
Inspección de modelos anatómicos en un entorno de Realidad Virtual inmersivo colaborativo

Universitat Politècnica de Catalunya
Mención en computación

Director

Pere Pau Vázquez Alcocer
pau@cs.upc.edu

Autor

Antonio García Alcón
antonio.garcia.alcon@estudiantat.upc.edu

Tutora

Eva Monclús Lahoya
emonclus@cs.upc.edu



**UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH**

*Dedico este trabajo a Pere Pau y Eva Monclús
por su ayuda y dedicación durante todo el desarrollo del proyecto,
a los trabajadores del ViRVIG por participar en el User Study
y a mi familia y amigos por su apoyo incondicional.*

Resumen

La Realidad Virtual es una tecnología que no para de crecer y que se ha mostrado de gran utilidad en ramas de conocimiento muy diversas. Esto se debe a su capacidad de ofrecer aproximaciones realistas del mundo que nos rodea, permitiéndonos interactuar con el entorno virtual de una forma más natural que desde un monitor. Es principalmente por esta razón que en las recientes décadas ha habido un aumento del uso de herramientas de Realidad Virtual para la educación y concretamente para la educación en ámbitos médicos.

Este proyecto tiene como principal objetivo participar en ese crecimiento, añadiendo nuevas metáforas de interacción a una aplicación colaborativa e inmersiva de Realidad Virtual, desarrollada en el Centro de Realidad Virtual del grupo de investigación ViRVIG, que simula una clase de anatomía donde varios estudiantes y un profesor inspeccionan un modelo anatómico virtual. Sin embargo, pese a que la aplicación está principalmente orientada para un uso académico en anatomía, su uso es extensible para todas aquellas áreas en las que un grupo de personas deban inspeccionar e interactuar de forma conjunta con un modelo tridimensional.

Resum

La realitat virtual és una tecnologia que no para de créixer i que s'ha mostrat molt útil en branques de coneixement molt diverses. Això és degut a la seva capacitat d'ofrir aproximacions realistes del món que ens envolta, permetent interactuar amb l'entorn virtual d'una manera més natural que des d'un monitor. És principalment per aquesta raó que en les dècades recents hi ha hagut un augment de l'ús d'eines de Realitat Virtual per a l'educació i concretament per a l'educació en àmbits mèdics.

Aquest projecte té com a principal objectiu participar en aquest creixement, afegint noves metàfores d'interacció a una aplicació col·laborativa i immersiva de Realitat Virtual, desenvolupada al Centre de Realitat Virtual del grup de recerca ViRVIIG, que simula una classe d'anatomia on diversos estudiants i un professor inspeccionen un model anatómic virtual. Tot i que l'aplicació està principalment orientada per a un ús acadèmic en anatomia, el seu ús és extensible per a totes aquelles àrees en què un grup de persones hagin d'inspeccionar i interactuar de manera conjunta amb un model tridimensional.

Abstract

Virtual reality is a technology that does not stop growing and that has been shown to be very useful in very diverse branches of knowledge. This is due to its ability to offer realistic approximations of the world around us, allowing us to interact with the virtual environment in a more natural way than from a computer. It is for this reason that in recent decades there has been an increase in the use of Virtual Reality tools for education and specifically for education in medical fields.

The main objective of this project is to participate in this growth, adding new interaction metaphors to a collaborative and immersive Virtual Reality application, developed at the Virtual Reality Center of the ViR-VIG research group, which simulates an anatomy class where several students and a teacher inspect a virtual anatomical model. However, despite the fact that the application is mainly oriented for academic use in anatomy, its use is extensible for all those areas in which a group of people must inspect and interact with a three-dimensional model.

Índice general

1.	Introducción	1
1.1.	Contextualización	1
1.2.	Formulación del problema	2
1.3.	Actores implicados	3
2.	Estado del arte	4
2.1.	Sistemas de Realidad virtual	4
2.2.	Realidad virtual en la educación	5
3.	Definición del alcance	6
3.1.	Objetivos y sub-objetivos	6
3.2.	Obstáculos y riesgos	6
4.	Metodología y rigor	8
4.1.	Metodología de trabajo	8
4.2.	Herramientas	8
4.3.	Métodos de validación	8
5.	Desarrollo de las metáforas	9
5.1.	Introducción al proyecto original	9
5.2.	Panel de interacción personal	11
5.2.1.	Componentes principales del PIP	12
5.2.2.	Interacción con el panel	17
5.3.	Photoportal acoplado	18
5.3.1.	Características	18
5.3.2.	Implementación	19
5.4.	Go Inside Model	19
5.4.1.	Características	20
5.4.2.	Diseño de la interfaz	20
5.4.3.	Implementación	21
5.5.	Pseudocódigo de los algoritmos principales	23
6.	Test de usabilidad	29
6.1.	Diseño del experimento	30
6.2.	Datos de los participantes	30
6.3.	Análisis de los resultados	31
6.3.1.	Panel de interacción personal	32
6.3.2.	Photoportal acoplado	33
6.3.3.	Go inside model	34
6.3.4.	Valoración general	34
7.	Conclusiones	37

8. GEP	38
8.1. Descripción de las tareas	38
8.1.1. GEP	38
8.1.2. Desarrollo del proyecto	38
8.1.3. Recursos	39
8.2. Diagrama de Gantt	40
8.3. Gestión del riesgo	42
8.4. Presupuesto	42
8.4.1. Identificación y estimación de los costes	42
8.4.2. Control de gestión	43
8.5. Informe de sostenibilidad	44
8.5.1. Impacto ambiental	44
8.5.2. Impacto económico	45
8.5.3. Impacto social	45
A. Tabla de partidas por tarea	50

Capítulo 1

Introducción

1.1. Contextualización

Las aplicaciones gráficas por computador son herramientas muy útiles en disciplinas como el diseño, la investigación y la experimentación ya que nos permiten construir virtualmente prototipos y llevar a cabo simulaciones que pueden ser difíciles de obtener en la vida real (ya sea por el elevado coste de construcción, la complejidad o la peligrosidad que tienen) [1, 2]. La Realidad Virtual (RV) nace de este tipo de aplicaciones siendo una tecnología que va un paso mas allá, ya que nos permite experimentar una visualización estereoscópica (visualización en 3D) del mundo virtual, como si se tratase de un cine en 3D donde el usuario puede interactuar con el medio virtual de forma activa.

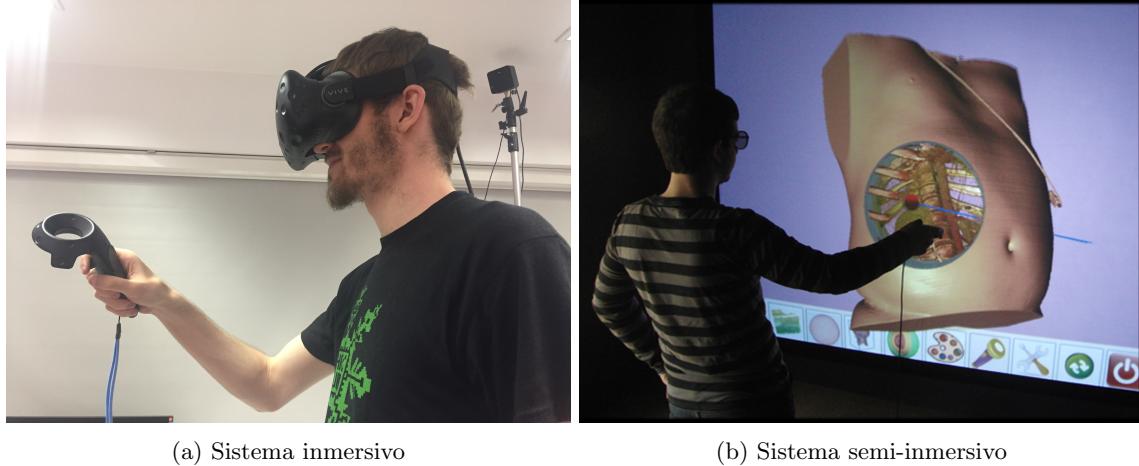
La RV tiene un gran potencial para ayudar al aprendizaje, entrenamiento y entendimiento de estructuras complejas en 3D. A diferencia de las aplicaciones de escritorio (en inglés *desktop*), ofrece una aproximación más realista del mundo que nos rodea y nos permite visualizar e interactuar con el medio virtual de forma natural, como se haría si se estuviera físicamente presente en ese medio. Es por ello que en las recientes décadas ha habido un aumento en el uso de aplicaciones de RV para la educación y concretamente para la educación en ámbitos médicos [3, 4].

Estas aplicaciones pueden ser tanto *inmersivas* (aquellas que trasladan al usuario a un entorno plenamente virtual) como *semi-inmersivas* (aquellas que permiten experimentar entornos virtuales tridimensionales a la vez que no se pierde la noción del mundo real [5]) (ver figura 1.1). Las aplicaciones de RV necesitan de una serie de componentes (audio envolvente, dispositivos de visualización inmersiva, equipos de seguimiento de posición) que interactúen y colaboren sincronizadamente [6, 7]. Como ejemplos de sistemas tenemos: el casco de Realidad virtual (Head-Mounted Display, *HMD*) para sistemas inmersivos y la caverna (*CAVE*) para sistemas semi-inmersivos (la figura 2.2 muestra un ejemplo de sistema CAVE y Powerwall).

Además, estas aplicaciones se denominan *colaborativas* [8, 9] si se tratan de aplicaciones multiusuario donde cada uno de ellos comparte la presencia de los otros dentro de la aplicación. El grado de colaboración dependerá de los objetivos tanto de cada usuario como del objetivo global de la aplicación. Un ejemplo muy conocido podrían ser los videojuegos en línea o el Metaverso [14].

En RV, este tipo de aplicaciones se clasifican en *collocated* si los usuarios comparten el mismo espacio físico o *remotas* si están ubicados en espacios físicos distintos.

Para este proyecto, partimos de una aplicación inmersiva colaborativa [10] donde varios estudiantes y un profesor interactúan con un modelo anatómico tridimensional. Por tanto, el objetivo principal de este proyecto es dotar a la aplicación ya existente de nuevas funcionalidades que amplíen los mecanismos de comunicación entre los usuarios e incorporen nuevas posibilidades de exploración y manipulación del modelo.



(a) Sistema inmersivo

(b) Sistema semi-inmersivo

Figura 1.1: Diferentes tipos de inmersión en RV, fuente: ViRVIG

1.2. Formulación del problema

Tal y como se ha dicho anteriormente, este proyecto se centrará en el desarrollo de diferentes metáforas de interacción (técnicas de interacción) que faciliten la comunicación de los usuarios en un entorno inmersivo colaborativo (ver figura 1.2) dónde observarán un mismo modelo anatómico tridimensional.

El proyecto parte del trabajo ya realizado en un TFG anterior titulado *Inspecció de models anatòmics en un entorn immersiu col·laboratiu: aplicació en anatomia* realizado por David Hernández Morales [10], y de código ya escrito por el grupo de investigación ViRVIG [11] (grupo dentro del cual se desarrollará este proyecto). Al ser una aplicación colaborativa, la interacción entre los usuarios es algo fundamental y, por tanto, en este proyecto nos centraremos en herramientas que faciliten la comunicación e interacción entre ellos así como la interacción que pueden tener con el modelo. Como puede ser el uso de pantallas virtuales adicionales donde se visualiza el punto de vista de otro usuario [9] (también conocido como *Photoportals*) para facilitar a los estudiantes la comprensión de lo que está viendo el profesor o tratar con el tamaño de visualización de las diferentes partes del modelo para enfatizarlas y/o facilitar su observación.



Figura 1.2: Ejemplo de aplicación inmersiva colaborativa, fuente: ViRVIG correspondiente a [10]

La inclusión y/o adaptación de nuevas metáforas implica la posibilidad de añadir nuevos elementos de interacción y visualización que deben ser útiles, manejables y fáciles de entender. Por ello, se llevará a cabo un estudio de usuario donde se decidirá qué metáforas y qué versiones son las más útiles y cómodas para el usuario.

1.3. Actores implicados

La aplicación está principalmente orientada para el uso académico en clases de anatomía, aunque su uso es extensible a todas aquellas áreas en las que un grupo de personas deban inspeccionar e interactuar de forma conjunta con un modelo tridimensional. Los métodos tradicionales para la enseñanza anatómica implican el uso de libros de texto, atlas y disección de cadáveres. No obstante, el realismo de estas experiencias es bastante limitado. Por ejemplo, el color y la textura de las estructuras anatómicas en cadáveres difiere del de los pacientes vivos. Además, el uso de cadáveres conlleva un alto coste y el uso que se le puede dar es muy corto. Gracias a esta aplicación, los estudiantes podrán inspeccionar modelos virtuales del cuerpo humano, donde podrán moverse libremente alrededor del modelo e inspeccionar las partes mas internas de éste. También, podrán posicionar etiquetas en las diferentes partes del modelo que les servirán tanto para estudio como para poner a prueba sus conocimientos. Todo de una forma mas barata, rápida y sostenible.

Esto quiere decir, que los beneficios directos serán para los estudiantes, que podrán aprender de una forma más realista y eficiente. Dado que los estudiantes tendrán una mejor formación, habrán médicos, enfermeros y fisioterapeutas mejor cualificados lo que conllevará a un beneficio social de forma indirecta. Además, al tratarse de un proyecto muy centrado en la investigación, futuras herramientas que utilicen este tipo de tecnología podrán verse beneficiadas independientemente de que estén enfocadas en el ámbito de la medicina o no, ya que las metáforas y funcionalidades que emplea se pueden extrapolar a cualquier otro contexto como podría ser el de un grupo de ingenieros visualizando un nuevo modelo de coche que están diseñando o reuniones telemáticas de una forma más natural.

Capítulo 2

Estado del arte

2.1. Sistemas de Realidad virtual

Habiéndose empezado a desarrollar como una herramienta de entrenamiento y simulación para uso militar, la Realidad virtual ha experimentado un gran avance en los últimos años [12, 13] gracias sobre todo al éxito comercial de proyectos relacionados con los videojuegos [14, 15]. Estos proyectos han provocado una disminución en el coste de los dispositivos de RV como los head-mounted displays (HMDs), haciendo que el público general tenga un acceso relativamente económico a ellos.

Para este proyecto utilizaremos el dispositivo HTC Vive [16]. Como se puede observar en la figura 2.1, este dispositivo cuenta con una pantalla (display) que, como su nombre indica (HMD) va montado en la cabeza del usuario y le permite ver el entorno virtual en RV. También cuenta con uno o más mandos que son la principal fuente de entrada de información. Estos dispositivos le permiten al usuario interactuar con el entorno en RV mediante el movimiento de su cabeza y el uso de los joysticks y botones incorporados en los mandos.



Figura 2.1: HMD: HTC Vive, fuente: Google imágenes

Existen muchos otros tipos de dispositivos de RV que no se centran tanto en un feedback visual. Algunos de estos dispositivos son los tracksuits de bHaptics [17] o los haptic gloves de haptx [18]. Estos dispositivos se centran en un feedback háptico, es decir, en la respuesta que se genera al entrar en contacto con alguna superficie.

Recientemente, ha aparecido el concepto de Metaverse [19], que consiste en un universo 3D virtual que conecta a los usuarios en todos los aspectos de su vida. Este concepto se apoya fuertemente en la RV ya que mediante esta tecnología será como los usuarios podrán habitar estos espacios virtuales. Este concepto lo que busca es que las personas no solo naveguen por mera diversión sino para llevar a cabo actividades

productivas. Sin embargo, el Metaverse es todavía un concepto poco definido y hay quienes se han aventurado a decir que no es más que un juego de palabras para describir las mejoras tecnológicas venideras.

2.2. Realidad virtual en la educación

Muchos estudios indican que los sistemas interactivos son para los estudiantes un medio valioso para su aprendizaje y además hacen un progreso sustancial en el entendimiento de relaciones espaciales [24, 25].

Los sistemas de RV proveen un entorno interactivo que refuerzan la sensación de inmersión dando lugar a experiencias más realistas y dinámicas. El alto grado de detalle que se puede conseguir de estructuras tridimensionales complejas hace que sea mucho más fácil la comprensión, memorización y aprendizaje de las mismas. Además, permite la interacción con los modelos tridimensionales de una forma natural, como si se estuviera presente con ellos físicamente. Es por esto que la RV es una gran herramienta para la formación en anatomía motivo por el cual se han creado varios tipos de sistemas de RV para su uso en este campo [28]. Un ejemplo de ellos podría ser *3D Puzzle in VR* [26], un sistema donde el estudiante puede escoger entre varias estructuras anatómicas y escalarlas al tamaño que le parezca conveniente.

Existe un conjunto de sistemas llamados *Cooperative VR systems*, en el que se podría encapsular este proyecto, donde varios usuarios inspeccionan un entorno virtual (pudiendo ser tanto de forma collocated como remota). Actualmente existen muchas aplicaciones basadas en este sistema [8, 29] dada su efectividad y el abaratamiento de las herramientas necesarias para llevarlas a cabo. Como ejemplo, Fairén [27] introduce un sistema de RV para la educación en anatomía que funciona a través de *Power walls* y en un sistema CAVE. (ver Figura 2.2)



(a) Sistema Powerwall

(b) Sistema CAVE

Figura 2.2: Ejemplo de sistemas semi-inmersivos, fuente: ViRVIG

Capítulo 3

Definición del alcance

3.1. Objetivos y sub-objetivos

El objetivo principal de este proyecto es desarrollar nuevas metáforas de interacción que permitan una mejor comunicación en un entorno de RV inmersivo colaborativo. Para ello, el proyecto se llevará a cabo dentro del grupo de investigación ViRVIG el cual se dedica a la investigación y formación en la visualización por computador, realidad virtual e interacción de gráficos. Este proyecto parte del trabajo realizado tanto por los investigadores del ViRVIG como de un TFG anterior donde ya se implementaron metáforas de interacción y funcionalidades básicas de comunicación entre los usuarios del sistema, introduciendo sistemas para evitar colisiones entre ellos y procedimientos para compartir información colocando etiquetas de texto sobre el modelo.

Dado que partimos de un trabajo previo, se incluirán nuevas metáforas y se adaptarán las partes pertinentes, sin modificar todas aquellas que no se consideren oportunas. Este proyecto consiste en un entorno de RV creado con Unity3D [20] pensado para su uso con el casco de RV HTC Vive aunque también dispone de una versión desktop tanto para su desarrollo como para su uso.

El dispositivo HTC Vive es un entorno totalmente desconocido para la mayoría de usuarios finales (profesores de anatomía y estudiantes), por tanto, como sub-objetivo se querrán desarrollar e investigar sistemas de interacción que sean naturales y cómodos para ellos además de útiles tanto para esta aplicación como para otras que utilicen una tecnología similar.

3.2. Obstáculos y riesgos

Los obstáculos más frecuentes vendrán a partir de las metáforas de interacción, es decir, los problemas que pueden tener los usuarios al visualizarlas, al comprender su correcto funcionamiento o al orientarse en el entorno virtual, pero para reducirlos lo máximo posible, se sopesarán distintas alternativas para determinar cuál es objetivamente la mejor.

Además, existe un pequeño porcentaje de personas sensibles a los dispositivos de RV y, por tanto, hay que ir con mucho cuidado para no corra el riesgo de que el usuario se maree al usarlos. También pueden haber problemas con las herramientas de RV, como la sensibilidad de los controles, fallos de precisión o errores de visualización. No obstante, estos problemas no dependen en gran medida de los usuarios y por tanto los podemos detectar en tiempo de desarrollo.

También, yo como desarrollador parto de conocimientos prácticamente nulos en el ámbito de la RV, por lo que deberé aprender a utilizar esta tecnología durante el desarrollo del trabajo, lo que hará que avance más lentamente. Sin embargo,uento con código ya hecho que puede servirme de referencia además de la ayuda de Eva Monclús, tutora de este proyecto y miembro del equipo de investigación ViRVIG, así como del director del TFG, Pere-Pau Vázquez.

En lo referente al código, como toda aplicación informática surgirán errores de código que habrá que solucionar y pueden ralentizar el desarrollo de la aplicación si no se detectan a tiempo.

Capítulo 4

Metodología y rigor

4.1. Metodología de trabajo

La metodología de trabajo se basará en una metodología ágil de ciclos cortos. Esto quiere decir que cada semana se planificarán reuniones con la tutora Eva Monclús para hacer un seguimiento del trabajo, establecer los objetivos a cumplir para los próximos días y resolver dudas . Además, el desarrollo del proyecto se llevará a cabo en el Centro de Realidad Virtual (CRV) de la UPC donde se encuentra el grupo de investigación ViRVIG [11] y el laboratorio de RV con los dispositivos a usar en el marco del proyecto, esto facilitará la comunicación con los profesores haciendo que esta sea directa y constante durante todo el proyecto y evitará pérdidas de tiempo en caso de avanzar por una vía errónea en el desarrollo.

4.2. Herramientas

- Slack: Canal de comunicación con los profesores donde se llevará a cabo la monitorización del desarrollo del proyecto [21]
- GitLab: Repositorio de control de versiones de todo el proyecto [22]
- Overleaf: generación de documentos online con LaTeX
- LaTeX: Generación de la memoria y los entregables
- Unity3D: Motor de desarrollo de la aplicación
- Visual Studio: Entorno de desarrollo integrado

4.3. Métodos de validación

La validación del proyecto se llevará a cabo durante las reuniones semanales donde se planificarán las tareas a cumplir y se validarán las hechas anteriormente.

Además, para validar el correcto funcionamiento del desarrollo, se realizarán sesiones de testing con los miembros del CRV, como estudiantes y becarios que trabajan en el laboratorio, donde se evaluarán diferentes propuestas y funcionalidades para alcanzar, de forma objetiva, la mejor versión para los usuarios.

Capítulo 5

Desarrollo de las metáforas

5.1. Introducción al proyecto original

Como ya se ha mencionado anteriormente (ver capítulo 3), este proyecto parte del resultado de uno anterior [10]. Por ello, es conveniente proporcionar una visión general del funcionamiento de la aplicación antes de los cambios realizados durante el mismo.

Se parte de una aplicación colaborativa inmersiva (ver sección 1.1), desarrollada en Unity3D, donde varios usuarios interactúan a la vez con un modelo anatómico tridimensional. Para ello se utiliza la arquitectura cliente-servidor UNet [42] de Unity. Un sistema de comunicación en red donde, en este caso, el servidor es también uno de los clientes (a este cliente se le llama *host*). Este sistema es el que permite transmitir, entre otras cosas, la posición y rotación de los objetos y usuarios en el espacio virtual, de manera que los usuarios pueden ver la posición de los demás y del modelo en todo momento.

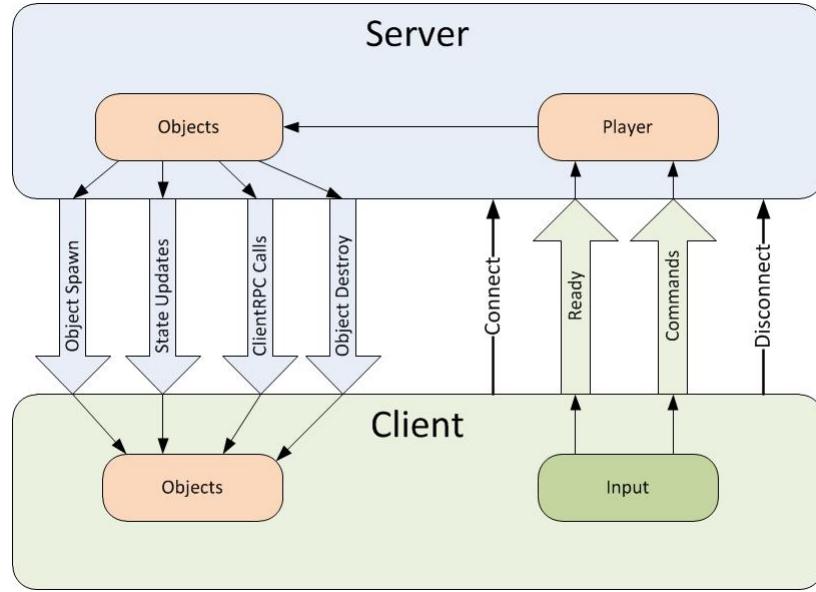


Figura 5.1: Esquema cliente-servidor de Unet. Fuente: Unity Documentation

En la aplicación existen 2 roles diferenciados para los usuarios: los alumnos, los cuales pueden interactuar con otros usuarios de distintas maneras, y el profesor, quien, aparte de ser el *host* de la sesión, tiene las mismas funciones que los alumnos y además puede cambiar el modelo a inspeccionar, añadirle etiquetas o cambiar la dinámica de la clase, entre otras funciones.

Los usuarios interactúan entre ellos a través del *Player Menu* (ver figura 5.2), un menú que esta situado sobre el avatar de cada usuario. Para visualizar dicho menú es necesario tener al usuario sobre el que se quiere interactuar en el campo de visión y seleccionarlo apretando el *Trigger* del mando (ver figura 5.3) apuntando hacia él. Esto es un factor limitante ya que es imposible interactuar con aquellos usuarios que se encuentren ocluidos por otros o por el modelo anatómico.

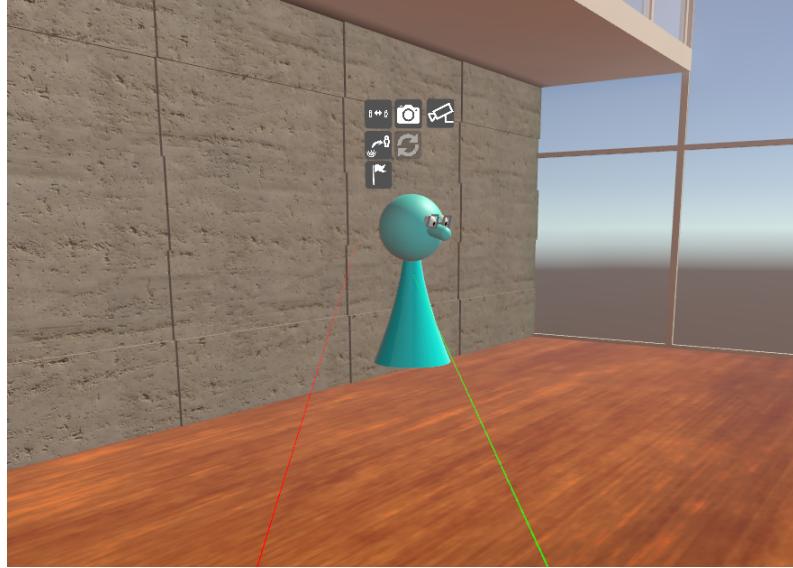


Figura 5.2: Visualización del Player Menu

El *Player Menu* está formado por distintos botones que permiten rotar el modelo para verlo desde el punto de vista del usuario propietario del menú (usuario objetivo), transportarse a su posición (esto es, situar en la aplicación al usuario local en la posición del usuario marcado) y/o abrir uno o varios *photoportals* [9] que muestre lo que esta viendo el usuario objetivo tanto de forma estática (como si fuera una imagen) como de forma dinámica (como si se estuviera viendo a través de una cámara de vídeo). Dichos *photoportals* aparecen en la dirección a la que está apuntando el mando del usuario y éste puede cambiar su tamaño (apuntando con el mando a los bordes del panel y arrastrando con el *Trigger* presionando) y su posición (apuntando con el mando al centro del panel y arrastrando con el *Trigger* presionado). Sin embargo, dado que los *photoportals* no siguen al usuario, es posible que el usuario olvide los *photoportals* que ha instanciado por el entorno si se mueve mucho por él y, por tanto, o bien tiene que regresar a por ellos, o bien tiene que generar otros nuevos.

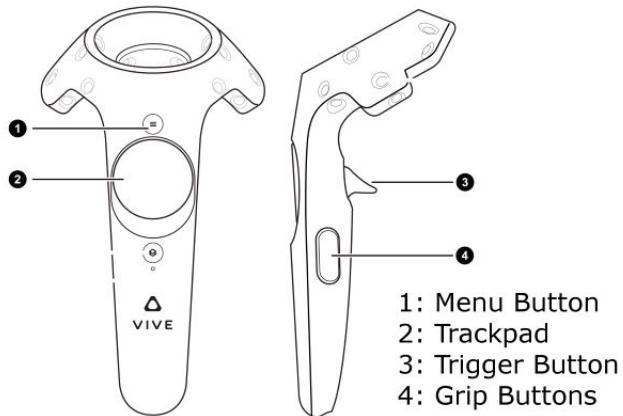


Figura 5.3: Esquema del mando HTC, fuente: Varwin's Documentation

La aplicación también permite a los usuarios moverse libremente por el entorno virtual. Esto permite la inspección del modelo desde cualquier punto accesible por parte del usuario, permitiendo observar el interior de los distintos modelos. Sin embargo, como no es posible modificar la escala del modelo esto limita la inspección de los modelos para aquellas partes y/o cavidades que no sean fácilmente accesibles o que, debido al tamaño, no pueden apreciarse con claridad.

Tal y como se presentó en el apartado 3, el objetivo principal de este proyecto es superar estas tres limitaciones básicas desarrollando nuevas funcionalidades que permitan la interacción fácil e intuitiva por parte de un usuario con el resto de los usuarios e incorpore un nuevo mecanismo que permita modificar la escala del modelo para una mejor inspección. A continuación, dedico un apartado para cada metáfora desarrollada durante el proyecto.

5.2. Panel de interacción personal

El Panel de interacción personal (PIP) es una interfaz activa, situada en el brazo izquierdo del usuario, que le aporta información y le permite interactuar con su entorno. Cada usuario tiene el suyo propio y no es visible por el resto de usuarios.

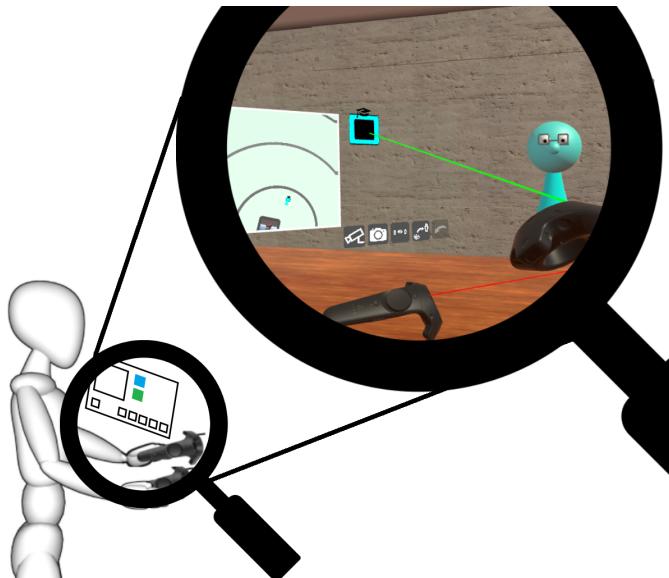
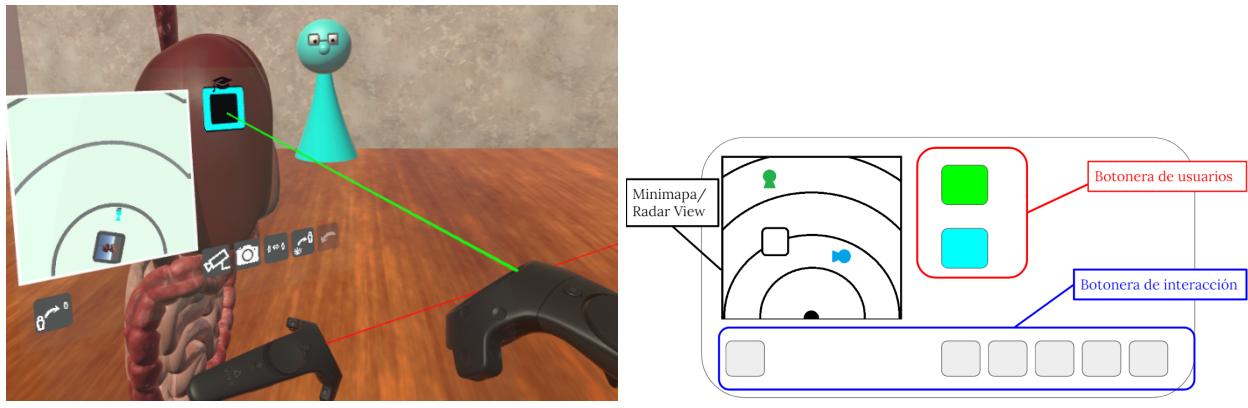


Figura 5.4: Representación esquemática de la interacción con el panel

El propósito de este panel es mejorar la experiencia del usuario dotándole de un menú de interacción acoplado a su brazo que le permite navegar con mayor comodidad por la escena (aportándole información de la misma) e interactuar con su entorno independientemente de su posición u orientación y la posición u orientación del resto de usuarios. Puesto que el panel cubre todas las funcionalidades del Player Menu (además de incluir otras nuevas), no tiene sentido seguir utilizándolo. Por tanto, el PIP sustituiría al Player Menu.

El panel dispone de 3 zonas bien definidas (ver figura 5.5):

- La zona del mapa, donde se representan las posiciones y orientaciones del resto de usuarios y del modelo.
- La botonera de usuarios que permite seleccionar al usuario con el que se quiere interactuar.
- La botonera de interacción que permite elegir qué interacción, de las explicadas en el apartado anterior, se quiere efectuar sobre el usuario escogido o si se quiere entrar en el modo *Go Inside Model* (ver apartado 5.4)



(a) Visualización del PIP en el entorno de RV

(b) Visualización esquemática del PIP

Figura 5.5: Visualización del PIP

5.2.1. Componentes principales del PIP

Para poder utilizar correctamente la aplicación, es necesario que el usuario utilice 2 mandos, uno para cada mano. El panel es un objeto que está situado en el brazo izquierdo del usuario, concretamente se sitúa encima del dorso del brazo con una pequeña inclinación para facilitar su inspección. De esta manera el usuario puede consultar el panel de forma cómoda con un movimiento de brazo similar al de mirar la hora. La posición del panel debe recalcularse en cada *frame*, ya que es una interfaz activa. Por ello se recalcula con cada llamada a la función *Update* (ver figura 5.6) del panel usando la función que se muestra en el fragmento de pseudocódigo de la figura 5.18. Puesto que la aplicación se puede ejecutar en modo simulación, es decir sin utilizar el casco de realidad virtual, es necesario comprobar cuando este está activo y cuando no. Ya que en este modo se necesita poder bloquear el panel en una posición determinada para poder interactuar con él (condición `!BlockedPanel`).

Botoneras

Como se ha comentado con anterioridad, el panel dispone de dos grupos de botones. Los botones de la parte superior representan a los usuarios conectados a la aplicación mientras que los inferiores sirven para interactuar con el usuario seleccionado o con el modelo.

Botonera de usuarios: El propósito de este conjunto de botones es permitir al usuario seleccionar con qué otro usuario quiere interactuar sin la necesidad de que se encuentre en su campo de visión. Por esta razón, es necesario que cada botón represente a un usuario en concreto.

En la aplicación los usuarios se representan mediante un avatar de un color distinto al resto, por ello cada botón toma el color del usuario al que representa para facilitar la asociación. Además, cada botón almacena a este usuario para, cuando son seleccionados, informar al PIP sobre que usuario se debe interactuar. También, el botón asociado al profesor se representa con un marco negro y un símbolo representativo para mejor distinción.

Al tratarse de una aplicación multiusuario, debe existir coherencia entre el número de usuarios conectados y el número de botones que los representan. Por esta razón, los botones se instancian o eliminan automáticamente cuando hay una variación de los usuarios conectados en la sesión (ver pseudocódigo de la figura 5.20). Para ello, cada usuario debe pedirle al servidor el número de usuarios conectados y comprobar en cada *Update* si ese número es distinto al que había en el *Update* anterior. Los botones se instancian como objetos *hijo* del panel, de esta forma los botones se mantendrán estáticos dentro del panel pero se moverán junto con él. Además, si se ha conectado un usuario a la sesión, el panel lo indica mediante un halo verde que lo recubre durante un breve tiempo.

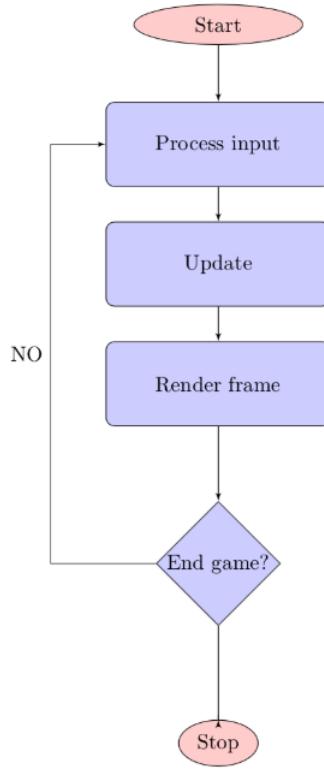


Figura 5.6: Esquema del GameLoop de unity

El panel es un objeto tridimensional que se encuentra representado sobre el plano XY y las posiciones de los botones son relativas a la posición del panel. Por ello, la posición de cada botón en el panel se calcula mediante un vector de 3 coordenadas de la siguiente forma:

- $posx = initialPosX + (n/4) * NextPosOffsetX$
- $posy = initialPosY - ((n - 1)mod3 * NextPosOffsetY)$
- $posz = 0$

Donde n representa el botón n-ésimo generado. Mediante esta función se consigue que los botones se generen por columnas con un máximo de 3 filas, de esta manera no existe un límite de usuarios a representar.

Botonera de interacción con el usuario: Este conjunto de botones permiten a los usuarios interactuar entre ellos y con el modelo y se encuentran en la parte inferior del panel (ver figura 5.5). El primer botón de todos, situado bajo el minimapa, permite entrar en el modo *Go Inside Model* del que se hablará con más detalle más adelante (ver sección 5.4). Los dos siguientes botones de interacción comenzando por la izquierda generan los *Photoports* dinámico y estático, respectivamente. Mientras que el cuarto y quinto botón modifican el punto de vista del usuario trasladándolo a la posición del usuario objetivo o reposicionando el modelo para visualizarlo como el usuario objetivo, respectivamente.

Las funciones que ejecutan estos últimos 4 botones ya estaban implementadas pero ha sido necesario modificarlas para su correcto funcionamiento con el panel. Para empezar, todas estas funciones dependían del Player Menu, por lo que esas dependencias se tuvieron que eliminar. Además, dado que con el Player Menu ya se interactuaba directamente con el usuario que lo contenía, para poder determinar con qué usuario se interactúa con el PIP ha sido necesario almacenar el usuario activo. Esto ha sido posible gracias a que cada botón instanciado almacena al usuario al cual hace referencia. Por último, ha sido necesario modificar el

código de cada función para que tomen como usuario a interactuar al usuario activo transmitido por el PIP, ya que el Player Menu utilizaba directamente el usuario que lo contenía.

Minimap/Radar View

Como se presentó al inicio de [5.2](#), queremos que el usuario pueda escoger con qué otro usuario quiere interactuar a través del PIP. Por ello, con tal de facilitar la comprensión de la posición y el punto de vista de cada usuario, se ha desarrollado un minimapa/Radar View dentro del PIP. El minimapa permite a los usuarios orientarse dentro del entorno virtual y decidir con qué otro usuario les es más conveniente interactuar para inspeccionar partes específicas del modelo, ya sea por la posición en la que se encuentran o su orientación.

Los elementos a representar en el minimapa son los usuarios y el modelo (en un entorno virtual más complejo, como por ejemplo un edificio con diferentes estancias, quizás sería necesario mostrar su estructura para ubicar al usuario). Puesto que para esta aplicación no es necesario mostrar mucha información a través del minimapa, se ha optado por una opción más simple y basada en el diseño de minimapas en videojuegos [\[40, 41\]](#). El minimapa es una imagen que muestra a los usuarios conectados y al modelo activo en sus posiciones relativas al usuario que lo contiene. La generación de esta imagen se realiza usando el propio pipeline renderizado de Unity. Para ello, se necesita de una cámara auxiliar que enfoca a los iconos de los usuarios y del modelo. Esta imagen mostrará en todo momento lo que la cámara está visualizando. De esta manera se pueden ver los cambios de posición y orientación de los demás usuarios en tiempo real. Dichos iconos:

- Se instancian con la creación de los usuarios a los que representan.
- Toman el color del usuario al que representan para mejor asociación.
- Están formados por un círculo que representa su posición y un triángulo invertido que representa su orientación o frustum de visión (Si se trata del profesor, contienen un ícono distintivo).
- Rotan siguiendo la orientación del usuario al que representan.

Con respecto a la manera de mostrar la información a través del minimapa se han barajado diversas opciones:

- **Minimap simple:** Consiste en una vista cenital fija de la escena.
- **Radar View centrada:** El usuario se sitúa en el centro del minimapa y los demás iconos se mueven respecto a él.
- **Radar View desplazada:** El usuario se sitúa en la parte inferior del minimapa y los demás iconos se mueven respecto a él.

A continuación, pasaremos a describir con más detalle cada una de estas opciones.

Minimap simple:

Consiste en una vista cenital fija de la escena. El usuario local está representado con un marco para distinguirlo de los demás (Basada en la explicada en [\[41\]](#)). Esta opción se descartó debido a que, tras probarla con varios usuarios, no se orientaban correctamente. Por ello se consideró que la mejor forma de ayudar al usuario a reconocer su posición y la de los objetos que le rodean es representar al modelo y a los demás usuarios en función de la posición y orientación del usuario local.

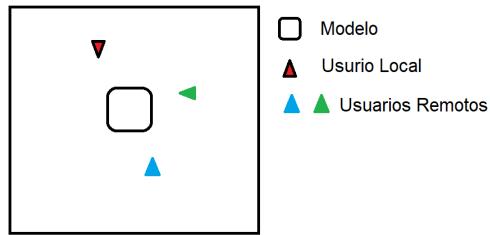


Figura 5.7: Representación esquemática del minimapa simple

Radar View centrada:

El usuario local se sitúa en el centro del minimapa. Por lo que la manera de mostrar la información es más similar a una Radar View, de ahí la dualidad del nombre de la metáfora. No obstante, no se consideraba útil mostrar lo que estaba situado detrás del usuario, ya que reducía el rango de visión de lo que se encontraba situado por delante de él. Además, los usuarios suelen estar distribuidos alrededor del modelo, inspeccionándolo. Lo que implica que, si un usuario se encuentra detrás de otro, su vista del modelo es peor porque está más lejos y puede estar oculto por el usuario situado delante suyo.

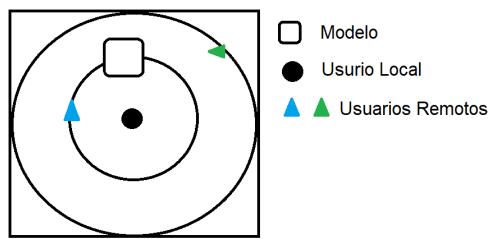


Figura 5.8: Representación esquemática de la Radar View centrada

Radar View desplazada:

Para evitar la problemática anterior, se decidió que el usuario local estaría siempre centrado en la parte inferior del recuadro que representa el Minimap. Los iconos que representan a los demás usuarios y al modelo activo cambian de posición respecto a su distancia y orientación con el usuario local. De esta manera se consigue una representación clara y personalizada de cada usuario. Para conseguir que los iconos se posicioneen respecto al usuario que representan, se ha creado una estructura de datos llamada PlayerMiniMap formada por el ícono que representa al usuario y el objeto del usuario. Mediante una lista de PlayerMiniMap se actualizan los iconos del minimapa de cada usuario.

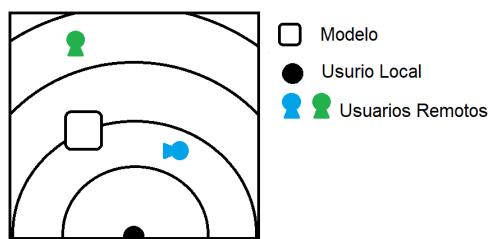


Figura 5.9: Representación esquemática de la Radar View desplazada

El minimapa muestra la posición física en la que se encuentra cada usuario. Se considera una posición como el conjunto de la situación en el espacio del usuario y su orientación. Sin embargo, si un usuario decide transportarse a la posición de otro, su posición dentro de la aplicación (posición virtual) es distinta a su posición física (ya que él físicamente no se ha movido pero dentro de la aplicación sí) y, por tanto, su punto de vista también lo es. En la mayoría de situaciones, la posición virtual se corresponderá con la posición física. Será solo cuando un usuario decida transportarse a otro cuando estas posiciones sean diferentes. Puesto que la intención principal del minimapa es mostrar las posiciones de los demás usuarios en el entorno virtual, se ha implementado una funcionalidad que permite observar tanto la posición física como la virtual de los usuarios. Para ello, cada usuario local en la aplicación debe enviar en cada *Update* al servidor la posición de su cámara (ya que la posición virtual de un usuario es, en esencia, lo que está viendo en todo momento en la aplicación) para que los demás usuarios, de forma remota, descarguen esta posición y puedan representarla a través del minimapa.

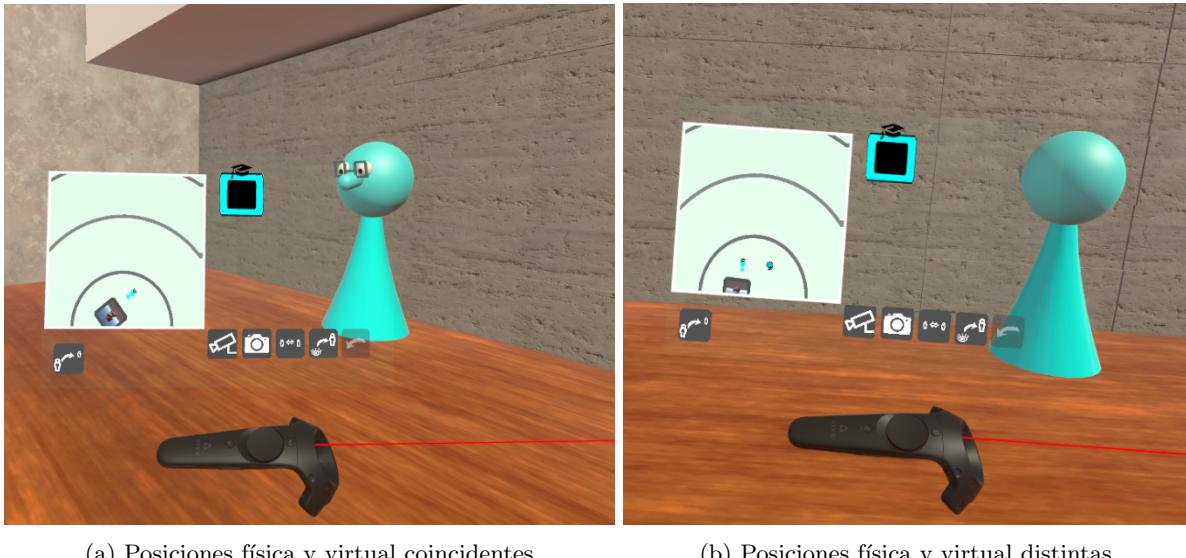


Figura 5.10: Visualización de la representación física y virtual en el minimapa



Figura 5.11: Visualización de la representación física y virtual de forma esquemática

Como se muestra en las imágenes de las figuras 5.10 y 5.11 la posición física del usuario está representada como un círculo con un reborde negro. Cuando esta posición se corresponde con la posición virtual, su representación contiene el triángulo invertido que representa el frustum de visión del usuario. No obstante, cuando estas posiciones difieren, como el frustum de visión del usuario se encuentra en su posición virtual, será el punto que la representa el que contenga el triángulo invertido.

Esto implica tener que añadir una variable más a la estructura PlayerMiniMap para almacenar el ícono que representa la posición virtual del usuario. Los iconos que representan las posiciones virtuales y físicas de los usuarios se modifican como muestra el fragmento de pseudocódigo de la figura 5.19.

5.2.2. Interacción con el panel

Para interactuar con el panel es necesario el mando de la mano derecha. Apuntar con el rayo marcador de este mando a los botones hará que cambie de color a verde, indicando que se trata de una entidad con la que se puede interactuar mediante el *Trigger* del mando. Para interactuar con un usuario es necesario marcar primero el botón que lo representa. Seleccionar un usuario hará que aparezca un recuadro negro en el interior del botón que lo representa y activará los botones inferiores, encargados de llevar a cabo las funciones descritas anteriormente (ver figura 5.12).

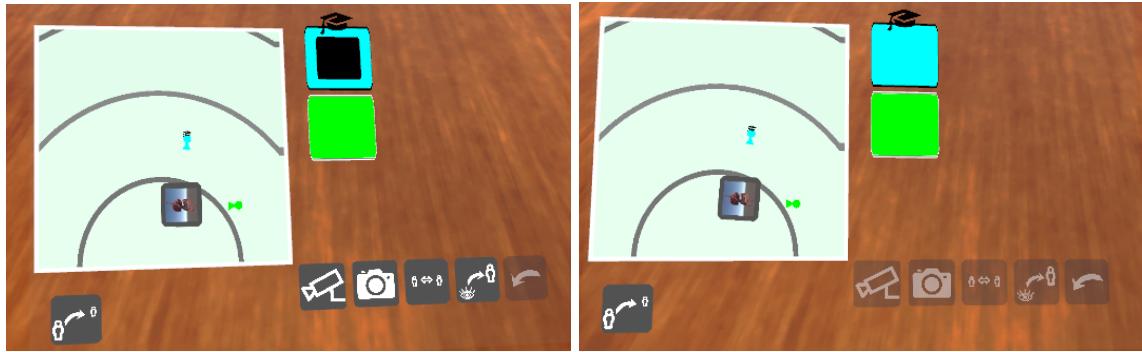


Figura 5.12: Selección de usuarios

Como se muestra en el diagrama de flujo de la figura 5.13, únicamente puede haber un usuario marcado a la vez por lo que seleccionar a otro implicaría desmarcar al primero. También es posible desmarcar a un usuario volviendo a pulsar su botón. Cabe destacar que activar el botón de teletransporte o el de rotar al modelo desactiva al otro ya que en esencia consiguen el mismo resultado: inspeccionar el modelo como lo haría el usuario marcado. También, activar uno de estos dos botones activa el último botón representado con una flecha. Dicho botón revierte las modificaciones hechas por los otros dos, devolviendo al usuario su punto de vista original. Finalmente, desmarcar al usuario objetivo o marcar a uno nuevo revierte las modificaciones automáticamente, volviendo al estado inicial, si no se hubiera hecho con anterioridad.

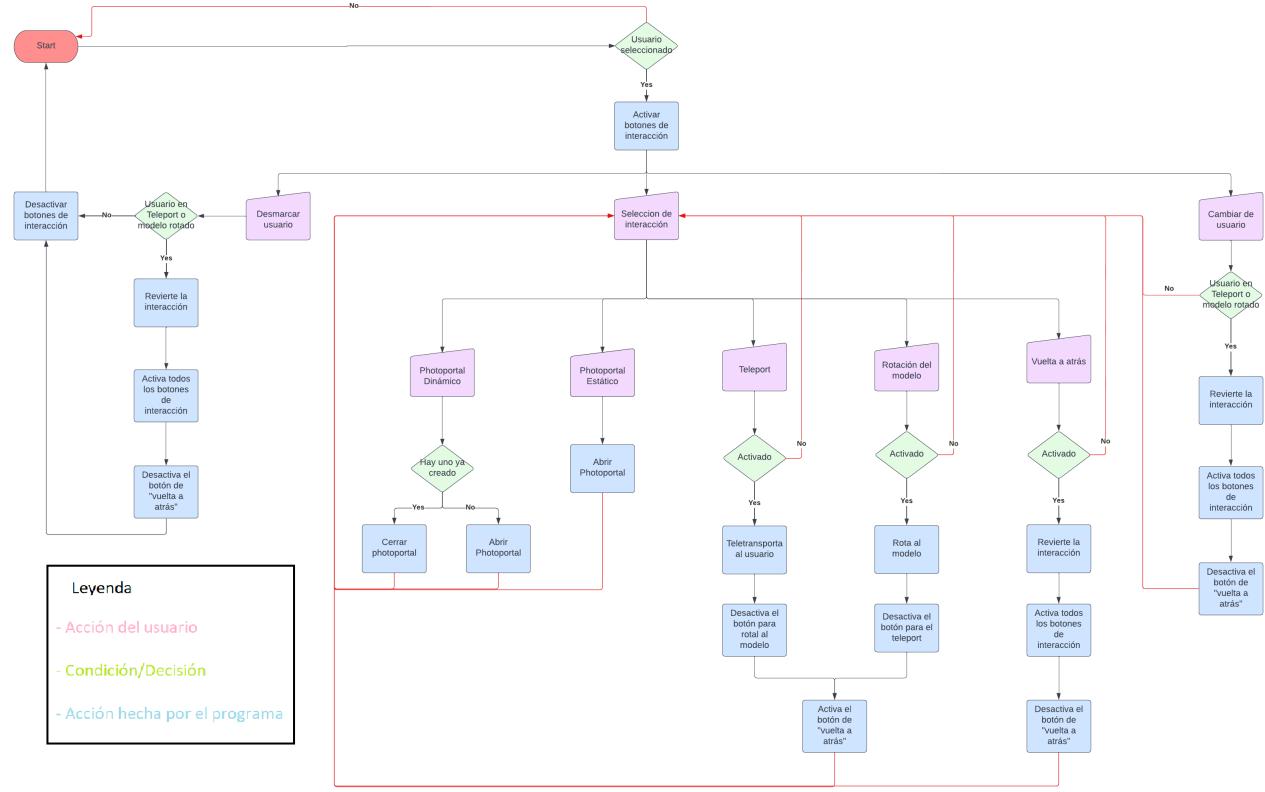


Figura 5.13: Diagrama de flujo de la botonera del panel

5.3. Photoportal acoplado

5.3.1. Características

En la aplicación anterior, los photoportals se instancian en la dirección a la que está apuntando el mando del usuario. En el caso de los photoportals estáticos, se generará uno cada vez que el usuario presiona su botón. Contrariamente, sólo puede haber un único photoportal dinámico en la escena a la vez. Esto tiene sentido ya que los primeros funcionan como fotografías que el usuario toma de la vista de otro, mientras que los segundos funcionan como una cámara de vigilancia posicionada en la cabeza de otro usuario, permitiéndole visualizar su punto de vista en todo momento sin tener que desplazarse hasta él.

Es por esto que se ha decidido acoplar únicamente el photoportal dinámico al PIP. Haciendo esto, el usuario tendrá en su brazo izquierdo el punto de vista de otro usuario y le permitirá comparar e inspeccionar lo que él está viendo sin perder la noción de su propio punto de vista (Ver figura 5.14) Además, con esta funcionalidad se elimina el problema de olvidar el lugar dónde se ha posicionado el photoportal si el usuario se mueve mucho por el entorno.

Como se ha mencionado en el apartado 5.1, es posible modificar tanto el tamaño como la posición de los photoportals mediante el mando de la mano derecha. También es posible desacoplar el photoportal dinámico del panel para fijar el photoportal en un lugar concreto del entorno. Una vez desacoplado, se activan todas las funcionalidades que ya permitía la aplicación anterior.



Figura 5.14: Photoportal acoplado al PIP

5.3.2. Implementación

Para implementar esto se han modificado las funciones de llamada y creación de photoportals para que admitan una variable booleana. Esta variable determina si la llamada a instanciar el photoportal ha sido desde el PIP. Hacer esto no sería necesario, ya que la única otra interfaz que podría generar photoportals sería el ya eliminado *Player Menu*. No obstante, se ha decidido hacerlo así por si en algún futuro se crea otra interfaz que también los genere.

A la hora de crear el photoportal, primero se debe comprobar si éste está acoplado al panel. En caso afirmativo, el photoportal se posiciona sobre el panel y lo toma como objeto *padre* (ver pseudocódigo de la figura 5.21). La relación de aspecto del photoportal y su tamaño se calcula mediante otra función como se muestra en la figura 5.22, aquí es necesario comprobar si el panel se encuentra acoplado o no. Esto es así porque la escala del photoportal es siempre relativa a su objeto *padre* y por tanto debe calcularse de forma diferente si tiene al panel como objeto *padre* (está acoplado) o no (está desacoplado). La manera de calcularlo consiste en revertir el escalado que se aplica al photoportal al asignarse como *hijo* del panel, esto es multiplicar su escala por 1/escalaDelPanel (en este caso se calcula mediante $1/(2*\text{escalaDelPanel})$ para que el tamaño del photoportal se adecúe más al del PIP).

Esto último es una casuística a tener en cuenta a la hora de desacoplar el photoportal por como está definida esta función. Para desacoplarlo o moverlo, el panel toma como objeto *padre* al mando que lo mueve mientras este tenga el *Trigger* presionado. Una vez que se suelta, el photoportal pasa a ser un objeto más de la escena sin objeto *padre*. Por tanto, se debe recalcular su tamaño (de la misma forma que se hacía con la relación de aspecto) para que se muestre en la escena con un tamaño igual o muy similar al que tenía cuando estaba acoplado.

5.4. Go Inside Model

El propósito de esta metáfora es permitir al usuario inspeccionar de una forma más precisa algunos puntos de interés del interior del modelo. Concretamente, el usuario es transportado al punto que ha seleccionado y el modelo se escala a un tamaño adecuado para simular que el usuario se encuentra en su interior a una escala miniaturizada (ver figura 5.15).

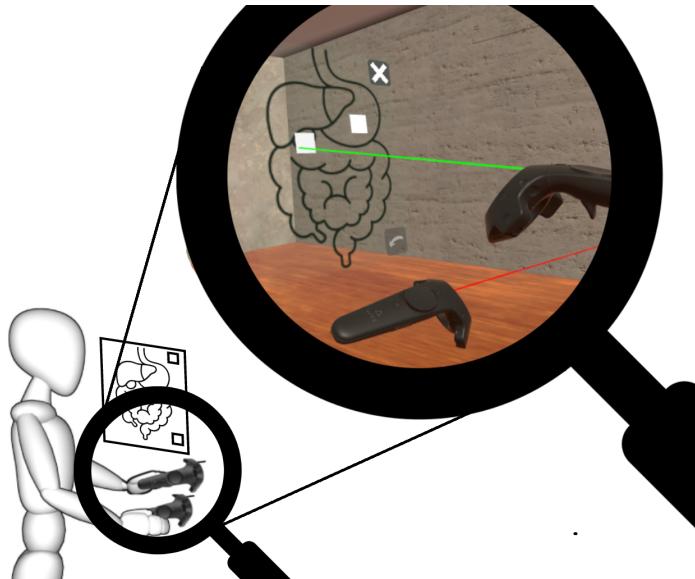


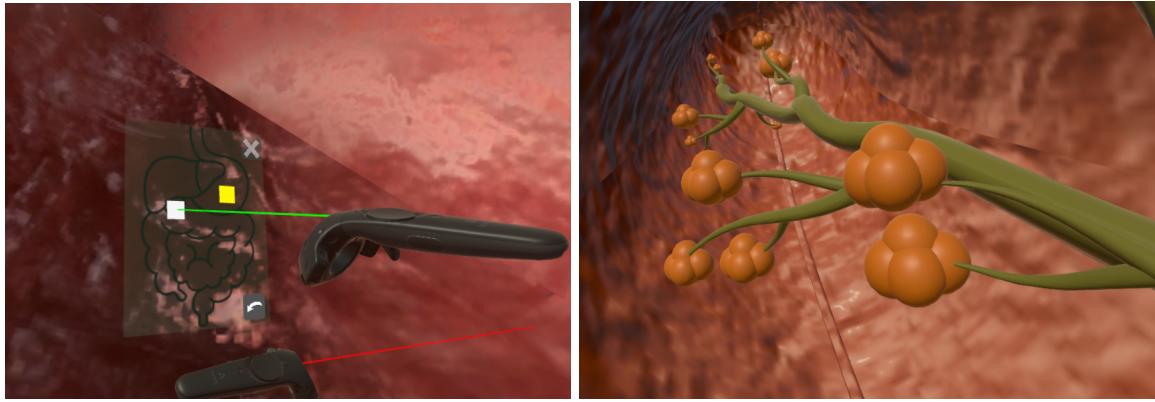
Figura 5.15: Esquema del uso del panel del GoInsideModel

5.4.1. Características

El Go Inside Model (GIM) es una interfaz activa, situada en el brazo izquierdo del usuario (al igual que el PIP) que muestra los distintos puntos de interés dentro del modelo a los que se puede trasladar. Como el PIP y el GIM están situados en el mismo lugar, cuando uno se encuentra activo, el otro deja de estarlo. También, para su funcionamiento, el GIM necesita de un fichero Xml [43] que le proporcione la información de los puntos a inspeccionar. Se ha optado por la utilización de ficheros Xml para proporcionar una mejor escalabilidad. De esta manera, si en un futuro se desea añadir nuevos puntos de interés, sólo sería necesario modificar este fichero. Los usuarios son trasladados a estos puntos de una forma similar al ya implementado teleport entre usuarios (de esto último se hablará con detalle más adelante).

5.4.2. Diseño de la interfaz

El PIP (ver sección 5.2) contiene un botón situado bajo el minimapa. Dicho botón provoca que se entre en el modo Go Inside Model. En este modo, el PIP es sustituido por el panel del GIM el cual consta de una imagen esquemática del modelo que se está inspeccionado actualmente y de un conjunto de botones situados en puntos específicos del esquema. Presionar uno de estos botones transporta al usuario dentro del modelo al punto representado en el esquema. Aquí dentro, el usuario puede inspeccionar con más detalle las cavidades internas y, también, puede o bien volver a su posición original o transportarse a otro punto de interés (ver figura 5.16). Finalmente, es posible salir del modo GIM, lo que hará que se sustituya nuevamente por el PIP.



(a) Selección del punto de interés del Páncreas

(b) Vista del usuario dentro del Páncreas

Figura 5.16: Usuario transportándose a un punto de interés (Páncreas) desde otro (Estómago)

5.4.3. Implementación

La aplicación consta, actualmente, de 4 modelos anatómicos distintos: El aparato digestivo, el corazón, el cerebro y el ojo. Puesto que cada modelo es distinto, cada uno de ellos tendrá puntos de interés a inspeccionar distintos. No obstante, los modelos del cerebro y el ojo no contienen cavidades internas de especial interés para su inspección. Por tanto, para este proyecto solo se han implementado los menús pertinentes para el aparato digestivo y el corazón.

Los puntos de interés se leen a través de un fichero XML. En este fichero la información está agrupada por modelos mediante un id que los representa (0 = Aparato digestivo y 1 = corazón). Cada modelo tiene sus puntos de interés (también diferenciados mediante un id) los cuales están formados por:

- **pos** : Posición a donde se transporta el usuario.
- **rot** : Posición del punto a donde se quiere que el usuario mire una vez transportado (*View reference point*).
- **size** : Factor que describe el escalado uniforme que se le aplica al modelo.
- **posbutton** : Posición donde se debe colocar el botón de este punto de interés en el esquema del panel del GIM.

Un ejemplo de como se vería un archivo para este código es el que se muestra en la figura 5.17. Para poder almacenar la información del fichero XML en la aplicación ha sido necesario crear dos estructuras de datos. La primera es la estructura *PointOfInterest* la cual está formada por los componentes listados anteriormente. La segunda es un diccionario, llamdo Points, que tiene como clave un entero, que representa el id de un modelo, y como datos almacenados una *LinkedList* [44] de *PointOfInterest*. De esta manera, cuando se quiera acceder a los puntos de interés de un modelo se puede hacer de forma sencilla mediante su identificador y, también, se pueden instanciar de forma cómoda todos los botones de los distintos puntos de interés gracias a la *LinkedList*.

La información se lee directamente del fichero XML, utilizando la extensión System.Xml [45], de la forma que se muestra en la figura 5.23.

El esquema del modelo y los botones que representan los puntos de interés se instancian en función del modelo activo y no son visibles ni interactivos hasta que el usuario entra en el modo GIM. En cada *Update* de la aplicación se comprueba el modelo activo en la escena. Si ese modelo es distinto al que había en el *Update* anterior se destruyen los botones instanciados, se cambia la imagen del esquema que representa al nuevo modelo y se instancian sus botones en sus correspondientes posiciones utilizando el diccionario Points, el cual ya contiene las posiciones de cada botón. Además, los botones instanciados toman como nombre el

```

<?xml version="1.0"?>
<models>
    <model id="0">

        <points>
            <point id="0">
                <pos>1.39,2.67,-1</pos>
                <rot>1.3,1.25,-3.215</rot>
                <size scale = "5"></size>
                <posbutton>120,122,0</posbutton>
            </point>

            <point id="1">
                <pos>1.22,3.36,0.067</pos>
                <rot>28.5,161.499,0</rot>
                <size scale = "10"></size>
                <posbutton>-102,50,0</posbutton>
            </point>
        </points>
    </model>

    <model id="1">
        <points>
            <point id="0">
                <pos>0.6,2.28,0.24</pos>
                <rot>0.57,1.68,-1.45</rot>
                <size scale = "2"></size>
                <posbutton>62,-257,0</posbutton>
            </point>

            <point id="1">
                <pos>0.46,2.97,-1.24</pos>
                <rot>-26,13,0</rot>
                <size scale = "2"></size>
                <posbutton>-148,-24,0</posbutton>
            </point>
        </points>
    </model>
</models>

```

Figura 5.17: Recorte del código Xml utilizado en este proyecto

identificador del punto al que representan. De esta manera, cuando son seleccionados, pueden indicarle al GIM a que punto en concreto debe trasladar al usuario. Finalmente, el usuario es trasladado dentro del modelo como se muestra en el pseudocódigo de la figura 5.24.

Como se puede apreciar, la posición original del usuario y la rotación que se le aplica se almacenan en variables globales (únicamente en los casos en los que el usuario quiera ir a un punto de interés sin estar ya en uno). Esto se hace para devolver al usuario a su posición original cuando desee salir del modelo. Además, se le aplica un fundido en negro (fade-in/fade-out) al campo de visión del usuario durante su transporte y escalado del modelo para conseguir una transición suave y evitar que el usuario se maree, abrume o sobresalte por un cambio repentino de escenario y posición.

5.5. Pseudocódigo de los algoritmos principales

Algorithm 1: setCoordsPIPUserAttached()

Result: out Vector3 pos, out Quaternion rot

```
//Si solo hay un mando conectado se posiciona en ese, si no
//se posiciona en la mano izquierda
if (leftHand) then
    | activeHand = VRInput.HandType.Left;
else
    | activeHand = VRInput.HandType.Right;
end
if (!useVR) then
    if (!BlockedPanel) then
        | pos = handPosition + handDirection * -0.1f;
        | pos = pos + handUp * 0.10f;
        | pos = pos - handRight * 0.2f;
        | rot = LookRotation(handDirection, handUp));
        | rot = AngleAxis(-90.0f, handUp) * rot;
        | rot = AngleAxis(40.0f, handDirection) * rot;
        //Se guarda la última posición y rotación del panel
        | gpos = pos;
        | grot = rot;
    else
        //Se bloquea el panel con la última posición y rotación
        | almacenada
        | pos = gpos;
        | rot = grot;
    end
else
    //Se calcula la posición respecto a los mandos
    | pos = handPosition + handDirection * -0.1f;
    | pos = pos + handUp * 0.10f;
    | pos = pos - handRight * 0.2f;
    | rot = LookRotation(handDirection, handUp));
    | rot = AngleAxis(-90.0f, handUp) * rot;
    | rot = AngleAxis(40.0f, handDirection) * rot;
end
```

Figura 5.18: Pseudocódigo que muestra como se reposiciona el panel en el brazo del usuario a partir de la ubicación del controlador izquierdo

Algorithm 2: UpdateMinimap()

Result: void

if *imInTP* **then**

//Si se entra en el TP se tiene que seguir a la camara del usuario

VirtualPlayer = FindGameObjectWithTag("VirtualCam");

else

VirtualPlayer = LocalPlayer;

end

if *Players.Count* != 0 **then**

i = 0;

while *i* != *Players.Count* **do**

i++;

//obtener posicion fisica del usuario remoto;

physicalpos = player.player.position; physicalrot = player.player.rotation;

//obtener posicion virtual del player;

virtualpos = player.player.VirtualPos; virtualrot = player.player.VirtualRotation;

//comprobar que el player remoto este o no en tp;

if *player.player.imInTP* **then**

| player.playerIcon.SetActive (true);

else

| player.playerIcon.SetActive (false);

end

//asignar posiciones relativas respecto al localPlayer

player.playerIcon.localPosition = LocalPlayer.position-physicalPos;

player.playerIcon.localRotation =

Quaternion.Euler (90f, physicalrot.eulerAngles.y-180, 180f);

//asignar posiciones virtuales respecto al localPlayer;

player.virtualPlayerIcon.localPosition = VirtualPlayer.position-virtualPos;

player.virtualPlayerIcon.localRotation =

Quaternion.Euler (90f, virtualrot.eulerAngles.y-180, 180f);

if TeacherNotFound and player.player.isTeacher () **then**

| EnableTeacherIcon(player);

| TeacherNotFound = false;

else

end

end

//asignar la posicion del modelo respecto al localPlayer;

ModelIcon.localPosition = LocalPlayer.position-Model.position;

ModelIcon.localRotation =

Quaternion.Euler (90f, Model.rotation.eulerAngles.y-180, 0f);

else

end

Figura 5.19: Pseudocódigo que muestra como se actualizan los iconos del minimapa

Algorithm 3: RefreshPlayers

Result: void

//Esta función también trata con los iconos del minimapa
RemoveButtons();
RemoveMinimapIcons();
ConnectedUsers = FindGameObjectsWithTag("PlayerPrefab");
n = 1;
i = 0;
while i != ConnectedUsers.length() **do**
 User = ConnectedUsers[i];
 //Si el usuario no es él mismo se añade al minimapa y a
 //la botonera
 if !User.isLocalPlayer **then**
 Minimap.AddPlayer(User);
 button = Instantiate(UserButton);
 //Calcula la posición para el n-ésimo botón
 button.position = positionForNButton(n);
 button.AssignPlayer(User);
 n++;
 else
 end
 i++;
end

Figura 5.20: Pseudocódigo que muestra como se actualizan los botones de los usuarios

Algorithm 4: initializeScreenPosition(bool PIPScreen)

Result: void

if PIPScreen **then**
 GameObject PIP = FindGameObjectWithTag("PIP");
 PIPAttached = true;
 this.setParent(PIP.transform);
 //posición relativa al PIP
 this.localPosition = AttachedPosition();
 //rotación relativa al PIP
 this.localRotation = AttachedRotation();
else
 Transform head = VRInput.head;
 this.rotation = head.rotation;
 //Calcula la posición del photoportal teniendo en cuenta
 //hacia donde mira el usuario y la distancia a la que se
 //quiere posicionar
 this.position = SpawnPosition();
end

Figura 5.21: Pseudocódigo que muestra como se inicializa el photoportal

Algorithm 5: setScreenAspectRatio(float aspectRatio,bool PIPScreen)

Result: void

if *PIPScreen* **then**

| PIPattached = true;

else

end

GameObject PIP = FindGameObjectWithTag("PIP");

//Si es mas ancho que alto

if (*aspectRatio* > 1) **then**

if (*PIPScreen*) **then**

| this.localScale =

| Vector3((*aspectRatio*/(2*PIP.localScale.x)), 1/(2*PIP.localScale.y), 1);

else

| this.localScale = Vector3(*aspectRatio*, 1, 1);

end

else

if (*PIPScreen*) **then**

| this.localScale =

| Vector3(1/(2*PIP.localScale.x), (*aspectRatio*/(2*PIP.localScale.y)), 1);

else

| this.localScale = Vector3(1, 1/*aspectRatio*, 1);

end

end

Figura 5.22: Pseudocódigo que muestra como se inicializa el photoportal

Algorithm 6: getPOIListFromXml()

Result: Dictionary<int, LinkedList<PointOfInterest>>

```
while reader.Read() do
    //Si el iterador encargado de leer encuentra un elemento
    //con nombre "model"
    if reader.NodeType == XmlNodeType.Element reader.Name == "model" then
        modelId = reader["id"];
        //Se declara la lista de puntos, tanto para reiniciarla
        //como para inicializarla
        ListOfPoints = new LinkedList<PointOfInterest>();
    else
        end
    if reader.NodeType == XmlNodeType.Element reader.Name == "point" then
        pointId = reader["id"];
    else
        end
    if reader.NodeType == XmlNodeType.Element reader.Name == "pos" then
        //Leer toda la información que hay en el Elemento
        while reader.NodeType != XmlNodeType.Text do
            value = reader.Value;
            Transformar el valor leido en un vector y almacenarlo en un vector pos;
        end
    else
        end
    Repetir estructura del if anterior para rot, size y posbutton (o más variables que haya)
    //Si se encuentra al final del elemento
    if reader.NodeType == XmlNodeType.EndElement reader.Name == "point" then
        PointOfInterest Point = new PointOfInterest(pointId, pos, rot, scale, posbutton);
        ListOfPoints.AddLast (Point);
    else
        end
    if reader.NodeType == XmlNodeType.EndElement reader.Name == "model" then
        DictionaryOfPoints.Add(modelId,ListOfPoints);
    else
        end
    end
return DictionaryOfPoints;
```

Figura 5.23: Pseudocódigo que muestra como se lee el fichero xml

Algorithm 7: GoInside(bool fromOtherPOI)

Result: void

//Si esto ocurre quiere decir que el player va al punto desde fuera del modelo

if *fromOtherPOI* **then**

| originalPos = LocalPlayer.transform.position;

| originalScale = Model.transform.localScale;

else

| aux = Currentbutton;

| //regresa a la posición original fuera del modelo

| GoBack();

| Currentbutton = aux;

end

Desactivar botón de la posición a la que se quiere ir

Currentbutton.GetComponent<Button>().interactable = false;

int i = 0;

while *i != ModelPoint.Count* **do**

| //Obtener la información del punto de vista al que se quiere ir mediante su id

| if *POI.point == Currentbutton.name* **then**

| | Point = POI;

| | break;

| | else

| | end

end

FadeIn();

//reescalar el modelo

Model.localScale = Point.modelScale;

//Hacer que el modelo mire hacia la dirección deseada

sourceDirection = VRInput.headDirection;

sourceDirection.y = 0;

targetDirection = Point.rot;

targetDirection.y = 0;

//Calcular el ángulo de incremento

teleportRotationYIncrement = SignedAngle(sourceDirection, targetDirection);

rotationReferencePoint = VRInput.headPosition;

//Rotar al usuario

VRInput.head.root.RotateAround(rotationReferencePoint, Vector3.up,

| teleportRotationYIncrement);

//reposicioanr al usuario

teleportUserPositionIncrement = Point.pos - rotationReferencePoint;

VRInput.head.root.Translate(teleportUserPositionIncrement, Space.World);

FadeOut();

Figura 5.24: Pseudocódigo que muestra como se transporta al usuario dentro del modelo

Capítulo 6

Test de usabilidad

Para evaluar el correcto funcionamiento y la usabilidad de las metáforas desarrolladas durante el proyecto, se ha diseñado un test de usabilidad. El test está dividido en 3 secciones (una para cada metáfora implementada) y para cada una de ellas se quiere evaluar distintos aspectos. A continuación pasamos a describirlos:

Panel de interacción personal (PIP)

- Posición preferente del usuario sobre el PIP en su brazo izquierdo.
- Asociación intuitiva de los botones con los usuarios que representan.
- Utilidad del minimapa.
- Correcta interpretación del minimapa (Orientación, simbología intuitiva...).
- Entendimiento de la representación de la posición virtual y física.

Photoportal acoplado

- Utilidad de la metáfora.
- Comodidad de la metáfora.

Go Inside model (GIM)

- Uso intuitivo del menú del GIM.
- Comodidad de la metáfora.
- Provocación de efectos adversos (malestar, náusea, fatiga...).

También, se evaluará de forma cuantitativa la eficacia del minimapa mediante un técnico encargado de registrar las veces que los usuarios utilizan el minimapa en lugar de alzar la vista para encontrar a algún usuario en concreto. Además, una vez realizadas estas tareas, los usuarios deberán responder a una serie de preguntas relacionadas con el grado de satisfacción, comodidad y utilidad de las diferentes metáforas para evaluarlas de forma cualitativa.

Finalmente, dado que se trata de un primer test de usuario y la aplicación es un prototipo que continúa en desarrollo, el test se ha hecho a una población de usuarios que no son necesariamente del grupo objetivo para el que está dirigida la aplicación final. No obstante, esto no quita valor a los resultados obtenidos ya que nos ha permitido un primer acercamiento de la aplicación a un grupo de usuarios externos para comprobar la viabilidad, utilidad y eficacia de las metáforas implementadas.

6.1. Diseño del experimento

Antes de comenzar con el test, a los usuarios se les pasará un formulario de consentimiento y, una vez lo respondan, se les hará una pequeña demostración del funcionamiento de la aplicación para que sepan lo que hace cada botón y como se puede interactuar con ellos. Una vez explicado esto, los participantes deben realizar varias tareas que nos permitirá valorar la eficacia en el uso de las metáforas implementadas:

- **Posición del PIP:** Se dará a elegir al usuario entre las 2 posibles posiciones del PIP (Sobre el brazo izquierdo o delante del mando izquierdo)
- **Uso de los botones y el minimapa:** Esta tarea estará dividido en 3 fases distintas:
 - **Selección de usuarios:** El usuario deberá seleccionar a varios usuarios en concreto para comprobar que se orienta y los distingue correctamente (Un usuario situado en un punto determinado, un usuario mirando a una dirección concreta o usuarios diferenciados por alguna simbología como el profesor). Para esta tarea, tomaré nota de cómo seleccionan al usuario que se les pide (alzando la vista o buscándolo en el minimapa) para puntuar la eficacia del minimapa.
 - **Entendimiento de la posición virtual/física:** En esta situación se le explicará la diferencia entre posición virtual y posición física y se desdoblará la posición de otro usuario para comprobar si el usuario distingue correctamente su posición física de su posición virtual.
 - **Interacción con el usuario:** El usuario deberá interactuar con el otro usuario seleccionado de distintas maneras (Rotar el modelo, photoportal estático, photoportal dinámico...)
- **Uso del Photoportal acoplado:** Se le pedirá al usuario que seleccione a otro para utilizar el Photoportal dinámico con él. Una vez que lo tenga activo se le preguntará acerca de lo que se ve a través de él y si le resulta cómodo tener la vista de otro usuario en esa posición. Finalmente, se le explicará cómo desacoplar el photoportal y se le pedirá que lo haga e interactúe con él.
- **Uso del GoInsideModel:** Se le pedirá al usuario entrar en el modo GoInsideModel y, una vez dentro, se le pedirá que se transporte dentro del páncreas y de ahí al estómago, para comprobar que la representación esquemática es suficientemente intuitiva para su utilización.

Al finalizar todo el experimento se les pasará un formulario donde evaluarán de forma cualitativa los aspectos explicados anteriormente que se quieren comprobar de cada metáfora y se les pedirá que den sugerencias o comentarios para cada metáfora.

6.2. Datos de los participantes

En este test ha habido un total de 9 participantes. A cada uno de ellos se les pedía responder a un formulario de consentimiento previo al experimento donde recogían los datos siguientes para tener una noción de la población utilizada:

- Nombre y apellidos.
- Nivel de conocimiento en Anatomía general, Realidad virtual y videojuegos (Nada, escaso, medio o alto).
- Frecuencia de uso de tecnologías de Realidad Virtual, ordenadores y videojuegos (Nada, escaso, medio o alto).

Cabe aclarar que la razón para pedir el nombre y apellidos de los usuarios es únicamente para vincular sus respuestas en el formulario de consentimiento con las del test de usuario. Los resultados obtenidos han sido los siguientes:

■ **Nivel de conocimiento en Anatomía general, Realidad virtual y videojuegos.**

	Anatomía General	Realidad Virtual	Videojuegos
Escaso	33,33%	22,22%	11,11%
Medio	55,56%	55,56%	44,44%
Alto	11,11%	11,11%	33,33%
Nada	0,00%	11,11%	11,11%

De la tabla anterior podemos extraer que la mayoría de los participantes tienen un conocimiento Medio-Escaso de Anatomía general y Realidad virtual, mientras que de videojuegos tienen un conocimiento Medio-Alto.

■ **Frecuencia de uso de tecnologías de Realidad Virtual, ordenadores y videojuegos.**

	Uso de Realidad Virtual	Uso de videojuegos	Uso de ordenadores
Nunca	44,44%	11,11%	0,00%
Alguna vez	22,22%	33,33%	0,00%
Muy a menudo	33,33%	22,22%	0,00%
Todos los días	0,00%	33,33%	100,00%

Como se puede observar, contamos con una población muy dada al uso de ordenadores, que juega a videojuegos ya sea de forma esporádica o asidua y que, en mayoría, o bien alguna vez ha utilizado alguna herramienta de Realidad Virtual o bien las utiliza muy a menudo.

6.3. Análisis de los resultados

Como se ha mencionado al inicio de este capítulo (6), el test de usuario esta dividido en 3 partes, una para cada metáfora de interacción implementada en el proyecto. Por ello, los resultados que se han obtenido en cada una de estas partes se analizarán de forma separada y al final se dará una valoración general de todos en conjunto.

El análisis de estos resultados sirven para valorar el grado de satisfacción de los usuarios al utilizar las diferentes herramientas de interacción (Comodidad, utilidad, fácil entendimiento y uso de las herramientas...) además de obtener un *feedback* de las posibles modificaciones o mejoras a añadir para la aplicación a futuro.

El grado de satisfacción se mide mediante una escala de Likert del 1 al 7 que representa:

- 1: Estoy totalmente en desacuerdo
- 2: Estoy en desacuerdo
- 3: Estoy un poco en desacuerdo
- 4: Neutral
- 5: Estoy un poco de acuerdo
- 6: Estoy de acuerdo
- 7: Estoy totalmente de acuerdo

6.3.1. Panel de interacción personal

Disposición del panel

Para empezar el test, se les preguntó a los usuarios que disposición del panel preferían: Sobre el brazo izquierdo (como si se tratase de un reloj) o delante de la mano izquierda.

Los resultados obtenidos están bastante equilibrados:

El 55,55 % de los usuarios prefieren observar el panel sobre el brazo izquierdo principalmente porqué, según ellos, tienen una vista más limpia de la aplicación y les permite centrarse más en el modelo a inspeccionar. Por otro lado, el 44,44 % de los usuarios preferían observar el panel delante de la mano ya que les parece una posición más natural y les permite un mayor descanso en el brazo.

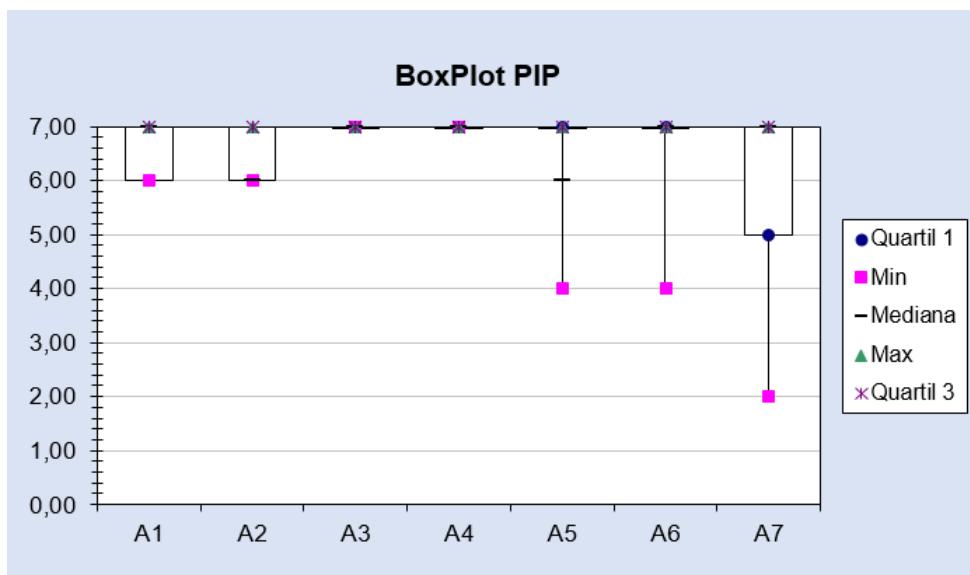
De aquí podemos extraer que ambas posiciones tienen sus ventajas y desventajas y, como bien muestran los resultados estadísticos, no hay una opción objetivamente mejor. Además, la mayoría de usuarios entrevistados aseguraron que dependiendo de la situación preferirían una posición u otra, por lo que tal vez sería interesante poder cambiar la disposición del panel mediante un botón de uno de los mandos.

Valoración del panel

Con tal de valorar la experiencia de los usuarios utilizando el panel, se les pasaron un conjunto de afirmaciones sobre las cuales debían mostrar su grado de conformidad con cada una de ellas siguiendo la métrica explicada en el apartado anterior. Las afirmaciones fueron las siguientes:

- A1) Me ha parecido fácil el uso del panel
- A2) El uso de la técnica ha sido cómodo
- A3) He asociado correctamente los botones con los usuarios a los que representan.
- A4) He distinguido correctamente al usuario seleccionado para interactuar mediante el panel.
- A5) Me he orientado correctamente mediante el minimapa.
- A6) He podido elegir al usuario que más me interesaba para interactuar mediante el minimapa.
- A7) He distinguido correctamente la posición virtual del usuario de su posición física

Los resultados obtenidos se muestran en el siguiente Boxplot:



Como se puede apreciar, todos los usuarios se sienten conformes con el uso del PIP (ya que sus valoraciones oscilan entre el 6 y el 7) con excepción de a la hora de distinguir la posición física de la posición virtual, donde se presenta mayor disconformidad. Durante el test el 44,44 % distinguió correctamente la una de la otra sin necesidad de explicarle que símbolo representaba a cual. Sin embargo, una vez conocían esto, les resultó sencillo distinguirlo en todo momento.

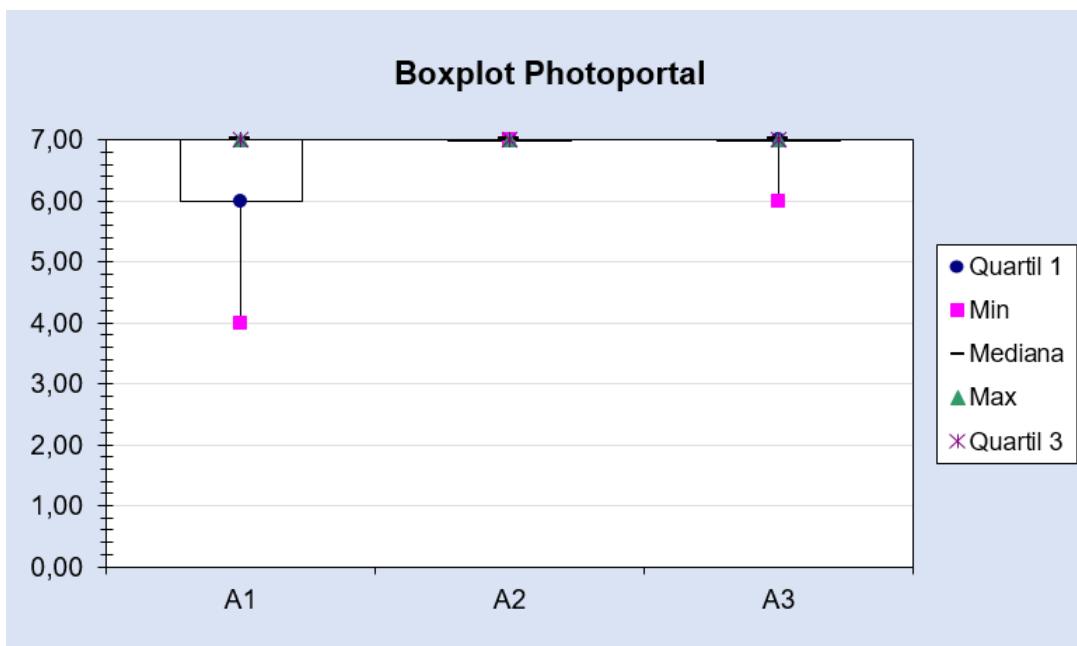
Además, se midió el uso que le daban los usuarios al minimapa. Para ello se les pidió seleccionar a un usuario en concreto en 4 ocasiones distintas durante el test y se anotó si, para determinar de que usuario se trataba, alzaban la vista para buscarlo en el entorno o si lo buscaban utilizando el minimapa. El 66,66 % de los usuarios utilizaban el minimapa para buscar a otro usuario que se encontraba oculto en su campo de visión. También, el 55,55 % de los usuarios observaban el minimapa para seleccionar al usuario que se correspondía con el profesor, es decir observaban antes el símbolo distintivo del panel que no la representación distintiva que tiene el modelo del profesor. No obstante, la mayoría de usuarios (el 66,66 %) no alternaban en el uso del minimapa con alzar la vista, es decir, o bien alzaban la vista en todas las ocasiones o bien miraban el minimapa en todas las ocasiones. Por tanto, podemos concluir que, para la situación que se les presentaba a los participantes del test, el uso del minimapa no era esencial y algunos se sentían más cómodos alzando la vista para seleccionar al usuario objetivo. Sin embargo, si el usuario era más complejo de identificar (ya sea por que estaba oculto o por que tenía algún elemento distintivo) les resultaba más cómodo observar el panel. Tal vez esta sea la razón por la que también hay una pequeña disconformidad en el Boxplot en cuanto al uso del minimapa.

6.3.2. Photoportal acoplado

Esta sección es la más reducida de las tres y lo que se quiere estudiar aquí es la utilidad de tener un photoportal acoplado en el brazo. Para ello, también se le pasaron una serie de afirmaciones a los participantes para medir su grado de conformidad. Las afirmaciones son las siguientes:

- A1) Me ha resultado útil tener el photoportal acoplado al panel.
- A2) Me ha parecido fácil desacoplar el photoportal del panel.
- A3) El uso de la técnica ha sido cómodo.

Los resultados obtenidos se muestran en el siguiente Boxplot:



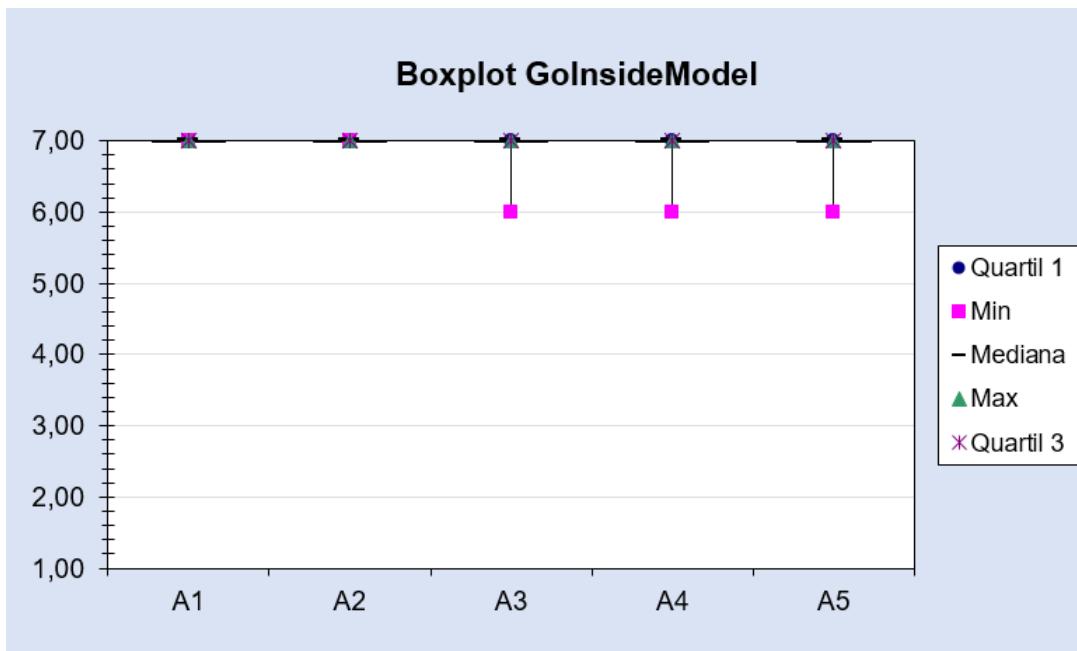
Como se puede observar, por lo general, los resultados son positivos. Los usuarios se sintieron cómodos al utilizar el photoportal y les pareció sencillo desacoplarlo. Sin embargo, se presenta un poco de disconformidad en cuanto a la utilidad de tener el photoportal acoplado al panel, las razones de esto han sido el tamaño del photoportal y la sensación de mareo que puede provocar, como bien puntualiza el siguiente usuario: "*El Photoportal me mareó un poco cuando el suelo de la pantalla coincidía con el suelo de la habitación que veía, me resultaba más cómodo desacoplarlo para que esto no pasara.*" Por otro lado, la mayoría de usuarios aseguraban durante el test, que dependiendo de la situación les sería más favorable tenerlo desacoplado o acoplado.

6.3.3. Go inside model

La valoración de esta sección se ha realizado de forma similar a las anteriores. Las afirmaciones que se le pasaron a los usuarios son las siguientes:

- A1) Me ha parecido sencillo el uso del GoInsideModel.
- A2) El uso de la técnica ha sido cómodo
- A3) La representación de los puntos de interés en el esquema me ha parecido intuitiva.
- A4) En todo momento me he sentido seguro/a utilizando la técnica.
- A5) El resultado de seleccionar un punto de interés ha sido el esperado.

Los resultados obtenidos se muestran en el siguiente Boxplot:



Como se puede observar los resultados son positivos. Los usuarios encontraron sencilla, útil, segura e intuitiva la metáfora. Además, gracias a los comentarios que nos dejaron los usuarios del test, sabemos que esta metáfora fue la que más disfrutaron por que era la que mayor inmersión proporcionaba.

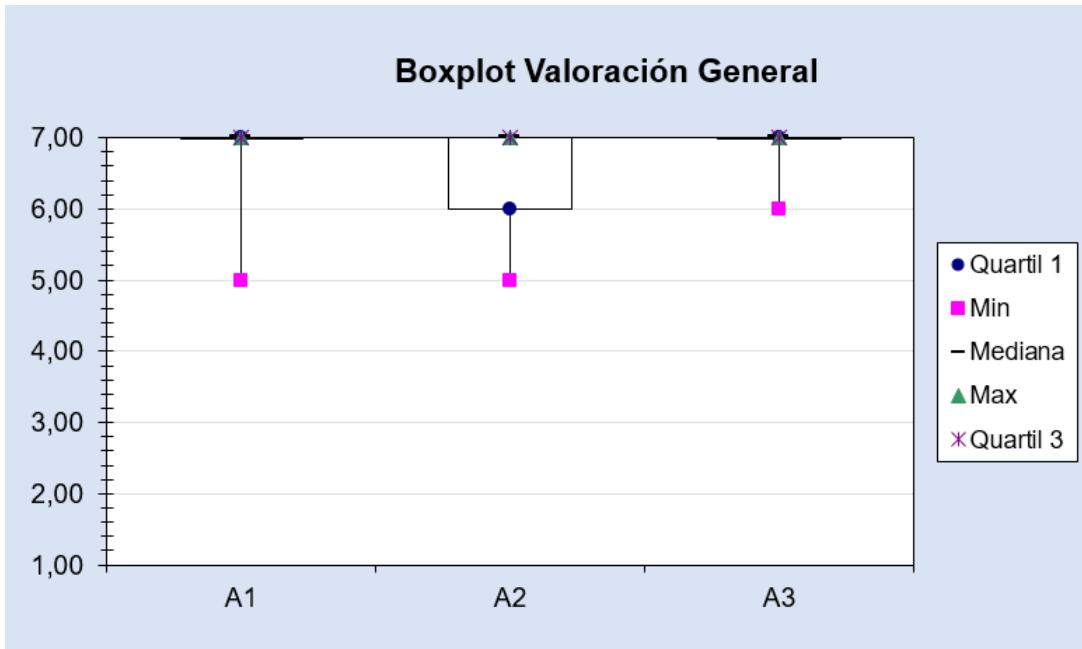
6.3.4. Valoración general

A modo de valoración general de las metáforas implementadas en el proyecto, a los participantes del test se les pidió que dejaran su opinión acerca de cada una de las metáforas, así como puntuarlas del 1 al 10 y

mostrar su conformidad con unas últimas 3 afirmaciones, como se ha hecho en los apartados anteriores. Las afirmaciones son las siguientes:

- A1) Considero que puedo utilizar las técnicas sin ayuda.
- A2) Considero que la mayoría de personas aprenderán el uso de las técnicas de forma rápida
- A3) Me siento seguro de mi mismo/a al utilizar las técnicas

Los resultados obtenidos se muestran en el siguiente Boxplot:



Como se puede observar, los resultados son positivos por lo general, habiendo un poco de discrepancia en la segunda afirmación sobre la velocidad de aprendizaje de las metáforas de interacción.

En cuanto a las puntuaciones para cada metáfora, mostramos el gráfico de la figura 6.1 que representa el recuento de las puntuaciones otorgadas por los usuarios a cada metáfora. Como se puede apreciar, las tres metáforas han sido valoradas muy positivamente.

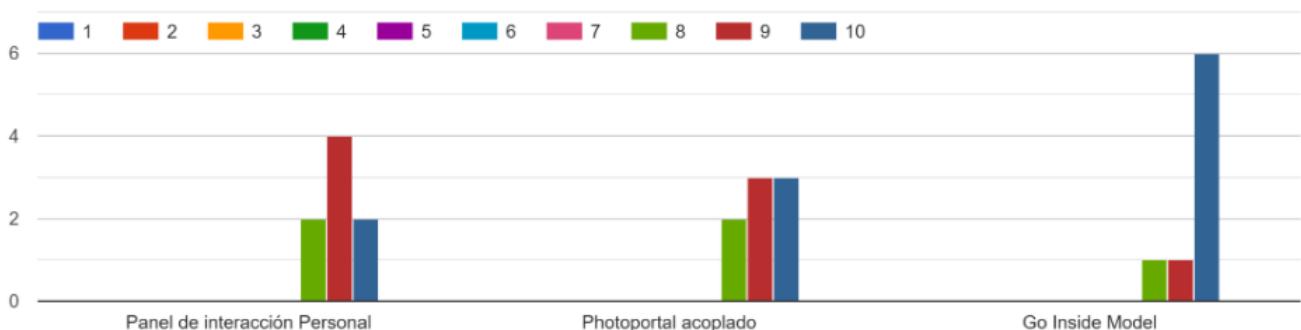


Figura 6.1: Recuento de puntuaciones para cada metáfora

Los comentarios que han dejado los usuarios han sido de gran utilidad y muy positivos en su mayoría. A continuación se presentan un conjunto de ellos en los que los usuarios sugieren posibles mejoras que podrían ser interesantes a explorar y tener en cuenta:

- Respecto al photoportal acoplado: *El photoportal está bien. Lo prefiero acoplado porque que se mueva contigo es muy cómodo, pero el poder desacoplarlo puede ser útil también. Igual podría estar bien poderlo volver a acoplar.*
- Respecto a la posición del PIP: *Para encontrar a los usuarios en el mapa creo que sería más sencillo si lo tuviera delante de la mano, porque solemos poner la mano hacia delante y la disposición se correspondería con lo que vemos de frente. Al mirar al brazo rotamos la cabeza y lo que vemos en el mapa cambia.*
- Respecto al minimapa del PIP: *La técnica de Panel me ha gustado mucho y solo añadiría una "leyenda"Para indicar qué significan los símbolos (posición física, virtual...).*
- Respecto al GoInsideModel: *El GoInsideModel es cómodo y hace lo que esperas. La transición me resultó cómoda. Lo único es que si el entorno colaborativo es con todas las personas en la misma sala, se pierde la referencia de dónde están porque tampoco se ve el minimapa y si hubiera gente podría causar algo de inseguridad moverse.*

Capítulo 7

Conclusiones

Durante este proyecto hemos desarrollado 3 técnicas de interacción en un entorno virtual colaborativo con el objetivo de mejorar la comunicación e interacción de los usuarios entre ellos mismos y con el entorno. Estas técnicas las hemos nombrado como *Panel de Interacción Personal (PIP)*, *Photoportal acoplado* y *GoInsideModel (GIM)*. Todas ellas cumplen con los objetivos que se habían fijado y se podrían utilizar perfectamente en cualquier ámbito (Ya sea en el contexto de una clase de anatomía o no) que requiera de un grupo de personas inspeccionando de forma colaborativa un modelo tridimensional en un entorno virtual. El PIP permite un mejor entendimiento del entorno y ofrece una interacción más cómoda a los usuarios, el Photoportal acoplado consigue tener una cierta versatilidad a la hora de observar lo que está viendo otro usuario sin perder la noción de lo que nosotros estamos observando y el GIM permite inspeccionar con mucho más detalle las partes interiores más importantes de cada modelo.

Sin embargo, tras evaluar los resultados del test de usuario, todavía quedan detalles a mejorar. Queda fuera del alcance de este proyecto y como desarrollo a futuro incorporar las sugerencias de los usuarios del test de usabilidad, así como continuar puliendo y mejorando la aplicación hasta su versión final. En concreto, se podría mejorar el minimapa del PIP para que muestre una pequeña leyenda que permita distinguir más fácilmente la posición virtual de la física. También sería interesante que la funcionalidad del Teleport transportara al usuario a la posición virtual del usuario objetivo, ya que tiene más sentido llevarlo a donde está mirando el otro usuario que a donde está posicionado físicamente. Finalmente, se podría añadir la funcionalidad de volver a acoplar el Photoportal al PIP.

A modo de conclusión, se han cumplido todos los objetivos de forma satisfactoria, dentro del tiempo esperado y quedamos muy contentos con los resultados obtenidos en el test de usuario y con el trabajo realizado en el proyecto.

Capítulo 8

GEP

8.1. Descripción de las tareas

En este apartado se intentará formalizar una planificación de las tareas a llevar a cabo durante todo el proyecto, así como el tiempo estimado que se necesitará para hacerlas y la carga de trabajo que implicará cada una.

El proyecto comienza el 21 de Febrero y el objetivo es considerarlo acabado el 27 de Mayo que se corresponde con el día límite de inscripción para el turno de lectura. Este intervalo consta de 96 días y se pretende dedicar entre 5 i 6 horas diarias. Además todas las tareas que conlleven el desarrollo de alguna herramienta para la aplicación necesitarán de un test de utilidad al final de su desarrollo.

El proyecto se dividirá en 2 fases: GEP y desarrollo del proyecto.

8.1.1. GEP

En esta fase, que consta de 4 entregas al tutor de GEP, se contextualizará y definirá el alcance del proyecto, se justificará la necesidad del mismo y se identificarán todas las herramientas necesarias para llevarlo a cabo. Se estima una dedicación de aproximadamente 116h.

8.1.2. Desarrollo del proyecto

Esta fase estará dividida en diferentes tareas. Se estima que el tiempo de dedicación total será aproximadamente de 375h.

Tarea 1. Formación y preparación

En esta tarea se dedicarán días a inspeccionar la arquitectura global (Elementos de RV, arquitectura cliente-servidor de Unity...) junto con la tutora Eva Monclús y se irán adquiriendo los conocimientos necesarios de forma incremental durante el desarrollo, a medida que me vaya familiarizando con el entorno de RV (Unity [20], HTC Vive [16]).

Pese a tener conocimientos de Unity gracias a otras asignaturas de la carrera, no tengo conocimientos de sistemas multiusuario ni de desarrollo de aplicaciones colaborativas en RV [8,9].

Por tanto, para esta tarea se prevé que se necesitarán aproximadamente 60h.

Tarea 2. Desarrollo de un panel de interacción personal

El objetivo de esta tarea es desarrollar un panel de interacción personal acoplado al brazo izquierdo del usuario. Se trata de una interfaz activa que deberá aportar información de los otros usuarios.

Para esta tarea será necesario estar familiarizado con el código y las herramientas de RV y diseñar una interfaz de usuario para que sea útil. Por tanto, se prevé que el tiempo estimado necesario será de 110h.

Tarea 3. Acoplar Photoportals al panel

En esta tarea se intentará acoplar un sistema de photoportals [9] al panel creado anteriormente para que sea posible visualizar el punto de vista de otro usuario.

Acoplar este sistema conlleva pensar algunas modificaciones en el panel para que sea más cómoda la visualización. Por ello, el tiempo esperado para esta tarea es de 85h.

Tarea 4. Desarrollo de una nueva metáfora de navegación que permita manipular la escala del modelo

Para esta tarea se tratará de modificar el tamaño del modelo (o de ciertas partes de él) haciéndolo más grande y así poder explorar mejor las cavidades y partes interiores de él.

Esta tarea conlleva crear una transición adecuada al escalado del modelo, determinar que puntos del modelo son de interés para su escalado y si se hace de forma individual para cada usuario o global. Por tanto, el tiempo esperado para finalizar esta tarea es de 60h.

Tarea 5. Test de usabilidad

El objetivo de esta tarea es el de hacer un test de usuario que permita modificar las metáforas creadas para que la aplicación sea objetivamente mejor para el usuario. Con esta tarea se finalizaría todo lo relacionado con la implementación y desarrollo de metáforas de interacción y su tiempo esperado es de 60h.

8.1.3. Recursos

Recursos humanos

Para este proyecto se definen 4 roles:

- Director del proyecto: Se encarga del curso que debe seguir el desarrollo del proyecto
- Investigador: Se encarga de hacer todo el trabajo de búsqueda de información.
- Desarrollador: Se encarga de implementar y diseñar la solución definida por la gestión del proyecto
- Evaluador: Se encarga de realizar las pruebas de validación de la aplicación

Recursos materiales

Como recursos materiales para este proyecto tenemos:

- PC: Es necesario disponer de ordenador propio para poder trabajar en el proyecto cuando no se esté en el centro de realidad virtual (CRV) del ViRVIG [11]
- GitLab: Repositorio de control de versiones de todo el proyecto [22]
- Overleaf: generación de documentos online con LaTeX
- Unity3D: Motor de desarrollo de la aplicación [20]
- HTC Vive: Dispositivo de RV inmersivo [16]

Nombre	Tiempo	Dependencia	Material	Personal
GEP	75h	-	PC, Overleaf, Gantt	DP, I
Desarrollo del proyecto	375h	GEP	-	-
Tarea 1	60h	-	PC, Unity3D, HTC Vive, GitLab	D
Tarea 2	110h	Tarea 1	PC, Unity3D, HTC Vive, GitLab	D, I
Tarea 3	85h	Tareas 1 y 2	PC, Unity3D, HTC Vive, GitLab	D
Tarea 4	60h	Tarea 1	PC, Unity3D, HTC Vive, GitLab	D, I
Tarea 5	60h	Tareas 1, 2, 3 y 4	PC, Unity3D, HTC Vive, GitLab	E

Cuadro 8.1: Tabla resumen de la planificación del proyecto. Leyenda: DP - Director del Proyecto, I - Investigador, D - Desarrollador, E - Evaluador

8.2. Diagrama de Gantt

Como se pueden ver las figuras siguientes [8.1](#), [8.2](#), [8.3](#) (generadas mediante GanttPRO [23]), se han marcado con un color rojo todas aquellas tareas que pueden suponer un riesgo para el correcto desarrollo del proyecto (De esto se hablará en más profundidad más adelante).

Siguiendo los horarios establecidos para el desarrollo del proyecto, se contará con 5 días de margen para posibles problemas inesperados. Se considera que se trabajará en el proyecto todos los días entre 5 y 6 horas (incluyendo las 8 horas semanales que se estará en el CRV del ViRVIG). No obstante, se han contado menos horas hasta el día 9 de Marzo ya que hay menos trabajo por hacer hasta ese día aproximadamente.

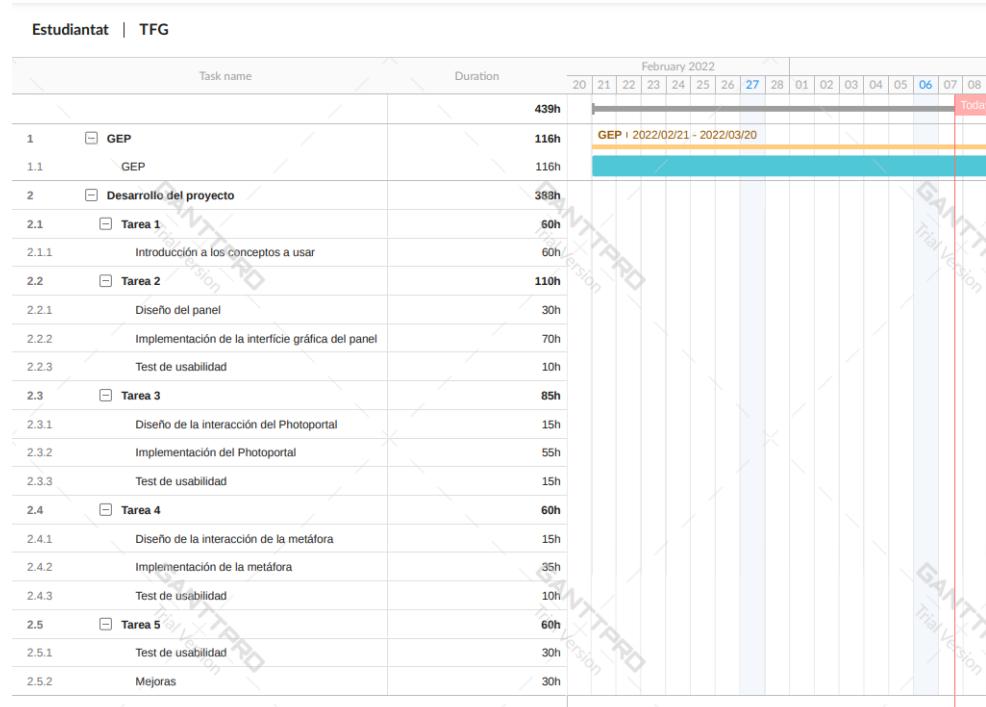


Figura 8.1: Diagrama de Gantt parte 1.

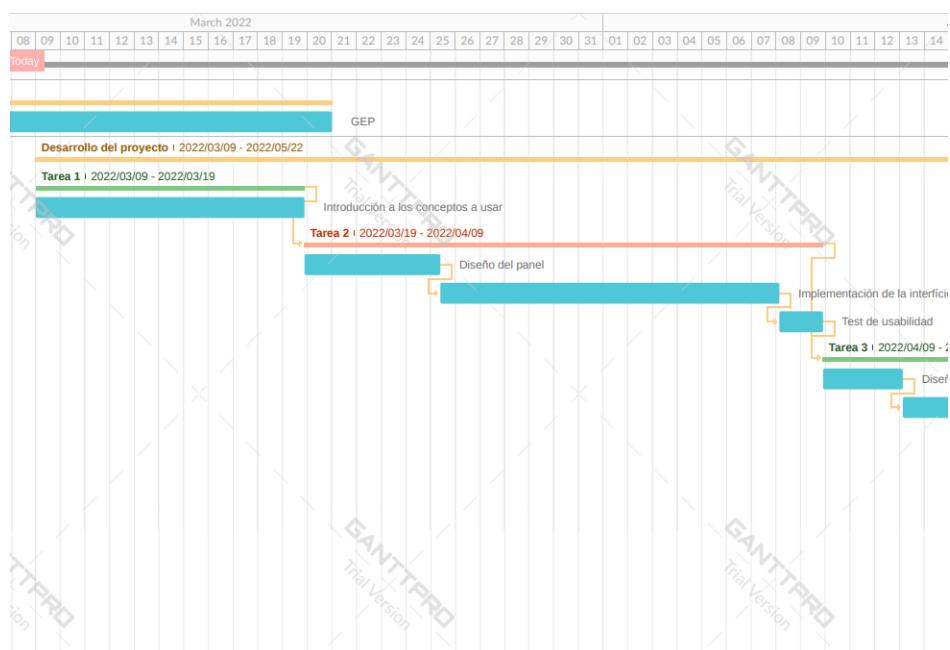


Figura 8.2: Diagrama de Gantt parte 2.

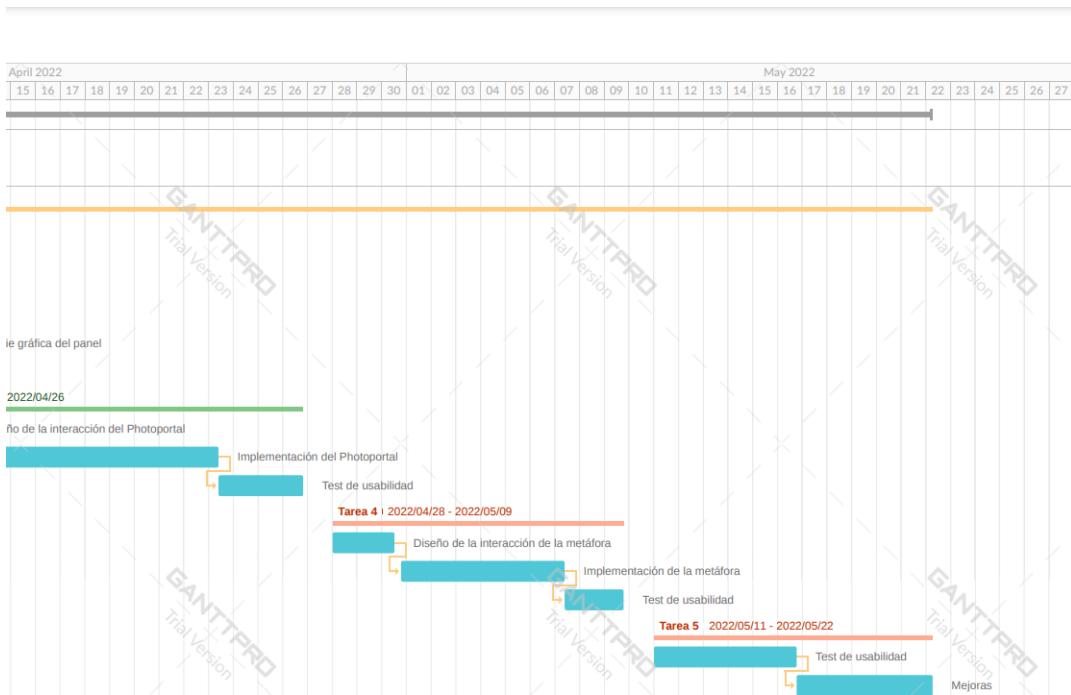


Figura 8.3: Diagrama de Gantt parte 3.

8.3. Gestión del riesgo

Algunas de las tareas suponen un riesgo para el correcto desarrollo del proyecto y se tiene que tener mucha precaución a la hora de hacerlas.

Para empezar, la tarea 2 puede considerarse de riesgo debido a la complejidad que tiene. Su implementación necesita de una interfaz de usuario totalmente nueva en la aplicación que sea cómoda y útil. Además, de esta tarea depende la tarea 3, por lo que es importante que se lleve a cabo correctamente para no arrastrar errores a otras partes del proyecto. Es por eso que se harán test de usuario para dar con la mejor versión posible antes de entrar en detalle con la siguiente tarea.

La tarea 4, supone un reto en cuanto a experiencia para el usuario ya que es la que puede dar a lugar más a problemas. Un mal diseño de esta implementación podría provocar mareos a los usuarios al agrandar el modelo o al transportarlos al interior del modelo agrandado, además de abrumarlos por ver el modelo demasiado grande delante suyo y provocarles una mala experiencia de usuario. Por suerte, la aplicación cuenta con la implementación testada de un transporte de un punto a otro que puede servir de referencia a la hora de implementar esta tarea. Al igual que para la tarea 2, se llevarán a cabo test de usuario para dar con la versión más adecuada para los usuarios.

Finalmente, los test de usuario también se consideran una tarea de riesgo, ya que es necesario que la aplicación cumpla con todos los requisitos de usabilidad antes de la inscripción al turno de lectura.

8.4. Presupuesto

8.4.1. Identificación y estimación de los costes

Recursos humanos

En esta sección del presupuesto se calcula el salario por hora de los diferentes componentes del equipo que se definieron para el proyecto. En la tabla 8.2 se detalla el salario de los 4 roles según el sondeo extraído de Indeed [30] y de talent [31].

Nombre	Salario (€/Hora)
Jefe de Proyecto	17,22
Investigador	11,33
Programador	13,85
Evaluador	26,15

Cuadro 8.2: Tabla de costes del personal

En la tabla A.1 figuran las partidas por tarea, teniendo en cuenta el coste del personal de la tabla 8.2 y de las tareas definidas con anterioridad. El coste total del personal incluyendo la cuota de la Seguridad Social (salario x 1,3) es de 11401,22 euros

Recursos Materiales

En esta sección del presupuesto se calcula el precio de los recursos materiales necesarios para llevar a cabo el proyecto que se definieron con anterioridad.

Como las empresas de software normalmente ofrecen tarifas de pago mensuales o anuales, optaremos por las suscripciones mensuales dado que la duración del proyecto es de 4 meses. Además, para este proyecto, se ha considerado que se necesitan como mínimo 2 HMD para poder probar correctamente la implementación de las metáforas de interacción entre varios usuarios, ya que se trata de una aplicación colaborativa. Finalmente, el precio del PC se ha extraído de un portal web donde se aproxima un valor medio para el ordenador dependiendo del rendimiento que se busca. Se ha considerado que el PC debe ser gaming (orientado a videojuegos)

Material	Unidades	Tiempo de uso (meses)	Precio unidad(€)
PC	1	4	1204 [36]
HTC Vive	2	4	949 [32]
Unity3D	1	4	150/mes [34]
Overleaf	1	4	14/mes [33]
GitLab	1	4	17,29/mes [35]

Cuadro 8.3: Tabla de costes de los recursos materiales

ya que debe funcionar correctamente con Unity y los dispositivos de RV, los cuales se utilizan en su mayoría para videojuegos.

Como se puede ver en la tabla 8.3, el coste de los recursos materiales sería de 181,29€ de costes mensuales y 2936,45€ de costes únicos. Esto nos deja un coste total en recursos materiales de 3661,61€.

Contingencias e imprevistos

Finalmente, se calculará un sobre-coste para cubrir posibles obstáculos e imprevistos. Estos imprevistos pueden deberse a demoras en el tiempo de implementación de alguna tarea, lo que conllevará un aumento de coste en los recursos humanos, o fallos en las herramientas utilizadas para el desarrollo del proyecto, lo que provocará un aumento en los recursos materiales. Con tal de tener todo imprevisto controlado se ha estimado un sobre-coste del 15 % del coste total.

Tipo	Coste(€)	Contingencias(€)
Recursos Humanos	11401,22	1710,18
Recursos Materiales	3661,61€	549,24
Total	15062,83	2259,42

Cuadro 8.4: Tabla de contingencias

Presupuesto inicial

Con los datos obtenidos en los apartados anteriores, ya podemos formalizar el presupuesto inicial del proyecto. El coste estimado total es de 17322,25€. En la tabla 8.5 se muestra el desglose del coste.

Tipo	Coste(€)
Recursos Humanos	11401,22
Recursos Materiales	3661,61€
Contingencias	2259,42€
Total	17322,25

Cuadro 8.5: Presupuesto inicial del proyecto

8.4.2. Control de gestión

En este apartado se describirán los mecanismos para controlar las desviaciones que puedan aparecer en relación con el presupuesto. Para ello, en las reuniones de supervisión y seguimiento será posible comprobar el tiempo real invertido en cada tarea, lo que nos permitirá calcular las desviaciones de coste. Además, se definirán el conjunto de indicadores numéricos que facilitarán el control.

- **Desviación del coste del personal por tarea:**

$$(\text{costeEsperado} - \text{costeReal}) \times \text{horasReales}$$
- **Desviación del precio por el uso de los recursos materiales:**

$$(\text{costeEsperado} - \text{costeReal}) \times \text{díasDeUsoReales}$$
- **Desviación en consumo del personal por tarea:**

$$(\text{horasEsperadas} - \text{horasReales}) \times \text{costeEsperado}$$
- **Desviación en consumo de los recursos materiales:**

$$(\text{díasDeUsoEsperados} - \text{díasDeUsoReales}) \times \text{costeEsperado}$$
- **Desviación total de los recursos humanos:**

$$(\text{desviaciónCostePersonal} + \text{desviaciónConsumoPersonal})$$
- **Desviación total de los recursos materiales:**

$$(\text{desviaciónPrecioRm} + \text{desviaciónConsumoRm})$$

8.5. Informe de sostenibilidad

Con tal de evaluar los objetivos de desarrollo sostenible (ODS) y las competencias en sostenibilidad del proyecto y mis conocimientos sobre ellos, se ha respondido a una encuesta que se realiza conjuntamente en 8 universidades del sistema universitario español. Dicha encuesta, incluye las 3 dimensiones de sostenibilidad: social, ambiental y económica y se considerará que el proyecto es sostenible si se tienen en cuenta todas ellas.

Tras realizar el cuestionario he podido extraer varias conclusiones. Durante la carrera, nos han ido dando pinceladas, en forma de competencias transversales, de los aspectos económicos, sociales y medioambientales de los productos y servicios TIC en el mundo. La mayoría de estas competencias se centraban en el impacto medioambiental, la fabricación de productos y las diferentes herramientas y componentes utilizados en este sector. Es por eso que he podido observar que tengo bastantes conocimientos en relación a las problemáticas sociales, económicas y/o ambientales de la sociedad actual, del impacto ambiental de los productos y servicios de mi ámbito profesional y de la gestión de recursos.

No obstante, he detectado que tengo bajos conocimientos en cuanto a las diferentes maneras de medir, describir y/o aproximar estos aspectos. Considero que esto se trata de un aspecto muy importante a tener en cuenta para el desarrollo de cualquier proyecto, ya que estos indicadores nos permiten mostrar crecimientos de interés en determinados productos [37] (dentro del ámbito social), el estado de la economía en un país [38] (dentro del ámbito económico) o los objetivos de desarrollo sostenible alcanzados para tomar futuras decisiones [39] (dentro del ámbito ambiental).

Es por esto que espero que este TFG me ayude a poder desarrollar un conocimiento más amplio en este aspecto.

A continuación, se llevará a cabo una reflexión sobre los impactos en las tres dimensiones de sostenibilidad respondiendo a las preguntas que figuran en la matriz de sostenibilidad de GEP.

8.5.1. Impacto ambiental

¿Has estimado el impacto ambiental que tendrá la realización del proyecto? ¿Te has planteado minimizar el impacto, por ejemplo, reutilizar recursos?

Considero que el proyecto no generará un impacto ambiental que necesite ser minimizado. Se trata de un proyecto que entra dentro de los estándares de consumo habituales al desarrollarse en un espacio de trabajo compartido [11] donde tiene lugar el desarrollo de varios proyectos al mismo tiempo.

¿Cómo se resuelve actualmente el problema que quieres abordar (estado del arte)? ¿En qué mejorará ambientalmente tu solución a las existentes?

Actualmente, el estudio de anatomía se hace mediante textos, atlas o disección de cadáveres. No obstante, el uso de cadáveres conlleva un alto coste y el uso que se le puede dar es muy corto. Además, la impresión de atlas y textos también conlleva un gasto de recursos materiales como el papel o la tinta. Con el uso de esta aplicación, sería posible ahorrarse este gasto dando un servicio más sostenible, ya que no sería necesario el uso de cadáveres ni impresiones, y reutilizable ya que los usuarios podrían utilizar la aplicación siempre que quieran.

Además, los conceptos utilizados en este proyecto pueden extrapolarse a otros ámbitos como el de la ingeniería, donde podrían simularse modelos de vehículos o maquinaria para inspeccionarlos en profundidad antes de comenzar con su creación, para así reducir el impacto ambiental que conlleva la creación de prototipos.

8.5.2. Impacto económico

¿Has estimado el coste de la realización del proyecto (recursos humanos y materiales)?

La estimación del coste en la realización del proyecto se puede apreciar con más detalle en el capítulo anterior titulado *Presupuesto*.

¿Cómo se resuelve actualmente el problema que quieres abordar (estado del arte)? ¿En qué mejorará económicamente tu solución a las existentes?

De la misma manera que como se ha comentado en el impacto ambiental, la utilización de esta aplicación conllevará un ahorro en el uso de recursos para la creación de textos y atlas y para el uso de cadáveres. Lo que abaratara los costes en la educación dentro de este ámbito. Es cierto que actualmente, los dispositivos de RV siguen siendo caros y el problema actual de abastecimiento de chips para dispositivos electrónicos no ayuda a la disminución del precio. Sin embargo, la esperanza de vida de un dispositivo de RV es de 2 a 3 años, por tanto el uso de RV nos permitiría ahorrar en los recursos mencionados durante este periodo de tiempo para, como mínimo, 1 estudiante.

8.5.3. Impacto social

¿Qué crees que te va a aportar a nivel personal la realización de este proyecto?

A nivel personal este proyecto me puede aportar un conocimiento más amplio en el uso y funcionamiento de herramientas de RV, herramientas colaborativas y de tratamiento de gráficos por ordenador más allá de los conocimientos básicos obtenidos durante la carrera. También, me permitirá ver de cerca como es trabajar en un entorno profesional orientado a la investigación. Por último, también me hace ver la importancia que tiene la informática y la gran utilidad que tienen los sistemas de RV para otros sectores.

¿Cómo se resuelve actualmente el problema que quieres abordar (estado del arte)? ¿En qué mejorará socialmente tu solución a las existentes?

Los métodos de educación en anatomía tradicionales, ya mencionados en los apartados anteriores, proveen una experiencia con un realismo bastante limitado, sobretodo los textos y atlas que son meramente teóricos. Por ejemplo, el color y la textura de las estructuras anatómicas en cadáveres difiere del de los pacientes vivos.

Con el uso de esta aplicación, los estudiantes podrán inspeccionar modelos virtuales del cuerpo humano de forma más realista, donde podrán moverse libremente alrededor del modelo e inspeccionar las partes mas internas de éste. También, podrán posicionar etiquetas en las diferentes partes del modelo que les servirán tanto para estudio como para poner a prueba sus conocimientos.

Dado que los estudiantes tendrán una mejor formación, habrán médicos, enfermeros y fisioterapeutas mejor cualificados lo que conllevará a un beneficio social de forma indirecta. Además, al tratarse de un proyecto

muy centrado en la investigación, futuras herramientas que utilicen este tipo de tecnología podrán verse beneficiadas independientemente de que estén enfocadas en el ámbito de la medicina o no, ya que las metáforas y funcionalidades que emplea se pueden extrapolar a cualquier otro contexto.

¿Existe una necesidad real del proyecto?

La existencia de herramientas de RV para ámbitos no meramente de entretenimiento significa un incentivo más para la investigación e innovación de los sistemas de RV, con el objetivo de ofrecer soluciones para el uso cotidiano de las personas.

Ya se ha hablado en apartados anteriores de los beneficios que puede provocar esta aplicación en concreto. Pero también las investigaciones e innovaciones que se lleven a cabo durante el desarrollo permiten perfeccionar esta tecnología y dar con nuevas metáforas y sistemas que faciliten y mejoren la calidad de vida de las personas, el aprendizaje, la investigación y la comunicación.

Bibliografía

- [1] Matt Cook, *Teaching and Learning with 3D Content*, [Consulta 24 febrero 2022] [En línea]. Disponible en: <https://library.harvard.edu/services-tools/teaching-and-learning-3d-content>
- [2] Qin, T.; Cook, M.; Courtney, M. *Exploring Chemistry with Wireless, PC-Less Portable Virtual Reality Laboratories*. J. Chem. Educ. 2021, 98, 521–529. [Google Scholar] [CrossRef]
- [3] D. Kaminska, T. Sapinski, S. Wiak, T. Tikk, R. E. Haamer, E. Avots, A. Helmi, C. Ozcinar and G. Anbarjafari *Virtual Reality and its application in education: survey*. información, 10(10):318, 2019.
- [4] Volbracht, S.; Shahrababaki, K.; Domik, G. y Fels, G.; (1996). *Perspective viewing, anaglyph stereo or shutter glass stereo?*. En: Proceedings IEEE Symposium on Visual Languages, pp 192 – 193
- [5] CyberPulse *Semi-Immersive Virtual Reality* [En línea] .[Consulta 24 Febrero 2022]. Disponible en: <https://www.youtube.com/watch?v=mcqXZI6VXtk&t=1s>
- [6] Cruz Neira, C.; Sandin, D. y Defanti, T.; Ken-Yon, R.; y Hart, J.; (1992). *The CAVE: Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE*. En: Proceedings of the 20th annual conference on Computer graphics and interactive techniques, SIGGRAPH 1993, pp. 135 - 142.
- [7] Cruz Neira, C.; Sandin, D. y Defanti, T.; (1993). *The CAVE: Audio Visual Experience Auto-matic Virtual Environment*. En: Communications of the ACM, Vol. 35(6), pp. 65-72
- [8] Zhenyi He, Ken Perlin *CollaboVR: A reconfigurable framework for multi-user to communicate in Virtual Reality*, 16 Dec 2019.
- [9] André Kunert, Alexandre Kulik, Stephan Beck, Bernd Froehlich *Photoportals: Shared references in space and time*, 15-19 Feb 2014
- [10] David Hernández Morales, *Inspecció de models anatòmics en un entorn immersiu col·laboratiu: aplicació en anatomia..* 2 de Julio de 2018
- [11] *Pagina web del ViRVIG* [En línea] .[Consulta 24 Febrero 2022]. Disponible en: <https://www.virvig.eu/>
- [12] Doug A. Bowman et al. *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004. I S BN: 0201758679.
- [13] Mazuryk, Tomasz, and Michael Gervautz. *Virtual reality-history, applications, technology and future*. (1996).
- [14] Oculus. *Oculus, Meta Quest* [Consulta 26 Febrero 2022][En línea] .Disponible en: <https://www.oculus.com/>
- [15] PlayStation Vr. *PlayStationVR* [En línea][Consulta 26 Febrero 2022] .Disponible en: <https://www.playstation.com/es-es/ps-vr/>
- [16] VIVE. *Discover Virtual Reality Beyond Your Imagination*. VIVE.[Consulta 26 Febrero 2022] [En línea] Disponible en: <https://www.vive.com/us/>.

- [17] bHamptic tracksuits. *bhaptic* [En línea] [Consulta 27 Febrero 2022]. Disponible en: <https://www.bhaptics.com/>
- [18] Haptx gloves. *haptx* [En línea] [Consulta 27 Febrero 2022] .Disponible en: <https://haptx.com/>
- [19] Lenis Alejandro, ¿Qué es el Metaverse y cómo funcionaría?, *hubspot* [En línea] 29 de Noviembre de 2021. [Consulta 27 Febrero 2022]. Disponible en: <https://blog.hubspot.es/marketing/que-es-metaverse>
- [20] Unity 3D. *Unity*[en línea] [Consulta 26 Febrero 2022] Disponible en: <https://unity3d.com/>.
- [21] Slack. *Where work happens — Slack* [En línea][Consulta 28 Febrero 2022].Disponible en: <https://slack.com/>
- [22] GitLab. *Iterate faster, innovate together— GitLab* [Consulta 28 Febrero 2022] [En línea]. Disponible en <https://about.gitlab.com/>
- [23] *GanttPRO* [En línea] [Consulta 5 de Marzo de 2022] .Disponible en: <https://app.ganttpro.com/>
- [24] Hackett, Matthew; Proctor, Michael. *Three-dimensional display technologies for anatomical education: a literature review.* *Journal of Science Education and Technology*, 2016, vol. 25, no 4, p. 641-654.
- [25] Azer, Samy A.; Azer, Sarah. *3D anatomy models and impact on learning: a review of the quality of the literature.* *Health professions education*, 2016, vol. 2, no 2, p. 80-98.
- [26] Ritter, Felix, et al. *Using a 3d puzzle as a metaphor for learning spatial relations.* En *Graphics Interface*. 2000. p. 171-178.
- [27] Fairén González, Marta, et al. *Virtual reality to teach anatomy.* En *Eurographics 2017: education papers*. European Association for Computer Graphics (Eurographics), 2017. p. 51-58.
- [28] Preim, Bernhard; Saalfeld, Patrick. *A survey of virtual human anatomy education systems.* *Computers and Graphics*, 2018, vol. 71, p. 132-153.
- [29] Schott, Danny, et al. *A vr/ar environment for multi-user liver anatomy education.* En *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2021. p. 296-305.
- [30] Indeed. *Ofertas de trabajo, bolsa de trabajo— Buscar empleo Indeed España* [Consulta 10 de Marzo 2022][En línea]. Disponible en: <https://es.indeed.com/?from=gnav-title-webapp>
- [31] Talent. *Talent* [Consulta 10 de Marzo] [En línea]. Disponibl en: <https://es.talent.com/>
- [32] Coste HTC Vive. *HTC Vive Pro Full Kit* [Consulta 10 de Marzo 2022] [En línea]. Disponible en: <https://www.vive.com/eu/product/vive-pro-full-kit/>
- [33] Coste Overleaf. *Plans and Pricing - Overleaf, Online Latex Editor* [Consulta 10 Marzo 2022] [En línea].Disponible en: <https://www.overleaf.com/user/subscription/plans>
- [34] Coste Unity. *Unity Pro Configure Plan* [Consulta 10 Marzo 2022] [En línea]. Disponible en: <https://store.unity.com/configure-plan/unity-pro>
- [35] Coste GitLab. *GitLab Pricing— GitLab* [Consulta 10 Marzo 2022] [En línea]. Disponible en: <https://about.gitlab.com/pricing/>
- [36] Coste PC. *Average cost of a gaming pc — The cost guys* [Consulta 10 Marzo 2022] [En línea]. Disponible en: <https://thecostguys.com/gaming/average-cost-of-a-gaming-pc#:~:text=A%20typical%20gaming%20PC%20will,pay%20as%20much%20as%20%242%2C000.>
- [37] Semrush .*SEO Social: la importancia de los indicadores sociales*[Consulta 11 Marzo de 2022] [En línea]. Disponible en: <https://es.semrush.com/blog/seo-social-importancia-indicadores-sociales/>

- [38] EBN. *La importancia de los indicadores economicos* [Consulta 11 Marzo de 2022] [En línea]. Disponible en: <https://www.ebnbanco.com/sinycon/blog/la-importancia-los-indicadores-economicos/#:~:text=Un%20indicador%20econ%C3%B3mico%20es%20un,pa%C3%ADses%20para%20tener%20m%C3%A1s%20perspectiva>.
- [39] Minam. *Indicadores ambientales* [Consulta 11 Marzo de 2022] [En línea]. Disponible en: <https://www.minam.gob.pe/investigacion/indicadores-ambientales/>
- [40] Gamarra Niño, Alejandro *Minimaps Personal Research*. [En línea]. [Consulta 24 Abril de 2022]. Disponible en: https://alejandro61299.github.io/Minimaps_Personal_Research/
- [41] CodeMonkey *How to make a simple minimap* [En línea]. [Consulta 24 Abril de 2022]. Disponible en: <https://www.youtube.com/watch?v=kWh0MJMihC0>
- [42] Unet. *Unity - Manual Multiplayer Overview*. [En línea]. [Consulta 2 Mayo de 2022]. Disponible en: <https://docs.unity3d.com/Manual/UNetOverview.html>
- [43] XML. *Introducción a XML*. [En línea]. [Consulta 6 Mayo de 2022]. Disponible en : https://developer.mozilla.org/es/docs/Web/XML/XML_introduction
- [44] LinkedList. *LinkedList; Clase*. [En línea]. [Consulta 6 Mayo de 2022]. Disponible en: <https://docs.microsoft.com/es-es/dotnet/api/system.collections.generic.linkedlist-1?view=net-6.0>
- [45] System.xml. *System.xml Espacio de nombres*. [En línea]. [Consulta 6 Mayo de 2022]. Disponible en:<https://docs.microsoft.com/es-es/dotnet/api/system.xml?view=net-6.0>

Apéndice A

Tabla de partidas por tarea

Nombre	Duración (h)	Recursos	Coste (€)	Coste con S.S (€)
GEP	118	-	2371,86	3083,41
Entrega 1	30	CP,I	856,5	1113,45
Entrega 2	40	CP	688,8	895,44
Entrega 3	30	CP	516,6	671,58
Entrega 4	18	CP	309,96	402,95
<i>Desarrollo del proyecto</i>	375	-	7352,85	8317,81
Tarea 1	60	-	1510,8	1964
(T1) Introducción a los conceptos a usar	60	I,P	1510,8	11964
Tarea 2	110	-	1986,4	1582,3
(T2) Diseño del panel	30	I,P	755,4	982
(T2) Implementación de la interfaz gráfica del panel	70	P	969,5	1260,4
(T2) Test de usabilidad	10	E	261,5	340
Tarea 3	85	-	1531,7	1991,21
(T3) Diseño de la interacción del photoportal	15	I,P	377,7	491
(T3) Implementación del photoportal	55	P	761,75	990,3
(T3) Test de usabilidad	15	E	392,25	509,93
Tarea 4	60	-	1123,95	1461,14
(T4) Diseño de la interacción de la metáfora	15	I,P	377,7	491
(T4) Implementación de la metáfora	35	P	484,75	630,17
(T4) Test de usabilidad	10	E	261,5	340
Tarea 5	60	-	1200	1560
(T5) Test de usabilidad	30	E	784,5	1019,85
(T5) Mejoras	30	P	415,5	540,15
Total RH	493	-	9724,71	11401,22

Cuadro A.1: Tabla de partidas por tarea. Leyenda: CP - Jefe de proyecto, I - Investigador, P - Programador, E - Evaluador