

E.T.S. de Ingeniería Industrial, Informática
y de Telecomunicación

Implementación y evaluación de menús en realidad virtual basados en la propiocepción



Grado en Ingeniería Informática

Trabajo Fin de Grado

Autora: Ainhoa Sofía Basadre Solana

Tutores:

Amalia Ortiz Nicolás

Óscar Ardaiz Villanueva

Pamplona, 06/09/2021

upna

Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Resumen y Palabras Clave

Este proyecto se ha centrado en investigar si el sentido de la propiocepción permite crear nuevas formas de interacción con los menús de los mundos virtuales más inmersivas e intuitivas.

Para ello se ha diseñado e implementado un conjunto de menús a partir de dos variables, posición (menú fijo, en la mano o en el antebrazo) y visibilidad (opciones de menú visibles o no visibles).

Estas interfaces han sido evaluadas con el objetivo de valorar si el sentido de la propiocepción podría usarse en diferentes plataformas de interacción de usuario para mejorar su experiencia y respetar más la inmersión sin entrar en conflicto con el buen funcionamiento del mismo.

Como conclusión final se ha obtenido que, gracias a la propiocepción, un menú situado en la mano es igual de eficiente que el menú fijo. Además, no se aprecian tampoco diferencias significativas en la mano a la hora de visualizar o no las opciones. Por lo tanto, se ha concluido que situar las opciones de un posible menú en la mano sin ser visualizadas puede presentar ventajas frente a un menú fijo, ya que no se percibe mejoría ni en precisión ni en tiempo y ofrece la ventaja de evitar la oclusión del escenario.

Palabras Clave: *hand tracking*, menú, Oculus Quest, propiocepción, Unity, realidad virtual

ÍNDICE

1. INTRODUCCIÓN	3
1.1. Motivación de elección	3
1.2. Motivación del proyecto.....	3
2. ANTECEDENTES.....	4
2.1. Conceptos	4
2.1.1. Propiocepción	4
2.1.2. Menús en Realidad Virtual	5
2.2. Estado del arte	7
2.2.1. Artículo propiocepción.....	7
2.2.2. Artículo interfaces en RV	9
3. OBJETIVOS.....	11
4. DESARROLLO DEL PROYECTO.....	12
4.1. Hardware y Software utilizado	12
4.1.1. Oculus Quest 2	12
4.1.1.1. Hand Tracking	14
4.1.2. C# y Visual Studio.....	15
4.1.3. Unity	16
4.1.3.1. Qué es y por qué lo he elegido	16
4.1.3.2. Objetos, componentes y paquetes usados	19
4.1.3.3. Vincular Oculus Quest 2 con Unity	22
4.1.3.4. Vincular scripts en C# con Unity	24
4.1.4. Inverse Kinematics.....	25
4.2. Diseño y desarrollo de escenarios.....	27
4.2.1. Planteamiento	27
4.2.2. Diseño	28
4.2.3. Desarrollo	31
4.2.4. Evaluación de las pruebas de usuario.....	34
4.2.4.1. Hipótesis	34
4.2.4.2. Metodología.....	35
4.2.4.3. Análisis de resultados.....	42
5. CONCLUSIONES	50
6. LÍNEAS FUTURAS DE TRABAJO	52
7. BIBLIOGRAFÍA.....	53
8. ANEXOS.....	55

1. INTRODUCCIÓN

A continuación, explicaré por qué he escogido este proyecto y cuál es la motivación del mismo.

1.1. Motivación de elección

He elegido este proyecto porque el mundo y entorno de realidad virtual me ha llamado siempre la atención, a pesar de no haber tenido interacción previa con unas gafas de realidad virtual. Me parece un campo muy interesante y del que todavía se puede sacar mucho partido.

Por otro lado, el sentido de la propiocepción se me hace muy llamativo también. Me pareció que en los mundos virtuales podría encajar mucho y ayudar a mantener la inmersión del usuario dentro de aplicaciones o videojuegos, además de luego poder aplicarse al ámbito de la rehabilitación, por ejemplo.

1.2. Motivación del proyecto

En este proyecto se pretende entender muy bien el sentido de la propiocepción (qué es, qué abarca y cómo funciona) además de estudiar diferentes formatos de interfaces de usuario dentro del mundo virtual. Se quiere comprobar si el sentido de la propiocepción puede brindar nuevas formas de interacción dentro de estas interfaces para mejorar la experiencia de usuario, ayudando a mantener la inmersión sin tener que renunciar a la sencillez o la comodidad de la interacción.

El proyecto se centra en diseñar y desarrollar unas interfaces muy sencillas con el principal objetivo de evaluar la interacción de los usuarios con las mismas y poder concluir si la propiocepción puede ser provechosa en este campo o, en su defecto, no. En caso de probar su utilidad, este diseño de interfaces podría llegar a ser aplicable en otros ámbitos como, por ejemplo, para acciones directas dentro de videojuegos o para ejercicios de rehabilitación neuromuscular.

2. ANTECEDENTES

En este apartado se expondrá qué estudios previos he tenido que realizar para entender bien el entorno, el contexto de trabajo.

2.1. Conceptos

Para poder diseñar y desarrollar interfaces que hagan uso del sentido de la propiocepción ha sido necesario realizar una investigación previa y entender qué es exactamente, cómo funciona y qué comprende.

2.1.1. Propiocepción

La palabra propiocepción deriva de las raíces latinas *propius*, que significa “de uno mismo”; y *capere*, que significa “tomar” o “agarrar”. Existen diferentes definiciones de la propiocepción, pero me he inclinado por la dada en [1]:

Proprioception, or kinesthesia, is the sense that lets us perceive the location, movement, and action of parts of the body. It encompasses a complex of sensations, including perception of joint position and movement, muscle force, and effort. These sensations arise from signals of sensory receptors in the muscle, skin, and joints, and from central signals related to motor output. Proprioception enables us to judge limb movements and positions, force, heaviness, stiffness, and viscosity. It combines with other senses to locate external objects relative to the body and contributes to body image. Proprioception is closely tied to the control of movement.

De esta definición podríamos entender que, gracias a la propiocepción, conocida comúnmente como “sexto sentido”, somos capaces de ubicar en el espacio nuestro cuerpo sin necesidad de utilizar los otros sentidos.

Los receptores que captan la información propioceptiva y la transmiten al cerebro son:

- Husos neuromusculares (*muscle spindle* en figura 1): situados en el vientre muscular.
- Órganos tendinosos de Golgi (*Golgi tendon organ* en figura 1): están en el tendón y en la unión del músculo y el tendón.
- Propioceptores capsuloligamentosos (*joint receptors* en figura 1): se encuentran en las cápsulas y ligamentos.

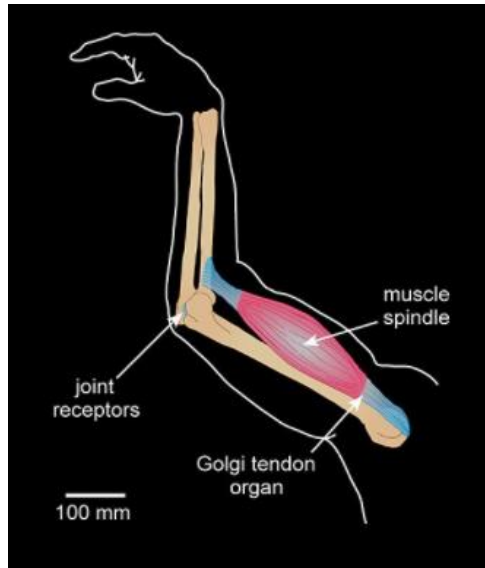


Figura 1. Dibujo que muestra receptores propioceptivos

2.1.2. Menús en Realidad Virtual

En [2] explican el concepto de realidad virtual de la siguiente manera:

Virtual reality (VR) is an advanced, human-computer interface that simulates a realistic environment. The participants can move around in the virtual world. They can see it from different angles, reach into it, grab it and reshape it. Cyberspace is thought of as the ultimate virtual reality environment.

Uno de los factores más importantes y por lo que más destaca la realidad virtual es la inmersión dentro del mundo virtual. Como se dice en [2], es un ambiente realista que se puede percibir desde diferentes ángulos y con el que se puede interactuar a otro nivel. Por este motivo es de vital importancia poder encontrar una interfaz de usuario que pueda mantener ese nivel de inmersión dentro del mundo virtual.

Dentro de la realidad virtual existen diferentes formatos de interfaces de usuario con los que interactuar. Los más comunes son:

- ***Floating menu***: son aquellos que están flotando en el entorno virtual, como el mostrado en la figura 2. Este formato es el más extendido. Tienden a romper la cuarta pared, por lo que conviene usarlos en situaciones en las que no importe sacar al usuario de la inmersión, como, por ejemplo, menús de pausa o menús principales.

- Controller menu: son los menús que se despliegan del controlador, como el que se ve en la figura 3. Este formato respeta más la inmersión del usuario en el entorno y conviene utilizarlo para selección de opciones que impactan directamente con el controlador, como selección de armas en un videojuego o selección de brochas en una aplicación de pintura.
- Environment menu: este tipo de menú está integrado con los objetos del entorno virtual, como el ordenador que se puede ver en la figura 4. Con este formato se puede escoger opciones que afectan directamente al entorno manteniendo la inmersión del usuario y dándole una impresión de interacción más profunda.

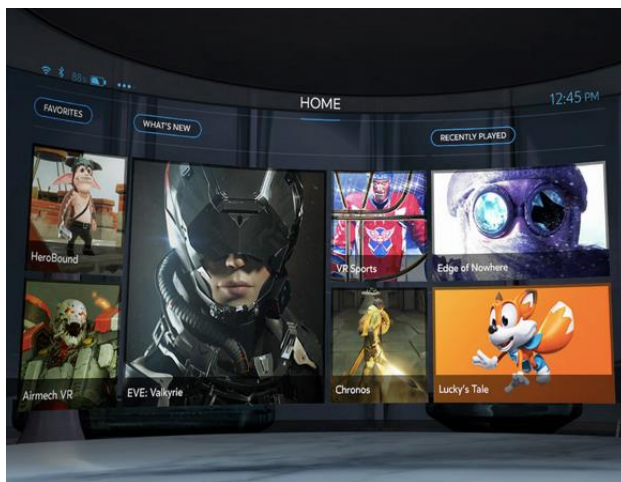


Figura 2. Floating menu

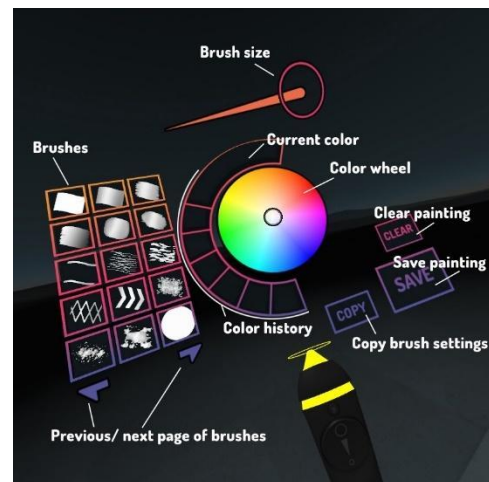


Figura 3. Controller menu



Figura 4. Environment Menu

Existen también otros formatos de menú, como aquellos que se despliegan de la mano, pero están menos estudiados y extendidos.

2.2. Estado del arte

Son muchas las investigaciones centradas en que la interacción de las personas usuarias con los mundos virtuales sea natural e intuitiva manteniendo la sensación de inmersión sin perjudicar la eficacia en la selección de opciones. Sin embargo, utilizar el sentido de la propiocepción como herramienta para el diseño de los menús es todavía es un campo por explorar.

A continuación, describo los dos artículos encontrados más relevantes y en los que me he basado para diseñar los menús que se evaluarán.

2.2.1. Artículo propiocepción

En este artículo se describe un experimento realizado por Wolf *et al.* [4]. En él investigan cuánta guía visual necesitan los usuarios para interactuar con la parte trasera de un dispositivo.

A los participantes del experimento se les solicitaba tocar puntos en la parte trasera del aparato que se mostraban en la pantalla superior. Los resultados obtenidos mostraban que los usuarios tocaban los puntos con exactitud sin necesidad del *feedback* visual. El esfuerzo sin *feedback* visual no aumentaba mucho y no afectó en gran medida a los parámetros medidos como, por ejemplo, la efectividad.

El objetivo de este experimento era averiguar si el *feedback* visual en esta clase de interacción es necesario o si puede hacerse uso de la propiocepción sin sacrificar efectividad, entre otras cosas.

Diseño del experimento

Se realizó un estudio del movimiento de los humanos y de su interacción con los ordenadores. Formularon la hipótesis de que, con la propiocepción, la falta de visibilidad no afectaría mucho.

Para cuantificar el impacto se realizó el experimento con 10 participantes diversos. Se juntaron dos dispositivos iPad pegados por la parte trasera (véase figura 5) para permitir la interacción del pulgar con la parte delantera y del resto de dedos con la parte posterior. Los usuarios tenían que seleccionar *targets* separados entre sí por 15mm.

Se plantearon 5 técnicas diferentes:

- Interacción delantera sin *feedback*

- Interacción trasera con *feedback*
- Interacción trasera sin *feedback*
- Interacción delantera y trasera con *feedback*
- Interacción delantera y trasera sin *feedback*

En aquellas técnicas con *feedback* aparecían puntos rojos en la posición donde los usuarios tocaban la pantalla trasera.

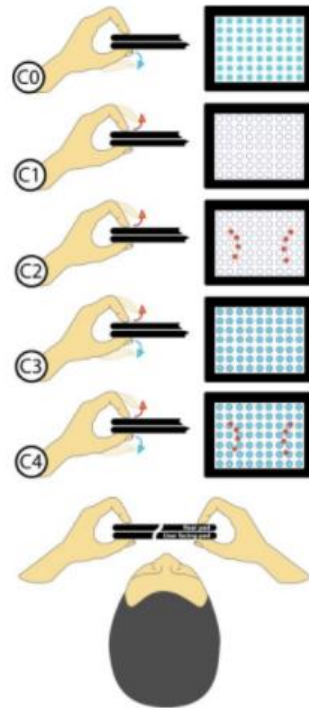


Figura 5. Prototipo interactivo que muestra targets en la pantalla frontal. Los dedos que tocan la pantalla trasera se visualizan como círculos rojos en la pantalla frontal

Los objetivos debían borrar círculos azules tocándolos con alguna de las técnicas mencionadas anteriormente. Se grabaron las pruebas y se hizo recuento del número de puntos azules borrados correctamente, además de fijarse en medidas más descriptivas como, por ejemplo, cuántos intentos les llevó borrar cada círculo.

Resultados

El rendimiento de cada usuario se definió por la efectividad (número de *targets* seleccionados correctamente) y por la eficiencia (cuántos intentos por *target*). Los resultados mostraron que la provisión de *feedback* visual no tenía un impacto significativo en el rendimiento cuantitativo o subjetivo.

Conclusión

En el artículo, tras realizar el experimento, se llega a la conclusión de que la propiocepción compensa la falta de *feedback* visual ya que, sin este, no cambian mucho los resultados. Si se mostraran los dedos podría llegar a considerarse redundante.

2.2.2. Artículo interfaces en RV

En [5] parten de la premisa de que las interfaces e interacciones convencionales en aplicaciones realidad virtual quedarán obsoletas. En consecuencia, han estudiado dos tipos de interfaces de menú: fijo y "manual"; con tres técnicas de selección diferentes: *hand pointing with button press* (Hand-BT), *head pointing with button press* (Head-BP) y *head pointing with dwell* (Head-DW). Es decir, han utilizado dos variables independientes para evaluar las diferentes interacciones: tipo de menú y técnica de selección.

Diseño del experimento

Como se ha mencionado anteriormente se utilizaron dos variables independientes: tipo de menú y técnica de selección. Al haber escogido dos formatos de menú y tres técnicas de selección distintas se hicieron un total de 6 escenarios donde se controlaban: el tamaño del *target*, la distancia entre los ítems, el *feedback* de apuntar y el de confirmación de selección.

Todos los menús tenían 16 ítems diferentes utilizando distintas formas y colores. Se trató de escalar el tamaño de las opciones del menú según la distancia del usuario a las mismas, como se muestra en la figura 6.

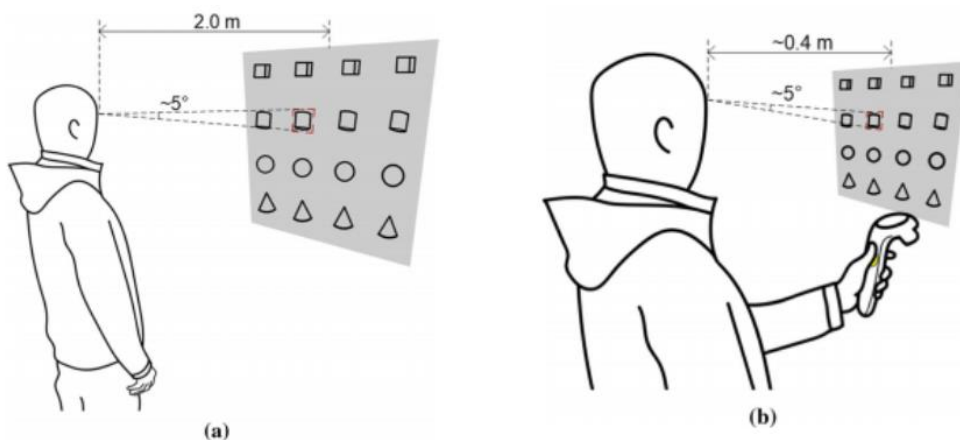


Figura 6. Escalado de opciones de menú fijo (a) y manual (b) para mantener el mismo tamaño

La interacción se realizaba con el *raycast* del controlador en la técnica de Hand-BP y con el *raycast* del HMD en el caso de las técnicas Head-BP y Head-DW. La selección en Hand-BP se confirmaba al pulsar el *touchpad* del controlador al igual que en Head-BP, pero en Head-DW, se confirmaba tras mantener la selección un umbral de tiempo de 780 ms.

Se midieron 4 métricas diferentes con sus respectivas variables dependientes: rendimiento (tasa de error, tiempo de realización), usabilidad percibida (facilidad de uso, capacidad de aprendizaje, eficiencia, precisión), carga de la tarea (exigencia mental, exigencia física, exigencia temporal, rendimiento, esfuerzo, frustración) y, por último, preferencia.

Cada menú constaba de un total de 16 selecciones y cada participante probó las seis interfaces en un orden aleatorio obtenido de un cuadro latino. Para cada selección una ruleta giraba y elegía la opción que el usuario debía elegir en el menú. Se pedía a los participantes que hicieran balance entre velocidad y precisión de forma que intentaran seleccionar la opción lo más rápido posible manteniendo el número de errores al mínimo. Tras completar los seis menús los usuarios rellenaron un cuestionario para evaluar diferentes aspectos de la interacción.

Resultados

La evaluación del rendimiento de cada usuario se basó en el tiempo de finalización de la tarea y en la tasa de error. Los resultados mostraron que la mejor técnica de selección para los menús fijos era Hand-BP, mientras que para aquellas interfaces manuales era mejor Head-BP.

3. OBJETIVOS

Para poder centrar y encaminar correctamente el proyecto he tenido que establecer una serie de objetivos, de manera que siempre tuviera claro qué paso se debía tomar. A continuación, enumero dichos objetivos que me han ayudado a dar forma al desarrollo del trabajo:

- Estudio conceptos:
 - Entender sentido de la propiocepción: comprender qué es exactamente para poder sacar partido de ella a la hora de diseñar el menú.
 - Estudiar tipos de interfaces de usuario en realidad virtual
- Aprendizaje de las herramientas a utilizar:
 - Oculus Quest 2: mirar especificaciones, entender cómo funcionan, saber manejarse con ellas.
 - Unity: estudiar entorno, cómo funciona y familiarizarse con él.
 - C#: aprender características principales del lenguaje, mirar scripts, hacer programas de prueba.
- Desarrollo proyecto:
 - Diseño del experimento
 - Diseño de los escenarios
 - Desarrollo de los escenarios
 - Evaluación de pruebas de usuario
 - Análisis de resultados
 - Obtención de conclusiones

4. DESARROLLO DEL PROYECTO

En este apartado se describen las herramientas utilizadas y las tareas llevadas a cabo para alcanzar los objetivos del proyecto.

Se comienza describiendo el estudio de las herramientas hardware y software utilizadas y se termina con la evaluación de las pruebas de usuario y el análisis de resultados correspondiente a las mismas.

4.1. Hardware y Software utilizado

A continuación, se explicará qué tecnologías software y hardware se han usado, qué son, su historia y cómo se han utilizado.

4.1.1. Oculus Quest 2

Oculus Quest 2 es un *headset* de realidad virtual creado por Oculus, una marca que pertenece a Facebook. Es el sucesor de Oculus Quest.



Figura 7. HMD y controladores de Oculus Quest 2



Figura 8. Entorno virtual de inicio de Oculus Quest 2

Las Oculus Quest 2, gracias a la tecnología integrada Oculus Insight, es capaz de realizar el seguimiento de cabeza y de manos en el espacio tridimensional, es decir, seguimiento de 6 grados de libertad. Hay disponibles dos versiones: una con 64GB de almacenamiento, y otra de 256GB.

Tiene dos paneles de visualización LCD (pantalla de cristal líquido) de cambio rápido con IPD (distancia interpupilar física) ajustable, de 58, 63 y 68mm. De esta manera se puede ajustar para tener una visión nítida y evitar fatiga. Tiene una resolución de pantalla de 1832x1920 por ojo y tiene una frecuencia de actualización de 72Hz o 90Hz, según la versión. Tiene un procesador Qualcomm® Snapdragon™ XR2 con 6GB de RAM. Tiene micrófono y altavoces integrados, y además es compatible con auriculares

de 3.5mm de entrada. Tiene conectividad inalámbrica por WIFI 6 y Bluetooth 5.1. Pueden usarse de forma independiente o pueden conectarse al ordenador a través de Oculus Link, usando el puerto USB tipo C del que dispone.



Figura 9. Headset Oculus Quest 2

Los controladores tienen unas dimensiones de 9x12 y un peso de 126g por controlador. Cada controlador dispone de un joystick, un botón de menú Oculus, dos botones de interacción, uno tipo *trigger* (gatillo) y otro tipo *grip* (de agarre). Usan baterías 2AA, una por controlador.

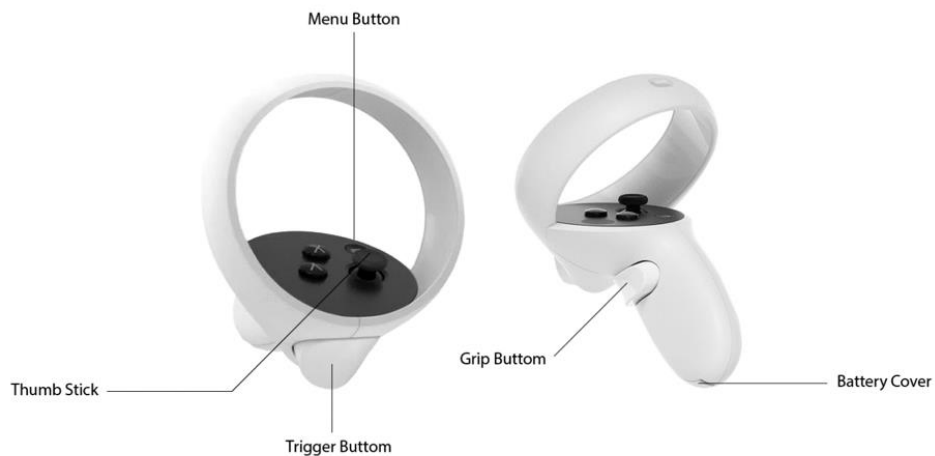


Figura 10. Controladores de Oculus Quest 2

Este dispositivo, además de hacer uso de los controladores *touch*, permite usar las manos a través del *hand tracking*.

4.1.1.1. *Hand Tracking*

El *headset* de las Oculus Quest 2 tiene 4 cámaras internas que detectan a tiempo real la posición y la orientación de puntos clave de las manos, como nudillos o yemas, y la configuración de los dedos. Teniendo esta información se aplican una serie de algoritmos de visión artificial para rastrear el movimiento de las manos y su orientación.

Si se usan las manos como entrada estas proyectan un puntero de cursor láser análogo al del controlador estándar con el que se puede realizar las interacciones dentro de las aplicaciones, como seleccionar o clicar opciones entre otras cosas.

Las Oculus Quest 2 (y Quest) reconocen por defecto una serie de gestos cuando se hace uso del *hand tracking*:

- *Point and Pinch* (apuntar y pellizcar): selección de ítems.
- *Pinch and Scroll* (pellizcar y arrastrar): hacer scroll hacia arriba, abajo, izquierda o derecha.
- *Palm Pinch* (pellizco de palma): pellizcar con la palma mirando hacia arriba y mantener para ir al menú inicial de Oculus.



Figura 11. Oculus Home usando *hand tracking*

La posibilidad de utilizar las manos para la interacción con las aplicaciones nos resultó muy interesante para usarlo en el proyecto. Este es el motivo por el que se eligió este

dispositivo. También hubo que tener en cuenta que presenta algunos problemas al superponer las manos, ya que se pierde el *tracking* de las manos.

4.1.2. C# y Visual Studio

Inicialmente Unity soportaba tres lenguajes de programación, todos ellos orientados a objetos: Boo, UnityScript y C#. Boo se declaró obsoleto para la versión de Unity 5 y UnityScript (versión de JavaScript) lo hizo en 2017 con el lanzamiento de Unity 2017.1. C# es el lenguaje más utilizado y del que más información hay, además de que Unity basa toda su documentación en él. Teniendo en cuenta todos estos factores, se eligió utilizar C# para la programación de los scripts.

"C#" es un lenguaje de programación multiparadigma desarrollado y estandarizado por la empresa Microsoft. Su lanzamiento fue en 1999. Es un lenguaje orientado a objetos.



Figura 12. Logo de C#

Visual Studio 2019 ha sido el entorno de desarrollo integrado (IDE) que he utilizado para programar los scripts en C#.

Es compatible con múltiples lenguajes de programación (C#, Visual Basic, Java...) y está preparado para Windows y macOS. Su primera versión fue lanzada en 1998.

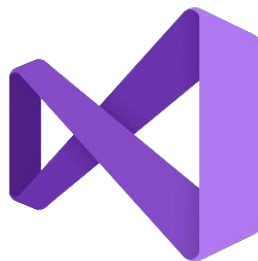


Figura 13. Logo de Visual Studio

4.1.3. Unity

4.1.3.1. *Qué es y por qué lo he elegido*

Unity ha sido el motor de creación de videojuegos que he utilizado. Unity, creado por la empresa Unity Technologies, es un motor de videojuego multiplataforma. Está disponible para Microsoft Windows, Mac OS y Linux, entre otras plataformas. Unity permite la creación de videojuegos en 2D y en 3D.

Inicialmente comencé a desarrollar el proyecto con Unreal Engine 4.0, pero Unity tenía -en mi opinión- una curva de aprendizaje más sencilla.

A continuación, se mostrarán algunas de las ventanas disponibles dentro de la interfaz de Unity usando como ejemplo una escena de un paisaje forestal [6] conseguido en el *Asset Store*.

Asset Store

Se introdujo *Unity Asset Store* dentro del editor de Unity en noviembre de 2010 para facilitar el acceso a los desarrolladores a todo su contenido.

Desde *Asset Store* se busca, descarga e importa el paquete deseado: desde modelos 3D a servicios en línea. En este caso descargamos [6], como se ve en la figura 14.

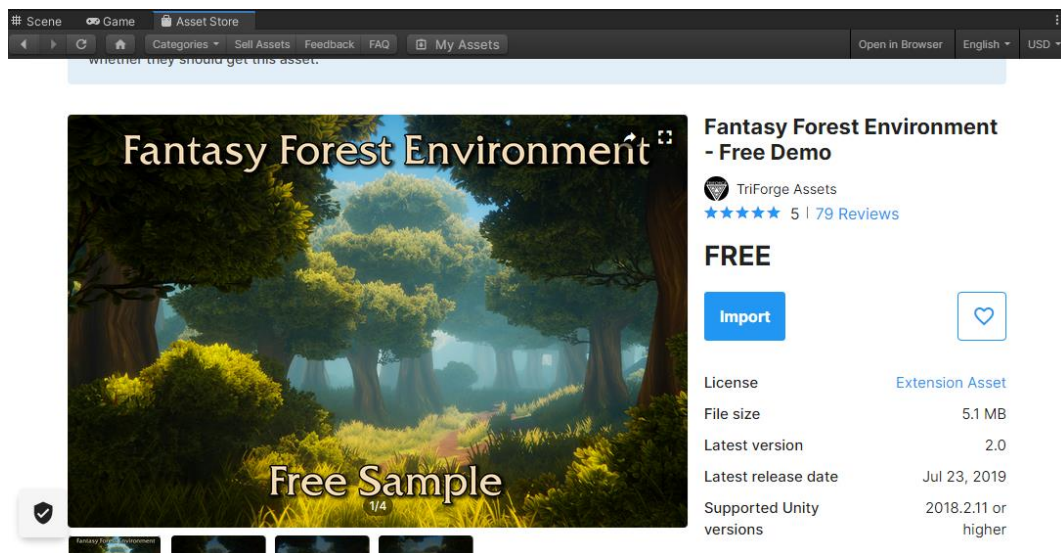


Figura 14. Ventana de Asset Store en Unity

Una vez importado el paquete se abre la escena de prueba (“demoScene_free”). Las escenas contienen todos los elementos del juego o la aplicación que se esté desarrollando.

Project View

En la sección llamada *Project* se puede ver y acceder a todos los directorios y ficheros que hay dentro del proyecto, como muestra la figura 15.

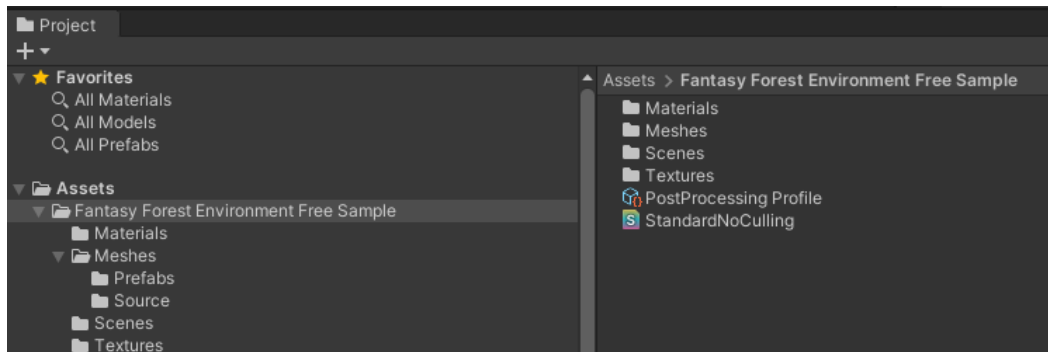


Figura 15. Ventana de Project en Unity

Hierarchy

La ventana *Hierarchy* muestra todos los elementos que hay dentro de la escena y la jerarquía que llevan, como puede verse en la figura 16. Desde aquí se pueden crear nuevos elementos, borrar los existentes, modificarlos o cambiar el orden de la jerarquía.

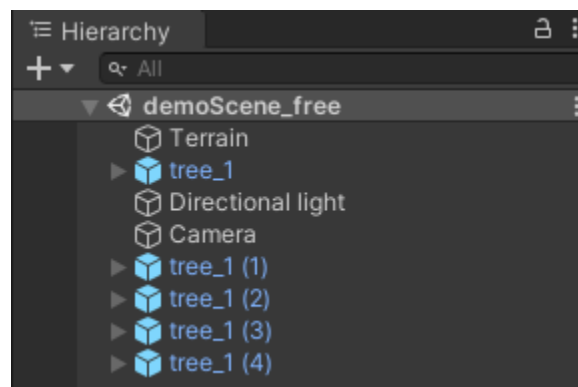


Figura 16. Ventana de Hierarchy en Unity

Scene View

Desde *Scene* (véase figura 17) se pueden ver todos los elementos dentro de la escena, el espacio que ocupan, la posición con respecto a otros objetos, etc. También permite redimensionar y recolocar los objetos, entre otras cosas, de una forma más visual y directa.

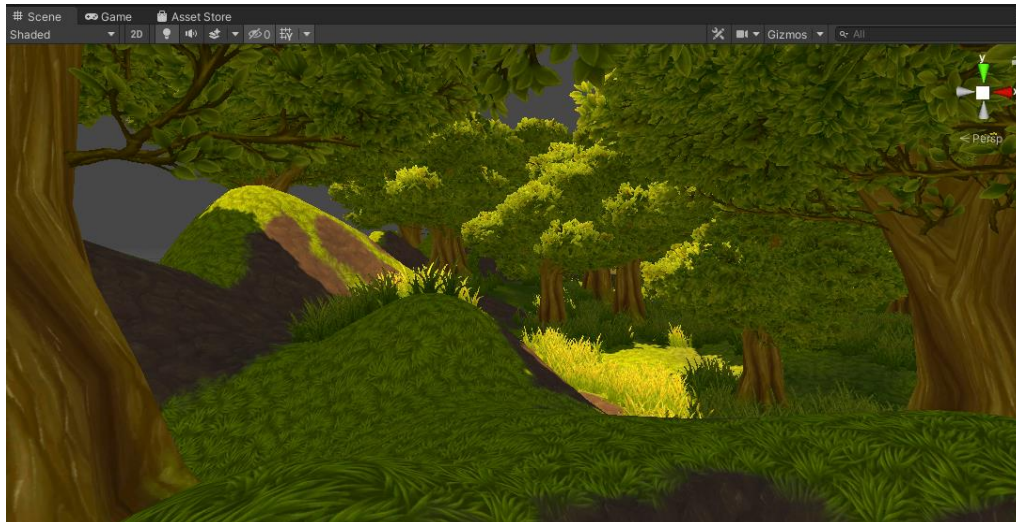


Figura 17. Ventana de Scene en Unity

Game View

En *Game*, como se puede ver en la figura 18, se puede visualizar todo el entorno de la escena de la forma en la que se vería dentro del videojuego o aplicación correspondiente. Cuando se ejecuta la escena se proyecta en esta ventana lo que se ve desde las gafas.

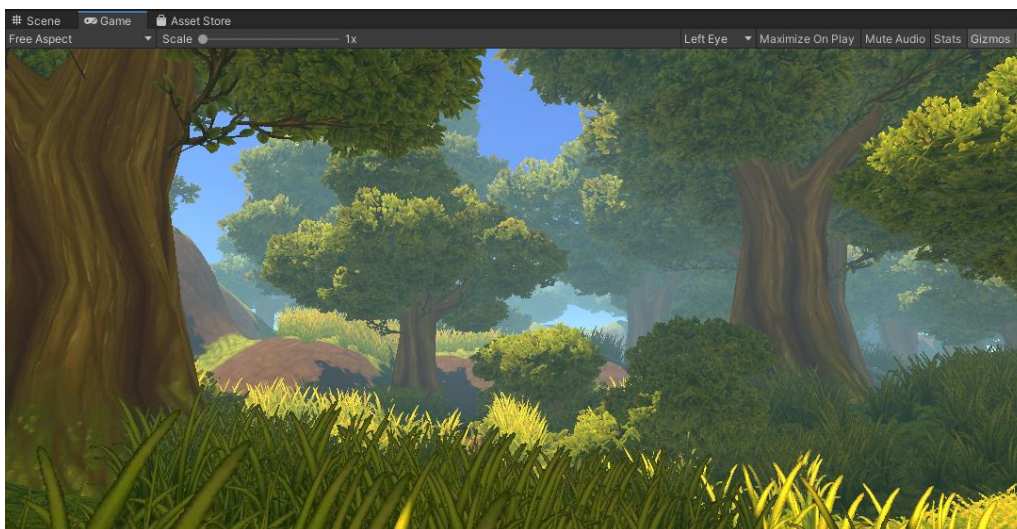


Figura 18. Ventana de Game en Unity

Inspector

En la pestaña *Inspector* se puede ver un panel con los atributos del *GameObject* seleccionado, como muestra la figura 19 habiendo seleccionado el objeto *Ground*, que corresponde al suelo de una escena. Desde ahí se pueden modificar las propiedades del objeto, añadir o borrar componentes.

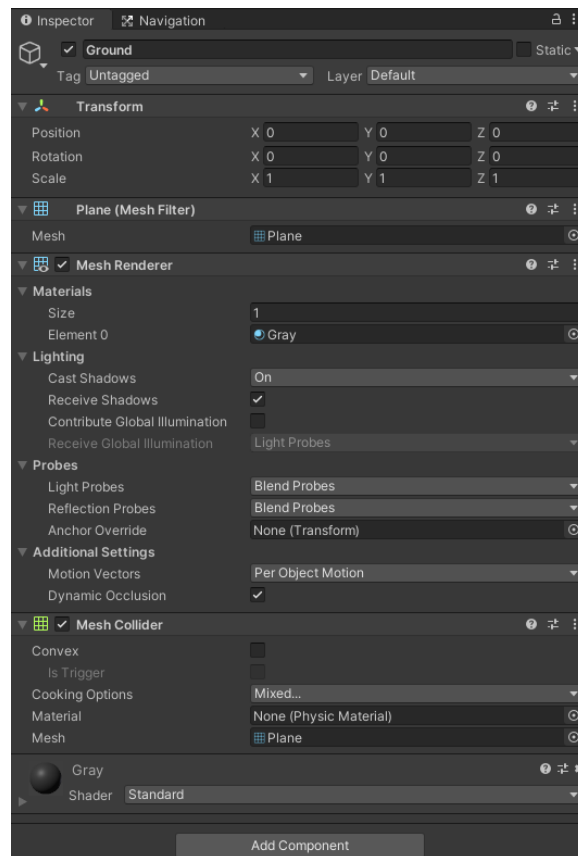


Figura 19. Ventana de Inspector en Unity

4.1.3.2. Objetos, componentes y paquetes usados

En este apartado expondré algunos de los objetos, componentes y paquetes más importantes que he utilizado en este proyecto.

Oculus Integration

El paquete más usado ha sido Oculus Integration [7]. Proporciona soporte para desarrollar aplicaciones de Oculus en Unity. Incluye scripts y prefabricados, entre otros elementos, para complementar el soporte integrado de Unity. Contiene, por ejemplo, funciones de renderizado y una interfaz para controlar el comportamiento de la cámara.

OVRCameraRig

Este objeto sustituye a la cámara principal dentro de la escena en Unity. Pertenece al paquete de Oculus Integration. Este objeto cámara es el más adecuado a utilizar si se quiere trabajar con las Oculus ya que tiene acceso al objeto *OVRManager* que es el que proporciona la interfaz al hardware de realidad virtual correspondiente y se encarga de optimizar el renderizado para la pantalla estereoscópica de las Oculus.

OVRHandPrefab

Es un objeto prefabricado de la mano que ofrece Oculus Integration. Se incluye dentro de la jerarquía del objeto *OVRCameraRig* y es el que construye las manos cuando se usa *hand tracking* dentro de la aplicación.

InteractableToolsSDKDriver

Este objeto es un prefabricado que también pertenece a Oculus Integration. Se ha usado para añadir unos sensores en las puntas de los dedos del prefabricado de las manos que detectan el contacto con otros objetos de la escena.

Primitive objects

Existen un tipo de objetos denominados “Objetos primitivos” que se pueden crear directamente dentro de la jerarquía de Unity. Estos objetos son: el cubo, la esfera, la cápsula, el cilindro, el plano y el cuádruple (véase figura 20). Para añadir estos objetos en la escena se puede hacer a través del menú *GameObject > 3D Object* y escoger el objeto deseado, o bien hacerlo desde un script de C#.

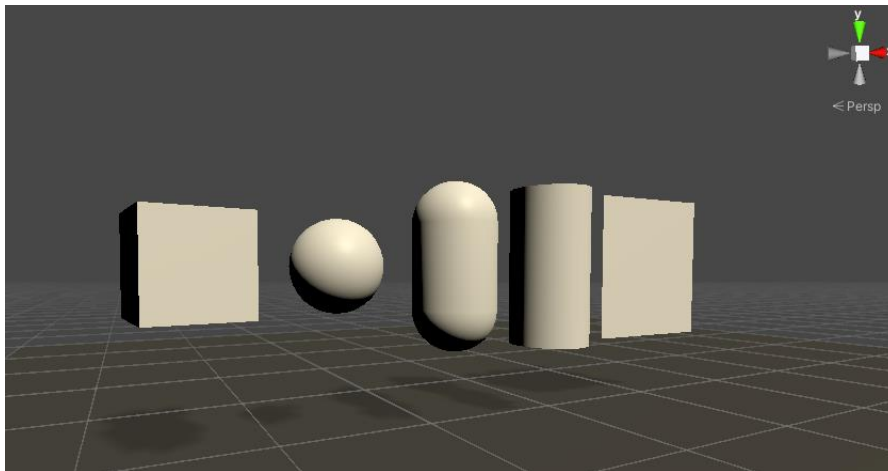


Figura 20. Escena con objetos primitivos

En este proyecto se han utilizado 3 tipos diferentes de objetos primitivos:

- Plano: los planos se han utilizado para representar las opciones de los menús visibles.
- Cubo: los cubos se han utilizado para detectar la selección de una opción en los menús fijos y del antebrazo.
- Esfera: las esferas se han utilizado para detectar la selección de una opción en los menús de los dedos.

Button Controller

Este componente se ha utilizado para detectar el contacto con alguna de las opciones y desencadenar alguna acción, en este caso el registro de la selección entre otras cosas.

Canvas

Canvas es el objeto que debe contener todos los elementos UI dentro de Unity. Si se crea un elemento UI (*button*, *text*, *scrollbar*, etc) automáticamente se crea también un *canvas* que lo contiene. Este elemento hace uso de un *EventSystem* que mantiene informado al sistema de mensajes de Unity.

Los elementos interactivos, como por ejemplo los botones o los *scrollbar*, no conseguí que funcionaran bien con Hand Tracking. Por este motivo el único elemento que se ha utilizado dentro del *canvas* en el desarrollo de este proyecto es el objeto texto.

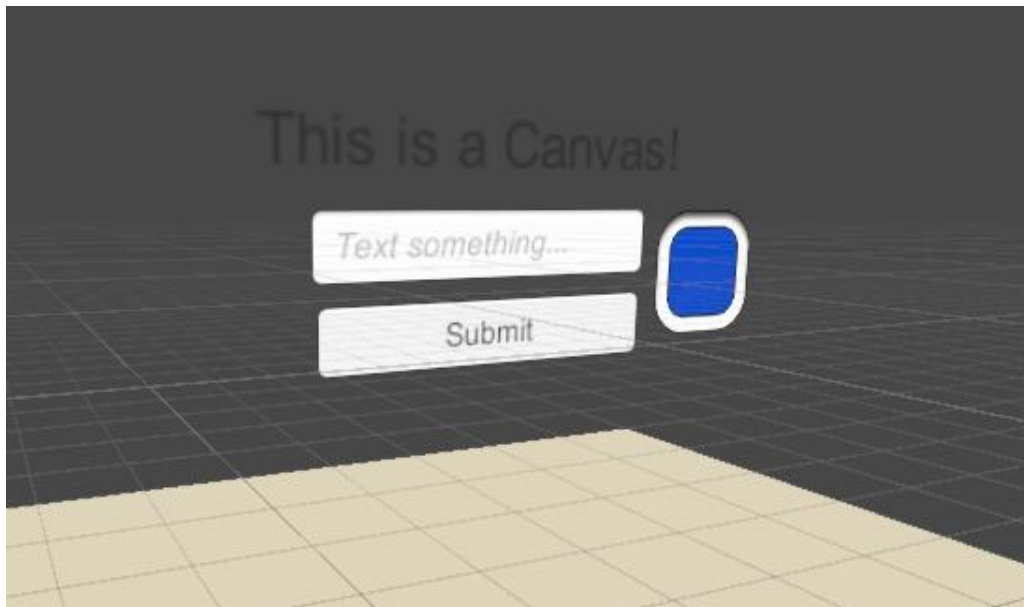


Figura 21. Escena con estructura canvas

Para usarlo en aplicaciones de realidad virtual hay que poner el modo de renderizado a "World Space" y en el atributo de cámara hay que añadir el objeto "CenterEyeAnchor" contenido en el prefabricado de *OVRCameraRig*, como se muestra en la figura 22.

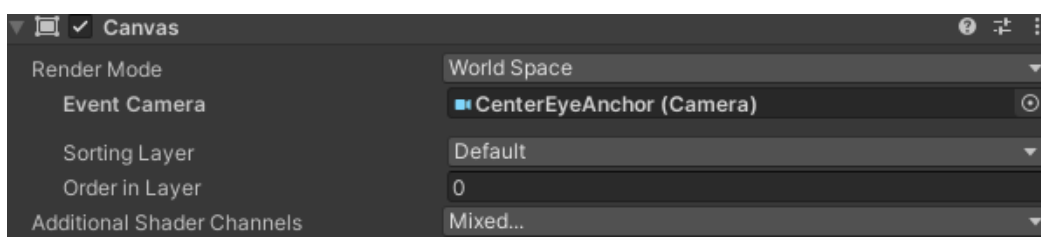


Figura 22. Captura componente canvas

4.1.3.3. Vincular Oculus Quest 2 con Unity

Para poder utilizar las Oculus Quest 2 en el entorno de Unity hubo que hacer uso de librerías del *Asset Store* y seguir una serie de pasos para la configuración que se explicarán a continuación.

El paquete de Oculus Integration es uno de los que contiene todos los prefabricados necesarios para poder usar las gafas con los controladores o las manos.

Para poder hacer uso de las gafas fue necesario:

1. Descargar el paquete "Oculus Integration" desde la *Asset Store* e importarlo al proyecto.
2. Activar modo de desarrollador (*developer mode*) desde la configuración de las Oculus. Conectar el headset al ordenador mediante el Oculus Link Cable y conceder permiso *USB debugging* al PC. Activar Oculus Link desde las gafas.
3. En Unity > *Build Settings*

1. Cambiar plataforma de PC a Android

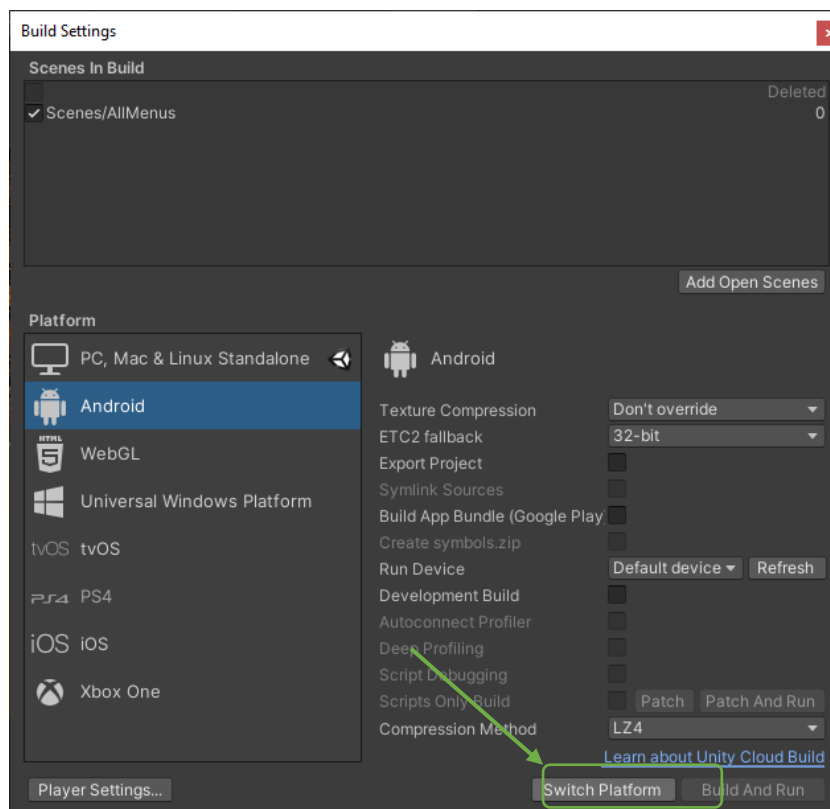


Figura 23. Build Settings de Unity

2. En *Player Settings*

a. Añadir Oculus en *Virtual Reality SDK's*

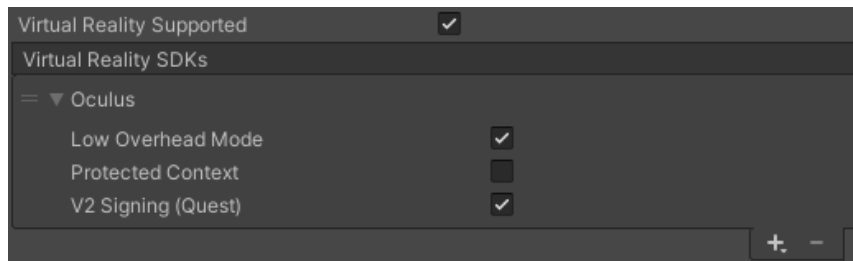


Figura 24. Configuración "Virtual Reality Supported" en Unity

b. Eliminar "Vulkan" de *Graphics APIs*

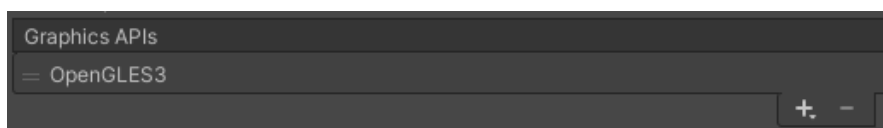


Figura 25. Configuración "Graphics APIs" en Unity

c. Poner versión mínima de Android: 4.4 Kit-Kat



Figura 26. Configuración "Minimum API Level" en Unity

1. Una vez la configuración de Unity está preparada incluir en la escena los elementos para poder utilizar las Oculus y el *hand tracking*:

1. Se incluye en la escena un objeto de tipo "OVRCameraRig"

a. En el componente OVRManager dentro del objeto:

1. *Tracking > Tracking Origin Type: Floor Level*

2. *Display > Color Gamut: Quest*

3. *Quest Features > Hand Tracking Support: Only Hands / Hands and Controllers.*

2. Dentro de OVRCameraRig añadir los prefabricados de las manos OVRHandPrefab en su correspondiente "Hand Anchor" y configurar las componentes para sus correspondientes manos: derecha o izquierda.

3. Añadir objeto de tipo "HandsManager" y asignar como parámetros de la mano derecha e izquierda los prefabricados añadidos en el paso anterior.

4. Para poder interactuar con otros objetos dentro de la escena es necesario incluir un objeto del tipo InteractableToolsSDKDrivers y añadir tantos

sensores como se deseen: se pueden colocar detectores en las puntas de los dedos o en formato *raycaster*.

- a. Para que aparezcan colocados correctamente en los dedos hay que marcar la celda de "Enable Physics Capsules" dentro del prefabricado de cada mano.

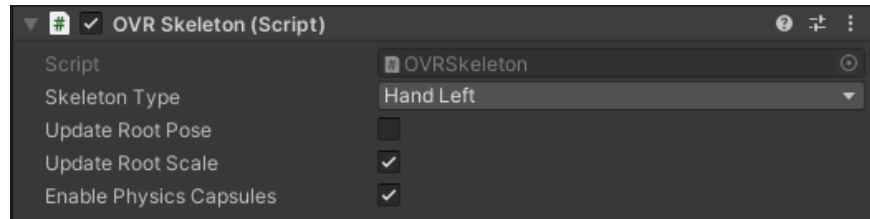


Figura 27. Configuración OVRSkeleton en Unity

- b. Para que esto vayan a la misma velocidad que los dedos hay que modificar lo siguiente: en *Edit > ProjectSettings > Time* asignar a "Fixed Timestep": 0.005 y a "Maximum Allowed Timestep": 0.05.

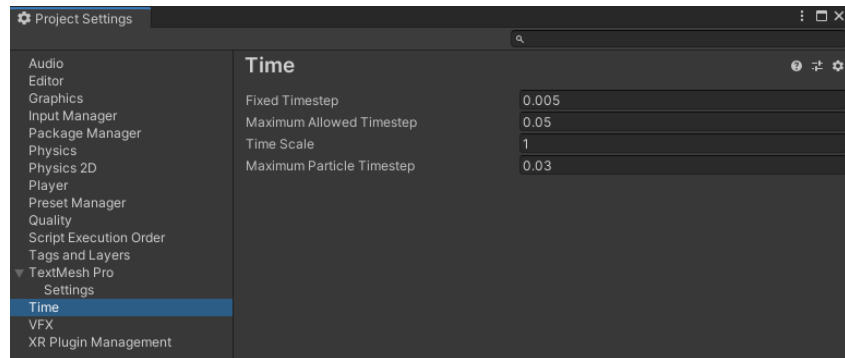


Figura 28. Configuración de Time en Project Settings

4.1.3.4. Vincular scripts en C# con Unity

Para poder asociar los scripts desarrollados en C# con Unity habría que crear un objeto en la escena (GameObject) al que se le añade el componente correspondiente al script. De esta forma ya queda asociado. Aquellas variables que se han declarado como públicas en el código aparecerán dentro del componente del objeto y podrán ser inicializadas con el valor deseado desde la ventana *Inspector* de Unity.

Si se crea algún objeto de una clase del script también podrán ser usadas las funciones públicas de la clase. De esta forma se puede definir la forma y el comportamiento del objeto.

4.1.4. Inverse Kinematics

Desde un principio planteamos la posibilidad de desarrollar una de las interfaces de usuario adherida a los brazos. Las Oculus Quest 2 solo tienen reconocimiento de manos, de brazos no, por lo que habría que construir una representación propia de los mismos, ya lo hiciéramos con los controladores o con *hand tracking*.

La representación del movimiento tendría que ser realista porque si íbamos a basar los menús en la propiocepción y el brazo virtual no correspondiera con el real entraría en conflicto. Dada esta situación pensamos en hacer uso de la cinemática inversa.

Cinemática Inversa

En Robótica, la Cinemática inversa (IK) es la técnica que permite determinar el movimiento de una cadena de articulaciones para lograr que un actuador final se ubique en una posición concreta. El cálculo de la cinemática inversa es un problema complejo que consiste en la resolución de una serie de ecuaciones cuya solución normalmente no es única [8].

La cinemática inversa trata de calcular los valores de las coordenadas de las articulaciones del robot para que su extremo tome la posición y orientación según una localización espacial determinada.

El movimiento de una cadena cinemática lo modelan las ecuaciones cinemáticas de la propia cadena, ya sea un robot o un personaje animado. Estas ecuaciones son las que definen la configuración de la cadena en función de sus parámetros.

Cinemática Inversa en Unity

Unity tiene su propio paquete llamado "Inverse Kinematics" que se puede conseguir en su *Asset Store*. Este paquete contiene funciones matemáticas basadas en la cinemática inversa.

Investigando en busca otros proyectos similares en los que hicieran uso de realidad virtual y esta librería encontré un proyecto de Unity [9] llamado "VRArmIK" y lo descargué. En él ha desarrollado una demo en un escenario en el que, usando los controladores de Oculus, se pueden visualizar unos brazos implementados con las librerías de Inverse Kinematics que reproducen bastante bien el movimiento natural de los mismos en la vida real. Desde su cuenta de YouTube ha subido un vídeo [10] donde muestra cómo funciona. Las capturas de la figura 29 pertenecen al mismo.

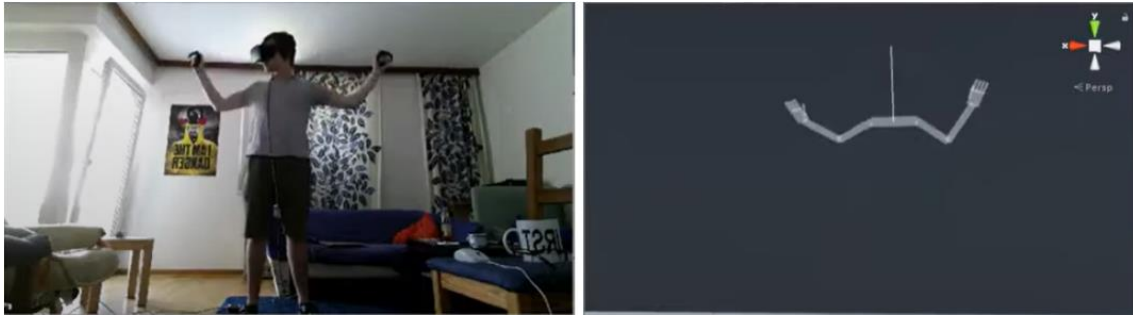


Figura 29. Captura obtenida del vídeo que muestra movimiento real del usuario y su reproducción por el paquete Inverse Kinematics

Una vez descargado tuve que hacer unos pequeños ajustes para que funcionara correctamente en mi entorno con los controladores: cambiar posición absoluta del Avatar, cambiar la cámara utilizada por OVRCameraRig, etc. El movimiento de los brazos era coherente, es decir, su comportamiento era muy similar a la realidad, pero no se ajustaba del todo al movimiento del brazo real.

De cualquier modo, intenté probar su funcionamiento con *hand tracking*. Para ello desactivé la visualización de las manos construidas con objetos primitivos, introduje en la escena los prefabricados de las manos y cambié los puntos de referencia que tenía el avatar y el objeto "Pose Manager". Como puede verse en la figura 30, funcionaba mal: tenía por defecto unos tamaños de *upper arm* y *forearm* definidos que, si no corresponden con el tamaño del usuario, el movimiento mostrado no corresponde en absoluto con el real. Conforme el tamaño del brazo se ajustaba más al real mejor funcionaba, pero sin llegar a hacerlo bien del todo. Para solucionar esto, se pensó en hacer una calibración previa (a la hora de evaluar) para obtener el tamaño del brazo del usuario y así que funcionara mejor, pero era demasiado complejo para los resultados que se podía llegar a obtener.

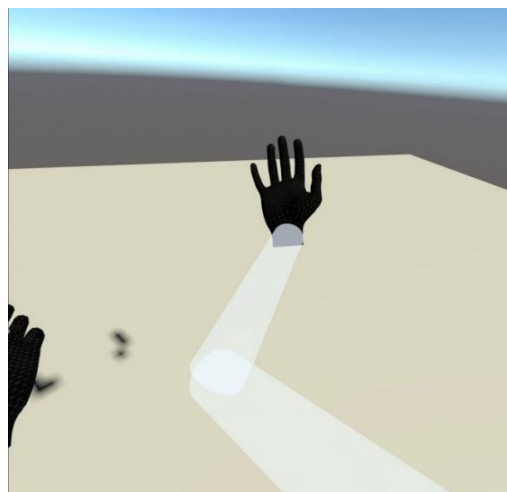


Figura 30. Brazo mal reproducido usando paquete inverse kinematics

Finalmente llegamos a la conclusión de que iba a ser contraproducente el uso de estas librerías y planteamos la posibilidad de obligar al usuario a mantener una posición determinada si se iba a hacer uso de una interfaz en el brazo.

4.2. Diseño y desarrollo de escenarios

4.2.1. Planteamiento

Partimos de la premisa de que teníamos que hacer uso del sentido de la propiocepción para seleccionar las opciones de un menú, por tanto, tuvimos que pensar qué formato de menú y de interacción con el cuerpo sería más apropiada. Planteamos diferentes opciones:

- Menú de cuerpo entero
- Menú con gestos
- Menú en la mano
- Menú en los dedos
- Menú en el brazo

Menú de cuerpo entero

Planteamos inicialmente

- usar parches corporales que al contacto detectaran la selección de una opción
- utilizar los acelerómetros de los controladores para ubicar la posición espacial de las manos además de otros aparatos para ubicar las otras partes del cuerpo

pero descartamos estas opciones porque habríamos necesitado hardware auxiliar y queríamos evitarlo. Lo único que se nos ocurrió sin necesidad de hardware auxiliar fue hacer una calibración previa de las partes del cuerpo que corresponderían a las opciones, guardar su posición absoluta y restringir el movimiento del usuario, lo que nos resultó poco natural y contraproducente.

Menú con gestos

A la hora de realizar gestos se hace uso de la propiocepción, pero parecía muy simple. Sin embargo, para líneas futuras de trabajo podría ser interesante combinar este formato con otros.

Menú en la mano

Las Oculus, como ya hemos comentado con anterioridad, tienen *hand tracking* por lo que no es necesario el uso de los controladores. De este modo, planteamos poner el menú a lo largo de la mano, pero esto suponía un problema ya que, al superponer una mano con la otra, se pierde el *tracking* y el menú no se mantiene.

Menú en los dedos

Habiendo visto el problema planteado con el menú en la mano pensamos que se podía reducir a opciones en las puntas de los dedos ya que al seleccionarlas no tendría por qué haber oclusión ni fallo en el *tracking*.

Menú en los brazos

Como ya he comentado con anterioridad, planteamos hacer un estudio para intentar integrar la librería de *inverse kinematics* y así conseguir reproducir el movimiento de los brazos a partir de las posiciones de las manos y de la cabeza. Si no funcionaba planteamos la opción de fijarlo a la mano y obligar al usuario a actuar en una posición restringida.

4.2.2. Diseño

Basándonos [5] donde se hacía un experimento con 6 escenarios diferentes con dos variables independientes (tipo de selección y tipo de menú) planteamos un experimento similar. Pensamos en 2 x 3 escenarios con dos variables independientes: visibilidad (menú visible o no visible) y tipo de menú (fijo, dedos y antebrazo).

FIJO VISUAL	MANO VISUAL	BRAZO VISUAL
FIJO NO VISUAL	MANO NO VISUAL	BRAZO NO VISUAL

Para poder tener en cuenta el factor de la oclusión pensamos que sería apropiado poner elementos dentro de la escena. Por ese motivo desde el *Asset Store* se descargó [11], un entorno de una habitación medieval.

A continuación, expongo las características principales del diseño del experimento.

Características generales

- El experimento consiste en la interacción del usuario con los 6 escenarios diferentes: aparece una instrucción con el número de la opción que el usuario tiene que tocar hasta que el usuario realiza la selección, entonces cambia. Son

bloques de 19 selecciones. Se indica el principio y el fin de cada bloque y se proporciona un *feedback* cuando el participante haga una selección.

- Todos los menús disponen de 4 opciones. Los paneles que representan las opciones de los menús no están numerados para que no supusiera una ventaja de los menús visibles sobre los no visibles.
- Las opciones en los menús de la mano son del tamaño de las yemas de los dedos puesto que son superficies discretas. En los menús fijo y del antebrazo, al ser superficies continuas, las opciones están escaladas siguiendo el mismo método de escalado que en [5], como muestra la figura 6. El antebrazo al tener un tamaño acotado es el que determina el tamaño de las opciones y es el menú fijo el que hace el escalado a partir de este.
- Todas las selecciones se realizan con el índice derecho y en la mano izquierda se coloca el menú de la mano y del brazo.
- La mano izquierda no se visualiza para que el sentido de la propiocepción no se vea influido por el de la vista.

Características menú fijo

En este caso, consideramos que las opciones del menú fijo debían estar ancladas a las gafas en vez de al escenario ya que esto simplificaría las cosas al usar el menú no visible, teniendo en cuenta que el usuario podría estar mirando en cualquier dirección. El ancho del menú no ocupa toda la visión ya que, como se ha comentado en las consideraciones generales, las opciones están escaladas siguiendo una técnica específica.



Figura 31. Captura del menú fijo del proyecto final

Características menú mano

Las opciones del menú están en las yemas de los dedos y ocupan esa superficie, de forma que hay que realizar las selecciones con la palma abierta mirando hacia el cuerpo. Como he mencionado antes, las opciones no se escalan al ser los dedos superficies discretas.



Figura 32. Captura del menú de los dedos del proyecto final

Características menú antebrazo

Como se ha mencionado anteriormente, no fue posible conseguir que funcionara bien el brazo construido con la librería de Inverse Kinematics por lo que optamos por obligar al usuario a mantener una posición determinada y anclar el menú a la mano de forma que encajara con el antebrazo. El set de opciones ocupa desde la muñeca hasta el codo aproximadamente.

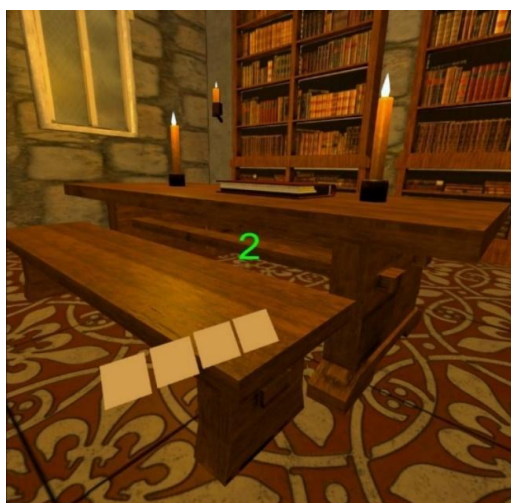


Figura 33. Captura del menú del antebrazo visible del proyecto final

4.2.3. Desarrollo

En este apartado describiré el desarrollo de las diferentes partes dentro del proyecto de Unity.

Desarrollo de scripts

Para poder registrar las selecciones, guardar datos de interacción y definir comportamiento del proyecto han tenido que programarse una serie de scripts que son los siguientes:

- He creado una clase llamada “**Registro**” que guarda el identificador del participante, el número de la escena, una lista de enteros con las instrucciones generadas, una lista de enteros con las selecciones realizadas, una lista de booleanos correspondiente a los aciertos y por último una lista de *float* para guardar el tiempo que tarda el usuario en realizar cada acción. Se ha creado la función “saveParticipantData()” que crea un objeto de la clase Registro y guarda todos los datos obtenidos durante la evaluación completa de un menú. En caso de no existir un fichero correspondiente al participante se crea y se guardan los datos en formato Json en el fichero log. En cuanto se completa el bloque de 19 *trials* del menú, se guardan los datos automáticamente.
- He creado la clase **Contador** que es la que genera las instrucciones, gestiona las selecciones y activa el *feedback* de selección. Cada vez que comienza la evaluación de un menú, se instancia un objeto de la clase *Random* que es la que genera las instrucciones. La semilla del objeto Random depende del tiempo. Esto se hace para que la secuencia de instrucciones sea diferente en cada escenario y que no pueda haber proceso de aprendizaje de una interfaz a otra. Cuando se toca una opción con el dedo índice derecho se registra en esta clase y en caso de que haya pasado un margen de tiempo mayor o igual a 2 segundos desde la anterior selección, la selección se guarda y se da por válida, en caso contrario se descarta la selección. Si se da por válida se activa el *feedback* visual y tras un segundo se muestra el icono de retirada de mano. Cuando vuelven a pasar los dos segundos, vuelve a estar preparado para registrar nuevas selecciones.
- El script “**FingerExtension**” asigna las opciones y las esferas que detectan la selección de una opción (en el caso del menú de la mano) a los huesos de los dedos. Recorre todos los huesos de la mano y coloca las opciones en los puntos correspondientes: las yemas de los dedos.

- El script “**ForearmExtension**” se encarga de ubicar las opciones y los cubos que detectan la selección de una opción (en el caso del menú del brazo). Recorre todos los huesos de la mano y coloca las opciones en los puntos correspondientes, poniendo la opción 1 cercana al codo y la número 4 cercana a la muñeca.
- La clase **Chooser** es la que gestiona el cambio de un escenario a otro. Los menús van numerados del 1 al 6, cuando se pulsa una de estas teclas se activa el menú correspondiente y con los contadores reseteados. En caso de pulsar la tecla F se desactiva el menú activo en el momento y se resetean los contadores. Si se realizan los 19 trials del menú, se guarda en el fichero log llamando a la función expuesta con anterioridad.

Configuración mano

Como ya he comentado en el apartado de diseño, se le obliga al usuario a utilizar únicamente el índice derecho para realizar las selecciones. Consideré que la mejor opción era colocar un sensor en la yema de este dedo únicamente. Por tanto, modifiqué el objeto creado en la escena de InteractableToolsSDKDrivers para poner como único detector de interacción el dedo índice derecho, es decir, FingerTipPokeToolIndex.

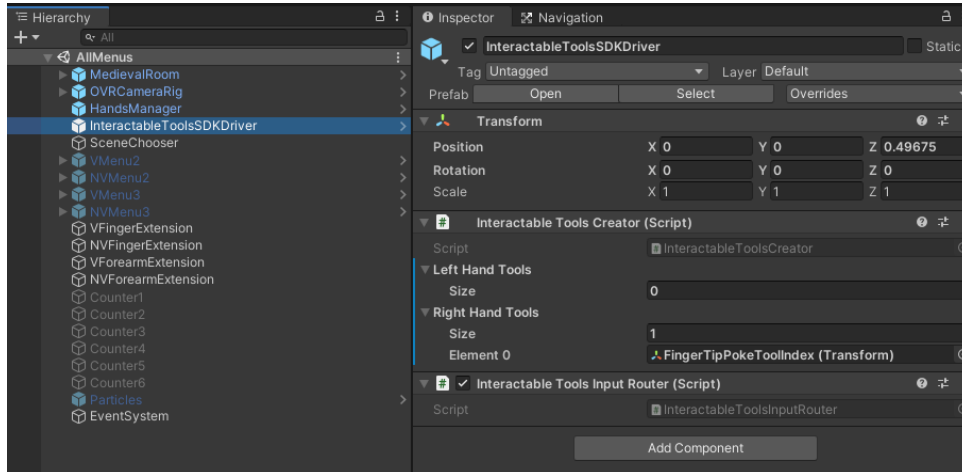


Figura 34. Imagen del objeto InteractableToolsSDKDriver de Unity

También tomamos la decisión de visualizar solo la mano derecha por lo que, en el prefabricado de la mano izquierda, se inhabilitó el componente de ‘mesh renderer’. De esta forma Unity sigue reconociendo la mano, pero no la muestra.

Preparación escena

Dentro de la escena, además de los elementos necesarios para utilizar las Oculus, la habitación medieval y los menús, se han creado una serie de elementos necesarios para el seguimiento del experimento:

- Objetos **Contador**: Estos objetos hacen uso del script “Contador” descrito anteriormente. Cada uno de ellos tiene como parámetros de entrada los objetos de las opciones del menú, por lo que cada menú tiene su propio contador. Además, tienen también los objetos pertenecientes a la interfaz como el panel con las instrucciones, el icono de la mano y el *feedback* de partículas.

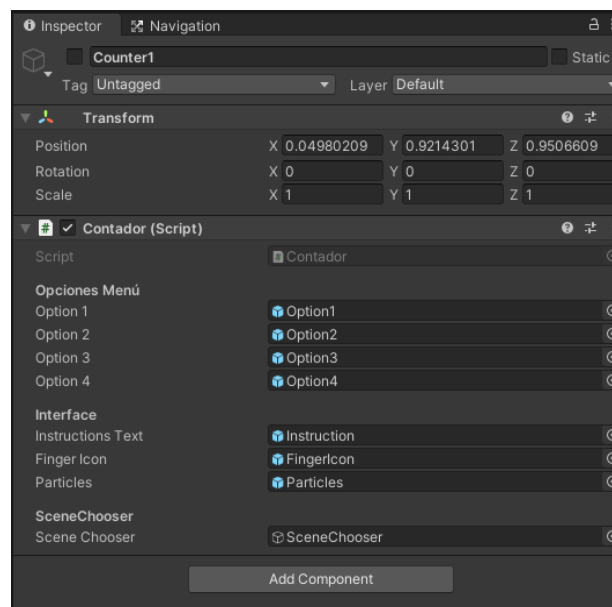


Figura 35. Imagen del objeto Contador en Unity

- Los objetos **FingerExtension** y **ForearmExtension**, que hacen uso de los scripts “FingerExtension” y “ForearmExtension”, contienen las opciones y detectores correspondientes a los menús de la mano y del brazo para poder asignarlos donde corresponde, tal como se ha descrito antes.
- Objeto **SceneChooser**: este objeto tiene como componente principal el script “Chooser” explicado con anterioridad. Es el gestor del experimento. A este objeto se le asigna el identificador del participante y todos los menús para poder controlarlos, es decir, activarlos y desactivarlos cuando corresponda.

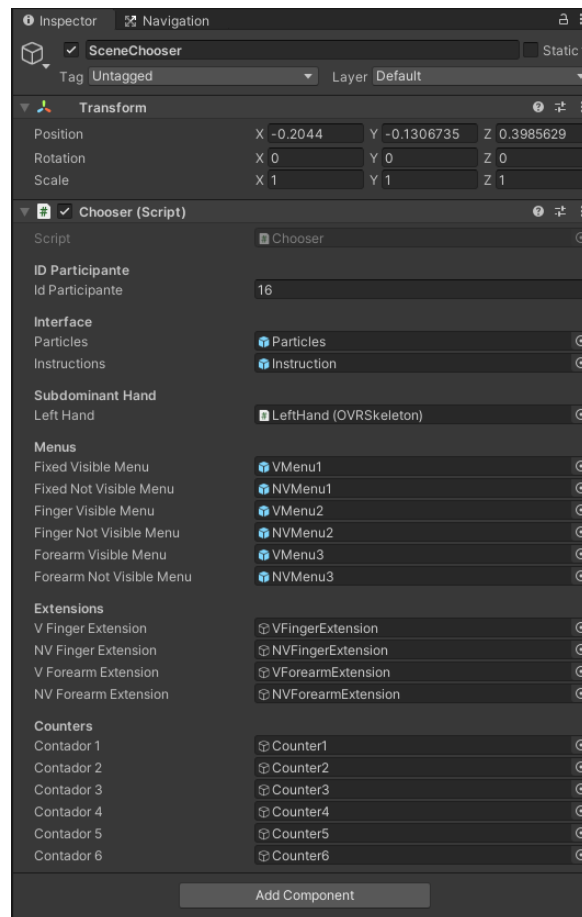


Figura 36. Imagen del objeto SceneChooser en Unity

4.2.4. Evaluación de las pruebas de usuario

4.2.4.1. Hipótesis

Basándonos en los artículos [4] y [5] y con la idea de probar qué formato de menús podrían ser más adecuados para un entorno virtual inmersivo, partimos de las siguientes hipótesis formuladas con respecto a diferentes parámetros:

- **H1: Respecto a la posición:** Partimos de la premisa de que el resultado obtenido por Wang *et al.* [5] en el que el menú fijo fue el más rápido, preciso y preferido por los usuarios se debe a que el menú manual se posicionó sobre el controlador y no directamente sobre el cuerpo. Nuestra hipótesis es que los menús corporales obtendrán resultados similares o mejores que el fijo gracias al sentido de la propiocepción.
- **H2: Respecto a la posición corporal:** Partimos de la premisa de que es importante definir correctamente dónde situar las diferentes opciones de menú y el formato

de las mismas. Nuestra hipótesis es que los puntos discretos van a ser ubicados de forma más precisa por el usuario que los situados en una superficie continua.

- H3: Respecto al opción visual: Basándonos en [4] la hipótesis formulada es que gracias al sentido de la propiocepción podemos eliminar completamente la visualización de las opciones del menú sin empeorar los resultados de velocidad y precisión. Si esta hipótesis es cierta, lograremos beneficiarnos de la ventaja que ofrece el menú corporal de reducir la oclusión del escenario virtual

4.2.4.2. Metodología

En este apartado se hará un resumen del procedimiento que hemos llevado a cabo para la realización del experimento

Participantes

Con el propósito de probar las hipótesis expuestas anteriormente hemos realizado el experimento a 18 personas diferentes de un rango de edades de 20 a 35 años, diestros, algunos con experiencia previa de ámbitos en realidad virtual y otros sin.

Se les ha pedido nombre, apellido, edad y sexo, y se les ha asignado un identificador único a cada uno de ellos.

Aparatos

Para la realización de las pruebas de usuario hemos necesitado:

- El ordenador con el que se ha programado el proyecto para ejecutarlo y hacer el seguimiento del experimento.
- Un ordenador portátil con las encuestas para que los usuarios que realizaran las pruebas las rellenaran.
- Las Oculus Quest 2 conectadas al ordenador sin hacer uso de los controladores, ya que se utiliza únicamente el *hand tracking* para la realización de las pruebas.



Figura 37. Fotografías sacadas durante el proceso de evaluación de usuarios

Diseño cuestionarios

Consideramos oportuno diseñar dos formatos de encuesta para los usuarios: una encuesta por menú, para poder valorarlo individualmente, y una encuesta final, para poder hacer una comparativa y evaluar la experiencia general.

❖ Encuesta por menú

Para recopilar las impresiones de los usuarios causadas por los diferentes formatos de interacción se preparó una serie de formularios. Se tomó como referencia UEQ+ [12], que es una extensión modular de UEQ (User Experience Questionnaire).

Valorando la naturaleza del proyecto y basándonos también en el artículo [5] mencionado anteriormente, se decidió escoger las variables: eficiencia, utilidad y uso intuitivo. De cada una de las variables se han formulado diferentes preguntas donde el usuario ha de responder evaluando en una escala del 1 al 7, donde el 1 sería el valor más "negativo" y 7 el más "positivo".

Eficacia

De la variable "eficacia" se formularon preguntas con respecto a la rapidez (lento/rápido), eficiencia (ineficiente/eficiente), practicidad (nada práctico/práctico) y la sencillez (recargado/sencillo). Con esta encuesta se quiere responder a las preguntas mostradas en la figura 38.

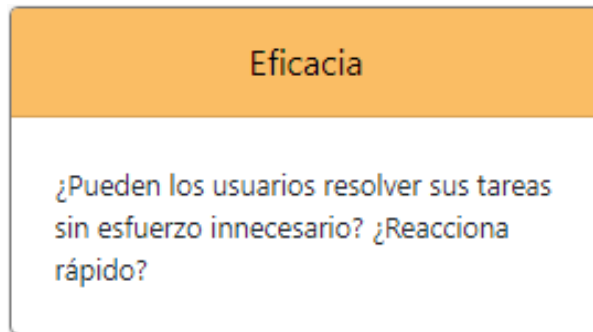


Figura 38. Captura del apartado eficacia de UEQ+

Uso intuitivo

De la variable "uso intuitivo" se formularon preguntas con respecto a la facilidad (árido/fácil), lógica (ilógico/lógico), evidencia (equivoco/evidente) y la coherencia (incoherente/coherente). Con esta encuesta se quiere responder a la pregunta mostrada en la figura 39.

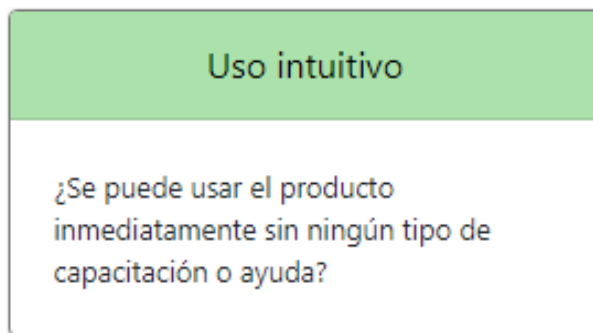


Figura 39. Captura del apartado de Uso Intuitivo de UEQ+

Utilidad

De la variable "utilidad" se formularon preguntas con respecto a la utilidad (inútil/útil), provecho (nada provechoso/provechoso), beneficio (nada beneficioso/beneficioso) y la rentabilidad (nada rentable/rentable). Con esta encuesta se quiere responder a la pregunta mostrada en la figura 40.

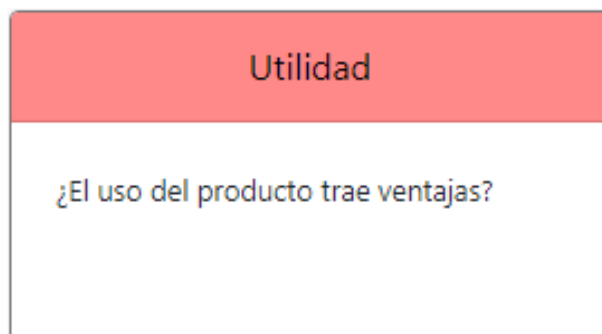


Figura 40. Captura del apartado Utilidad de UEQ+

Inmersión

Además de estas variables también consideramos importante tener en cuenta y que el usuario valorara el nivel de inmersión de cada uno de los menús y así poder sacar conclusiones a nivel de preferencia de los participantes. De este modo se formulamos dos afirmaciones distintas en las que el usuario tenía que mostrar su nivel de acuerdo del 1 al 7 donde el 1 es “Totalmente en desacuerdo” y el 7 es “Totalmente de acuerdo”. Las afirmaciones eran las siguientes:

- *Me he sentido inmerso en el mundo virtual*
- *Mientras realizaba la selección en el menú me he sentido inmerso en el mundo virtual.*

De esta manera se podía comprobar si el usuario sentía que el formato del menú impactaba positiva o negativamente en la inmersión en el mundo virtual.

Exigencia física

Consideramos interesante que los usuarios también evaluaran el nivel de actividad física que exigía la selección de las opciones de cada menú.

❖ **Encuesta final**

También se diseñó una encuesta final en la que se le pedía al participante hacer una comparativa entre los 6 modos de interacción diferentes, es decir, que los ordenaran del que más les hubiera gustado al que menos. Del mismo modo, se pedía que resaltarán lo que más les hubiera gustado del favorito y lo que menos del menos apreciado.

Finalmente, también se consideró relevante preguntar acerca de la impresión general de la inmersión dentro del mundo virtual durante el experimento y cómo afectaba la visibilidad de los menús.

Además, se facilitó un apartado donde pudieran dejar otro tipo de observaciones, impresiones o sugerencias.

Procedimiento

Al realizar las pruebas de usuario hemos llevado una estructura fija para así hacer todas las evaluaciones en igualdad de condiciones. La estructura es la siguiente:

- 1) Explicación del experimento

Se les explica a los usuarios el funcionamiento general de la prueba.

- Cómo son los escenarios: se explica que hay 6 escenarios diferentes:

1. Menú fijo visible
2. Menú fijo no visible
3. Menú dedos visible
4. Menú dedos no visible
5. Menú antebrazo visible
6. Menú antebrazo no visible

y que cada uno tenía 4 opciones distintas ordenadas de izquierda a derecha en el caso del fijo y en el antebrazo y el de los dedos el índice la opción número 1 y el meñique la opción número 4. Para la mejor comprensión de la disposición del menú se le enseña imágenes de los menús visibles con los paneles numerados y sin numeración.

- Contexto del experimento: se da una explicación breve para que puedan contextualizar el motivo del proyecto sin llegar a condicionar sus respuestas. Se les dice que queremos saber si, en un contexto de un videojuego o una aplicación, una interfaz no visible puede ayudar a mantener la inmersión en el mundo virtual al usuario o en su contra puede resultar más complicado o confuso.
- Características de los menús: se describe brevemente la posición y el funcionamiento de cada menú:
 - Menús fijos: Están anclados a las gafas y no ocupan todo el rango visual. Está aproximadamente a 40 cm de distancia. Enseñar qué tamaño aproximado.
 - Menús dedos: Hay que colocar la mano izquierda en horizontal (mostrar cómo) y tocar las opciones desde el lateral intentando no superponer una mano con la otra. Las opciones están en las yemas de los dedos.
 - Menús brazo: Para la correcta postura de este menú, hay que colocar el brazo apoyado con la palma abierta y pegada en la superficie formando un ángulo de 120° aproximadamente y mantener la muñeca recta, sin

moverla. El panel número 1 está alrededor del codo y el número 4 de la muñeca.

- Explicación del funcionamiento general del experimento: Se describe el procedimiento de la siguiente manera:

Se avisa de qué menú es el siguiente. Se activa el menú correspondiente y aparece un mensaje de “*Begin*” en magenta. Seguidamente aparece un número verde indicando el número de panel a seleccionar. Con el dedo índice de la mano derecha se selecciona alguna de las opciones, intentando hacer un balance entre velocidad y precisión. Cuando se seleccione alguna de las opciones (sea la correcta o no) el número de la instrucción cambia de verde a magenta y aparece un *feedback* de partículas de *fingerprints* en la posición de la opción seleccionada. Después aparece un icono de un dedo índice levantado indicando que hay que retirar el dedo de la opción. Una vez desaparezca el icono aparecerá inmediatamente después la siguiente instrucción en verde. Las selecciones van en bloques de 19. Cuando se realizan todas aparece un mensaje de “*End*”. Este procedimiento se llevará a cabo con cada uno de los 6 menús en un orden determinado.

- Cuestiones a resaltar: temas con los que hacer hincapié:
 - Solo detecta selección con el dedo índice derecho.
 - Solo acepta una selección por cada instrucción, sea correcta o no.
 - Retirar la mano de la opción seleccionada cuando aparezca el icono del dedo índice levantado.
 - Tratar de hacer balance entre velocidad y precisión.
 - No superponer las manos para no perder el trackeo.
 - Hacer movimientos exagerados, sin hacer barridos.
- Dudas: se le da la opción al participante de preguntar las posibles dudas que pueda tener.

2) **Firma del consentimiento**

El participante ha de firmar un consentimiento en el que se detalla

- Qué es y qué persigue el estudio

- Quién realiza la prueba
- Información que se recoge
- Contacto para solicitar más información

y con el que declara que

- Se le han explicado las características y los objetivos del estudio y se le ha dado tiempo y oportunidad para realizar preguntas entre otras cosas
- Autoriza la medición de datos y la grabación de audio y vídeo de las pruebas
- Es libre de retirarse del estudio en cualquier momento

3) **Formulario con datos del participante**

Los usuarios dan su nombre y apellido, su sexo y su edad y se le asigna un identificador único.

4) **Preparativos**

Se prepara el entorno: el usuario se prueba las gafas, se las ajusta, prueba gestos... Se conectan las gafas y se activa Oculus Link, se introduce identificador de participante en el objeto SceneChooser y se le da a play.

5) **Realización del experimento**

- a. **Se selecciona el escenario correspondiente:** cuando el proyecto está en ejecución se pulsa la tecla del escenario que corresponde (número de 1 al 6).
- b. **El participante realiza la interacción con el menú:** el usuario realiza el bloque de 19 selecciones del menú correspondiente.
- c. **El usuario completa la encuesta específica del escenario:** desde el portátil auxiliar el participante rellena el formulario correspondiente al menú con el que ha interactuado.

Para que no hubiese patrones de aprendizaje de un escenario a otro en la realización de las pruebas se diseñó un cuadro latino para determinar el orden de los escenarios para cada participante. Se hizo siguiendo el patrón mostrado en la figura 41, obtenido de [13].

1	2	n	3	n-1	4	...
2	3	1	4	n	5	...
3	4	2	5	1	6	...
⋮	⋮	⋮	...	⋮	⋮	⋮
n-1	n	n-2	1	n-3	2	...
n	1	n-1	2	n-2	3	...

Figura 41. Orden al diseñar cuadro latino

En la figura 42 puede observarse el resultado final del cuadro latino, teniendo $n=6$ al haber un total de 6 escenarios diferentes. Esto determinó que el número de participantes de la evaluación debía ser un múltiplo de 6.

		ORDEN					
		1º	2º	3º	4º	5º	6º
PARTICIPANTES	1	1	2	6	3	5	4
	2	2	3	1	4	6	5
	3	3	4	2	5	1	6
	4	4	5	3	6	2	1
	5	5	6	4	1	3	2
	6	6	1	5	2	4	3

Figura 42. Cuadro latino seguido para el orden de las evaluaciones de usuario

6) Se realiza la encuesta general

El participante hace la encuesta final en la que hace una comparación de los 6 menús diferentes, hace una valoración de qué es lo que más le ha gustado del menú favorito y qué es lo que menos del menú menos valorado. También se le hace preguntas con respecto a la sensación de inmersión dentro del experimento y si se valora más la inmersión al no ver los paneles o, en su contra, afecta negativamente.

4.2.4.3. Análisis de resultados

A continuación, se comentarán los resultados teniendo en cuenta las medidas objetivas (tiempo de selección y precisión) y las medidas subjetivas (obtenidas de las encuestas a los usuarios).

Medidas objetivas

Para el análisis de resultados, tal como se había planteado desde un inicio, de los bloques de 19 *trials* por menú, las tres primeras selecciones no se tienen en cuenta. Es decir, se dejó un margen de aprendizaje de 3 selecciones por menú al participante y son

las 16 últimas selecciones las que se han tenido en cuenta para las medidas calculadas para el análisis de resultados que se exponen a continuación.

Descriptives				
Posicion	V1	Mean	SD	N
Brazo	NoVisual	3.197	0.462	18
	Visual	3.251	0.525	18
Fijo	NoVisual	3.654	0.812	18
	Visual	3.159	0.288	18
Mano	NoVisual	3.353	0.525	18
	Visual	3.349	0.375	18

Figura 43. Tabla de medias y desviaciones estándar de tiempos

La figura 43 muestra todos los menús con sus respectivas medias y desviaciones estándar con respecto al tiempo de selección de los usuarios. Se puede observar que las medias oscilan entre los 3.15 y los 3.65 segundos aproximadamente y las desviaciones entre los 0.29 y los 0.81 segundos.

El menú fijo no visible es el que mayor media tiene, por lo que podríamos deducir que es el que más le “ha costado” ubicar y seleccionar al usuario. Por otro lado, la menor media y desviación estándar es el menú fijo visible por lo que se asume que las selecciones en el mismo han sido rápidas y constantes.

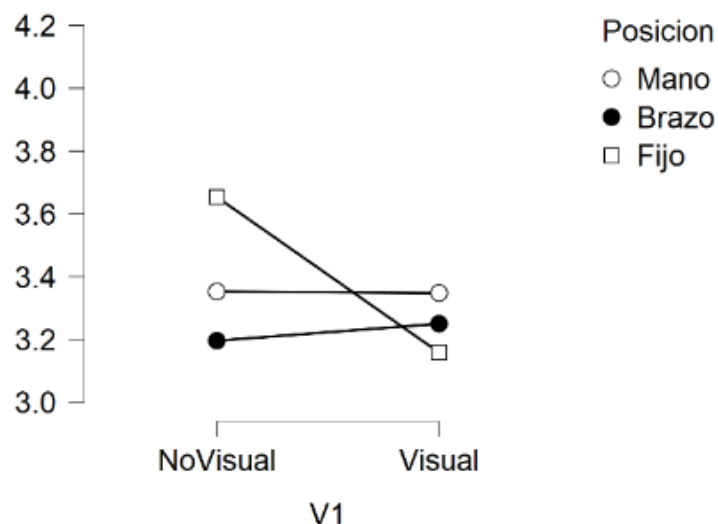


Figura 44. Gráfica relación tipo menú / tiempo

En la figura 44 podemos observar que, en los menús corporales, es decir, de la mano y del brazo, no hay una gran diferencia entre el tiempo de selección siendo visibles o no visibles. En cambio, en el menú fijo hay casi medio segundo de diferencia. De esto

podríamos deducir que, de ser un menú o forma de interacción no visible, la propiocepción ayuda a la selección en gran medida.

Descriptives				
Posicion	V1	Mean	SD	N
Brazo	NoVisual	0.604	0.228	18
	Visual	0.882	0.132	18
Fijo	NoVisual	0.899	0.096	18
	Visual	0.993	0.020	18
Mano	NoVisual	0.878	0.110	18
	Visual	0.993	0.020	18

Figura 45. Tabla de medias y desviaciones estándar de la precisión

La figura 45 muestra todos los menús con sus correspondientes medias y desviaciones estándar de la precisión de selección.

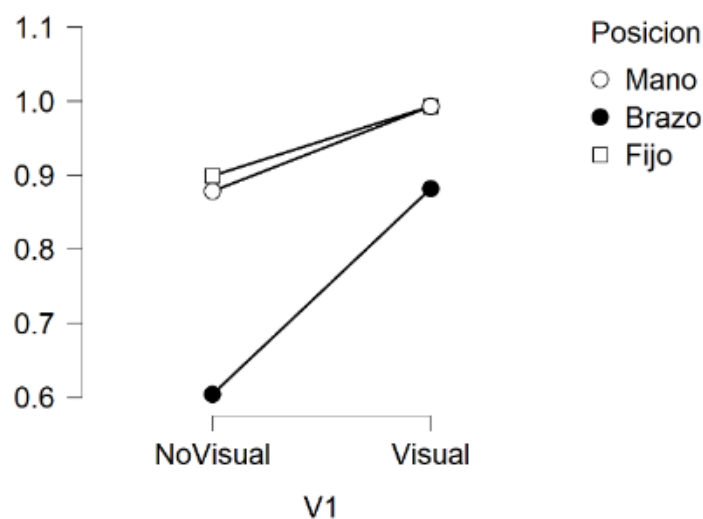


Figura 46. Gráfico relación tipo menú / precisión

En la figura 46 podemos ver que los menús del brazo tienen una tasa de acierto menor, el no visible con una diferencia más notable. En cambio, los menús fijos y de mano tienen valores similares. En los no visibles tienen una tasa de acierto cerca del 0.9 y en los visibles cerca del 1.0.

Medidas subjetivas

Para el análisis de las medidas subjetivas tomadas de UEQ+ [12], se utilizó de cada medida una única variable, la considerada más significativa. Por cada atributo se ha obtenido una gráfica que representa la media de respuestas (valoraciones del 1 al 7) de cada menú. En el eje x están los menús y, en el eje y, su media correspondiente.

- Eficiencia

En el caso de la *eficiencia* se escogió la variable “Ineficiente-Eficiente” puesto que era la más representativa del concepto.

Como se puede observar en la figura 47 el menú fijo está en ambos extremos: el menú fijo no visible es considerado el más ineficiente y el fijo visible el más eficiente, seguido de la mano visual. Según lo comentado por los usuarios el no tener ninguna referencia a la que atenerse para interactuar con el fijo no visible lo hacía poco práctico.

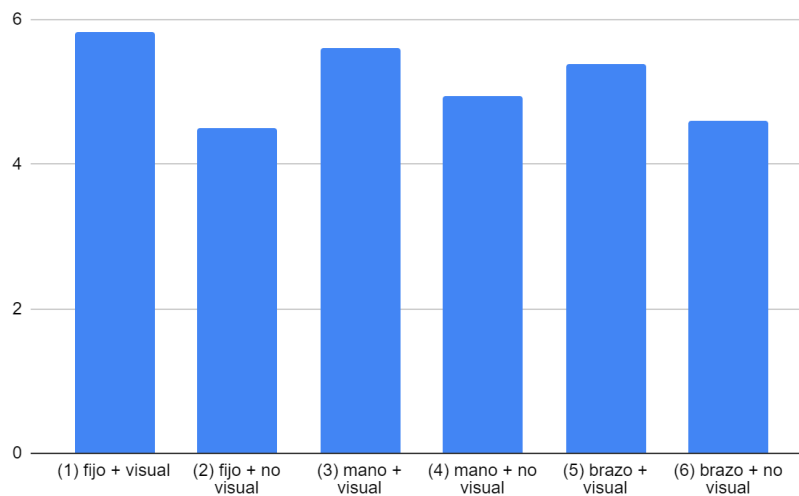


Figura 47. Nivel de ineficiente-eficiente valorado por el usuario

- Utilidad

En el atributo *utilidad* se utilizó la variable “Inútil-Útil” siendo también la más representativa del concepto. La figura 48 muestra que el considerado más inútil fue el menú fijo no visible y el más útil el de mano visible. Al parecer de los usuarios el menú de los dedos visibles resultaba muy útil ya que tiene *feedback* háptico y es sencillo de entender y con el que interactuar ya que sabes dónde está en todo momento y no produce oclusión con el resto del entorno.

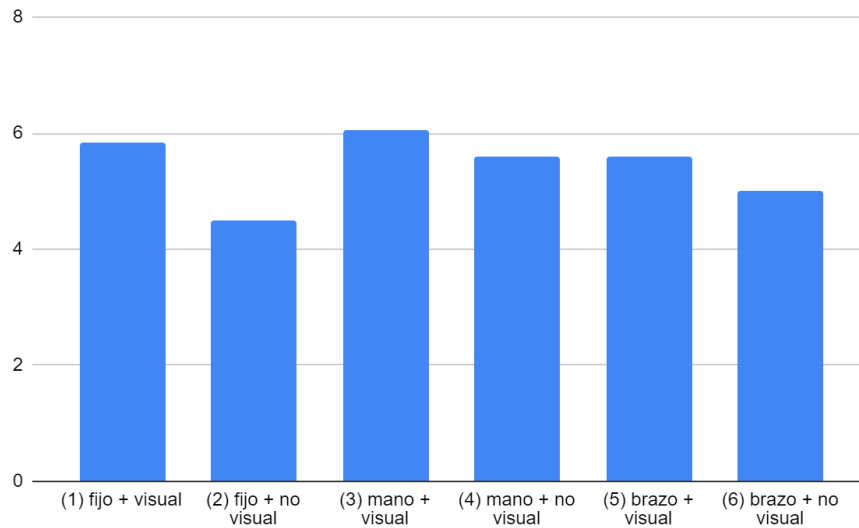


Figura 48. Nivel de inútil-útil valorado por el usuario

- Uso intuitivo

Para el atributo de *uso intuitivo* hemos elegido la variable “Equívoco-Evidente” que consideramos que es el que mejor la representa. Como se aprecia en la figura 49, el más perjudicado ha sido el menú fijo no visual al no tener puntos de referencia con lo que ubicar el menú o al no saber si se ha seleccionado correctamente o no. El menú considerado más evidente ha sido el fijo visual siendo el formato más conocido y utilizado.

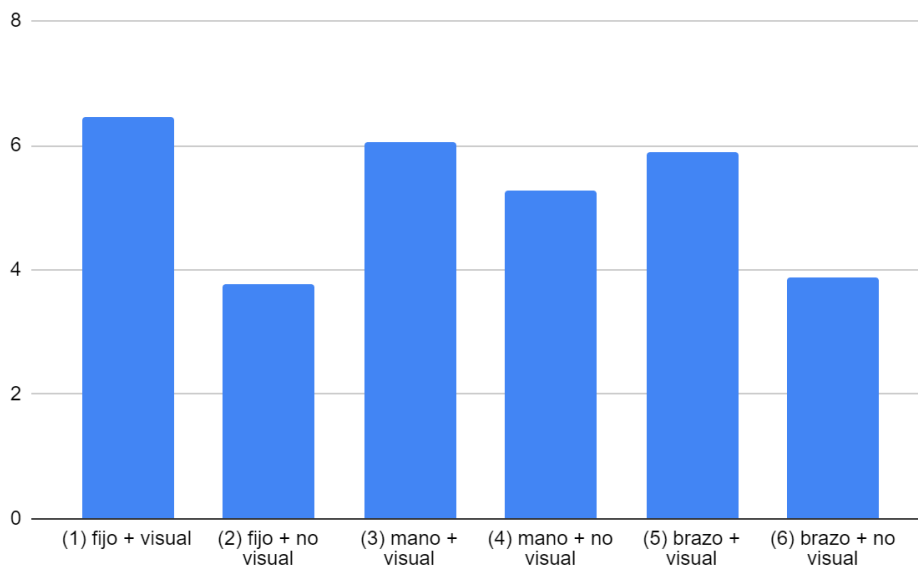


Figura 49. Nivel de equívoco-evidente valorado por el usuario

Todos ellos han tenido un orden similar de ordenación dejando en últimas posiciones el menú fijo y el de brazo no visuales y en primeras el de mano y fijo visuales.

A continuación, analizaré las respuestas dadas a los atributos de esfuerzo y de inmersión que, aunque no pertenezcan a UEQ+, consideramos interesantes de evaluar.

- Esfuerzo

En la figura 50 muestra una gráfica con los menús ordenados de menor a mayor según la tasa de esfuerzo. Se puede observar que los menús fijos son los que más esfuerzo consideran que ha requerido y los del brazo los que menos. Podemos deducir que esto se debe a que, para el menú del brazo, se le ha obligado al participante a mantener el brazo apoyado en una superficie, por lo que este resultado no es muy revelador.

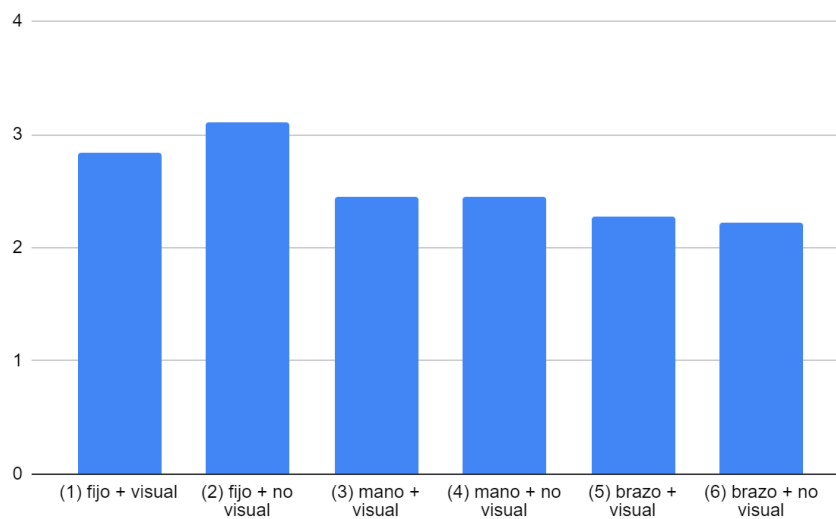


Figura 50. Tasa de esfuerzo valorado por el usuario

- Inmersión

Para esta variable lo que se ha considerado conveniente ha sido ordenar según la diferencia de las respuestas obtenidas en la cuestión “Inmersión dentro del mundo virtual” y la cuestión “Inmersión dentro del mundo virtual durante selección del menú”. De esta manera podemos obtener el grado en el que la interacción con el menú perjudicaba la sensación de inmersión.

Como se puede observar en la gráfica de la figura 51 el menú que menos perjudicaba la sensación de inmersión dentro del mundo virtual es el menú *mano visual*. Según las respuestas dadas por los usuarios esto es porque han considerado que es un menú sencillo, intuitivo, con el que no era necesario tener que pensar mucho.

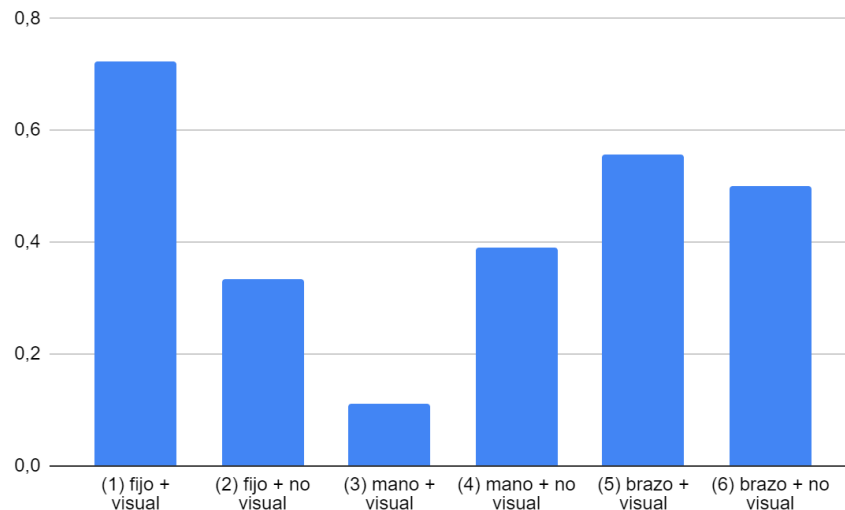


Figura 51. Nivel de inmersión valorado por el usuario

Para finalizar el análisis de resultados se resaltarán algunos de los comentarios realizados por los usuarios con respecto a los menús. En las encuestas finales de las pruebas de usuario se les pidió a los participantes que ordenaran los escenarios según preferencia, siendo el 1 el menú preferido y el 6 el de menor agrado.

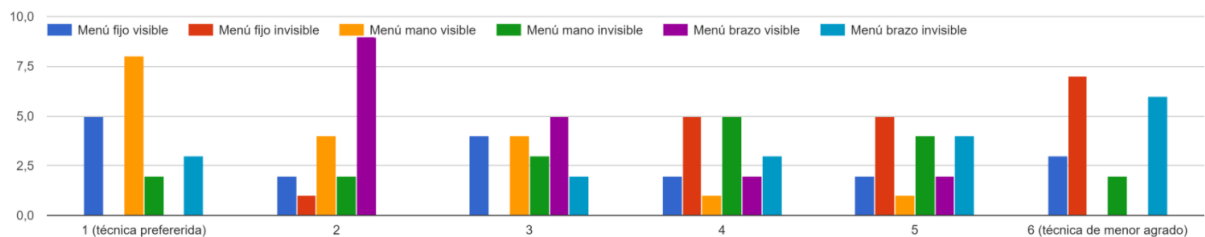


Figura 52. Gráfica con preferencias de los menús

En la figura 52 puede verse el recuento de valoraciones. Se aprecia que el menú preferido por más personas es el “Menú mano visible” y el de menor agrado para más participantes es el “Menú fijo invisible”.

Los usuarios que eligieron “Menú mano visible” como primera opción destacaron que:

- Resulta muy útil tener *feedback* háptico
- Es ventajosa la selección sobre superficies discretas
- El menú es sencillo de utilizar y cómodo
- Se tiene una ayuda propioceptiva a la hora de seleccionar
- Es innovador
- Respeta la inmersión en el mundo virtual

Se les pidió a los usuarios que dejaran comentarios destacando qué es lo que más les había gustado de su menú preferido. Algunos de los comentarios con respecto al “Menú manos visible” fueron los siguientes:

- *“Que es muy sencillo de utilizar y que la interfaz visible no molesta ni dificulta la inmersión.”*
- *“Es lo suficientemente inmersivo para no distraerse del entorno virtual sin ser demasiado complejo o poco intuitivo. Además, el feedback del tacto es muy ventajoso.”*
- *“Es una técnica sencilla de entender pero con muy buena precisión.”*

Los usuarios que eligieron “Menú fijo invisible” como última opción destacaron que:

- Resulta muy poco natural
- Es complicado interactuar con él al no tener referencia para ubicarlo
- Requiere demasiada atención y puede llegar a causar fatiga

También se solicitó que resaltaran qué es lo que menos les había gustado del menú de menos agrado. Estos fueron algunos de los comentarios realizados sobre el “Menú fijo invisible”:

- *“Exige demasiada atención para acertar.”*
- *“Que no sé dónde están las opciones del menú y al no ser como con el antebrazo o la mano que sé dónde las tengo, no sé a dónde tengo que apuntar.”*
- *“Resulta poco natural pulsar un menú que no ves, por no saber dónde se encuentra cada casilla ni si has pulsado la correcta.”*

5. CONCLUSIONES

Tras la realización de las pruebas de usuario, su consiguiente análisis de resultados y revisión de las encuestas han podido obtenerse una serie de conclusiones.

Conclusiones por hipótesis

- Conclusión con respecto a H1 (posición del menú): No hay una diferencia significativa ni en la precisión ni en el tiempo de selección entre el menú de la mano y el fijo. Esto prueba la hipótesis H1, es decir, un menú corporal situado en puntos discretos obtendrán resultados similares que el menú fijo si se diseña teniendo en cuenta el sentido de la propiocepción.
- Conclusión con respecto a H2 (opción continua o discreta): Existe una diferencia significativa en la precisión entre los menús de la mano y los del brazo de forma que aquellas con una superficie continua (brazo) tienen una tasa de error mayor. Por lo tanto, tal y como se intuía, es importante a la hora de diseñar un menú corporal tener en cuenta que el sentido de la propiocepción es menos exacto a la hora de ubicar puntos situados en una zona continua.
- Conclusión con respecto a H3 (opción visual): Con respecto al tiempo de selección no hay diferencia entre los menús de mano y fijos visuales y no visuales. En cuanto a la precisión sí existe una tendencia a mejorar la tasa de acierto cuando se visualiza, pero esta diferencia no es significativa, por tanto, se podría aprobar la H3, es decir, se puede eliminar la visibilidad de los menús sin empeorar significativamente los resultados obtenidos.

Como conclusión final se determina que, gracias a la propiocepción, un menú situado en la mano es igual de eficiente que el menú fijo. Además, no se aprecian tampoco diferencias significativas en la mano a la hora de visualizar o no las opciones. Por lo tanto, se ha concluido que situar las opciones de un posible menú en la mano sin ser visualizadas puede presentar ventajas frente a un menú fijo, ya que no se detecta mejoría ni en precisión ni en tiempo y ofrece la ventaja de evitar la oclusión del escenario.

Conclusiones por opiniones usuarios

El menú preferido por más gente ha sido el menú de mano visible y el de menos agrado para la mayoría ha sido el menú fijo no visible, como muestra la figura 52.

Por lo general a los participantes de las pruebas de usuario les ha parecido interesante el uso de la propiocepción para el desarrollo de los menús. Les ha gustado también que

las opciones estuvieran sobre el cuerpo, sobre todo en el caso de la mano ya que ofrecía *feedback* háptico y al ser una parte discreta se sabía si se seleccionaba bien o no

Por tanto, podemos concluir que la preferencia de los usuarios cambia con respecto al resultado obtenido en [5] si en el diseño se incluye el sentido de la propiocepción. Además, parece que, aunque objetivamente la diferencia de los resultados obtenidos entre opciones visuales y no visuales no es significativa, los usuarios se sienten más cómodos si visualizan el menú.

6. LÍNEAS FUTURAS DE TRABAJO

Una vez concluido el desarrollo y también gracias a sugerencias de los participantes de las pruebas de usuario surgen nuevas líneas de trabajo futuro. Algunas de ellas son las siguientes:

- Diseñar una interfaz con las opciones distribuidas por todo el cuerpo (brazos, torso, etc) y de forma que estas sean discretas.
- Explorar diferentes tecnologías como inverse kinematics, parches corporales, controladores, visión, etc.. con el fin de situar de forma exacta las opciones de menú en otras partes discretas del cuerpo. Conseguir una correcta reproducción del brazo para poder sacar más partido a interfaces que lo utilicen.
- Explorar el uso de partes continuas como submenús que requieran la selección entre un rango de opciones (por ejemplo, en una aplicación de pintura, selección de color en una paleta de colores, escoger grosor de pincel...).
- Estudiar la posibilidad de combinar menús con gestos corporales que también hacen uso de la propiocepción.
- Diseñar un menú para una aplicación real (videojuego, visita virtual...) para explorar otros aspectos como:
 - la transición entre navegar por el mundo y acceder al menú
 - número de opciones que el usuario sería capaz de recordar más allá de las cuatro de la mano
 - diferentes formas de selección del menú, por ejemplo, en el menú manual seleccionar con el pulgar de la misma mano para no tener que involucrar ambas manos.

7. BIBLIOGRAFÍA

- [1] J.L. Taylor, "Proprioception", *Encyclopedia of Neuroscience*, 1143-1149, 2009. <https://doi.org/10.1016/B978-008045046-9.01907-0>
- [2] J.M. Zheng, K.W. Chan and I. Gibson, "Virtual reality", *Doi.org*, 1998. [En línea]. Disponible en: <https://doi.org/10.1109/45.666641>.
- [3] L. Carpenter, "VR menu patterns and use cases", *Medium*, 2018. [En línea]. Disponible en: <https://bit.ly/3jreLMk>.
- [4] K. Wolf, C. Müller-Tomfelde, K. Cheng and I. Wechsung, "Does proprioception guide back-of-device pointing as well as vision? | CHI '12 Extended Abstracts on Human Factors in Computing Systems", *Doi.org*, 2012. [En línea]. Disponible en: <https://doi.org/10.1145/2212776.2223702>.
- [5] Y. Wang, Y. Hu y Y. Chen, "An experimental investigation of menú selection for immersive virtual environments: fixed versus handheld menus". *Virtual Reality* 25, 409–419, 2021. <https://doi.org/10.1007/s10055-020-00464-4>
- [6] "Fantasy forest environment - Free demo: 3D Fantasy: Unity asset store," *3D Fantasy | Unity Asset Store*, 23-Jul-2019. [En línea]. Disponible en: <https://assetstore.unity.com/packages/3d/environments/fantasy/fantasy-forest-environment-free-demo-35361>.
- [7] "Oculus integration: INTEGRATION: Unity asset store," *Integration | Unity Asset Store*, 30-Aug-2021. [En línea]. Disponible en: <https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022>.
- [8] "Cinematika inversa - Wikipedia, la enciclopedia libre", *Es.wikipedia.org*. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Cinem%C3%A1tica_inversa.
- [9] M. Parger, "GitHub - dabeschte/VRArmIK: Unity Inverse Kinematics solution for arms in VR", *GitHub*, 2018. [En línea]. Disponible en: <https://github.com/dabeschte/VRArmIK>.
- [10] M. Parger, "VRArmIK - using inverse kinematics for animating arms in VR," *YouTube*, 05-Jun-2018. [En línea]. Disponible en: <https://www.youtube.com/watch?v=dHgH6Xsi3JQ>.

- [11] "Free medieval room: 3D ENVIRONMENTS: Unity asset store," *3D Environments / Unity Asset Store*, 05-Dec-2018. [En línea]. Disponible en: <https://assetstore.unity.com/packages/3d/environments/free-medieval-room-131004>.
- [12] M. Schrepp and J. Thomaschewski, "UEQ+", *Ueqplus.ueq-research.org*, 2017. [En línea]. Disponible en: <http://ueqplus.ueq-research.org/>.
- [13] S. Glen, "Latin Square Design: Definition and Balanced Latin Square Algorithm", *Statistics How To*, 2017. [En línea]. Disponible en: <https://www.statisticshowto.com/latin-square-design/>.

8. ANEXOS

Sección 1 de 6

Formulario Menu

Descripción del formulario

Id participante *

Texto de respuesta corta

Tipo menu *

1. (1) fijo + visual
2. (2) fijo + no visual
3. (3) mano + visual
4. (4) mano + no visual
5. (5) brazo + visual
6. (6) brazo + no visual

EFICIENCIA: Para lograr mis objetivos, el menú me parece:



Descripción (opcional)

Lento/Rápido *

	1	2	3	4	5	6	7	
lento	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	rápido

Ineficiente/Eficiente *

	1	2	3	4	5	6	7	
ineficiente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	eficiente

Impráctico/Práctico *

	1	2	3	4	5	6	7	
nada práctico	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	muy práctico

Recargado/Sencillo *

	1	2	3	4	5	6	7	
recargado	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sencillo

MANEJO INTUITIVO: El manejo del menú me parece:

Descripción (opcional)

Arduo/Fácil *

	1	2	3	4	5	6	7	
arduo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	fácil

Ilógico/Lógico *

	1	2	3	4	5	6	7	
ilógico	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	lógico

Equivoco/Evidente *

	1	2	3	4	5	6	7	
equivoco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	evidente

Incoherente/Coherente *

	1	2	3	4	5	6	7	
incoherente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	coherente

UTILIDAD: En cuanto a la posibilidad de utilizarlo, el menú me parece:



Descripción (opcional)

Inútil/Útil *

	1	2	3	4	5	6	7	
Inútil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Útil

Nada provechoso/Provechoso *

	1	2	3	4	5	6	7	
Nada provechoso	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Provechoso

Nada beneficioso/Beneficioso

	1	2	3	4	5	6	7	
Nada beneficioso	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Beneficioso

Nada rentable/Rentable *

	1	2	3	4	5	6	7	
Nada rentable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Rentable

Inmersión



Descripción (opcional)

Me he sentido inmerso en el mundo virtual. *

1 2 3 4 5 6 7

Totalmente en desacuerdo Totalmente de acuerdo

Mientras realizaba la selección en el menú me he sentido inmerso en el mundo virtual. *

1 2 3 4 5 6 7

Totalmente en desacuerdo Totalmente de acuerdo

Después de la sección 5 Ir a la siguiente sección



EXIGENCIA FÍSICA



Descripción (opcional)

¿Cuánta actividad física fue necesaria? *

1 2 3 4 5 6 7

Baja Alta

Preguntas Finales

Descripción del formulario

Id del participante

*

Texto de respuesta corta

En el contexto de una tarea de selección de elementos de un menú como la que has hecho durante este estudio, ¿Cuál de los 6 menús prefieres? Ordénalos del 1 al 6 (la 1 es la de mayor preferencia) *

Menú fijo vis... Menú fijo inv... Menú mano ... Menú mano i... Menú brazo ... Menú brazo i...

1 (técnica pr...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6 (técnica de...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

¿Qué es lo que más valoras de tu primera opción?

Texto de respuesta larga

¿Qué es lo que menos valoras de tu última opción?

Texto de respuesta larga

¿Crees que el hecho de no ver las opciones en los menús no visuales mejora la sensación de inmersión? ¿Por qué?

Texto de respuesta larga

¿Crees que merece la pena no ver el menú para mantener la visibilidad total del entorno virtual?

Texto de respuesta larga

Otras observaciones

Texto de respuesta larga

CONSENTIMIENTO INFORMADO PARA PARTICIPACIÓN EN ESTUDIO DE EVALUACIÓN DE TECNOLOGÍAS DE INTERACCIÓN PERSONA-ORDENADOR

1. ¿Qué es y qué persigue el estudio?

El estudio forma parte de un proyecto de investigación que tiene como objetivo evaluar tecnologías de interacción persona ordenador. Te pediremos que pruebes una tecnología de interacción persona ordenador con diferentes modalidades de interacción durante aprox. 30-60 minutos. Este experimento no entraña peligro alguno y ayudará a crear mejores sistemas de interacción persona-ordenador.

2. ¿Quién me va a realizar la prueba?

Investigadores del Departamento de Estadística, Informática y Matemáticas de la Universidad Pública de Navarra (UPNA).

3. Información que recogeremos

Observaremos y grabaremos en video como interactúas con tecnologías de interacción persona ordenador, mediremos como realizas esas pruebas y guardaremos esos datos, y te pediremos que rellenes un cuestionario con datos personales (sexo, edad, ocupación) y otro cuestionario sobre las pruebas que realizarás y experiencias previas con tecnologías de interacción persona-ordenador.

4. Contacto

Para solicitar más información sobre el proyecto y/o tratamiento de sus datos puede contactar con el equipo de investigación a través de oscar.ardaiz@unavarra.es

DECLARACIÓN DEL PARTICIPANTE

Se me ha entregado una copia de este consentimiento informado, fechado y firmado. Se me han explicado las características y los objetivos del estudio y los posibles beneficios y riesgos que puedo esperar. Se me ha dado tiempo y oportunidad para realizar preguntas. Todas las preguntas fueron respondidas a mi entera satisfacción. Puedo ponerme en contacto con el equipo investigador en cualquier momento para realizarle preguntas sobre el mismo.

Autorizo la medición de datos, la grabación de audio y video de mis pruebas para el estudio. Y el uso de breves extractos en la presentación de los resultados del estudio y en material docente.

Soy libre de retirarme del estudio en cualquier momento.

Por tanto, YO _____, con DNI _____, tras comprender toda la información que se me ha proporcionado, consiento participar voluntariamente en este estudio.

Fecha _____

Firma del participante _____

Firma del investigador _____

CONSENTIMIENTO INFORMADO PARA EL TRATAMIENTO DE DATOS

1. Existencia de tratamiento de datos de carácter personal

La manifestación de consentimiento a participar en el proyecto descrito implica la tramitación de los siguientes datos: nombre, apellidos, firma, consentimiento en participar y participación en el proyecto en cuestión.

Asimismo, y en la medida que resulten vinculables a cada uno los participantes en el proyecto, también se tratarán los siguientes datos: edad, sexo, ocupación, experiencias previas con tecnología interacción persona-ordenador, mediciones de las tareas, cuestionarios sobre las tareas y grabaciones en audio y video de la realización de las tareas.

2. Normativa aplicable al tratamiento de datos previsto

Reglamento General de Protección de Datos (RGPD) y 2 de la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos ([LOPDGDD](#))

3. Legitimación para el referido el tratamiento

Conforme a los artículos 6 y 89 del RGPD y 6 de la [LOPDGDD](#), sus datos serán tratados de acuerdo con su propio consentimiento informado y para fines de investigación científica.

4. Garantía adecuadas

El equipo investigador precisa, por un lado, conservar los datos de su consentimiento de participación en el proyecto y, por otro, las variables resultantes de dicha información. No obstante, dicho equipo se compromete a conservar ambas referencias de forma disociada. De este modo, las variables obtenidas en el proyecto no podrán vincularse directa e inmediatamente a las identidades de cada uno de los participantes en el proyecto. En todo caso, se trata de una disociación de valor relativo ya que sólo 12 personas participan en el proyecto.

5. Derechos de información, acceso y otros en materia de protección de datos

El Responsable del Tratamiento de sus datos es la Universidad Pública de Navarra (UPNA), ostentando el carácter de personas autorizadas a tratar datos por cuenta de la UPNA los miembros del equipo investigador que coordina el presente proyecto.

Para conocer los detalles del tratamiento de sus datos puede contactar a través de la siguiente dirección electrónica oscar.ardaiz@unavarra.es

Para ejercitar formalmente los derechos de acceso, información, rectificación, supresión y otros en materia de protección de datos, contacte con delegado.protecciondatos@unavarra.es