje de Unity | 0 referencias
29 void Update()
30 {
31     if (OVRInput.GetDown(OVRInput.Button.PrimaryIndexTrigger))
32     {
33         RaycastHit hit;
34         if (Physics.Raycast(rayOrigin.position, rayOrigin.forward, out hit))
35         {
36             {
37                 GameObject instance = Instantiate(selectedPrefab, hit.point, Quaternion.identity);
38                 AddInteractionComponents(instance);
39             }
40
41             void AddInteractionComponents(GameObject instance)
42             {
43                 if (instance.GetComponent<Rigidbody>() == null)
44                 {
45                     instance.AddComponent<Rigidbody>();
46                 }
47
48                 if (instance.GetComponent<XRGrabInteractable>() == null)
49                 {
50                     instance.AddComponent<XRGrabInteractable>();
51                 }
52             }
53         }
54     }
55 }
56
57
```

Figura 1: Código Left Controller Raycast

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

@ Script de Unity (1 referencia de recurso) | 0 referencias
public class PrefabSelector : MonoBehaviour
{
    public LeftControllerRaycast leftControllerRaycast;
    // Start is called before the first frame update
    0 referencias
    public void SelectPrefab(int index)
    {
        leftControllerRaycast.selectedPrefab = leftControllerRaycast.Prefabs[index];
        Debug.Log("Prefab with index " + index + "selected");
    }
}
```

Figura 2: Código Prefab Selector

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.EventSystems;
5  using UnityEngine.XR.Interaction.Toolkit;
6
7  // Script de Unity (2 referencias de recurso) | 0 referencias
8  public class SpawnCircle : MonoBehaviour
9  {
10     public List<GameObject> circlePrefabs;
11     public Transform rayOrigin;
12     public List<OVRSpatialAnchor> AnchorList;
13     public int count=0;
14     Vector3 subida = new Vector3(0, 0.0001f, 0);
15
16     public XRRayInteractor rightRay;
17
18     // Mensaje de Unity | 0 referencias
19     void Start()
20     {
21     }
22     // Update is called once per frame
23     // Mensaje de Unity | 0 referencias
24     void Update()
25     {
26
27         bool isOverUI = IsRayHoveringOverUI(rightRay);
28         if (OVRInput.GetDown(OVRInput.Button.One) && !isOverUI)
29         {
30             RaycastHit hit;
31             if (Physics.Raycast(rayOrigin.position, rayOrigin.forward, out hit) )
32             {
33                 if (count == 0)
34                 {
35                     GameObject vaseInstance = Instantiate(circlePrefabs[0], hit.point+subida, Quaternion.identity);
36                     count++;
37                 }
38                 else if (count == 1)
39                 {
40                     GameObject vaseInstance = Instantiate(circlePrefabs[1], hit.point+subida, Quaternion.identity);
41                     count++;
42                 }
43                 else { }
44             }
45         }
46
47         if (OVRInput.GetDown(OVRInput.Button.SecondaryHandTrigger))
48         {
49             foreach (OVRSpatialAnchor anchor in AnchorList)
50             {
51                 anchor.SaveAnchorAsync();
52             }
53         }
54     }
55
56     // 1 referencia
57     private bool IsRayHoveringOverUI(XRRayInteractor rayInteractor)
58     {
59         RaycastResult uiRaycastResult;
60         if (rayInteractor.TryGetCurrentUIRaycastResult(out uiRaycastResult))
61         {
62             return uiRaycastResult.isValid;
63         }
64         return false;
65     }
66 }
67
68
69

```

Figura 3: Código Spawn Circle

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

Script de Unity | 0 referencias
public class HideCanvas : MonoBehaviour
{
    public Canvas canvas;

    Mensaje de Unity | 0 referencias
    void Start()
    {
        canvas.gameObject.SetActive(true);
    }

    Mensaje de Unity | 0 referencias
    void Update()
    {
        if (OVRInput.GetDown(OVRInput.Button.Three))
            canvas.gameObject.SetActive(false);
    }
}
```

Figura 4: Código Hide Canvas

Anexo II: Cuestionarios usabilidad y carga de trabajo

SUS

1. Creo que me gustaría utilizar este sistema con frecuencia
2. Encontré el sistema innecesariamente complejo
3. Pensé que el sistema era fácil de usar
4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema
5. Encontré que las diversas funciones de este sistema estaban bien integradas
6. Pensé que había demasiada inconsistencia en este sistema
7. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente
8. Encontré el sistema muy complicado de usar
9. Me sentí muy seguro usando el sistema
10. Necesitaba aprender muchas cosas antes de empezar con este sistema

Anexo II: Cuestionarios usabilidad y carga de trabajo

NASA TLX

Rating Scale Definitions

Place a mark at the desired point on each scale:

Title	Descriptions	
MENTAL DEMAND	How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?	<p>MENTAL DEMAND</p> <p>Low High</p>
PHYSICAL DEMAND	How much physical activity was required (e.g., pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?	<p>PHYSICAL DEMAND</p> <p>Low High</p>
TEMPORAL DEMAND	How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?	<p>TEMPORAL DEMAND</p> <p>Low High</p>
PERFORMANCE	How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?	<p>PERFORMANCE</p> <p>Good Poor</p>
EFFORT	How hard did you have to work (mentally and physically) to accomplish your level of performance?	<p>EFFORT</p> <p>Low High</p>
FRUSTRATION LEVEL	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?	<p>FRUSTRATION</p> <p>Low High</p>

Figura 5: Cuestionario NASA TLX