
Herramienta de Accesibilidad para Mejorar la
Comunicación de Personas con Discapacidad Auditiva en
Entornos Sociales de Realidad Virtual

*Accessibility Tool to Improve the Communication of People with
Hearing Impairment in Social Environments of Virtual Reality*



Trabajo de Fin de Grado
Curso 2021–2022

Autor

Antonio Cardona Costa

Director

Federico Peinado Gil

Grado en Desarrollo de Videojuegos
Facultad de Informática
Universidad Complutense de Madrid

Herramienta de Accesibilidad para Mejorar la
Comunicación de Personas con Discapacidad Auditiva en
Entornos Sociales de Realidad Virtual
*Accessibility Tool to Improve the Communication of People with
Hearing Impairment in Social Environments of Virtual Reality*

Trabajo de Fin de Grado
Dpto. de Ingeniería del Software e Inteligencia Artificial

Autor
Antonio Cardona Costa

Director
Federico Peinado Gil

Convocatoria: *Junio 2022*

Grado en Desarrollo de Videojuegos
Facultad de Informática
Universidad Complutense de Madrid

30 de mayo de 2022

Agradecimientos

A Nuria de Pedraza, por ser siempre la luz de mi vida, que me acompaña y anima a seguir adelante, creyendo en mí y recordándome el camino que me ha llevado hasta el presente. Gracias por darme fuerzas cuando no me quedan.

A mi madre, Maria Costa Serra. que sin ella nada de esto sería posible, desde su humildad ha luchado siempre sola con una fuerza que me ha inspirado toda la vida.

Resumen

La realidad virtual ha cambiado mucho en los últimos años. En sus inicios se trataba de un producto enfocado a investigadores y entusiastas, pero esos primeros kits de desarrollo han dado paso a productos enfocados al público general con varias generaciones de fabricación que han dado vida a un sector con cada vez más empresas. HTC, Valve, Sony, HP, Samsung y muchos otros se han unido a la carrera iniciada por Oculus. Hoy podemos encontrar dispositivos tan potentes como Quest 2 en los escaparates de grandes superficies a precios muy asequibles.

Muchas han sido las mejoras experimentadas gracias a la evolución de esta tecnología, todas encaminadas a proporcionar una inmersión más profunda. Una mayor tasa de refresco evita los mareos, se alcanzan resoluciones que aspiran a eliminar el efecto de rejilla, los sistemas de sonido integrados permiten experimentar un sonido tridimensional, etc.

Se trata de una ventana a nuevos mundos virtuales que explorar, interactuando con todo lo que se encuentra a través de nuestro avatar. Es especialmente importante poder relacionarse con otras personas en entornos sociales, como son muchos videojuegos, donde la comunicación entre usuarios es fundamental para disfrutar y conseguir mejores resultados.

En medio de esta frenética carrera tecnológica no debemos olvidar a aquellas personas que no tienen fácil acceso a estas experiencias. En este trabajo se presenta el desarrollo de una herramienta que facilita la comunicación de personas con dificultades auditivas, permitiendo la comunicación mediante traducción de algunos signos o de la voz a texto plano, de modo natural y sin perjudicar en exceso la inmersión.

Se puede acceder al código de la aplicación a través del repositorio público (Antonio Cardona, 2022).

Palabras clave

Accesibilidad, Lengua de Signos, Realidad Virtual, Comunicación, Desarrollo de Videojuegos, Entornos Sociales, Multijugador, Unity.

Abstract

Virtual Reality scene has changed a lot in the last few years. In its beginnings it was a product targeted to enthusiasts and researchers, but those first development kits brought a new line of products targeted to the general public. Generation by generation this sector has come to life with more and more investors and manufacturers. HTC, Valve, Sony, HP, Samsung and many others joined this race started by Oculus. Today it's easy to find powerful devices like Quest 2 in any mall for an affordable price.

Many improvements came with the evolution of this technology, aiming for a deeper user immersion. A higher refresh rate prevents dizziness, display resolutions are reaching a point where screen effect is close to disappear, integrated sound systems give a better 3D sound awareness. Step by step technology is reaching a better immersion.

This is about providing an access window to new virtual worlds to explore and interact with through our avatar. It is of most interest to give the users the ability to communicate with others in these new social environments, being videogames or just social applications, so everyone can enjoy and interact seamlessly.

Developers can not afford to forget about those who can't have an easy access to this experiences. This work develops a tool that facilitates communication for people with Hearing Impairment, enabling communication through the translation of voice to text and sign language to text.

The code can be found in this repository (Antonio Cardona, 2022).

Keywords

Accessibility, Sign Language, Virtual Reality, Communication, Videogame Development, Social Environments, Multiplayer, Unity.

Índice

1. Introducción	1
1.1. Motivación	2
1.2. Propósito y alcance	3
1.3. Asignaturas relacionadas	4
1.4. Estructura del trabajo	5
2. Estado de la cuestión	7
2.1. Mundos virtuales y juegos sociales	7
2.2. Realidad virtual	10
2.2.1. Mundos virtuales y realidad virtual	14
2.3. Accesibilidad en videojuegos	15
2.3.1. Discapacidad auditiva y realidad virtual	18
2.4. Reconocimiento de gestos	18
2.4.1. Reconocedores visuales	19
2.4.2. Captura con marcadores o sensores	20
2.4.3. Algoritmos aplicados al reconocimiento	21
3. Objetivos y especificación	23
3.1. Objetivos propuestos	23
3.2. Especificación	24
3.2.1. Funcionalidades	24
3.2.2. Limitaciones	26
4. Metodología y herramientas	29
4.1. Metodologías de desarrollo	29
4.1.1. Reuniones quincenales	29
4.1.2. Planificación	29
4.2. Herramientas utilizadas	30
4.2.1. Google Suite	30
4.2.2. Slack	31
4.2.3. Github	31

4.2.4.	Github Projects	31
4.2.5.	Git for Windows	31
4.2.6.	Unity	31
4.2.7.	Visual Studio	31
4.2.8.	SideQuest	32
4.2.9.	OBS Studio	32
4.2.10.	Notepad++	32
4.2.11.	Overleaf	32
4.2.12.	Paquetes externos	32
5.	Diseño y desarrollo de la herramienta	35
5.1.	Análisis y diseño	35
5.1.1.	Ofuscación entre manos	35
5.1.2.	Ofuscación de los dedos	35
5.1.3.	Signos complejos	36
5.1.4.	Signos con la mano mirando hacia abajo	36
5.1.5.	Signos en movimiento	36
5.2.	Interfaces virtuales	36
5.2.1.	Botones	37
5.3.	Reconocimiento de gestos	38
5.3.1.	Controlador del jugador	38
5.4.	Persistencia	39
5.4.1.	Base de datos	39
5.4.2.	Ruta de almacenamiento	40
5.4.3.	Propiedades de los gestos	41
5.4.4.	Carga, captura y almacenamiento	45
5.4.5.	Proceso de reconocimiento	45
5.4.6.	Gestión de la base de datos	47
5.5.	Tutorial	52
5.6.	Modo multijugador	52
5.6.1.	Escenas para el modo multijugador	53
6.	Resultados	57
6.1.	Pruebas	57
6.2.	Discusión de los resultados del formulario	59
7.	Conclusiones	63
7.1.	Objetivos alcanzados	63
7.2.	Trabajo futuro	64
7.3.	Reflexión final	65
A.	Introduction	67

A.1. Motivation	68
A.2. Purpose and scope	69
A.3. Related subjects	69
A.4. Structure of the work	70
B. Conclusions	73
B.1. Goals complete	73
B.2. Future tasks	74
B.3. Final thoughts	75
B.4. Cuestionario y resultados	76
Bibliografía	83

Índice de figuras

1.1. Dispositivos de RV más utilizados en los últimos años según Steam Analytics.	2
2.1. Un sala de Habbo Hotel que reproduce una playa.	8
2.2. Second Life.	9
2.3. Club Penguin.	10
2.4. Primeros modelos de Oculus.	11
2.5. HTC Vive de Steam.	11
2.6. Modelo comercial Playstation VR.	12
2.7. Meta Quest.	12
2.8. Meta Link permite utilizar Quest para jugar a juegos de PCVR.	13
2.9. Comparación de ventas estimadas. Por Superdata Research.	14
2.10. Valve index en acción.	14
2.11. Cartel promocional de Half Life Alyx.	15
2.12. Imagen promocional de VRChat.	15
2.13. Imagen capturada y proyección al modelo 3D de la propuesta Digit Eyes.	19
2.14. Un ejemplo de captura con cámaras binoculares.	20
3.1. Concepto inicial presentado en la fase de diseño.	24
3.2. Diseño de la interfaz.	25
5.1. Esferas colisión en cada hueso.	37
5.2. Prefab de botón virtual que sirve de plantilla.	37
5.3. Código de un botón personalizado que hereda de la plantilla.	38
5.4. Vista de Inspector de Unity del OVRPlayerController.	39
5.5. Datos de posición de los huesos del OVRHand.	40
5.6. Un gesto en la base de datos.	40
5.7. Estructura de datos que almacena la información de un gesto.	41
5.8. Struct con los datos de un hueso en concreto.	42
5.9. Posición y Rotación de un hueso en el gesto almacenado.	42
5.10. Enumerado para establecer el uso de manos en un gesto.	43

5.11. Enumerado para establecer la categoría del gesto.	44
5.12. Enumerado para establecer la Fase del gesto.	44
5.13. Flujo del Reconocimiento.	45
5.14. Consola de Captura.	48
5.15. El Nombre es rellenado con el teclado.	48
5.16. Teclado en la consola de captura.	49
5.17. Botones para configurar que mano se va a capturar.	49
5.18. Selección de categoría.	49
5.19. Selección de fase.	50
5.20. Cuenta atrás para capturar el gesto.	50
5.21. El nuevo gesto ha sido capturado.	51
5.22. Lista de gestos y botón para eliminar.	51
5.23. Paso 5 del tutorial.	53
5.24. Lobby de bienvenida de la aplicación.	53
5.25. Selector de salas multijugador con distintas escenas.	54
5.26. Escena exterior multijugador.	54
5.27. Escena laboratorio UCM, ordenadores.	54
5.28. Escena laboratorio UCM, pupitre del profesor.	55
6.1. Pruebas realizadas en la Facultad de Informática de la UCM.	58
A.1. Dispositivos de RV más utilizados en los últimos años según Steam Analytics.	68

Capítulo 1

Introducción

La Realidad Virtual esta en clara expansión y cada vez son más los estudios que desarrollan para estas plataformas que trasladan al jugador a nuevos mundos virtuales. El mundo de los videojuegos siempre ha tenido un gran componente de escapismo del mundo real, por ello cuando el usuario se sumerge en la realidad virtual toma el control de su avatar virtual. Estos son capaces de todo tipo de proezas, podemos generar objetos de la nada, escalar muros imposibles, atravesar el cielo volando, dar grandes saltos o levantar cualquier tipo de objetos, usar magia, transformar cuerpo en cualquier tipo de forma. Teniendo esta capacidad de ser algo distinto, debe plantearse seriamente cómo esta tecnología puede ayudar a superar barreras que en el mundo real sufren personas con distintos tipos de discapacidades. En el caso de personas con discapacidad auditiva, el videojuego tradicional ofrece interfaces visuales con subtítulos descriptivos en canales de chat. En el mundo virtual la mayoría de aplicaciones recurren a la comunicación vía voz y los usuarios no disponen de un teclado. Es necesario crear una herramienta capaz de superar este nuevo reto y ayudar a que estas personas se integren de la forma más natural posible, ya que de ello depende su disfrute de la inmersión en un mundo virtual.

La Realidad Virtual (RV) comercial aparece en 2016 y no ha parado de crecer. El mundo ha sido testigo de una batalla entre los distintos dispositivos, algunos autónomos, otros conectados al móvil, otros al ordenador. Desde el inicio de la pandemia esta competición se ha recrudecido, enfrentando a las grandes empresas tecnológicas por intentar recabar la mayor parte de este creciente sector y el interés del público general. Valve, HTC, Meta (antes Facebook), Sony, Samsung o HP, cada una ha presentado su particular propuesta, desde simples soportes para habilitar los teléfonos móviles de última generación como dispositivos de RV, hasta periféricos de videoconsolas que cuentan con equipos de desarrollo enteros destinados a generar contenido exclusivo para esas plataformas, pasando por soluciones generalistas para ordenadores con capacidades técnicas muy potentes, con suficiente capacidad

computacional para reproducir los juegos en dispositivos de RV conectados a través de un cable. Aunque en los primeros años los dispositivos para consola fueron los de mayor éxito, el mercado se ha decantado a día de hoy por la solución más accesible y de coste asequible, los cascos autónomos de RV.

Estos dispositivos no requieren de hardware adicional para funcionar, sus sistemas de seguimiento "dentro a fuera" no requieren la instalación de balizas de posicionamiento en una habitación, dando a los usuarios total libertad para vivir las experiencias de RV en cualquier lugar y entorno. El precio a pagar por esta independencia es su potencia y autonomía: dependen de baterías que limitan su tiempo de uso y sus juegos tienen gráficos algo menos detallados. Estos dispositivos autónomos permiten ahora ser conectados a ordenadores para aprovechar sus capacidades, como ocurre con el cable Link, dando una solución versátil capaz de adaptarse a las necesidades de cada usuario. El éxito de esta filosofía es patente en otros sectores como el de los videojuegos, el Libro Blanco de los Videojuegos 2021 muestra cómo la consola para la que más desarrollan las empresas y estudios españoles es Nintendo Switch (DEV, 2021), que además es líder en ventas de consolas de la actual generación (VGChartz, 2022), siendo esencialmente una plataforma de menor potencia, pero mayor versatilidad, con capacidades móviles y de dispositivo sobremesa. En la RV, el dispositivo de éxito y líder de ventas actualmente es Meta Quest 2 (Meta, 2019), como muestran las estadísticas de Steam Valve (2022).

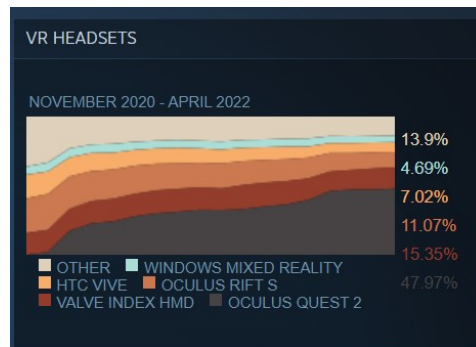


Figura 1.1: Dispositivos de RV más utilizados en los últimos años según Steam Analytics.

1.1. Motivación

Este nuevo paradigma en el mundo del entretenimiento debería ser accesible para cualquier usuario. La misma filosofía de proyectar al usuario en un avatar del mundo virtual debería permitir, del mismo modo, dejar atrás aquellas barreras del mundo real que son fácilmente superables tecnológica-

mente. Por ello, las herramientas de accesibilidad son tan importantes para derribar esos muros.

En este proyecto se pretende crear una herramienta que ayude a cualquier usuario con problemas de audición a comunicarse de la manera más natural posible. El objetivo es que un usuario con discapacidad auditiva pueda expresarse y entender a los demás, de la forma que le resulte más natural.

Meta Quest permite reconocer directamente las manos reales del usuario y proyectarlas en el mundo virtual, además incorpora micrófonos y altavoces. Se dispone de todas las herramientas necesarias para desarrollar una herramienta capaz de superar todas estas barreras.

1.2. Propósito y alcance

Según la Confederación Estatal de Personas Sordas (CNSE), cerca del 3% de la población española sufre algún tipo de pérdida auditiva (CNSE, 2022).

El propósito de este trabajo es aprovechar la tecnología disponible para crear una herramienta de inclusión que ayude al colectivo de personas con discapacidad auditiva a integrarse en el futuro del metaverso, los videojuegos y demás aplicaciones sociales de RV. Ahondar en el desarrollo de aplicaciones de RV para dispositivos móviles con base Android también forma parte de lo que se busca. Crear una herramienta libre, fácilmente implementable en cualquier proyecto de RV creado en el popular entorno de desarrollo Unity, es también una de las metas.

Si bien existen algunas aplicaciones desarrolladas para interpretar gestos, estas se quedan en simples demostraciones del alfabeto dactilológico de la lengua de signos americana (ASL). Este proyecto se centra en la Lengua de Signos Española, su alfabeto dactilológico y también la capacidad de capturar, almacenar y reconocer los signos pertenecientes a un diccionario de gestos.

La comunidad con discapacidad auditiva existe en la RV, pero tienen que aprender a expresar sus respectivas lenguas de signos, superando la barrera que supone interactuar con el mundo virtual a través de mandos que limitan enormemente la capacidad de gesticular y realizar ciertos signos. Este proyecto pretende que esas personas puedan comunicarse utilizando sus propias manos, sin necesidad de aprender qué botón sirve para extender o contraer que dedo virtual, o qué combinación de estos botones crean un gesto predefinido.

1.3. Asignaturas relacionadas

A lo largo de la carrera han sido muchas las asignaturas que aportan los conocimientos necesarios para poder emprender a día de hoy un proyecto de estas características. En esta sección destacaré aquellas que han sido las más relevantes para este proyecto.

- **Diseño de Videojuegos.** Esta asignatura de primer curso ofrece las habilidades y herramientas necesarias para iniciar un proyecto, además de las pautas adecuadas de diseño.
- **Metodologías Ágiles de Producción.** Enseña todo lo relacionado con la organización de proyectos de desarrollo de software.
- **Proyecto I.** Este proyecto introdujo el desarrollo utilizando el entorno de desarrollo Unity, también usado en este trabajo.
- **Proyecto III.** Enseña a estructurar adecuadamente los proyectos e integrar APIs externas.
- **Estructuras de Datos y Algoritmos.** Saber como manipular datos es fundamental en este proyecto, en el cual se gestiona mucha información.
- **Tecnologías de la Programación de Videojuegos.** Enseña la correcta aplicación de los patrones de diseño comunes en este tipo de desarrollo como el Observer, Singleton, Update Method o Game Loop.
- **Videojuegos para Dispositivos Móviles.** Da las bases de desarrollo para Android en el entorno de desarrollo Unity.
- **Usabilidad y Análisis de Videojuegos.** Esta asignatura enseña cómo diseñar y afrontar pruebas con usuarios y realizar formularios.

Un desarrollo de estas características no es posible sin invertir tiempo en comprender la dificultad de la tarea y preparar las aptitudes necesarias. Por ello, se han realizado varios cursos externos a la universidad. Es muy recomendable la asignatura optativa de Desarrollo en Realidad Virtual, que no pudo ser cursada.

Se empezó centrando el aprendizaje sobre el desarrollo de aplicaciones de RV en Unity, cursando "VR Development Fundamentals With Oculus Quest 2 and Unity" (DEMİRBAŞ, 2020b) y "Multiplayer Virtual Reality (VR) Development With Unity" (DEMİRBAŞ, 2020a), ambos impartidos por Tevfik Ufuk a través de la plataforma de enseñanza Udemy. Estos cursos aportaron las habilidades necesarias para empezar a programar la base necesaria y así exportar la aplicación a los equipos de RV.

El siguiente paso requería obtener un entendimiento inicial de la Lengua de Signos, con este fin se realizó el curso "Lengua de Signos Española A1" de la UNED (Uned, 2017), a través de la plataforma Signo Campus. Esto fue fundamental para entender cuál es la complejidad real de la tarea, tener claros qué retos plantea y cuáles son las limitaciones tecnológicas que aplican al caso.

1.4. Estructura del trabajo

El presente trabajo se estructura como sigue:

- Capítulo 1 de introducción, donde se presenta el propósito de este Trabajo de Fin de Grado, el contexto en que se realiza, la motivación, alcance y que asignaturas de la carrera tienen relación con este.
- En el Capítulo 2 se revisa el estado de la cuestión, viendo el momento que vive la realidad virtual, qué supone ser un usuario con discapacidad auditiva en el mundo de los videojuegos y la evolución del reconocimiento de gestos.
- En el Capítulo 3 se presentan los principales objetivos del trabajo, así como una explicación de todos los requisitos necesarios para el desarrollo de la herramienta.
- En el Capítulo 4 se explica la metodología seguida para la realización del proyecto, además de una enumeración y justificación de las herramientas usadas.
- En el Capítulo 5 se realiza una explicación detallada del diseño e implementación del proyecto, así como de los distintos obstáculos y problemas con los que se ha tenido que trabajar para su completo desarrollo.
- En el Capítulo 6 se recopilan las opiniones acerca del proyecto y pruebas realizadas con usuarios reales.
- En el Capítulo 7 se da una conclusión al proyecto, repasando si se han cumplido las propuestas y objetivos previamente definidos, así como planteando posibles ampliaciones y líneas de trabajo a futuro.

Capítulo 2

Estado de la cuestión

El propósito del proyecto es diseñar una herramienta capaz de facilitar la comunicación entre personas con escucha y aquellas con deficiencia auditiva, cuando estas interactúan en entornos sociales de realidad virtual, ya sean juegos u otro tipo de experiencias. La intención es que este código se pueda encapsular en una herramienta que pueda ser integrada en cualquier juego de RV, siendo usada fácilmente por los desarrolladores.

Se aportan dos ejemplos con los cuales probar la tecnología. El primero es una escena con una zona de testeo del reconocimiento de signos y la conversión de voz a texto. El segundo ejemplo consta de una escena que se puede compilar directamente en una aplicación que cargar directamente en Unity. En ella se puede jugar en un pequeño entorno multijugador para poner a prueba los sistemas entre dos jugadores conectados en línea. Antes de entrar a hablar del desarrollo de la herramienta, es importante hablar de todos los conceptos que se van a tratar. Se da un breve repaso a la historia reciente de la Realidad Virtual, definiendo el concepto de mundos virtuales y en que se diferencia de los videojuegos multijugador clásicos. Además, se indaga en los distintos estudios sobre las tecnologías utilizadas en el proyecto, principalmente la captura de movimiento de manos y los motores de conversión de lenguaje hablado a texto.

2.1. Mundos virtuales y juegos sociales

Los juegos de rol multijugador fueron los primeros en ofrecer mundos abiertos que permitieran tener un avatar con el que relacionarse con el de otros jugadores. Ultima Online(1997), RunesCape (2001), Ragnarok Online (2002) o World of Warcraft (2004) son ejemplos de juegos que tuvieron muchísimo éxito entre la comunidad rolera. En estos se disfruta de una experiencia en la que se pueden formar grupos, gremios, alianzas y guerras entre jugadores. Pero más allá de eso, uno se puede sentar en la ciudad a hablar, o visitar algún escenario bonito. La experiencia trasciende las mecánicas, se

vuelve una forma más de socializar y expresarse a través del ciberespacio. Esta filosofía trascendería los juegos de rol y daría nacimiento a los mundos virtuales.

Los mundos virtuales son entornos sin apenas mecánicas, a diferencia de los videojuegos clásicos. Su principal cometido es generar un entorno donde se antepone la relación social entre los jugadores. Algunos optan por simular situaciones del día a día, otros aportan minijuegos para entretener a los usuarios, algunos incluso permiten la modificación de un escenario personal, que después puede ser compartido con la comunidad. En resumen, son redes sociales donde el jugador tiene un avatar con el cual explorar un entorno e interactuar con él junto al resto de jugadores.

Este tipo de juegos se popularizaron a principios de siglo, tras la llegada de internet de forma masiva a los hogares de todo el mundo. En esta primera década se publicaron los títulos más famosos y conocidos, gozando de comunidades grandes y de mucho éxito. Empezar poniendo por ejemplo Habbo Hotel (Oy, 2022), lanzado en agosto del año 2000.

La mayoría de Mundos Virtuales permiten personalizar la apariencia de tu propio avatar. La intención es permitir al usuario proyectarse sobre el personaje y trasladarle a estos mundos. Al igual que muchos juegos de rol online multijugador masivos (MMORPG de ahora en adelante por sus siglas en inglés), los mundos virtuales se monetizan a través de suscripciones o microtransacciones, bien sea para adquirir bienes digitales como accesorios para resaltar entre los jugadores u obteniendo acceso a zonas exclusivas. Habbo Hotel explota estas estrategias, creando toda una escalera social dentro del juego, donde la personalización de los espacios personales y la apariencia del avatar repercuten en la popularidad del usuario.



Figura 2.1: Un sala de Habbo Hotel que reproduce una playa.

Otro ejemplo interesante de Mundo Virtual fue el caso de Second Life (Junio 2003).

Second Life (Linden Research, 2022), como su nombre sugiere, pretende crear un mundo paralelo donde sus usuarios tengan una segunda vida con su casa, un puesto de trabajo y zonas de ocio. Todo ello apoyado sobre un mercado abierto con su propia divisa. Dada la madurez del concepto, este juego está restringido a usuarios mayores de 18 años. La llamada Linden Dollar, en referencia al estudio de desarrollo, tiene conversión directa con el dolar real, oscilando entorno a 248 linden dollar por 1 dolar americano real. Se estima que los jugadores llegaron a ingresar unos 60 millones de dolares y que este mundo llegó a amasar un PIB de unos 500 millones de dolares.

Este mundo permite crear y comerciar con propiedad virtual, utilizando herramientas propias de diseño 3D, así como lenguajes de programación propios para generar scripts que permiten darle comportamientos a estos recursos virtuales llamados "prims". La propiedad intelectual inicialmente esta protegida por los términos de servicio de la plataforma, aunque con el tiempo estos derechos fueron retirados. Los bienes inmobiliarios también tienen mucho peso, se pueden poseer terrenos y alquilar viviendas. Esto llegó a generar una élite de jugadores, aventajados terratenientes que llenaban sus bolsillos especulando con bienes inmuebles virtuales. Varias empresas realizaron encuentros y eventos virtuales dentro de Second Life, así como conciertos de grandes grupos como los irlandeses U2.



Figura 2.2: Second Life.

Otro exitoso mundo virtual fue creado teniendo en mente a los más pequeños de la casa. Club Penguin (Disney, 2022) (Octubre 2005) ofrece un entorno seguro, lleno de minijuegos donde los más jóvenes pueden relacionarse y jugar en línea. Comprado por Disney en 2007, su popularidad explotó y llegó a contar con una enorme base de jugadores. Se caracteriza por un férreo control de la comunicación entre jugadores y un flujo constante de nuevo

contenido. El acceso a esta plataforma es libre, aunque existe una suscripción pagada que ofrece más opciones de personalizar el avatar del jugador, además de acceso a ciertos minijuegos, mascotas y otros.

A grandes rasgos se puede ver que estos mundos comparten una serie de puntos clave: la proyección del jugador en el mundo a través de un avatar personalizable con el cual expresarse, libertad de movimiento por un entorno virtual y un foco centrado en la socialización por encima de la gamificación. Dada la naturaleza de estos entornos, la plataforma por excelencia era el PC y la comunicación entre jugadores se da mediante el interfaces de chat de texto, donde se teclea el mensaje a enviar por distintos canales de mayor o menor difusión en el mundo virtual. Este punto es fundamental, dado que en este caso de estudio se pierde por completo el concepto del teclado.



Figura 2.3: Club Penguin.

2.2. Realidad virtual

Durante la siguiente década (2010 en adelante) los mundos virtuales perdieron fuerza y fueron cayendo en el olvido. Las distintas plataformas fueron cerrando o restringiendo sus servicios hasta el punto de no ser atractivos para el público. El boom de las redes sociales como Facebook, Twitter, Youtube han introducido nuevas formas de conocer gente en las redes de un modo más simple y eficaz.

En 2010, Palmer Luckey iba a lanzar una carrera tecnológica que llevaba dormida desde los 90 en el mundo de los videojuegos. Oculus Development Kit 1 fue presentado como una plataforma que buscaría revivir la fiebre por la realidad virtual. La tecnología permitía por primera vez, con un casco no muy aparatoso conectado a un ordenador potente, traer la realidad virtual a casa.

Una comunidad de desarrolladores y entusiastas se concentró para darle un empujón, sacando a la luz una plétora de demostraciones técnicas con las que experimentaban el límite de la tecnología. El buen recibimiento llamó la atención de grandes empresas como Steam, Sony o incluso Facebook, que acabó comprando la firma Oculus en 2014. Gigantes del desarrollo como John Carmack se involucraron en el proyecto y finalmente salió a la venta el primer producto enfocado al usuario final, llamado Oculus Rift, en 2016.



Figura 2.4: Primeros modelos de Oculus.

Steam vio el gran potencial de esta tecnología y se alió con el fabricante de dispositivos móviles HTC. Juntos desarrollaron el dispositivo HTC Vive, presentado en el Mobile World Congress de 2016, con prestaciones que superaban a los modelos de Oculus, aunque a mayor precio de salida. Este modelo exclusivo de PC requería una potencia gráfica enorme, aunque a cambio tenía la mejor respuesta y calidad de imagen.



Figura 2.5: HTC Vive de Steam.

Por su parte Sony sacó a la venta Playstation VR como complemento a Playstation 4. A pesar de su menor potencia, el modelo de Playstation gozó de gran éxito, ya que contaba con uno de los mayores catálogos de juegos en su época. De pronto, el mercado estaba repleto de dispositivos de Realidad Virtual, nacía un nuevo paradigma en el videojuego.

Sin embargo, la Realidad Virtual seguía siendo percibida como un ni-



Figura 2.6: Modelo comercial Playstation VR.

cho. Esto vino propiciado por varios motivos: su elevado precio, un catálogo limitado carente de juegos triple A desarrollados de forma nativa, la dificultad de conexión y altos requisitos tecnológicos. Reflejo de esto es el éxito de Playstation VR. Fue el dispositivo con más unidades vendidas y conseguía paliar los problemas descritos arriba. Su precio era considerablemente más bajo que el de sus competidores, 499 dólares en su lanzamiento. En comparación resaltan los 699 dólares de las Oculus Rift o los 799 dólares de HTC Vive. Además, el usuario no tenía que preocuparse de configurar el dispositivo, pues solo necesitaba conectarlo a la consola. Con los beneficios que una plataforma cerrada aporta, se consiguió un catálogo de juegos amplio e interesante. Sony contaba con la plataforma con más usuarios y esto atrajo a los desarrolladores. Con ello, PlaystationVR ha conseguido vender más de 5 millones de dispositivos.

Siguiendo el camino marcado por Sony, Facebook preparaba el lanzamiento de una plataforma que explotara esos puntos fuertes, e incluso los mejorara. Fue en Septiembre de 2018 cuando se presentó el primer dispositivo completamente autónomo de Realidad Virtual, que supuso una revolución, Oculus Quest, hoy en día llamadas Meta Quest.



Figura 2.7: Meta Quest.

Meta Quest es un dispositivo Android completamente autónomo. No requiere ser enchufado a un computador o consola para jugar y su sistema de rastreo de dentro hacia fuera no necesitaba cámaras o balizas de posicionamiento. La simplicidad de colocarse los casos, definir una zona de seguridad y jugar coronaba a las Quest como la experiencia más amigable de cara al usuario. Cuenta con su propia tienda donde Facebook hace de publisher, copiando el modelo de Steam. El HUB principal de la consola permite gestionar el lanzamiento de juegos y aplicaciones de forma sencilla. El único inconveniente es que, al ser un dispositivo autónomo, su potencia palidecía frente a los dependientes de un PC o consola.

El rotundo éxito y favorable recibimiento hizo palidecer a Oculus Rift, la versión de los cascos dependiente de PC, más aún cuando se anunció el cable Oculus Link que permitía conectar Quest a un ordenador para aprovechar la potencia gráfica y jugar a títulos exclusivos de PC, convirtiendo a Quest en un dispositivo híbrido.



Figura 2.8: Meta Link permite utilizar Quest para jugar a juegos de PCVR.

Su precio de salida fue de 399 dólares en su versión de 64 Gb de memoria interna y 499 dólares la de 128 Gb en mayo de 2019. No hay datos oficiales de ventas, pero según los cálculos sobre los reportes de beneficios de Facebook respecto a las fechas de lanzamiento y por cuartos, contrastando con las ganancias obtenidas de la tienda exclusiva de Meta Quest, se estima que en su primer año vendió entre 600.000 y 800.000 unidades, generando más de cien millones de dolares en venta de contenido para esta plataforma.

A partir de este punto la Realidad Virtual se consolidaba como un sector del mercado a tener en cuenta, las empresas seguían empujando la tecnología a velocidades de vértigo. En Junio de 2019 Steam presenta su buque insignia, Valve Index. Este cuenta con un panel con la mayor resolución hasta el momento (1440x1600 píxeles por ojo), mayor ángulo de visión (120 a 130°) y tasa de refresco (120 hz). Además destacaba su sistema de sonido, con altavoces a la altura de las orejas del usuario y unos mandos con capacidad

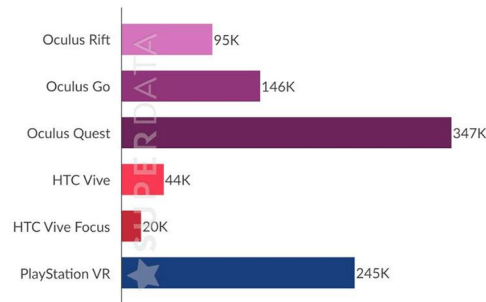


Figura 2.9: Comparación de ventas estimadas. Por Superdata Research.

de detectar la posición relativa de los dedos, sujetándose a la palma con una cinta, eliminando la necesidad de agarrar el mando con los dedos.



Figura 2.10: Valve index en acción.

La carrera tecnológica del hardware iba a un ritmo frenético, pero parecía que el software se quedaba atrás. A pesar de que ya existían juegos muy famosos como Beat Saber o Super Hot, no había ninguna experiencia pura para la realidad virtual, solo adaptaciones de grandes títulos como Resident Evil VII, que permitía experimentar el juego en primera persona. Valve cambiaría esto, poco después de presentar sus Index, aparece el primer título de gran calibre exclusivo para la realidad virtual: Half Life Alyx. Este título se desarrolló con la intención de explotar al máximo las capacidades de Valve Index para maximizar la inmersión del usuario y sirvió como otro gran reclamo, siendo su salida en Marzo de 2020.

La generación actual (Meta Quest 2, HP Reverb G2, Valve Index)

2.2.1. Mundos virtuales y realidad virtual

En 2017 nace la que a día de hoy es la aplicación social más popular en la Realidad Virtual, VRChat.



Figura 2.11: Cartel promocional de Half Life Alyx.

2.12



Figura 2.12: Imagen promocional de VRChat.

VRChat comparte la filosofía vista previamente en otros mundos virtuales. La capacidad de utilizar avatares personalizables, la navegación libre por un entorno virtual y un enfoque puramente social. Esta vez, desde un punto de vista en primera persona, el jugador puede ver su cuerpo digital, puede moverse en el espacio, agarrar objetos y charlar cara a cara con otros de viva voz. La inmersión es casi absoluta, las ventanas de chat y el teclado desaparecen en favor de la comunicación por voz y determinada por la cercanía, como sucede de forma natural.

2.3. Accesibilidad en videojuegos

En España, la Ley Orgánica 51/2003 (de España, 2003) define el derecho a la igualdad de oportunidades de las personas con discapacidad. Este

derecho declara que debe haber una ausencia de toda discriminación hacia este colectivo, de forma directa o indirecta, en todos los ámbitos sociales. Los videojuegos, sobre todo aquellos con un componente multijugador, son elementos culturales en los cuales se desarrollan actividades sociales. Por lo tanto, es responsabilidad de los desarrolladores intentar crear experiencias accesibles para todos.

En 2018 los miembros de la Universidad de Alcalá publicaron un documento Aguado-Delgado et al. (2020) en el que dan un repaso a las publicaciones relacionadas con accesibilidad. En el mismo determinan que se ha demostrado poco interés por parte de los desarrolladores de crear títulos que puedan ser jugados por cualquier usuario. Sin embargo, en los últimos años han empezado a emerger toda una serie de propuestas e investigación relacionada con esta.

Bierre et al. (2005) realiza un estudio sobre las herramientas de accesibilidad presentes en juegos utilizados por jugadores con discapacidades. Mediante una encuesta online, preguntan a jugadores con discapacidades a qué juegos juegan para estudiar sus géneros y qué herramientas de accesibilidad tienen implementadas. Destacan que un pequeño porcentaje de estos juegos son multijugador y que la mayoría de herramientas se centran en problemas visuales, solo un pequeño porcentaje da ayudas para usuarios con baja audición o sordas.

Yuan et al. (2011) estudia las distintas discapacidades que sufren los jugadores y analiza las herramientas utilizadas en varios juegos. Para personas con discapacidad auditiva expone como ejemplo los títulos XIII, The Sims, Doom3 CC, Torque CC, Zork: Grand Inquisitor, Half Life 2, Sin Episodes, Smile y Copycat. En estos se utilizan las siguientes herramientas:

- Sonidos transcritos: como si de un comic se tratara, XIII permite mostrar los sonidos como bocadillos de texto.
- Marcar visualmente emisores de sonido: Los Sims muestran pistas visuales para estos elementos, como notas saliendo de una radio. Otros utilizan radares de sonido.
- Subtítulos para sordos: conocidos en inglés como Close Caption (CC) Subtitles, estos subtítulos añaden al diálogo cualquier sonido relevante en escena, como el sonido de una puerta abriéndose. Cada vez es más común ver este tipo de subtítulo en juegos comerciales como The Last of Us Part II.

Ian Hamilton presenta su ponencia (Hamilton, 2019), durante el evento Game Developers Conference (GDC) llevado a cabo en San Francisco, California, en 2019. En este habla de lo importante que es para todos el uso de subtítulos y la mala implementación de los mismos por parte de industria de los videojuegos. Declara tres grandes problemas que suelen ser el foco de

quejas por parte de los usuarios: tamaño, contraste y cantidad de texto en pantalla. Habla de una serie de pautas y recomendaciones para mejorar cada uno de estos problemas, utilizando ejemplos correctos e incorrectos, siempre sugiriendo otorgar opciones al usuario para ajustar la experiencia a su gusto y necesidades. Estas pautas, inspiradas en las propias de la BBC(BBC, 2021) son:

- **Tamaño:** Los textos deben tener un tamaño adecuado y configurable.
- **Contraste:** Dar la capacidad de resaltar el texto del subtítulo por encima de la escena, usando trazos y más preferiblemente cajas de texto con transparencia regulable.
- **Cantidad de texto:** Un máximo de 2 líneas (3 en excepciones) con un límite de 38 caracteres por línea.
- **Precisión:** Evitar erratas, diferencias de diálogos, pasar los subtítulos por una revisión de calidad.
- **Comprensión:** Asegurarse de que todas las partes del juego están subtituladas. Tener los subtítulos activos por defecto o dar la opción de hacerlo en el momento.
- **Centrado:** Colocar los subtítulos en el centro de la parte inferior de la pantalla.
- **Añadir desde abajo:** Toda nueva línea de texto añadida al subtítulo debe posicionarse debajo de las anteriores.
- **Indicar quien habla:** Poner el nombre del personaje que dice la frase. El color puede ser útil, pero se dejan de lado a personas daltónicas.
- **Mostrar dirección:** dar pequeñas indicaciones visuales del origen del sonido subtulado, como una flecha que apunte en esa dirección.
- **Dar tiempo:** es importante dar suficiente tiempo para leer los subtítulos. Se recomienda 1 segundo para subtítulos de una línea y de 2 a 2,5 segundos para subtítulos más largos, teniendo en cuenta la pauta de cantidad de texto previa.
- **Fuentes claras:** la fuente utilizada para el subtítulo debe ser fácil de leer, como Clear sans-serif font. Utilizar mayúsculas de forma normal, no forzar textos en mayúsculas.
- **Subtitular sonidos importantes:** agregar en los subtítulos aquellos sonidos que tengan relevancia para el jugador como apertura de puertas, sonidos mecánicos, explosiones, disparos, etc.

- **Opciones:** Dar la capacidad de modificar cada uno de los parámetros desde el menú de configuración del juego.

Ian describe a *Farcry: New Dawn*, *The Division 2*, *Shadow of the Tomb Raider* como grandes ejemplos de una correcta implementación de herramientas de accesibilidad para personas con dificultades auditivas. Finalmente, cierra la conferencia hablando de la Realidad Virtual. Plantea las dificultades de pasar a un mundo tridimensional, donde hay que pensar mejor como presentar los subtítulos, evitar la oclusión de elementos, manejar el tamaño, distancias con el emisor y localización de estos. El mejor modelo planteado hasta ahora es uno en el que los subtítulos están colocados en la posición emisor, mientras este permanezca en la línea de visión. En caso contrario, que el subtítulo generado por el emisor aparezca en un espacio definido dentro del campo de visión del jugador, indicando la dirección de su origen.

2.3.1. Discapacidad auditiva y realidad virtual

Helping Hands (Hands, 2022) es una comunidad fundada en el juego VRChat por un pequeño grupo de personas, algunas oyentes y otras con discapacidad auditiva, que querían conocer otros jugadores en las mismas circunstancias o curiosos por aprender lengua de signos, para así crear una comunidad cercana de amigos que se entendieran entre ellos a través de la lengua de signos.

Esta comunidad se ha expandido con el tiempo, originalmente se comunicaban con ASL (Lengua de Signos Americana), pero a día de hoy se encuentran usuarios de todo tipo de Lenguas, como la lengua de signos japonesa (JSL), coreana (KSL), filipina (FSL) o británica (BSL). Además se suelen organizar clases básicas de lengua de signos, impartidas únicamente por sus miembros con discapacidad auditiva e interpretes, que cubren la adaptación a realidad virtual de cada lengua de signos, que debe adaptarse a los movimientos que permite realizar cada mando de realidad virtual, los cuales cambian de una plataforma a otra y no esta estandarizado.

2.4. Reconocimiento de gestos

El reconocimiento de manos ha sido un tema ampliamente estudiado desde hace décadas y ha tenido toda una serie de aproximaciones que han intentado hacer uso de la tecnología disponible en cada momento para intentar trasladar la posición y gestos de una mano real a un entorno virtual. Esta sección busca hacer un breve repaso a distintos estudios sobre la materia, qué soluciones se han dado y cuáles son los problemas que estas presentan.

2.4.1. Reconocedores visuales

Este método de captura es el menos invasivo, no requiere marcadores, sensores ni guantes. Toma como entrada de datos una imagen, generalmente frontal, de las manos a capturar. Este método suele requerir un entorno muy controlado donde la mano sea fácilmente distinguible del fondo y cuente con una iluminación adecuada. Algunos ejemplos de publicaciones que utilizan esta metodología son Rehg y Kanade (1994a), Rehg y Kanade (1994b), Heap y Samaria (1995), Nirei et al. (1996), Schlattman y Klein (2007).

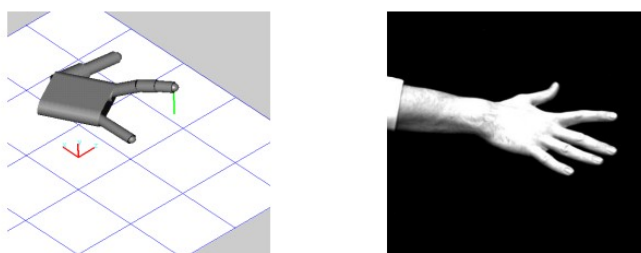


Figura 2.13: Imagen capturada y proyección al modelo 3D de la propuesta Digit Eyes.

2.4.1.1. Reconocedores físicos

- **Cámaras simples** Capturan una sola imagen de la mano. Suelen ser cámaras estacionarias que apuntan a un fondo homogéneo y que permiten identificar la mano fácilmente. Melax et al. (2013), Pan et al. (2010)
- **Cámaras Binoculares** Capturan dos imágenes con intención de obtener información del gesto desde dos ángulos distintos o bien extrapolando profundidad Nirei et al. (1996), Schlattman y Klein (2007). La siguiente figura muestra una captura del estudio Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking Rehg y Kanade (1994b), donde se utiliza la profundidad de campo (Depth of Field, DOP):
- **Microsoft Kinect** Kinect es un sistema de reconocimiento de profundidades que salió a la venta originalmente como un periférico de Xbox 360 en 2010. Sharp et al. (2015), Adewale y Olamiti (2018)
- **Leap Motion** Este sensor, que podía ser utilizado en ordenadores y dispositivos de realidad virtual a modo de periférico, fue lanzado al público en 2013. Koul et al. (2016)

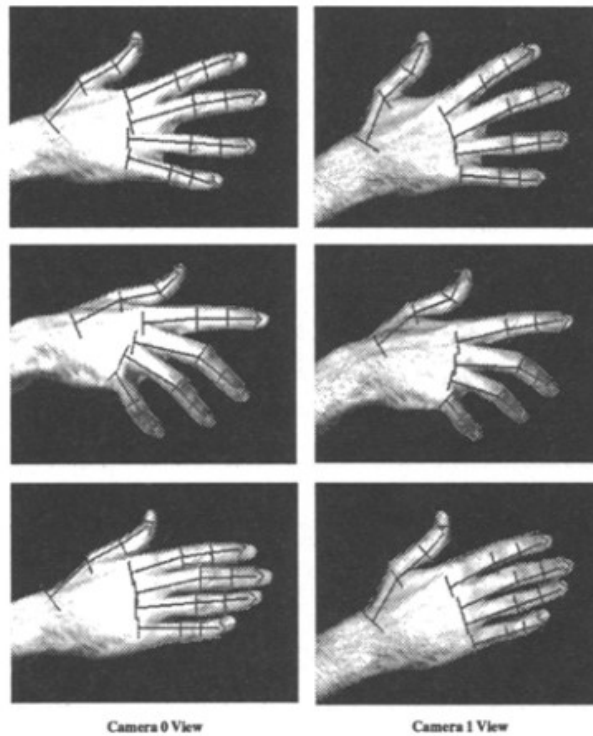


Figura 2.14: Un ejemplo de captura con cámaras binoculares.

2.4.2. Captura con marcadores o sensores

Este tipo de captura suele utilizar cámaras externas que rastrean los marcadores colocados sobre la superficie a la que se desea hacer un seguimiento. Es una opción descartada actualmente debido a que es muy invasivo y requiere que el usuario lleve constantemente un dispositivo o prenda.

- **Guantes con sensores** Su origen se establece en el motion capture, los guantes estos cuentan con unas esferas reflectoras que permiten rastrear su posicionamiento, sin embargo aportan poca información detallada de elementos más pequeños como los dedos.
- **Guantes con marcadores** Wang y Popović (2009), El-Sawah et al. (2008) Guantes con distintos marcadores que el software de seguimiento es capaz de rastrear.
- **Pulseras magnéticas** Ma et al. (2011) Esta aproximación poco ortodoxa lee la diferencia de intensidad en las señales magnéticas dependiendo del gesto realizado.

2.4.3. Algoritmos aplicados al reconocimiento

También hay una gran diversidad de aproximaciones a la hora de afrontar el reconocimiento de gestos, aquí unos ejemplos.

- **Algoritmos Genéticos (GAN)** Mueller et al. (2018)
- **Redes Neuronales (RNs)** Adewale y Olamiti (2018), Koul et al. (2016)
- **Inverse Kinematics** Schröder et al. (2014)
- **Modelos Jerárquicos Bayesianos** Stenger et al. (2006)

Capítulo 3

Objetivos y especificación

El propósito de este proyecto, como se refleja en el Capítulo 1, es desarrollar una herramienta de accesibilidad que pueda ser integrada en cualquier entorno social de RV y permita derribar las barreras de comunicación entre personas oyentes y personas con alguna discapacidad de tipo auditivo.

Se pretende establecer un canal común de texto a la que ambas partes puedan tener acceso, transcribiendo los mensajes que cada comunicador envía por el medio de comunicación que le resulta más natural. Así pues, tanto en lengua española como la lengua de signos española, deben ser transcritas a texto. Con ello se quieren derribar dos barreras de comunicación a la vez, desde el punto de vista de la persona con la discapacidad. La primera sería la barrera "Dentro a Fuera" por la cual no es posible expresar lo que se quiere comunicar a una persona oyente que desconoce la Lengua de Signos Española. La segunda sería la barrera "Fuera a Dentro" por la cual la persona con discapacidad auditiva no puede entender bien lo que una persona oyente trata de comunicarle mediante su voz.

3.1. Objetivos propuestos

Una vez conocido cuál es el estado de la cuestión, con las oportunidades y retos que esto implica, este proyecto se propone una serie de objetivos concretos:

- Diseñar y desarrollar un entorno social en forma de aplicación de RV para dispositivos de última generación.
- Implementar un componente centralizado para gestionar el canal común de comunicación en tiempo real basada en texto.
- Integrar en ese componente un sistema de reconocimiento de voz en idioma español con conversión a texto.

- Integrar en ese componente un sistema de reconocimiento de gestos de la Lengua de Signos Española con conversión a texto. Este sistema tendrá una base de datos de gestos que el usuario podrá gestionar directamente, para tener un mayor control sobre su repertorio de signos.
- Validar la utilidad y usabilidad de la aplicación mediante pruebas con usuarios reales.

3.2. Especificación

La idea inicial del proyecto se ha definido en la fase inicial de especificación. El objetivo estaba claro: crear una transcripción de LSE a texto y Voz a texto, generando un canal común donde una persona con discapacidad auditiva y otra oyente pudieran comunicarse.

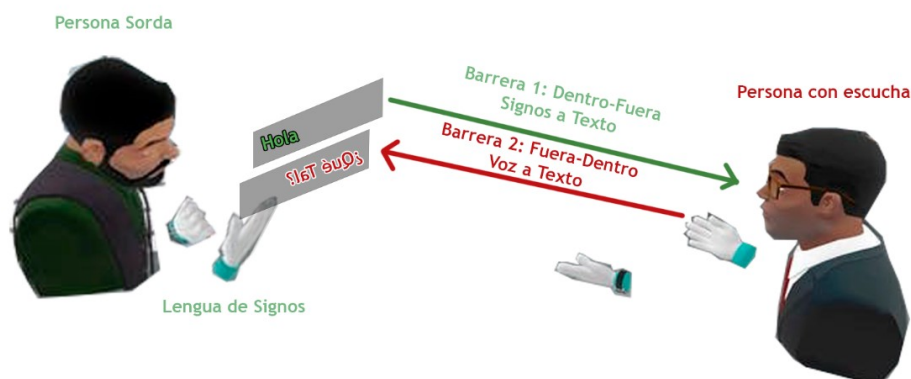


Figura 3.1: Concepto inicial presentado en la fase de diseño.

3.2.1. Funcionalidades

La idea es generar una interfaz común entre los dos usuarios. Se han valorado varias ideas, como bocadillos sobre los jugadores con lo que acababan de decir, pero eso plantea el problema de qué ocurre si una persona sorda no está mirando al hablante, ya que no puede percibir que alguien se dirige a ella.

Además la persona signante tiene que poder comprobar si el mensaje es correcto antes de transmitirlo, por lo que se necesita una forma de construir la frase, corregirla y decidir en qué momento enviarla al canal común.

Finalmente se optó por una interfaz personal de cada usuario, en la cual se proyectara la transcripción de todo lo dicho y signado en la sala. Esta interfaz debe seguir al usuario, estar colocada en una posición que no moleste demasiado, ya que obstruirá parte del campo de visión del jugador.

Esta interfaz contará con 3 ventanas, una naranja y grande que mantendrá el registro de lo dicho en el chat. Una gris donde se irá realizando la construcción de la frase y finalmente una verde que mostrará el estado del reconocimiento en tiempo real.

Colocar objetos visuales adecuadamente en realidad virtual no es tan simple como hacerlo en el espacio de una pantalla. Hay que tener en cuenta el ángulo de visión del usuario, además de que la interfaz debe ser hija de algún elemento del espacio de rastreo para que su posición se mantenga a pesar del movimiento. Por suerte Oculus Integration, contiene un objeto OVRPlayer que sitúa al jugador dentro de la escena virtual, y en este se puede encontrar una posición central a los ojos que sirve como ancla para estos elementos.



Figura 3.2: Diseño de la interfaz.

3.2.1.1. Chat

El chat es la mayor ventana de la interfaz de usuario. En ella se va publicando el registro de lo que se habla. Para poder identificar mejor cuál es la fuente del texto introducido, el gestor de textos añade un icono a la izquierda del texto indicando si la fuente es voz o lengua de signos.

3.2.1.2. Buffer de escritura

En la ventana gris de la interfaz se puede construir la frase deseada utilizando lengua de signos. Existen unos gestos especiales de tipo comando que se describen en las siguientes secciones y que permiten realizar ciertas acciones como modificar lo escrito o enviar la frase construida al chat.

3.2.1.3. Información del reconocimiento

La ventana inferior informa del estado del reconocimiento. Puede mostrar varios mensajes, dependiendo de si hay algún problema en la detección de manos, el signo es reconocido o si no reconoce nada.

3.2.2. Limitaciones

A continuación se explican algunas limitaciones del proyecto planteado.

3.2.2.1. Extensión

Dada la extensión de la lengua de signos española, no se plantea crear una herramienta que contenga todos los posibles signos, sino una que permita al usuario, capturar, reconocer y usuario gestionar la base de datos de gestos capturados, para así poder añadir gestos poco a poco.

3.2.2.2. Expresividad del avatar virtual

La Lengua de Signos no se compone únicamente de gestos con las manos, sino que además aprovecha distintos gestos corporales, como la inclinación del torso, expresiones faciales o gesticulación de las palabras con los labios. La tecnología de Meta Quest 2 solo permite capturar las manos, no las expresiones faciales o de la boca. Sin embargo, en un futuro cercano es posible que aparezcan dispositivos que incorporen esa funcionalidad de forma nativa, ya que se ha visto interés de compañías como HTC, que ya presentó un periférico llamado Vive Facial Tracker para sus cascos HTC Vive. Meta por su parte planea añadir esta capacidad, además de el seguimiento de ojos, en las futuras Meta Cambria.

3.2.2.3. Ofuscación

El gran problema del reconocimiento de manos es la ofuscación. Este término tiene que ver con el campo visual de las cámaras frontales y su capacidad de reconocer las manos dentro de estas, por lo tanto, no es posible reconocer aquello que la cámara no ve, como los dedos que quedan ocultos detrás de la palma de la mano, o un gesto que realice una mano cubierta por la otra.

3.2.2.4. Construcción avanzada de frases

La LSE no tiene una correspondencia directa con la lengua española, esto implica que hay signos que pueden leerse de varias formas, un ejemplo es el signo Yo/Mi, que depende del contexto de la frase para determinar el significado apropiado. Poder realizar una construcción de frases completa es

todo un reto en sí mismo y no entra en las pretensiones de este trabajo, pero queda anotado como un trabajo a futuro.

Capítulo 4

Metodología y herramientas

4.1. Metodologías de desarrollo

El proyecto se ha llevado a cabo utilizando metodologías ágiles aprendidas en el primer curso de la carrera. Estas metodologías han ido siendo refinadas en los proyectos anuales que han sido realizados en los tres primeros cursos. Debido a la circunstancia de que este proyecto solo cuenta con un desarrollador, metodologías con proyección grupal como las reuniones diarias no se han realizado.

4.1.1. Reuniones quincenales

Se han llevado a cabo reuniones periódicas entre el alumno y director del TFG. Aplicando la metodología Scrum, en cada reunión se presentan los avances al director a modo de revisión, se comparten ideas y establecen objetivos. Las tareas propuestas se añaden a un Backlog, se hace una planificación de cuáles van a desarrollarse en el sprint actual, durante este se implementan y en la siguiente reunión se revisan. Es común comentar otros trabajos relacionados, compartir información relevante al proyecto y hablar del avance dentro del Hito actual, que se explica en la siguiente sección.

4.1.2. Planificación

El proyecto se ha dividido en tres grandes hitos a lo largo del año. Cada hito marca una etapa en el proyecto, cuyas fechas de cierre son: Navidad, inicio de Semana Santa y la etapa de fin de curso, este año a finales de mayo. A su vez, cada hito se ha dividido en tiempos de desarrollo llamados sprints de dos semanas de duración, coincidiendo con las reuniones entre el director del TFG y el alumno que lo ha desarrollado.

- **Hito de Navidad.** Durante este hito se da forma a la idea que quiere desarrollarse, estudiando cuáles son las metas que pueden alcanzarse de

forma razonable. Se preparan las herramientas necesarias y desarrolla un prototipo que sirva de prueba de concepto. También se ha estudiado el estado del arte en materia de reconocimiento de manos y herramientas de accesibilidad para usuarios de lengua de signos, reflejado en el segundo capítulo de esta memoria.

- **Hito de Semana Santa.** El desarrollo de la aplicación se ha concentrado en este periodo de tiempo, a la par que se han llevado a cabo los cursos para obtener las habilidades necesarias.
- **Hito Final.** En este periodo se ha trabajado en depurar una demo final, con todas las funcionalidades y un tutorial que enseñe al usuario a utilizar la herramienta. Se han realizado una serie de pruebas con voluntarios en la facultad de informática y se les ha solicitado rellenar un cuestionario. El resto del periodo se ha dedicado a cumplimentar esta memoria.

4.2. Herramientas utilizadas

Para realizar este trabajo de fin de grado se ha utilizado un amplio repertorio de herramientas, tanto de desarrollo como de documentación, que se nombraran a continuación:

4.2.1. Google Suite

Google ofrece toda una serie de portales dedicados a distintas herramientas y utilidades requeridas en la organización del proyecto. Se enumeran las secciones utilizadas:

- **Google Calendar** sincroniza los horarios de distintas cuentas y ha permitido organizar las reuniones.
- **Google Meet** es un portal web que permite realizar reuniones telemáticas con voz y vídeo.
- **Google Forms** esta herramienta ha sido utilizada para realizar el cuestionario que ha sido distribuido a los participantes de las pruebas en la universidad, también ha sido distribuido por varios canales online.
- **Google Drive** ofrece una nube de almacenamiento de ficheros utilizada para compartir las presentaciones y documentos relativos al proyecto.

4.2.2. Slack

Slack es una herramienta de comunicación, organizado en servidores con distintos canales de texto y voz. A través de este se ha mantenido una comunicación con los miembros de Narratech.

4.2.3. Github

Github es el portal web donde se gestionan los proyectos de desarrollo, enfocado principalmente en aportar un sistema de control de versiones, mediante repositorios.

4.2.4. Github Projects

Este apartado de Github permite gestionar un proyecto aplicando metodología ágil. En este se especifican una serie de columnas, a modo de categorías, en las cuales se cuelgan tareas, organizándolas según el estado de desarrollo de estas. De izquierda a derecha se colocan primero las tareas a desarrollar, seguido por columnas que detallan el estado por el que se irá moviendo dicha tarea. La tarea empezará en un backlog, una vez iniciada por el desarrollador pasa a una fase de "En progreso". Una vez terminada, en un equipo de desarrollo de varias personas, se catalogaría como "En Revisión" y debería ser revisada por los otros miembros del equipo. Como este TFG ha sido desarrollado por una única persona, esa tarea de revisión ha recaído en el director del proyecto. Una vez verificada la tarea, se desplaza finalmente a la columna "Terminado" donde la tarea ya se considera totalmente implementada.

4.2.5. Git for Windows

Git for Windows es una consola de comandos que permite gestionar este sistema de control de versiones, sincronizando el código del repositorio local con el remoto, mediante comandos escritos.

4.2.6. Unity

Unity es el entorno de desarrollo de videojuegos utilizado. Entre muchos otros, permite el desarrollo de aplicaciones para el sistema operativo Android utilizado en Meta Quest.

4.2.7. Visual Studio

Visual Studio es el entorno de desarrollo por excelencia en sistemas operativos Windows. Cuenta con toda una gama de herramientas para el desarrollo de código, teniendo integración con Unity.

4.2.8. SideQuest

Esta herramienta permite comunicar el ordenador con las Meta Quest, además de instalar aplicaciones y gestionar los archivos internos de Meta Quest.

4.2.9. OBS Studio

Utilizado para documentar con vídeo las transmisiones realizadas de las Meta Quest al ordenador.

4.2.10. Notepad++

Editor de texto avanzado, con reconocimiento de lenguajes de programación. Utilizado para gestionar el archivo generado con la base de datos de gestos en formato XML.

4.2.11. Overleaf

Portal web que permite editar proyectos LaTeX, utilizado para realizar la memoria.

4.2.12. Paquetes externos

Durante el desarrollo se han utilizado una serie de paquetes externos, algunos son paquetes de integración necesarios para los objetivos del desarrollo, otros son recursos visuales y modelos 3D gratuitos utilizados en la creación de los escenarios multijugador. Todos han sido obtenidos a través de la plataforma Unity Asset Store.

4.2.12.1. Paquetes de integración

- **Oculus Integration Package.** Paquete de integración de tecnología Meta, necesario para desarrollar juegos de RV en Unity. Este paquete es específico para dispositivos Meta, pero da el soporte necesario para trabajar con la captura de manos. Desarrollado por Meta.
- **Speech Recognition System.** Para el reconocimiento de voz y su conversión a texto, se utiliza el paquete Speech Recognition System (Studio, 2022) del equipo Stendhal Syndrome Studio. Se ha optado por esta herramienta por que aporta varias ventajas, es capaz de capturar y transcribir voz de manera inmediata, reconociendo el idioma español, sin necesidad de estar conectado a la red. Tiene un alto nivel de calidad en su transcripción. Este motor se ha integrado en la solución del proyecto, conectando el texto generado por la captura a las

distintas interfaces de chat. No es necesario utilizar este producto para hacer funcionar el proyecto, cualquier motor de conversión voz a texto puede ser utilizado con la misma finalidad.

- **PUN 2, Photon Engine Multiplayer Backend.** Paquete que habilita el desarrollo de aplicaciones multijugador, dando una interfaz de comunicación con los servidores Photon.

4.2.12.2. Paquetes de recursos visuales y modelos 3D

- **Magic Mirror Lite - Reflection for Unity.** Paquete con elementos reflectantes tipo espejo. Desarrollado por Digital Ruby.
- **Down Town Pack.** Paquete con modelos 3D urbanos de pocos polígonos (Low Poly). Desarrollado por GameMag Creation Studio.
- **Fantasy Skybox.** Paquete de texturas para el skybox del juego. Desarrollado por Render Knight.
- **Free Street Props.** Paquete con modelos 3D urbanos esculpidos con pocos polígonos (Low Poly). Desarrollado por Ribrado.
- **Low Poly Cars.** Paquete con modelos 3D de vehículos de pocos polígonos. Desarrollado por Broken Vector.
- **Mountain Terrain + Rock + Tree.** Paquete con modelos 3D. Desarrollado por CG Creative Sets.
- **Planes and Choppers - PolyPack.** Paquete con modelos 3D de aeronaves. Desarrollado por Alstra Infinite.
- **Realistic Tree 9 [Rainbow Tree].** Paquete con modelos 3D de árboles. Desarrollado por Pixel Games.
- **Low Poly Office Pack.** Paquete con modelos 3D de material de oficina. Desarrollado por Poligonal Mind.
- **Tileable Pack 01.** Paquete con texturas para suelos. Desarrollado por Mafubash.
- **CITY Package.** Paquete con modelos 3D de edificios y calles. Desarrollado por 255 Pixel studios.

Capítulo 5

Diseño y desarrollo de la herramienta

Este capítulo profundiza en el desarrollo de la herramienta, describiendo las partes más importantes del desarrollo de la aplicación que sirve como prueba de concepto de la herramienta propuesta. Se analiza cómo se ha desarrollado el núcleo de la aplicación, que es el motor de reconocimiento de signos.

5.1. Análisis y diseño

Los retos encontrados en el análisis de la especificación son los siguientes:

5.1.1. Ofuscación entre manos

Hay casos en los que los gestos de la Lengua de Signos Española utiliza ambas manos y produce contacto entre ellas. Por desgracia, el reconocimiento de manos de Meta Quest deja de funcionar cuando hay una ofuscación de este tipo entre ambas manos. Este es un límite tecnológico que no se puede superar actualmente. Los signantes seguirán teniendo que adaptar algunos de sus signos para que puedan ser reconocidos por el dispositivo. Es suficiente con separar ligeramente las manos para que no exista dicho punto de contacto.

5.1.2. Ofuscación de los dedos

Los signos de la Lengua de Signos Española están diseñados para ser vistos de frente, pero las cámaras de Meta Quest se sitúan justo detrás del sentido de signado. Esto implica que puede haber ofuscación de los dedos detrás de la palma y algunos signos pueden tener dificultades a la hora de

ser detectados correctamente, ya que la posición de los dedos pasa a ser estimada.

5.1.3. Signos complejos

Algunos signos pueden ser muy parecidos unos con otros, esto puede provocar falsos positivos. Un ejemplo son la O y la S, muy similares. Es importante que a la hora de capturar estos signos, o al percibir que se empieza a obtener falsos positivos entre estos, se debe borrar esas entradas de la base de datos con un editor de texto y se volver a capturarlas intentando remarcar aquello que las diferencia.

5.1.4. Signos con la mano mirando hacia abajo

Signos como la M parecen tener más dificultad a la hora de ser reconocidos por Meta Quest, ya que su algoritmo de captura no tiene tanto entrenamiento en posturas donde la mano apunta hacia el suelo. Incorporando pequeñas modificaciones al signo en la captura se pueden sortear estas dificultades, como por ejemplo cambiando la rotación de la mano ligeramente hasta encontrar un punto donde el reconocimiento es consistente.

5.1.5. Signos en movimiento

Existen Signos que realizan un movimiento que modifica el gesto de la mano. Estos pueden detectarse fácilmente capturando el gesto inicial y final de dicho signo. Sin embargo, el mayor problema se presenta con signos cuyo gesto se mantiene, los huesos no se mueven, pero la posición relativa de la propia mano varía. Estos gestos no son reconocibles actualmente. Un ejemplo es el gesto Z, que es igual al gesto I, lo único que varía es que se dibuja una Z con la punta del dedo en el aire. Queda como trabajo a futuro el agregar un método de reconocimiento para este tipo de signos.

5.2. Interfaces virtuales

Esta aplicación ha sido diseñada para ser utilizada únicamente con el reconocimiento de manos. Por ello es necesario una forma de interactuar con el entorno virtual de la forma más natural posible. Para ello se han desarrollado una serie de botones virtuales que pueden ser accionados de forma física, pulsando con los dedos virtuales.

Para lograr esto, ha sido necesario crear unas pequeñas esferas, capaces de generar colisiones, en la punta de los dedos de la mano virtual. Para que su funcionamiento sea correcto, tienen como componente un script que actualiza su posición con la punta del dedo que tenga configurado. Este script es el TouchBallScript.cs. En la siguiente figura se aprecia la posición de cada uno

de los huesos al activar el renderizado de las esferas. Únicamente la punta de los dedos genera colisiones con los objetos con los que comparte capa.

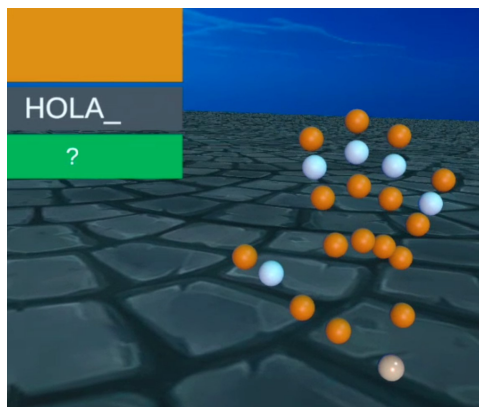


Figura 5.1: Esferas colisión en cada hueso.

5.2.1. Botones

Para los botones se ha creado un prefab llamado VRButton. Este botón sirve de plantilla para cualquier comportamiento deseado. Cuenta con una parte roja que puede colisionar con las manos virtuales. Ese botón tiene un límite de desplazamiento en un único eje, pasado el cual se considera accionado, cambiando de color a verde. El botón siempre volverá a su posición de descanso si no esta siendo tocado.

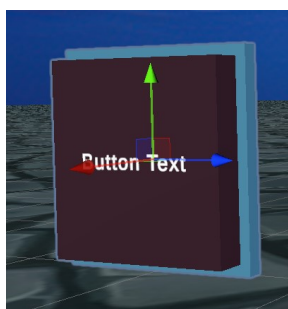


Figura 5.2: Prefab de botón virtual que sirve de plantilla.

Al ser accionado, se puede personalizar qué se quiere que ejecute el botón. Lo único que se debe hacer es borrar el componente plantilla que tiene incluido cuyo nombre es VR_Button_Template. Crear un nuevo script que herede de este e implementar el método OnClick como se desee. La plantilla contiene el código común que maneja el movimiento del botón. Aquellas clases que hereden de él pueden incluir nuevas variables para ser usadas en el editor Unity.

En la siguiente figura se aprecia un ejemplo de una clase que hereda de la plantilla. Todos los botones del proyecto funcionan del mismo modo.

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

Script de Unity (2 referencias de recurso) | 0 referencias
public class LeaveRoomButton : VR_Button_Template
{
    2 referencias
    public override void OnClick()
    {
        MultiplayerManager.Instance.LeaveRoomAndLoadHomeScene();
    }
}
```

Figura 5.3: Código de un botón personalizado que hereda de la plantilla.

5.3. Reconocimiento de gestos

Para poder reconocer los gestos se necesita primero capturar la información relativa a la mano proyectada. Es importante remarcar que la mayoría de juegos y aplicaciones desarrollados en realidad virtual no tienen soporte para captura de manos.

Esto es debido a que en la actualidad se dispone de dos maneras de integrar la realidad virtual un proyecto:

- XR Integration Kit: Este es el kit de integración multiplataforma aportado por Unity. Su filosofía es brindar una interfaz compatible con la mayoría de dispositivos, por eso es el más popular. Sin embargo, no soporta el reconocimiento de manos de Meta.
- Oculus Integration: Este paquete se puede obtener en la Unity Store esa API nativa que da Meta para el desarrollo de aplicaciones VR en Unity. Es el único que soporta el reconocimiento de manos.

5.3.1. Controlador del jugador

Oculus Integration brinda una serie de Prefabs y scripts que dan al desarrollador las herramientas necesarias para crear aplicaciones en VR. Estos prefabs tienen nombres que empiezan con OVR.

- **OVRPlayerController**

El prefab más importante de todos es el OVRPlayerController. Proporciona las cámaras de cada ojo y toda una serie de anclas para incorporar elementos dentro del espacio de reconocimiento.

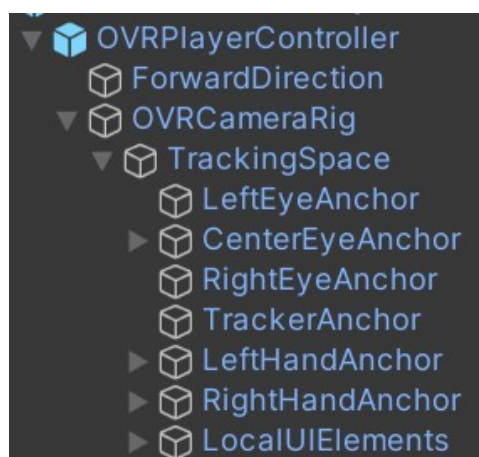


Figura 5.4: Vista de Inspector de Unity del OVRPlayerController.

- **OVRHandPrefab**

Este objeto prefab puede ser incorporado al OVRPlayerController, en las dos anclas de las manos. El mismo prefab puede ser configurado para ser usado como mano derecha o izquierda.

- **OVRSkeleton y OVRHand**

Estos scripts asociados al OVRHandPrefab contienen los datos de cada uno de los huesos que componen la mano. Estos datos son los que son extraídos para realizar el reconocimiento.

5.4. Persistencia

En esta sección se trata el tema de la persistencia. Se explora qué forma final toma la base de datos que almacena los signos capturados, dónde se almacena en el producto y qué información ha sido utilizada para capturar los gestos realizados, configurando las propiedades necesarias de cada signo.

5.4.1. Base de datos

Uno de los métodos más sencillos para serializar los datos en un archivo es convertir la información de las clases en un archivo XML. Es de interés que esta base de datos sea accesible y modificable por los usuarios, tanto dentro como fuera de la aplicación. La intención es mantener una filosofía de transparencia donde cualquier persona pueda acceder a la base de datos para ayudar a ampliar y mejorar el proyecto. Así pues, se utilizarán los paquetes XmlSerializer y TextWriter para serializar la base de datos. La clase encargada de serializar y deserializar la base de datos es Persistence.

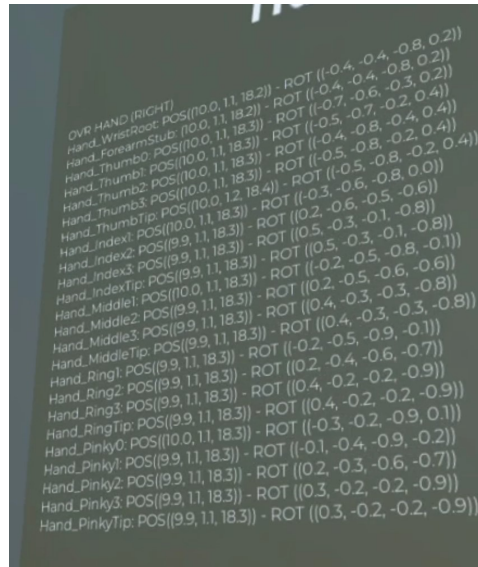


Figura 5.5: Datos de posición de los huesos del OVRHand.

```

3 | <Gesture>
4 | <gestureName>LSE-RH-A</gestureName>
5 | <RHBoneInfo>
463 | <LHBoneInfo />
464 | <usedHand>RIGHT_HAND_ONLY</usedHand>
465 | <gCategory>GESTURE_LETTER</gCategory>
466 | <gPhases>
467 | | <eGesturePhase>GESTURE_SIMPLE</eGesturePhase>
468 | </gPhases>
469 | <singleTranscription>a</singleTranscription>
470 | <composedTranscription />
471 | </Gesture>

```

Figura 5.6: Un gesto en la base de datos.

5.4.2. Ruta de almacenamiento

- En el sistema operativo Windows, la base de dato se guardará como un archivo llamado LSE_DB.xml en la ruta:

C:/Usuarios/NombreUsuario/AppData/LocalLow/
es_ucm_tfg_cardona/TFGCardona/LSE_DB.xml

- Es posible acceder al archivo de la base de datos almacenado dentro de las Meta Quest en la siguiente ruta:

/android/data/es.ucm.tfgcardona/files/LSE_DB.xml

5.4.3. Propiedades de los gestos

La Lengua de Signos Española es un sistema muy complejo de signado, tal y como describe previamente en la sección Retos encontrados. Cada gesto tiene una serie de características bien definidas para ayudar al reconocedor a diferenciarlos. La imagen muestra cuáles son y qué importancia tienen.

```
public struct Gesture
{
    public string gestureName;
    public List<BoneData> RHBoneInfo;
    public List<BoneData> LHBoneInfo;
    public eHandUsage usedHand;
    public eGestureCategory gCategory;
    public List<eGesturePhase> gPhases;
    public string singleTranscription;
    public List<string> composedTranscription;
};
```

Figura 5.7: Estructura de datos que almacena la información de un gesto.

5.4.3.1. Nombre del gesto

Cada gesto tendrá un nombre único a modo de identificador y para construirlo se debe seguir una sintaxis específica. Para entender bien su uso se plantea el siguiente ejemplo, el gesto con nombre **LSE-RH-A-2**. Todo gesto en la base de datos tendrá un nombre que empezará con las siglas de la lengua a la que pertenece, en este caso **LSE** (Lengua de Signos Española). Acto seguido, separado por un guión, aporta información acerca de qué mano es utilizada para realizar el gesto, en este caso **RH**, que implica el uso de la mano derecha (Right Hand). El identificador para la mano izquierda sería LH o si es un gesto que requiere ambas manos, utilizaría 2H. El siguiente parámetro del nombre indica el nombre del signo en cuestión, siendo fieles al diccionario de dicha Lengua de Signos, en este caso la letra **A**. Por último es posible que sea interesante tener un mismo gesto repetido dentro de la base de datos, bien sea por que varios usuarios han almacenado su particular forma de signar dicho gesto. Es importante recalcar que estos signos deben coincidir con lo establecido en el diccionario de la Lengua de Signos, pero el reconocedor de gestos es muy sensible a las pequeñas variaciones que puedan aparecer entre distintos usuarios, debido a ello se permite almacenar más de un gesto para cada signo y este último número permite diferenciar las variantes. Este identificador final se agrega de forma automática cuando un gesto es creado con un nombre ya existente en la base de datos.

5.4.3.2. Datos de los huesos

La siguiente información a almacenar serán los datos de posición y rotación de cada uno de los huesos de la mano o manos utilizadas. Estos datos se presentan en forma de float y son los que se comparan con el gesto reconocido para determinar cuáles tienen menor distancia. Se utilizan dos listas de Structs BoneData, uno por mano, que contienen el identificador interno de cada hueso, su posición y rotación. Únicamente se almacenará la información de la mano o manos utilizadas en el gesto.

```
public struct BoneData
{
    public OVRSkeleton.BoneId id;
    public Vector3 position;
    public Quaternion rotation;
}
```

Figura 5.8: Struct con los datos de un hueso en concreto.

```
<BoneData>
  <id>Hand_Thumb0</id>
  <position>
    <x>-0.0200692657</x>
    <y>-0.0115541453</y>
    <z>-0.0104965251</z>
  </position>
  <rotation>
    <x>0.7068734</x>
    <y>0.374449641</y>
    <z>-0.122005761</z>
    <w>0.587564468</w>
  <eulerAngles>
    <x>67.22579</x>
    <y>136.279465</y>
    <z>94.31285</z>
  </eulerAngles>
</rotation>
</BoneData>
```

Figura 5.9: Posición y Rotación de un hueso en el gesto almacenado.

5.4.3.3. Identificador del uso de manos

Cada gesto contiene un parámetro que determina que manos se utilizan en el signo. Esto ayuda a filtrar gestos durante el reconocimiento y almacenar los datos necesarios según la mano utilizada. Se utiliza un enumerado con este fin:

```
public enum eHandUsage
{
    NOHAND,
    LEFT_HAND_ONLY,
    RIGHT_HAND_ONLY,
    BOTH_HANDS
};
```

Figura 5.10: Enumerado para establecer el uso de manos en un gesto.

5.4.3.4. Categoría del gesto

Los gestos pueden pertenecer a distintas categorías, dependiendo del signo o signos a los que puedan pertenecer. Esto es muy importante en la fase de reconocimiento y validación del gesto, así que se establecen las siguientes categorías:

- **Letra:** este tipo de gestos solo se transcriben como una letra, esto implica que únicamente contarán con una transcripción simple y no se le deben añadir espacios en el buffer.
- **Letra o Palabra:** un mismo gesto puede ser una letra concreta o ser parte del signo de otra palabra o letra completamente distintas. Este tipo de gestos son especialmente complejos y requieren ser marcados para manipularlos adecuadamente en el reconocimiento y validación.
- **Palabra:** estos gestos se transcriben como palabras, por lo que hay que añadir espacios para separarlas de otros gestos en la transcripción.
- **Comandos:** existen un tipo específico de gestos, que no pertenecen a la Lengua de Signos, pero permiten al usuario introducir una serie de comandos para manipular el texto, bien sea para enviar el texto introducido en el buffer al chat, borrar letras, activar o desactivar el reconocimiento de gestos de la lengua de signos o limpiar completamente el buffer. Son procesados de manera distinta y no generan ninguna transcripción.

5.4.3.5. Fase del gesto

Los gestos pueden diferenciarse en 3 fases en función de si es un gesto estático o en movimiento. Los signos estáticos son los más sencillos de reconocer, por ello se les clasifica como fase simple. El problema viene cuando se reconoce un signo que se realiza a través de un movimiento. Para identificar


```
public enum eGestureCategory
{
    GESTURE_WORD,
    GESTURE_LETTER,
    GESTURE_LETTER_OR_WORD,
    GESTURE_COMMAND,
    NONE
}
```

Figura 5.11: Enumerado para establecer la categoría del gesto.

estos signos, se almacenan dos gestos relativos al signo, uno que contiene el estado de las manos en su posición inicial (BEGIN) y otro que contiene el estado de las manos al final del signo (END). Hay que tener en cuenta que un mismo gesto puede ser un signo por si solo mientras esta estático, pero a su vez ser parte del inicio o final de otro signo en movimiento. En estos casos, si el gesto es validado en su fase estática, utilizando la transcripción simple, en caso contrario, se utiliza el primer elemento almacenado en la transcripción compuesta.

```
public enum eGesturePhase
{
    GESTURE_SIMPLE, //
    GESTURE_BEGIN, //
    GESTURE_END, //
    NONE
}
```

Figura 5.12: Enumerado para establecer la Fase del gesto.

5.4.3.6. Transcripción simple

La transcripción simple es una variable tipo String que contendrá la transcripción directa del signo. Si un gesto ha sido validado como simple, siempre se usa la transcripción simple.

5.4.3.7. Transcripción compleja

La transcripción compleja almacena una serie de variables tipo String, las cuales almacenan las posibles transcripciones de un Signo según el contexto de la frase para aquellos signos como Yo/Mi.

5.4.4. Carga, captura y almacenamiento

Al inicio de la ejecución la clase Persistence intentará encontrar y deserializar el archivo de base de datos. Si no lo encuentra, simplemente generará una base de datos vacía lista para ser rellenada por el usuario.

La captura será invocada por el método SaveGesture() de la clase GestureRecognizer(). Este método es público y es llamado desde el método OnClick() de los botones virtuales que tienen componente el script Capture-Button.cs. Save Gesture recibe los parámetros establecidos en la instancia del botón que le llama, toma los datos necesarios del estado de las manos en el momento de la llamada, e invoca el método SaveGesture() de la clase Persistence.

Persistence recibe el nuevo gesto a almacenar, lo añade a la copia de la base de datos en memoria y acto seguido la serializa en el archivo LSE_DB.xml.

5.4.5. Proceso de reconocimiento

El proceso de reconocimiento cuenta con una serie de fases en las que se va decidiendo qué hacer con la información obtenida. La clase encargada de realizar esta función es **GestureRecognizer**.

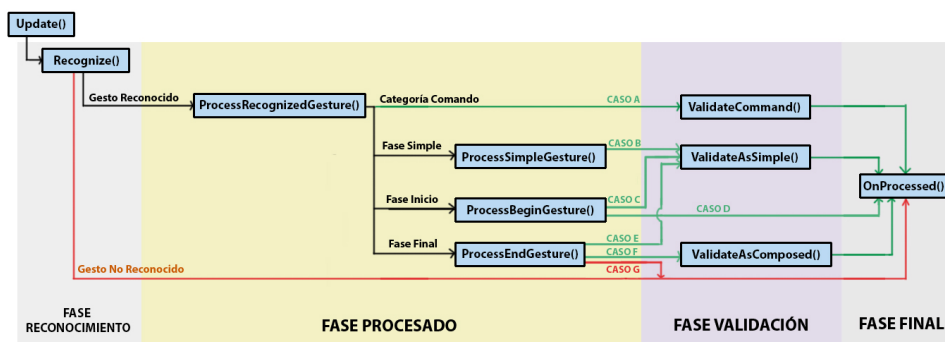


Figura 5.13: Flujo del Reconocimiento.

5.4.5.1. Fase de reconocimiento

La fase de reconocimiento compara los valores leídos de la posición de los huesos de la mano en la escena, contra la información almacenada sobre esos mismos huesos en cada uno de los gestos existentes en la base de datos.

El método para realizar este reconocimiento es mediante el calculo de diferencia mínimo de distancias. Se comparan el gesto actual contra cada uno de los gestos almacenados, obteniendo el valor absoluto de la resta del valor de las posiciones de cada hueso en ambos. A continuación, se hace un sumatorio de todos los valores resultantes de las restas y se selecciona el gesto

almacenado en la base de datos que da el valor más bajo. Para que no existan falsos positivos, es necesario establecer un umbral máximo. Si el sumatorio de valores resultantes de las restas está por encima de ese umbral, no se da el gesto como reconocido. Sin embargo si algún gesto cae por debajo de ese umbral, se entiende que el gesto realizado es el mismo gesto almacenado en la base de datos y se procede a procesarlo.

Este proceso se realiza 10 veces por segundo, aunque su frecuencia puede ser modificada en código en la clase `GestureRecognizer`, que se ocupa de todo el proceso de reconocimiento.

5.4.5.2. Fase de procesamiento

El siguiente paso procesa el gesto reconocido. Dependiendo de la categoría del gesto o la fase a la que pertenece, se debe almacenar dicho gesto como último reconocido y enviarlo al proceso de validación para que determine como transcribir el texto del signo reconocido.

Hay que recordar que los signos en movimiento cuentan con dos fases, una fase inicio que almacena el estado de las manos al inicio del signo, además de una fase final que almacena el estado de las manos al final del movimiento. También puntualizar que un gesto puede pertenecer a dos signos distintos dependiendo de si se usa de forma estática o en movimiento, de ahí que puedan contener dos tipos de fases distintas, aunque una de ellas siempre será simple.

Es muy importante hacer un seguimiento de cuál ha sido el anterior gesto reconocido, ya que en la fase de procesado determinará qué camino se debe seguir con el gesto reconocido actual. Los gestos de fase inicio siempre se quedan a la espera de ser validados por uno de fase final (Caso F). Si además contienen una fase simple, pueden ser validados si el siguiente gesto de fase simple o fase inicio (Caso C).

- **Caso A** El gesto es de categoría comando, por lo que se manda a la fase de validación directamente.
- **Caso B** El gesto únicamente contiene el parámetro fase simple. Comprobará si el gesto anterior necesita ser validado, después se validará como gesto simple.
- **Caso C** El anterior gesto reconocido contiene el parámetro fase inicio además de fase simple. Este se encuentra a la espera de ser validado. Dependiendo de los parámetros del gesto actual pasará a ser validado en su componente simple, a no ser que el gesto recibido sea del mismo signo con parámetro fase final.
- **Caso D** El gesto contiene el parámetro fase inicio, por lo que pasa directamente esperar en la posición de último gesto reconocido.

- **Caso E** El gesto contiene el parámetro fase final y el componente simple pasa a ser validado en su transcripción simple si el gesto anterior no es del mismo signo y de componente inicio.
- **Caso F** El gesto contiene el parámetro fase final es validado como un signo compuesto si el gesto reconocido previamente coincide con el mismo signo pero en fase inicio.
- **Caso G** El gesto contiene el parámetro fase final es descartado si no tiene una componente simple y el gesto previo reconocido no es del mismo en signo con él o coincide pero no corresponde a la fase inicio.

5.4.5.3. Fase de validación

La fase de validación es la encargada ejecutar la función del gesto reconocido. Si el gesto es de categoría comando, lo ejecutará inmediatamente. Si el gesto ha sido validado como un signo simple, se transcribe a la ventana de construcción de frases el valor de la variable String almacenada en su transcripción simple. Si el gesto ha sido validado como un signo compuesto, se transcribe el primer valor de la lista de Strings almacenado en su transcripción compuesta. Por decisión de diseño se permite tener más de una transcripción en la lista, aunque ahora mismo no sea usado. De este modo si más adelante se implementa algún método que ayude a determinar qué transcripción usar dependiendo del contexto de la frase, teniendo ya esta parte implementada.

5.4.5.4. Fase final

La fase final simplemente cierra todo el proceso de reconocimiento y habilita el inicio de un nuevo proceso de reconocimiento en el siguiente ciclo de ejecución. Si las opciones de debug están activas también almacenará información relevante al resultado del reconocimiento.

5.4.6. Gestión de la base de datos

Existe una escena adicional que permite gestionar la base de datos de gestos, en ella se encuentra una consola con los botones necesarios para establecer los parámetros necesarios de un gesto, capturarlo y almacenarlo. Además se pueden eliminar gestos que ya estén almacenados.

5.4.6.1. Nombre del gesto y teclado

En la parte superior de la consola se puede apreciar una casilla azul oscuro, con letras blancas. En ella aparece el nombre que se le dará al gesto, este hará las veces de transcripción simple del gesto, mientras que realmente el

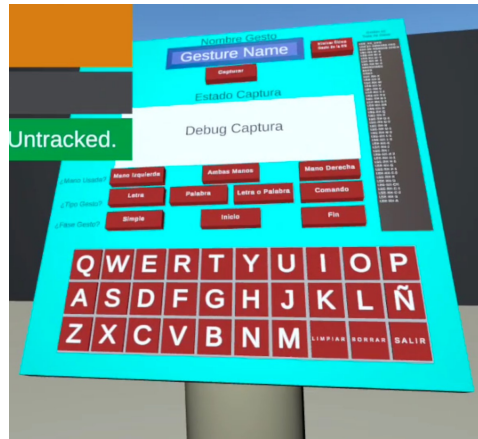


Figura 5.14: Consola de Captura.

nombre identificador del gesto será rellenado de forma automática, siguiendo las pautas descritas en el apartado 5.3.3.1 Nombre del Gesto.



Figura 5.15: El Nombre es rellenado con el teclado.

Para establecer el nombre se cuenta con un teclado en la parte inferior de la consola. Este teclado cuenta con botones para borrar letras o limpiar el nombre por completo. Es posible ver como el nombre se rellena a medida que se pulsa en el teclado.

El teclado también cuenta con el botón Salir para volver a la escena inicial de la aplicación.

5.4.6.2. Botones de uso de manos

La primera fila de botones permitirá elegir qué mano será capturada y en la cual debe ser realizado el signo.

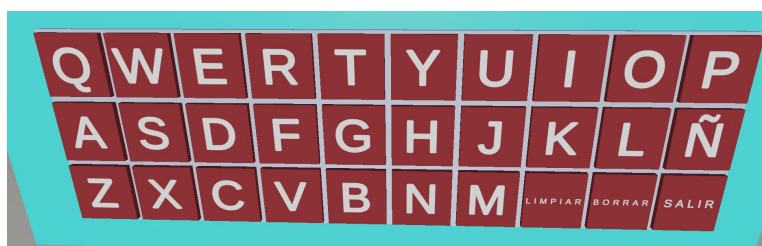


Figura 5.16: Teclado en la consola de captura.

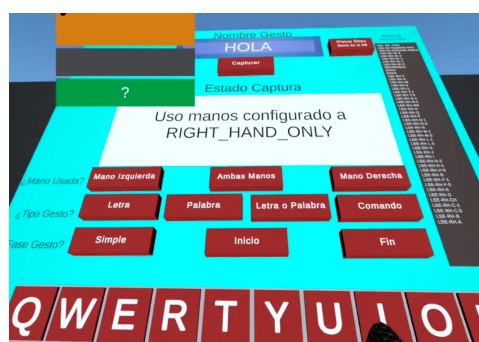


Figura 5.17: Botones para configurar que mano se va a capturar.

5.4.6.3. Botones de categoría

La segunda fila permite establecer la categoría del gesto.

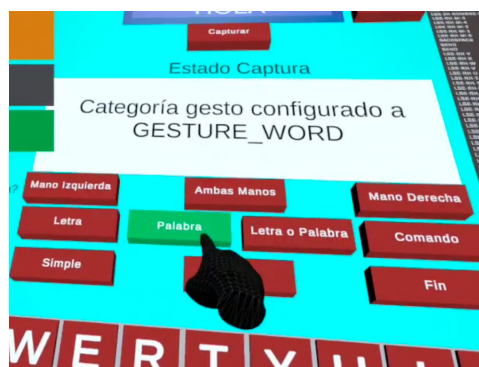


Figura 5.18: Selección de categoría.

5.4.6.4. Botones de fase

Y finalmente la tercera fila permite establecer la fase a la que pertenece el gesto, si es un gesto estático se utiliza la fase simple. Si es un gesto en movimiento, se debe almacenar primero el gesto configurado con la fase ini-

cio y posteriormente almacenar un nuevo gesto con el mismo nombre pero configurado con la fase fin.



Figura 5.19: Selección de fase.

5.4.6.5. Estado de captura

La ventana blanca indicará la cuenta atrás una vez se pulse el botón capturar y el momento en el que el gesto se almacene en la base de datos. También muestra cuándo se cambia la configuración de captura, indicando la selección de de mano utilizar, categoría o fase.

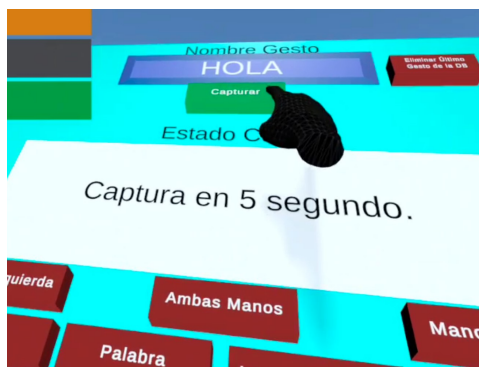


Figura 5.20: Cuenta atrás para capturar el gesto.

5.4.6.6. Captura

Al pulsar botón Capturar empezará una cuenta atrás, se dispone de 5 segundos para colocar la mano o manos en la posición deseada. Es importante hacer el gesto con las manos que se han especificado en la configuración del gesto, o no será correctamente almacenado ni reconocido. Si todo va bien se indica inmediatamente cómo el gesto ha sido almacenado y es automáticamente reconocido.

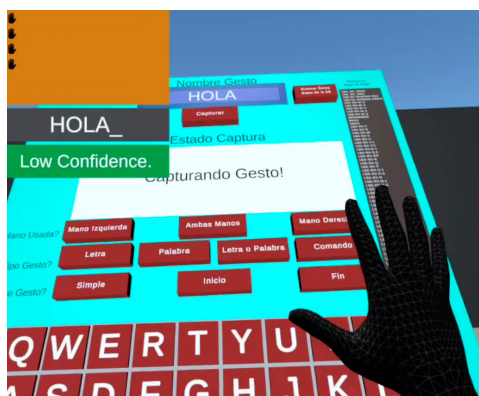


Figura 5.21: El nuevo gesto ha sido capturado.

5.4.6.7. Lista de gestos

En la parte derecha de la consola es posible ver una lista de los gestos almacenados en la base de datos, los más recientes aparecerán en lo más alto.

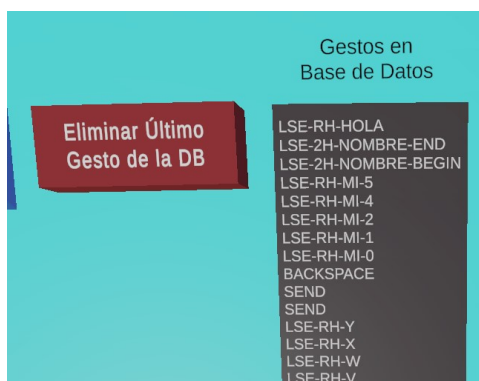


Figura 5.22: Lista de gestos y botón para eliminar.

5.4.6.8. Eliminación de gestos

Es posible eliminar el último gesto pulsando el botón que hay al lado de la lista. Únicamente se borrará el que esté en la parte más alta de la lista. También puede modificarse la base de datos editando con un procesador de texto el archivo LSE_DB.xml

5.5. Tutorial

La escena tutorial consiste en un pequeño carrusel de carteles donde se explica el funcionamiento a nivel usuario de la herramienta, haciendo especial énfasis en el reconocimiento de signos, construcción de frases y cómo utilizar los comandos. Cuenta con 9 pasos:

- **Paso 1.** Presentación de la herramienta
- **Paso 2.** Explicación de la interfaz
- **Paso 3.** Presentación del canal común de texto
- **Paso 4.** Transcripción de voz a texto
- **Paso 5.** Transcripción de LSE a texto
- **Paso 6.** Comando de envío
- **Paso 7.** Comando de borrado
- **Paso 8.** Estado del reconocimiento
- **Paso 9.** Otras escenas y cierre.

5.6. Modo multijugador

Desde el Lobby inicial de la aplicación se dispone de la opción de conectarse a una sala online. Al pulsarlo se establece una conexión con los servidores de Photon y el programa recibe sus callbacks.

La siguiente escena cargada es un selector de salas online. Se han desarrollado dos escenas a las que se puede acceder desde la selección de salas. Estas muestran si hay algún jugador conectado, por lo que se puede comprobar que, efectivamente, se ha realizado la conexión a un servicio en línea utilizando un segundo dispositivo de realidad virtual o el propio editor de Unity. Se han implementado botones en el editor para llamar a los callbacks de la clase que administra el componente multijugador.

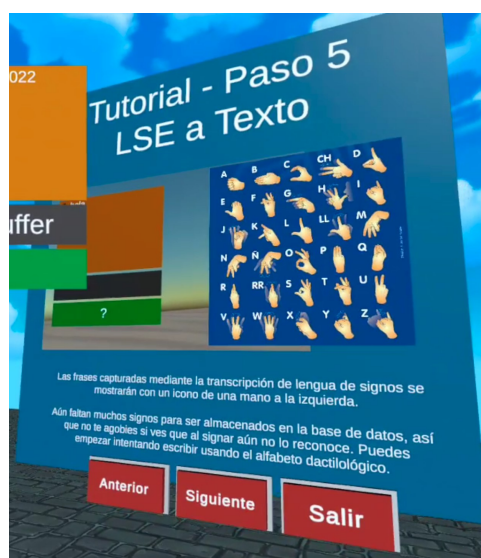


Figura 5.23: Paso 5 del tutorial.

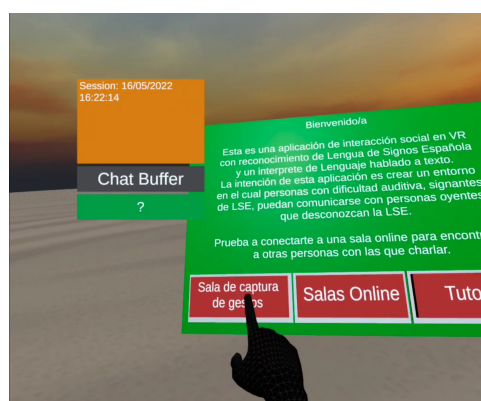


Figura 5.24: Lobby de bienvenida de la aplicación.

5.6.1. Escenas para el modo multijugador

5.6.1.1. Exterior

Esta escena pretende recrear un entorno a las afueras de una pequeña ciudad. Se ha diseñado con la intención de establecer conversaciones con las que hacer pruebas de calidad de comunicación entre ambos jugadores. ¿Cuántos coches están circulando?, ¿De qué color es el avión?, etc.

5.6.1.2. Aula

Esta escena se ha diseñado con intención de simular un espacio familiar para los voluntarios, recreando un laboratorio de la facultad de informática.



Figura 5.25: Selector de salas multijugador con distintas escenas.



Figura 5.26: Escena exterior multijugador.

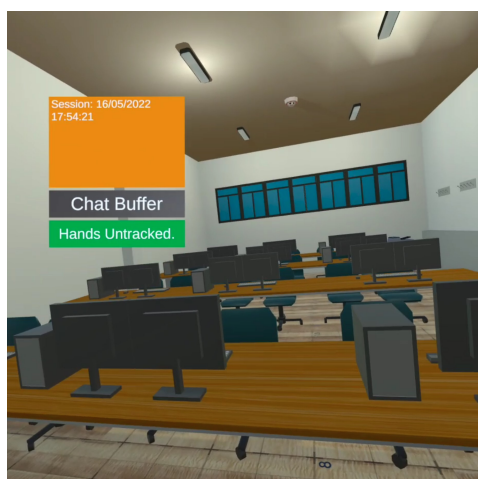


Figura 5.27: Escena laboratorio UCM, ordenadores.



Figura 5.28: Escena laboratorio UCM, pupitre del profesor.

Capítulo 6

Resultados

En esta sección se describen las pruebas realizadas con voluntarios en la facultad de informática y se discute brevemente la información obtenida de las respuestas en el formulario.

Se ha publicado el instalador para Meta Quest 1 y 2 en el siguiente enlace:
<https://narratech.com/es/tool/accessible-sign-voice-translator/>

6.1. Pruebas

La prueba se ha realizado en la entrada de la Facultad de Informática de la Universidad Complutense de Madrid. Se ha habilitado un espacio cerrado por un cordón de seguridad de unos 3 metros cuadrados, donde el voluntario realizaría la prueba con el dispositivo de realidad virtual de forma segura. El alumno guiaría al usuario a través de la demostración, una vez acabada esta se solicita al voluntario que rellene una encuesta en una mesa situada al lado de la zona de pruebas. Allí se ha preparado un portátil, para que el voluntario pudiera iniciar sesión en su correo y responder el cuestionario. La prueba se realizó en un dispositivo Meta Quest 2.

La prueba consiste en 4 partes:

- **Captación y preparación del entorno.** Se capta la atención del posible voluntario y se pregunta si disponen de unos minutos para la prueba. Se explica el propósito de la prueba y la finalidad del trabajo. Si este acepta, es llevado a la zona de seguridad y se le pregunta si ha tenido alguna experiencia con realidad virtual, se le avisa de los posibles efectos de esta, como mareos. Si el usuario requiere utilizar gafas durante la prueba, se adaptará el visor con el extensor para lentes. Se inicia la aplicación y se ayuda al voluntario a ajustarse el dispositivo, asegurando que ve nítidamente y tiene el casco bien ajustado a la cabeza con las cintas. Se explican los pasos de la demostración y se le anima a realizar el tutorial.

- **Tutorial** En este paso se deja al usuario continuar libremente, preguntando de forma recurrente si todo va bien y si es capaz de navegar por las ventanas explicativas. Se resolverá cualquier duda o problema que tenga. Aquí aprenderá a entender la interfaz de usuario, como capturar su voz y como mandar texto al chat usando Lengua de Signos. Una vez terminado este paso, se explica el siguiente paso, en el que será llevado a una escena donde podrá gestionar los gestos reconocidos.
- **Captura de un nuevo gesto.** Ayudar al voluntario a navegar por las escenas y abrir el gestor de gestos. Preguntar si son capaces de ver la consola de captura y se le anima a acercarse. En este paso se explica el funcionamiento de dicha consola para que aprendan como funciona la gestión de la base de datos de gestos. Como ejercicio, se anima al usuario a añadir una palabra de su elección, tecleando el nombre, configurando las opciones necesarias y finalmente capturando el gesto. Una vez conseguido, se pide que verifiquen si el sistema reconoce el gesto de forma inmediata. Una vez verificado, se informa al usuario de que la demostración ha terminado.
- **Opinión y rellenado del formulario.** Se ayuda al voluntario a retirar las gafas y se pregunta por la experiencia. Se invita al voluntario a salir de la zona de seguridad y a rellenar el formulario. Con ello acaba la prueba, se agradece al voluntario y empieza la búsqueda de otra persona que pueda probar la demostración.



Figura 6.1: Pruebas realizadas en la Facultad de Informática de la UCM.

Durante el día de las pruebas pudieron realizar la demostración 17 personas. Principalmente alumnos de la facultad, aunque también se pudo obtener la ayuda de algún profesor y visitantes de la facultad de mediana edad. Por desgracia, no se pudo contar con la opinión de ningún voluntario con discapacidad auditiva, sin embargo, sí se han obtenido respuestas en el formulario de personas en esta situación.

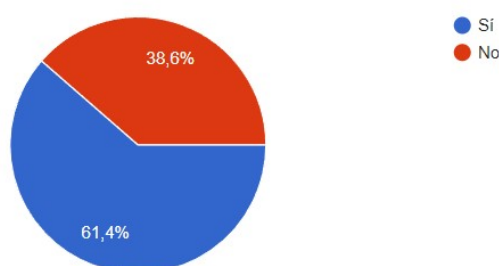
6.2. Discusión de los resultados del formulario

Se presenta un formulario a todos los voluntarios de las pruebas, además se difunde por varios canales de comunicación online, alcanzando una cifra de 57 respuestas. Se puede consultar el formulario completo y las respuestas obtenidas a cada pregunta en el apéndice B.4.

El 61,4 por ciento de los encuestados contaban con experiencias previas en realidad virtual, este dato es significativo ya que demuestra que un gran número de personas ha tenido en algún momento acceso a esta tecnología, sin embargo contrasta con que solo el 22,8 por ciento tienen en posesión algún dispositivo de realidad virtual.

1. ¿Tienes alguna experiencia con la realidad virtual?

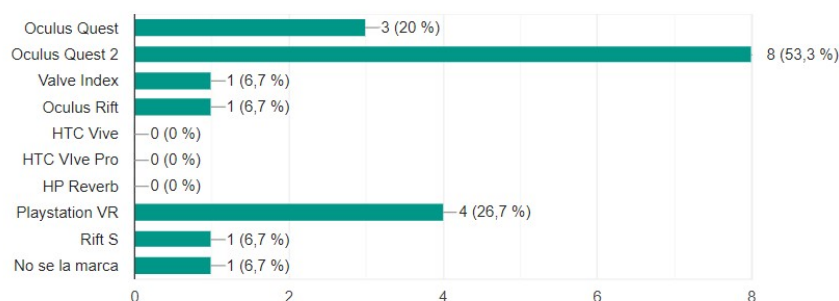
57 respuestas



A pesar de que la muestra es limitada, del cuestionario pueden obtenerse algunos datos interesantes sobre la experiencia a nivel usuario con la realidad virtual, además de qué dispositivos son los más utilizados. Se certifica que Meta Quest es la plataforma más común, acumulando un 73,3 por ciento entre los dispositivos Meta Quest 1 y Meta Quest 2, seguido muy por detrás por PlaystationVR, que aparece en el 26,7 por ciento de las casas encuestadas.

3. Si la respuesta anterior es afirmativa, ¿podrías decirnos cual?

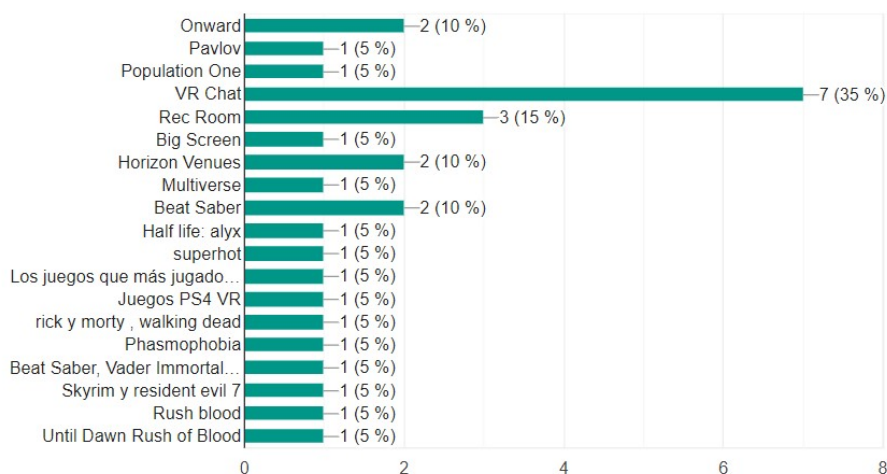
15 respuestas



También es muy relevante que el juego social más popular entre los encuestados sea VRChat, jugado con un 35 por ciento de los usuarios. Se confirma que los entornos sociales en realidad virtual son atractivos. Al mismo tiempo, la mitad de los encuestados quieren aprender o son usuarios de la Lengua de Signos Española, demostrando la relevancia de esta lengua y la cantidad de usuarios que podría ayudar esta herramienta.

4. ¿Has jugado en realidad virtual a alguno de estos juegos?

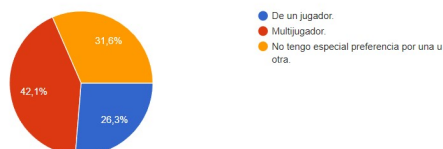
20 respuestas



Jugar a juegos multijugador es la preferencia entre los encuestados, ya que solo un 26,3 por ciento de estos prefiere las experiencias de un jugador. Aún así, aplicaciones puramente sociales representan un nicho, ya que solo han sido utilizadas por un 17,5 por ciento de los encuestados. Este dato contrasta con la importancia que se le da a las interacciones sociales mientras se juega, ya que el 80,7 por ciento considera que este componente social es muy importante.

8. ¿Preferirías experiencias de un solo jugador o multijugador?

57 respuestas

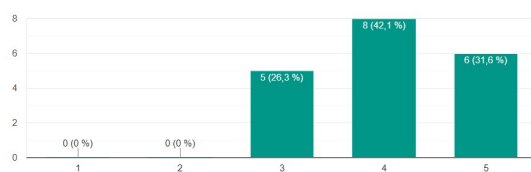


Se pregunto a los encuestados que herramientas de accesibilidad echan en falta para mejorar la comunicación entre jugadores, en sus respuestas aparecen tecnologías que se han implementado en esta herramienta, como el Speech to Text en tiempo real, o la capacidad de reconocer lenguas de

signos. Esto demuestra que hay una demanda por parte de los jugadores por ver implementadas estas herramientas. Finalmente, la recepción de la herramienta ha sido muy buena, con buenos resultados en las preguntas que hacen referencia a su utilidad y potencial de ser aplicada en proyectos educativos. Una petición común entre los entrevistados es mejorar la interfaz de usuario, haciendo que esta sea más intuitiva y que siga las pautas descritas en el capítulo 2.

21. Si un juego implementase esta herramienta, ¿Crees que te ayudaría a comunicarte mejor?

19 respuestas



Capítulo 7

Conclusiones

Se cierra esta memoria poniendo en valor el trabajo realizado, qué objetivos han sido cumplidos, además de definir cuáles son las futuras mejoras y ampliaciones del proyecto.

7.1. Objetivos alcanzados

Se analizan punto por punto los objetivos planteados en el capítulo 3 para comprobar si han sido cumplidos:

- Diseñar y desarrollar un entorno social en forma de aplicación de RV para dispositivos de última generación.

Se ha desarrollado una aplicación de Android (.apk) compatible con cualquier dispositivo Meta Quest 1 o 2, ya que ha sido probada en ambos y todas sus funcionalidades se ejecutan adecuadamente en ambas plataformas. Esta aplicación sirve como prueba de concepto de la herramienta. Contiene un tutorial de uso básico, una escena para gestionar y añadir signos a la base de datos, además de dos escenas multijugador.

- Implementar un componente centralizado para gestionar el canal común de comunicación en tiempo real basada en texto.

Ha sido necesario idear una interfaz visual especial para realidad virtual, donde el usuario pueda leer claramente los mensajes enviados. Esto ha podido realizarse gracias a la cámara central que se puede encontrar en el modelo de jugador OVRPlayer de Oculus Integration(Oculus, 2022). En él los objetos de interfaz de usuario colocados se ven correctamente en ambos ojos del reproductor. Para la gestión de las distintas entradas de texto, tal y como esta descrito en el capítulo 5, se ha implementado un script que gestiona todas las entradas de texto y da

formato al chat, incluyendo elementos visuales que ayudan a identificar si la entrada proviene de la captura de signos o el VoiceToText.

- Integrar en ese componente un sistema de reconocimiento de voz en idioma español con conversión a texto.

Tras analizar varias soluciones profesionales y su implementación en Unity, como las herramientas SpeechToText en la nube que ofrecen Google Cloud o Microsoft Azure, se decide utilizar un paquete de pago de Unity Store. *Speech Recognition System* (Studio, 2022) que es capaz de capturar voz y transcribir a texto, todo ello de forma local, sin conexión y consumiendo muy pocos recursos.

- Integrar en ese componente un sistema de reconocimiento de gestos de la Lengua de Signos Española con conversión a texto. Este sistema tendrá una base de datos de gestos que el usuario podrá gestionar directamente, para tener un mayor control sobre su repertorio de signos.

Se ha desarrollado una herramienta funcional, configurable y expansible. Esta es capaz de capturar gestos estáticos y móviles, almacenarlos como signos y reconocerlos al momento. Además se da la libertad al usuario de ampliar de forma sencilla la base de datos, añadiendo aquellos signos que sean necesarios o adecuando el reconocimiento a su forma natural de signar.

- Validar la utilidad y usabilidad de la aplicación mediante pruebas con usuarios reales.

Se ha llevado a cabo una jornada de pruebas en la facultad de informática, creando un espacio donde realizar las pruebas, además de un formulario para obtener datos y opiniones sobre la aplicación. Se han realizado pruebas a 17 voluntarios, obteniendo cerca de 57 respuestas al formulario.

7.2. Trabajo futuro

Dada la envergadura del proyecto planteado, se sugieren las siguientes líneas de trabajo para el futuro:

- Añadir más palabras del diccionario de la LSE a la base de datos.
- Adaptar la aplicación a las pautas descritas en el capítulo 2.2.
- Mejorar la captura de los gestos más complejos.
- Mejorar las salas multijugador para permitir el acceso a más jugadores de forma simultánea, mejorando la sincronización de datos a través del servidor.

- Estudiar la implementación de una Red Neuronal y comparar su rendimiento con el modelo actual de clasificación por diferencia de distancias.
- Construcción avanzada de frases. Implementar un sistema que permita decidir que transcripción utilizar para signos que tengan varios significados según el contexto de la frase.
- Ampliar el soporte a varias bases de datos, permitiendo al usuario seleccionar la que desea utilizar. Con esto se habilita el uso de distintas bases de datos para cada idioma.
- Mejor colocación de objetos, permitir transparencia en la interfaz de usuario, incluir el nombre del jugador que ha enviado cada mensaje en el chat, y muchos más.

7.3. Reflexión final

Este proyecto ha sido un auténtico reto a nivel personal, me quedo satisfecho con haber desarrollado una herramienta capaz de ayudar a personas con riesgo de exclusión social debido a las incapacidades que padecen. Este trabajo de fin de grado ha supuesto una gran inversión de tiempo, recursos y formación, pero considero que lo compensan con creces los conocimientos que he adquirido y las relaciones sociales que he podido entablar en el proceso.

Appendix A

Introduction

Virtual Reality is in clear expansion and more and more studios are developing for these platforms that take us to new virtual worlds. The world of video games always had a large component of escapism from the real world, so when diving into virtual reality the user takes control of his virtual avatar. These are capable of all kinds of feats, it can generate objects from nothing, climb impossible walls, fly across the sky, make big jumps or lift any kind of objects, use magic or transform our body into any kind of shape. Having this ability to be something different, it of most importance to consider how this technology can help us overcome barriers that people with different types of disabilities suffer in the real world. In the case of people with hearing impairment, the traditional video game offers us visual interfaces with descriptive subtitles and chat channels. In the virtual world, most applications rely on voice communication and users do not have access to a keyboard. Therefore, it is necessary to create a tool capable of overcoming this new challenge and help these people to be able to integrate in the most natural way possible, as their ability to enjoy immersion in a virtual world depends on it.

Virtual Reality (VR) has been around for a while now, since 2016 the market never stopped growing and it have witnessed a battle between different devices, some standalone, others connected to smartphones or to the computer. Since the beginning of the pandemic this competition has flared up, facing up big tech companies against each other as they try to get the biggest share of this growing sector and the interest of the general public. Valve, HTC, Meta (formerly Facebook), Sony, Samsung or HP, each one has presented its particular proposal, from simple cardboard cases to enable the latest generation of cell phones as VR devices, to game console peripherals with entire development teams aimed at generating exclusive content for these platforms, to generalist solutions for computers with very powerful technical capabilities and enough computational capacity to play games on VR devices connected through cable. While console devices were the most

successful in the early years, the market has now moved on to the most accessible and affordable solution, the standalone VR headset.

These devices do not require additional hardware to operate, their 'inside-out' tracking systems do not require positioning beacons to be installed in a room, giving users total freedom to live VR experiences anywhere and in any environment. The price to pay for this independence is their power and autonomy: they rely on batteries that limit their usage time and their games have somewhat less detailed graphics. These autonomous devices can now be connected to computers to take advantage of their capabilities, as is the case with the Meta Link cable, providing a versatile solution capable of adapting to the needs of each user. The success of this philosophy is evident in other sectors such as video games, the Libro Blanco de los Videojuegos 2021 shows how the console for which Spanish companies and studios are developing the most is Nintendo Switch (DEV, 2021), which is also the leader in sales of consoles for the current generation (VGChartz, 2022), being essentially a platform of lower power, but greater versatility, with mobile and desktop device capabilities. In VR, the current sales leader and success device is Meta Quest 2 (Meta, 2019), as shown by Steam statistics Valve (2022).

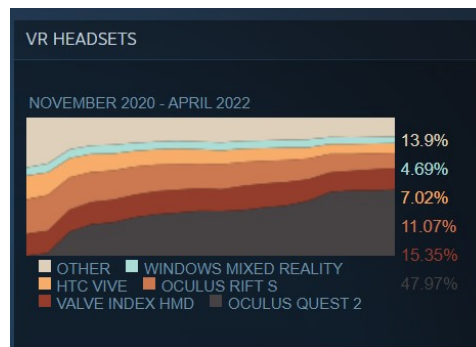


Figure A.1: Dispositivos de RV más utilizados en los últimos años según Steam Analytics.

A.1. Motivation

This new paradigm in the world of entertainment needs to be accessible to any user. The same philosophy of projecting ourselves in an avatar in a virtual world should also allow us to leave behind those barriers of the real world that can be easily overcome in the virtual one. That is why accessibility tools are so important so those walls can be broken down.

This project aims to create a tool that helps any hearing impaired user to communicate in the most natural way possible. The goal is for a hearing impaired user to be able to express themselves and understand others in the

most natural way possible for them.

Meta Quest allows us to directly recognize the user's real hands and project them into the virtual world, it also incorporates microphones and speakers. All necessary requirements to develop a tool capable of overcoming this communication problems are met.

A.2. Purpose and scope

According to the Confederación Estatal de Personas Sordas (CNSE), about 3 percent of the Spanish population suffers some kind of hearing loss (CNSE, 2022).

The purpose of this work is to take advantage of the available technology to create an inclusion tool that helps that group of people with hearing impairment to be part of the future of the metaverse, video games and other VR social applications. Delving into the development of VR applications for Android-based mobile devices is also part of the goal, since it is the operating system running in Meta devices. Creating a free tool, easily deployable in any VR project created in the popular Unity development environment, is also one of the goals.

While there are some applications developed to interpret gestures, these remain simple demonstrations for recognizing the American Sign Language (ASL) dactyl alphabet. This project focuses on Spanish Sign Language (LSE), its dactylogical alphabet and also the ability to capture, store and recognize signs belonging to the sign dictionary.

The hearing impaired community exists in VR, but they have to learn to express their respective sign languages, overcoming the barrier of interacting with the virtual world through controllers that greatly limit the ability to gesture and perform certain signs. This project aims to enable these people to communicate using their own hands, without the need to learn which buttons are used to extend or contract the virtual fingers of their avatar, or which combination of these buttons creates a predefined gesture.

A.3. Related subjects

Throughout the career there have been many subjects that provide the necessary knowledge to be able to undertake a project of these characteristics. In this section I will highlight those that have been the most relevant for this project.

- **Game Design.** This first-year course teaches skills and tools needed to start a project, as well as proper design guidelines.
- **Agile Production Methodologies.** It teaches everything related to

the organization of software development projects.

- **Project I.** This project introduced development using the Unity development environment, also used in this work.
- **Project III.** It teaches how to properly structure our projects and integrate external APIs.
- **Data Structures and Algorithms.** Knowing how to manipulate data is fundamental in this project in which a lot of information is managed.
- **Technologies of Video Game Programming.** Teaches the correct application of design patterns common patterns in this type of development such as Observer, Singleton, Update Method or Game Loop.
- **Video Games for Mobile Devices.** Gives the basics of Android development in the Unity development environment.
- **Usability and Analysis of Video Games** This course teaches how to design and conduct user testing and forms.

Such a development is not possible without investing time in understanding the difficulty of the task and preparing the necessary skills. For this reason, I had to take several courses outside the university. My training began focusing in VR application development in Unity, taking the ‘VR Development Fundamentals With Oculus Quest 2 and Unity’ (DEMİRBAŞ, 2020b) and ‘Multiplayer Virtual Reality (VR) Development With Unity’ (DEMİRBAŞ, 2020a) courses, both taught by Tevfik Ufuk through the Udemy teaching platform. These courses provided the necessary skills to start programming the necessary foundation and be able to export the application to VR rigs.

The next step required obtaining an initial understanding of Sign Language, to this end it was coursed the UNED ‘Spanish Sign Language A1’ course (Uned, 2017), taken through the Signo Campus platform. This was essential to understand the real complexity of the task, to be clear about the challenges and the technological limitations that apply to the case.

A.4. Structure of the work

This paper is structured as follows:

- chapter 1 presents the purpose of this Final Degree Project, the context in which it is carried out, the motivation, scope and which subjects of the degree are related to it.

-
- chapter 2 reviews the state of the art, looking at the current state of virtual reality, what it means to be a hearing impaired user in the video game world and the evolution of gesture recognition.
 - In chapter 3 the main objectives of the work are presented, as well as an explanation of all the requirements necessary for the development of the tool.
 - chapter 4 explains the methodology followed for the realization of the project, as well as an enumeration and justification of the tools used.
 - chapter 5 presents a detailed explanation of the design and implementation of the project, as well as the different obstacles and problems that had to be dealt with for its complete development.
 - chapter 6 feedback about the project and tests conducted with real users will be collected.
 - In chapter 7 conclusion to the project will be given, reviewing whether the previously defined proposals and objectives have been met, as well as suggesting possible extensions and lines of work for the future.

Appendix B

Conclusions

To close this memory lets look back upon the work done, which goals have been accomplished and define the future improvements or expansions that can be done to this project.

B.1. Goals complete

Let's analyze point by point all goals established in chapter 3 and see if they are completed:

- Design and develop a social environment in the shape of a VR app for last gen devices.

A Android app has been developed that is compatible with Meta Quest and Meta Quest 2 devices. It has been tested in both devices and all it's functionalities work as intended. All functionalities have been tested and work as intended in both platforms. This app works as a proof of concept of the tools developed. It contains a tutorial scene to learn the tool's basic usage, a DB management scene where new signs can be added and also two multiplayer scenes.

- implement a centralized component that manages the shared text channel of communication in real time.

It was necessary to devise a special visual interface for VR, where the user can clearly read all sent messages. This has been accomplished thanks to the central camera anchor that can be found in the OVR-Player Model that is part of the Oculus Integration package Oculus (2022). In it, any user interface placed is correctly projected to both eye cameras. To manage the diferent kind of text input, as described in 5, a script was implemented that formats all chat entries, adding visual elements that help identify the input source, whether it's sing language or voice.

- Integrate in that component a system that can recognize spanish spoken words and convert them into text.

A premilimar study was made, checking proffesional solutions and how to implement them in Unity Engine. Some examples are Google Cloud or Microsoft Azure online services, but I finally settled for a Unity Package that is sold in Unity Store. *Speech Recognition System Studio* (2022) is capable of capturing voice and translate it into text locally, without network connection and consuming very little resources.

- Integrate in that component a system that can recognize sings from the LSE and convert them into text. This system will have a DB of Signs that the user can manage directly, to give more control over the sign repertoire.

A functional tool have been developed that can be configured and expanded. It is capable of capture either moving or static gestures, store them as signs and recognize them on real time. Also the user is given the freedom to easily expand the data base, adding signs or recapturing them in the way that it's natural for him to sign. There is much left to improve in this recognition system, although it is perfectly functional, it still have to solve some problems regarding the most complex gestures.

- Validate the utility and usability of the application through test with real users.

A testing event was performed in the computer science faculty, stablishing a safety space where test in VR can be run. Volunteers have been given a form to answer for us to obtain data and feedback about the application. About 20 volunteers step up to test the app and it received 57 answers on the form.

B.2. Future tasks

This project still has a lot that can be polished and ways to be expanded:

- Add more LSE signs from its dictionary into to the preset data base.
- Adapt the app to the guidelines describe in 2.
- Improve gesture recognition to allow the more complex signs.
- Finish the implementation of multiplayer capability.
- Consider implementing a Neural Network and compare its performance to the actual distance difference model.

- Implement advanced phrase building. Add a system that allows to decide the transcription of the sign depending on the context of the conversation.
- Add support for various data bases, allowing the user to select which one to use. This enables the use of different databases depending on language.
- General polish, better object placement, allow transparency to GUI, include player names before the transcribed message in chat.

B.3. Final thoughts

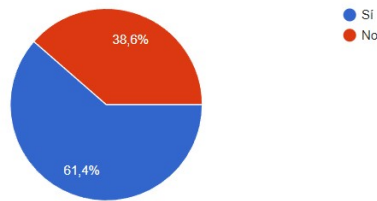
This project has been a tough personal challenge. I'm satisfied with this tool I have developed, seeing that it can be to somebody's help. I'm aware that there is much work ahead that can be done. This final project has been a great investment in time, resources and formation for myself, but it was all worth given the knowledge that I have obtained and the charming people I've found in the way.

B.4. Cuestionario y resultados

El cuestionario ha sido realizado utilizando la herramienta Google Forms. Fue distribuido por las redes el mismo día que se realizaron las pruebas y ha obtenido respuestas de un total de 57 usuarios. A continuación compartiré las preguntas y los resultados:

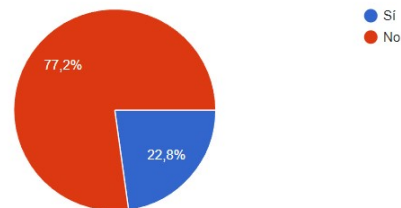
1. ¿Tienes alguna experiencia con la realidad virtual?

57 respuestas



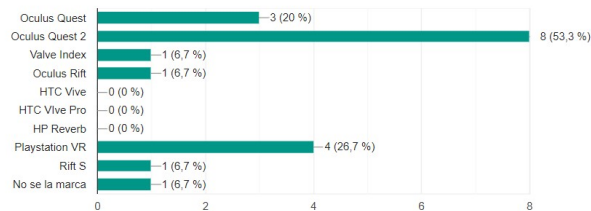
2. ¿Tienes algún dispositivo de realidad virtual de tu propiedad?

57 respuestas



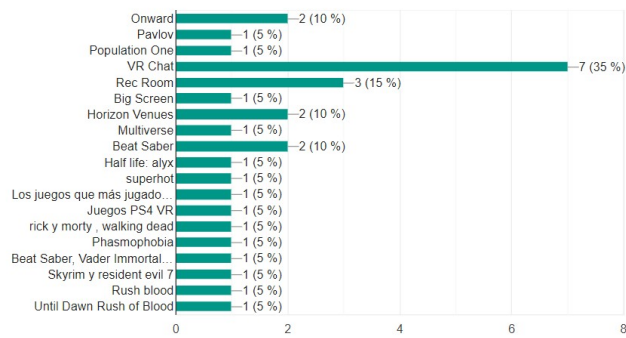
3. Si la respuesta anterior es afirmativa, ¿podrías decirnos cual?

15 respuestas



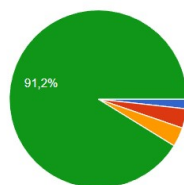
4. ¿Has jugado en realidad virtual a alguno de estos juegos?

20 respuestas



5. ¿Sufres algún tipo de discapacidad auditiva?

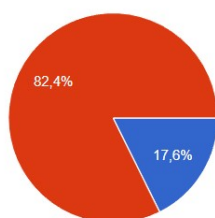
57 respuestas



- Si, soy sordo/a
- Si, estoy dentro del espectro hipoacúsico
- No, pero sufro otro tipo de discapacidad
- No

6. Si padeces alguna discapacidad, ¿afecta esta a tu habilidad de comunicarte con otros jugadores en videojuegos multijugador?

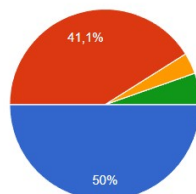
17 respuestas



- Si
- No

7. ¿Conoces la Lengua de Signos Española?

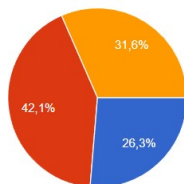
56 respuestas



- Solo he oído hablar de ella
- Quiero aprender algún día
- La estoy estudiando
- Se signar con fluidez

8. ¿Prefieres experiencias de un solo jugador o multijugador?

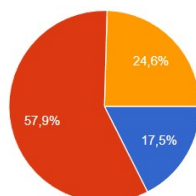
57 respuestas



- De un jugador
- Multijugador
- No tengo especial preferencia por una u otra

9. ¿Usas aplicaciones o videojuegos principalmente sociales como VRChat?

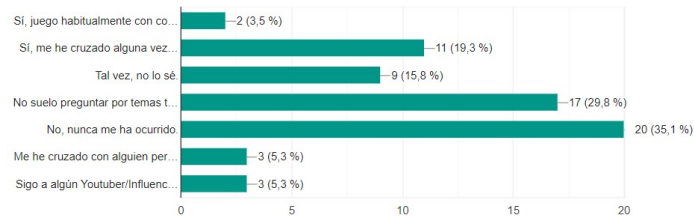
57 respuestas



- Si
- No
- No, pero me gustaría probarlo

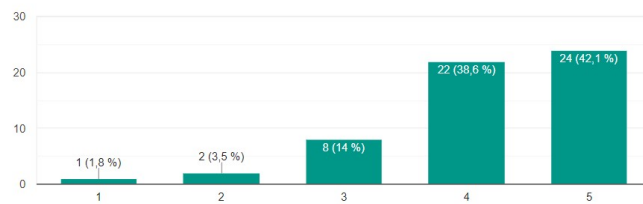
10. ¿Has conocido a otros jugadores que sufran de alguna discapacidad que dificulte su manera de comunicarse?

57 respuestas



11. ¿Qué grado de importancia le das a la interacción social dentro de juegos multijugador?

57 respuestas



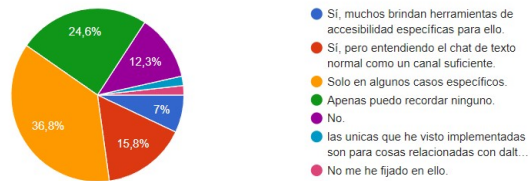
12. ¿Has utilizado las herramientas de accesibilidad de algún videojuego?

57 respuestas



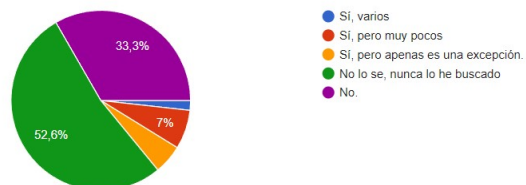
13. hay muchos ejemplos de herramientas de accesibilidad para mejorar la jugabilidad, sin embargo, ¿crees que los videojuegos brindan herramientas de accesibilidad suficientes para superar barreras de comunicación entre jugadores?

57 respuestas



14. ¿Conoces algún juego de realidad virtual que incorpore herramientas con dicha finalidad?

57 respuestas



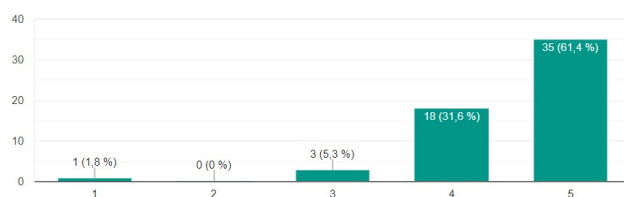
15. Qué herramientas o ideas te gustaría ver implementadas para facilitar la comunicación entre jugadores?

11 respuestas

videochat
Speech to text, aunque en tiempo real podría ser muy costoso técnicamente hablando
Soporte in game para comunicación media sin necesidad de chat (ni texto ni voz). Apex legends (desde mi punto de vista) consiguió algo muy bueno con un sistema de pings muy sencillo e intuitivo, lo que resultaba muy gratificante en juegos con desconocidos
No lo se. ¿Se puede mejorar?
Lenguaje de gestos
La verdad es que nunca me he parado a pensarlo porque con el chat de texto o el de voz de toda la vida nunca he tenido necesidad de utilizar otros.
Transcripción de juego y chat generalizado (text to speech, voice to text)
Voz a texto
Una IA con capacidad de aprender el lenguaje de signos y que haga de traductor virtual a la gente para que se puedan comunicar entre la gente con discapacidad y la gente que no tiene
Cualquier opción de accesibilidad nueva, por pequeña que sea, es bienvenida. La propia idea de este TFG me parece una forma genial de ayudar a las personas con problemas auditivos.
Cualquier herramienta que facilite la comunicación entre dos personas que de por si fuera difícil (personas con discapacidad auditiva severa, etc)

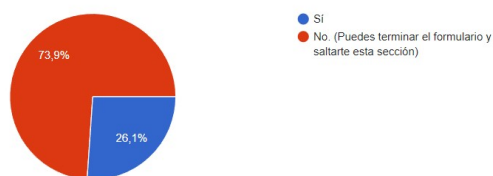
16. ¿Cuán importante es que se invierta esfuerzo en utilizar la tecnología para evitar la discriminación de colectivos con dificultades auditivas en el nuevo paradigma del mundo virtual?

57 respuestas



17. ¿Has probado la demo del TFG en la facultad de informática de la Universidad Complutense de Madrid?

46 respuestas



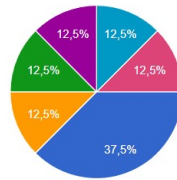
18. Si eres oyente... ¿Podrías entender lo que una persona signante quiere comunicarte?

21 respuestas



19. Si eres signante... ¿Podrías entender bien a una persona que se comunicó solo hablando?

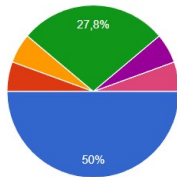
8 respuestas



- Si
- Si, pero necesaria que más lenguaje corporal fuera transmitido.
- Me costaría un poco saber que me esta diciendo.
- Solo algunas cosas.
- Difícilmente entendería nada.
- No soy signante pero creo que ellos nos entenderían con más "facilidad" que n...
- creo que no siempre tambien creo qu...

20. ¿Sientes que el mensaje que querías comunicar ha sido transmitido adecuadamente a la interfaz escrita (chat)?

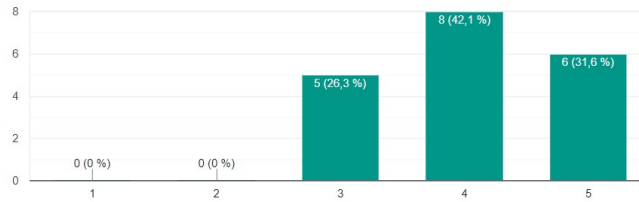
18 respuestas



- Si, en ambos sentidos
- Solo de voz a texto.
- Solo signo a texto.
- En general bien pero la construcción de frases puede mejorar.
- Me cuesta un poco decir exactamente lo que quiero
- No he conseguido comunicar lo que quería.
- No he probado la demo.

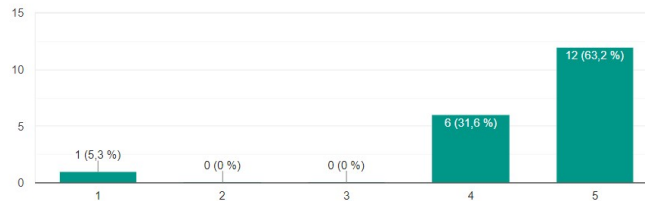
21. Si un juego implementase esta herramienta, ¿Crees que te ayudaría a comunicarte mejor?

19 respuestas



22. ¿Crees que tiene el potencial de ser una herramienta útil?

19 respuestas



23. ¿Qué añadirías?

5 respuestas

Si tuviese que describirlo en una sola palabra probablemente sería "original"

Es muy buena, simplemente genial

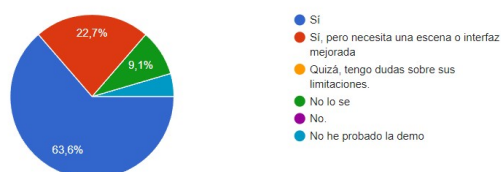
Ampliar base de datos con frases/expresiones básicas

Creo que a la herramienta le falta entrenamiento.

Añadir nada, más que cambios en la GUI para hacerla más "pulida" y accesible, pero es totalmente secundario ahora. Me centraría primero en asegurar la correcta funcionalidad, como está haciendo el autor del TFG.

24. ¿Crees que podría servir como una herramienta para el aprendizaje de lengua de signos?

22 respuestas



25. Si tienes algún comentario o sugerencia aquí puedes escribirla:

5 respuestas

¡Ánimo! El proyecto tiene una gran iniciativa en la relación entre Realidad Virtual y discapacidades en personas de hoy en día.

Que esta aplicación debería aplicarse a todos los juegos de realidad virtual online, y en terapias e incluso podría implementarse en clases escolares

Está perfecto

Como persona con hipoacusia mixta profunda, que ha trabajado junto a personas totalmente sordas, y que ha visto y vivido en muchas ocasiones la marginación que produce la incapacidad de utilizar con comodidad (o no poder usar en absoluto) los programas típicos de comunicación (skype, discord, etc.), considero que estas iniciativas son fundamentales para romper estas barreras y acercar a las personas con diversidad funcional al resto de la comunidad.

Debe tenerse en cuenta que, ya de base, las personas con algún tipo de discapacidad cuya dolencia las aísla de la sociedad sufren de trastornos graves como depresión, problemas de concentración, ansiedad e incluso de personalidad. Los seres humanos somos sociales por naturaleza y dependemos de la comunicación con los demás para una salud mental óptima.

Lógicamente en mi caso hablo especialmente de la sordera porque es uno de los muros más grandes que impide la sociabilización desde el nivel más básico (la propia familia) y cuyos niveles y rangos son tan variables que resulta muy complicado de comprender por quien no la sufre y lleva, en muchos casos, a la burla o al aislamiento. Desgraciadamente muchas veces por pura comodidad, ya que parece que es más fácil dejar de lado al/a la que le cuesta comunicarse que intentar poner el esfuerzo de buscar un canal común.

Considero que todas las herramientas que se puedan desarrollar para dar apoyo a los más vulnerables y ayudarlos a integrarse mejor deben contar por el soporte de todos/as y dedicarse todos los recursos necesarios para su estudio y potenciación. Gracias por pensar en nuestro colectivo para la realización de este proyecto.

Me parece bastante increíble que solo una persona pueda hacer una utilidad con tanto potencial. Enhorabuena.

Bibliografía

- ADEWALE, V. A. y OLAMITI, A. O. Conversion of sign language to text and speech using machine learning techniques. 2018.
- AGUADO-DELGADO, J., GUTIERREZ-MARTINEZ, J.-M., HILERA, J. R., DE MARCOS, L. y OTÓN, S. Accessibility in video games: a systematic review. *Universal Access in the Information Society*, vol. 19(1), páginas 169–193, 2020.
- ANTONIO CARDONA, F. P. Repositorio de este proyecto. github.com/Narratech/TFGCardona, 2022.
- BBC. Bbc subtitle guidelines. bbc.github.io/subtitle-guidelines/, 2021.
- BIERRE, K., CHETWYND, J., ELLIS, B., HINN, D. M., LUDI, S. y WESTIN, T. Game not over: Accessibility issues in video games. En *Proc. of the 3rd International Conference on Universal Access in Human-Computer Interaction*, páginas 22–27. 2005.
- CNSE. Confederación estatal de personas sordas. www.cnse.es, 2022.
- DEMİRBAŞ, T. U. Multiplayer virtual reality (vr) development with unity. udemy.com/course/multiplayer-virtual-reality-vr-development-with-unity, 2020a.
- DEMİRBAŞ, T. U. Vr development fundamentals with oculus quest 2 and unity. udemy.com/course/oculus-quest-development-with-unit, 2020b.
- DEV, A. E. D. E. P. Y. D. D. V. Y. S. D. E. Libro blanco del desarrollo de videojuegos 2021. www.dev.org.es/es/publicaciones/libroblancodev2021, 2021.
- DISNEY. New club penguin. 2022.
- EL-SAWAH, A., GEORGANAS, N. D. y PETRIU, E. M. A prototype for 3-d hand tracking and posture estimation. *IEEE Transactions on Instrumentation and Measurement*, vol. 57(8), páginas 1627–1636, 2008.

- DE ESPAÑA, G. Ley 51/2003, de 2 de diciembre, de igualdad de oportunidades, no discriminación y accesibilidad universal de las personas con discapacidad. www.boe.es/buscar/act.php?id=BOE-A-2003-22066, 2003.
- HAMILTON, I. Gdc vault - subtitles are changing, don't be left behind. www.gdcvault.com/play/1025738/Subtitles-Are-Changing-Don-t, 2019.
- HANDS, H. Discord server for helping hands vr community. discord.com/invite/helpinghands, 2022.
- HEAP, T. y SAMARIA, F. Real-time hand tracking and gesture recognition using smart snakes. *Proc. Interface to Human and Virtual Worlds, Montpellier, France*, vol. 50, 1995.
- KOUL, M., PATIL, P., NANDURKAR, V. y PATIL, S. Sign language recognition using leap motion sensor. *International Research Journal of Engineering and Technology (IRJET)*, vol. 3(11), páginas 322–325, 2016.
- LINDEN RESEARCH, I. Secondlife. <https://secondlife.com/>, 2022.
- MA, Y., MAO, Z.-H., JIA, W., LI, C., YANG, J. y SUN, M. Magnetic hand tracking for human-computer interface. *IEEE Transactions on Magnetics*, vol. 47(5), páginas 970–973, 2011.
- MELAX, S., KESELMAN, L. y ORSTEN, S. Dynamics based 3d skeletal hand tracking. En *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, páginas 184–184. 2013.
- META. Oculus quest 2. store.facebook.com/es/es/quest/products/quest-2/, 2019.
- MUELLER, F., BERNARD, F., SOTNYCHENKO, O., MEHTA, D., SRIDHAR, S., CASAS, D. y THEOBALT, C. Gnerated hands for real-time 3d hand tracking from monocular rgb. En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, páginas 49–59. 2018.
- NIREI, K., SAITO, H., MOCHIMARU, M. y OZAWA, S. Human hand tracking from binocular image sequences. En *Proceedings of the 1996 IEEE IECON. 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, vol. 1, páginas 297–302. IEEE, 1996.
- OCULUS. Oculus integration. packages/tools/integration/oculus-integration-82022, 2022.
- OY, S. Habbohotel. <https://www.habbo.es/>, 2022.
- PAN, Z., LI, Y., ZHANG, M., SUN, C., GUO, K., TANG, X. y ZHOU, S. Z. A real-time multi-cue hand tracking algorithm based on computer vision.

- En *2010 IEEE Virtual Reality Conference (VR)*, páginas 219–222. Ieee, 2010.
- REHG, J. M. y KANADE, T. Digiteyes: Vision-based hand tracking for human-computer interaction. En *Proceedings of 1994 IEEE Workshop on Motion of Non-rigid and Articulated Objects*, páginas 16–22. IEEE, 1994a.
- REHG, J. M. y KANADE, T. Visual tracking of high dof articulated structures: an application to human hand tracking. En *European conference on computer vision*, páginas 35–46. Springer, 1994b.
- SCHLATTMAN, M. y KLEIN, R. Simultaneous 4 gestures 6 dof real-time two-hand tracking without any markers. En *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, páginas 39–42. 2007.
- SCHRÖDER, M., MAYCOCK, J., RITTER, H. y BOTSCH, M. Real-time hand tracking using synergistic inverse kinematics. En *2014 IEEE International Conference on Robotics and Automation (ICRA)*, páginas 5447–5454. IEEE, 2014.
- SHARP, T., KESKIN, C., ROBERTSON, D., TAYLOR, J., SHOTTON, J., KIM, D., RHEMANN, C., LEICHTER, I., VINNIKOV, A., WEI, Y. ET AL. Accurate, robust, and flexible real-time hand tracking. En *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, páginas 3633–3642. 2015.
- STENGER, B., THAYANANTHAN, A., TORR, P. H. y CIPOLLA, R. Model-based hand tracking using a hierarchical bayesian filter. *IEEE transactions on pattern analysis and machine intelligence*, vol. 28(9), páginas 1372–1384, 2006.
- STUDIO, S. S. Speech recognition system. `packages/tools/audio/speech-recognition-system-187171`, 2022.
- UNED. Cursos lse a1, a2. uned.es/cuid/oferta_didiomas/guias2017/lse_a1-a2.pdf, 2017.
- VALVE. Steam store hardware survey. store.steampowered.com/hwsurvey/, 2022.
- VGCHARTZ. Consoles hardware sales. <https://www.vgchartz.com/>, 2022.
- WANG, R. Y. y POPOVIĆ, J. Real-time hand-tracking with a color glove. *ACM transactions on graphics (TOG)*, vol. 28(3), páginas 1–8, 2009.
- YUAN, B., FOLMER, E. y HARRIS, F. C. Game accessibility: a survey. *Universal Access in the information Society*, vol. 10(1), páginas 81–100, 2011.

