



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Museo de vestimentas del s. XVI en Realidad Virtual

16th century clothing museum in Virtual Reality

Cristian Rodríguez Navarro

La Laguna, 6 de septiembre de 2021

Dña. **Isabel Sánchez Berriel**, con N.I.F.42.885.838-S profesor Titular de Universidad adscrito al Departamento de Nombre del Departamento de la Universidad de La Laguna, como tutor

D. **Fernando Pérez Nava**, con N.I.F. 42.091.420-V profesor Titular de Universidad adscrito al Departamento de Nombre del Departamento de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

"Museo de vestimentas del s. XVI en Realidad Virtual"

ha sido realizada bajo su dirección por D. **Cristian Rodríguez Navarro**, con N.I.F. 79.080.964-H.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 6 de septiembre de 2021

Agradecimientos

En primer lugar quiero agradecer a mi tutora Isabel Sánchez Berriel por haber estado conmigo en este proyecto y haber sido de tanta ayuda, proporcionándome los materiales y la asistencia necesaria y por haber estado atenta de cualquier problema que pudiera surgir. También quiero agradecer a mi cotutor Fernando Pérez Nava por su colaboración para realizar este proyecto y por las horas dedicadas.

Agradecer también a Cecile Meier que ha proporcionado todos los modelos de los personajes con los que trabajo en este proyecto. Agradecer a Joshua García Expósito por haber proporcionado el modelo del museo. Agradecer también a Kevin Estévez Expósito y a Miguel Aurelio García González los cuales sentaron grandes precedentes y me allanaron el terreno. Por último agradecer a Carolina Méndez Hernández por haber prestado su voz para el desarrollo del proyecto.

Agradecer a todos los profesores que me han impartido clases y han hecho posible que haya llegado hasta aquí.

Por último dar gracias a mi madre, a mi padre y a mi hermana por ser una fuente constante de apoyo durante todos estos años y proporcionarme la seguridad y confianza necesaria para estar aquí hoy.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-
NoComercial-CompartirIgual 4.0 Internacional.

Resumen

El objetivo de este trabajo ha sido desarrollar una aplicación de Realidad Virtual en donde el usuario pueda visualizar las ropas de la población que habitaba una ciudad española del s. XVI como podía ser San Cristóbal de La Laguna. El desarrollo presta especial atención a cómo algunos de estos ropajes son afectados por el movimiento natural de un ser humano, como por ejemplo el caminar, los gestos que podemos hacer cuando hablamos o cuando cambiamos todo nuestro peso de una pierna a otra. Adicionalmente, cada personaje tendrá un panel de información en donde se enseñará un resumen de las características de sus vestidos y el estrato social al que representan. Toda esta experiencia será dentro de un modelo 3D del Museo Cayetano Gómez Felipe situado en San Cristóbal de La Laguna, haciendo uso de sus zonas interiores para exponer a los personajes.

Para el desarrollo de esta aplicación se ha usado Unity 3D, de manera complementaria, herramientas como Blender para la exportación de las animaciones, Marvelous Designer para la simulación de las ropas, el asset Vertex Animation Tools de Unity encargado de implementar las animaciones y los modelos de las vestimentas dentro de la aplicación y las gafas Oculus Quest 2, con el objetivo posterior de incluir el proyecto final en la tienda de Oculus.

Palabras clave: Realidad Virtual, Unity 3D, C#, Blender, Marvelous Designer, Oculus Quest 2, Personajes, Siglo XVI, San Cristóbal de La Laguna

Abstract

The main objective of this work have been developing a Virtual Reality application where the user can visualize the population clothing that lived on the 16th century in San Cristóbal de La Laguna, also it will be possible to see how these clothes are affected by the natural movement of a human being, as for example a walk, the gestures that we can make when we speak or when we shift all our weight from one leg to the other. Additionally, each character will have an information panel where it will be shown what is the history of that job, what were their objectives, how did they fit in society. . . All of this experience will be inside a 3D model of the Museum Cayetano Gómez Felipe located in San Cristóbal de La Laguna, making use of its interior areas to expose the characters.

For the development of this application Unity 3D has been used, also tools such as Blender for the exportation of the animations, Marvelous Designer for the simulation of the clothes, Unity Vertex Animation Tools asset which has been in charge of implementing the animation and the models of the characters to the application and Oculus Quest 2 headset, with the posterior objective of including the final project in the Oculus Store.

Keywords: Virtual Reality, Unity 3D, C#, Blender, Marvelous Designer, Oculus Quest 2, Character, 16th century, San Cristóbal de La Laguna

Índice general

Índice de Figuras	III
1. Introducción	1
1.1. Antecedentes y estado actual	3
1.2. Objetivos	6
2. Herramientas y Tecnologías	7
2.1. Oculus Quest 2	7
2.2. Unity (v2019.4.9f1)	8
2.2.1. C#	9
2.2.2. Paquetes de Realidad Virtual	9
2.2.3. Vertex Animation Tools (v3.41.0)	10
2.3. Marvelous Designer 9 (v5.1.477.28705)	11
2.4. Blender (v2.92.0)	11
3. Desarrollo	13
3.1. Preparación inicial	13
3.2. Paquetes de Realidad Virtual	14
3.2.1. Oculus Integration Package	14
3.2.2. XR Interaction Toolkit (v0.10.0)	15
3.2.3. Interacción	18
3.3. Personajes	19
3.3.1. Modelos femeninos	22

3.3.2. Modelos masculinos	26
3.4. Museo	32
3.5. Iluminación	34
3.6. Clases implementadas	37
3.7. Aplicación	40
3.7.1. Ejecución	40
3.7.2. Información	45
3.7.3. Audio	46
4. Dificultades e investigación	48
4.1. Paquetes de Realidad Virtual	48
4.2. Modelos de los personajes	49
4.3. Colliders	52
4.4. Rendimiento	53
5. Conclusiones y líneas futuras	55
5.1. Conclusiones	55
5.2. Líneas de trabajo futuro	57
6. Summary and Conclusions	59
6.1. Conclusions	59
6.2. Future Lines	61
7. Presupuesto	63
7.1. Presupuesto	63
7.2. Justificación	63
8. Glosario	65

Índice de Figuras

1.1. El primer plano de La Laguna, obra de Leonardo Torriani y extraído del libro publicado por el Colegio Oficial de Arquitectos de Canarias [1]	2
1.2. Vista del Museo de Valentino Garavani desde el interior	5
1.3. Vista del Museo Virtual de Brooklyn desde el interior	5
2.1. Controladores de las gafas Oculus Quest 2 [8]	8
3.1. Modelo del jugador	16
3.2. Rayo interactuador predeterminado	18
3.3. Rayo interactuador haciendo contacto con un botón de un canvas	19
3.4. Modelo de un cortesano de San Cristóbal de La Laguna del siglo XVI en Marvelous Designer	21
3.5. Resultado del modelo del cortesano de San Cristóbal de La Laguna del siglo XVI en Unity	22
3.6. Modelo de una cortesana en Unity	23
3.7. Diferentes modelos de la mujer común	24
3.8. Modelo de una mujer principal en Unity	25
3.9. Modelo de una mujer humilde en Unity	26
3.10 Modelo de un cortesano en Unity	27
3.11 Modelo A de un soldado en Unity	28
3.12 Modelo B de un soldado en Unity	28

3.13	Modelo médico en Unity	29
3.14	Modelo de un monje franciscano en Unity	30
3.15	Modelo de un monje dominico en Unity	31
3.16	Modelo A de un clérigo en Unity	32
3.17	Modelo B de un clérigo en Unity	32
3.18	Réplica en 3D del Museo Cayetano Gómez Felipe realizada por Joshua García Expósito	33
3.19	Ejemplo de iluminación en tiempo real [14]	35
3.20	Ejemplo de iluminación horneada [14]	35
3.21	Ejemplo de luz diurna con luz precomputada [14]	36
3.22	Ejemplo de luz nocturna con luz precomputada [14]	36
3.23	Skybox usado en el proyecto	37
3.24	Jerarquía de clases de la entidad jugador	38
3.25	Jerarquía de clases de la entidad guía	39
3.26	Primera vista obtenida al comenzar la aplicación	40
3.27	Cartel de información	41
3.28	Primer grupo de modelos	42
3.29	Guía avanzando de un grupo de modelos al siguiente	43
3.30	Escena final	44
3.31	Grupo de modelos antes de cambiar	45
3.32	Grupo de modelos después de cambiar	45
3.33	Podio antes de desplazar a través de la información	46
3.34	Podio después de haber desplazado	46
4.1.	Ejemplo del número de vértices de un modelo	50

Capítulo 1

Introducción

San Cristóbal de la Laguna fue fundada en 1497, los primeros colonos, la mayoría soldados, no recibieron ninguna parcela definida. La zona urbana no fortificada fue considerada como un espacio público en el que cada uno podía construir. De esta forma, los colonos fueron construyendo pequeñas casas alrededor de la Iglesia de la Concepción, en el mayor desorden, sin plano general, en la ciudad alta.

Esta situación fue regularizada en 1502 mediante una planificación basada en los planos que Leonardo Da Vinci había realizado para la ciudad de Imole y que fue adaptado por el Adelantado para la zona comprendida entre su residencia oficial y la Iglesia (Ver Figura 1.1). Largas calles principales enlazaban espacios públicos abiertos y formaban una cuadrícula sobre la que se insertaban las calles más pequeñas. La ciudad baja así creada se extendió rápidamente, atrayendo a las clases dirigentes de la isla y en 1515 ya contaba con más de mil habitantes. Fue la primera ciudad colonial sin murallas que conserva prácticamente intacto su trazado original del siglo XV, que sirvió de modelo para muchas ciudades coloniales.



Figura 1.1: El primer plano de La Laguna, obra de Leonardo Torriani y extraído del libro publicado por el Colegio Oficial de Arquitectos de Canarias [1]

San Cristóbal de La Laguna conservó esta posición política, religiosa y comercial dominante durante los siglos XVII y XVIII. La prosperidad obtenida se refleja en los edificios construidos en esta época. Fue declarada Bien Cultural y Patrimonio de la Humanidad por la UNESCO en 1999.

Gracias a toda esta historia, La Laguna aporta a todos sus visitantes un ambiente y una experiencia similar a la que podrían haber tenido los habitantes en sus inicios, siendo uno de los destinos turísticos más visitados de Tenerife.

Esto nos lleva al siguiente planteamiento, ¿Qué podríamos hacer para compartir el patrimonio de La Laguna con todo el mundo?, ¿Cómo podríamos llevar la experiencia de conocer un poco mejor nuestra ciudad a las personas en el otro extremo del planeta de la manera más inmersiva posible sin que tengan que trasladarse miles de kilómetros?, aquí es donde entra en juego el núcleo de nuestro TFG, la Realidad Virtual y las gafas

1.1. Antecedentes y estado actual

Este TFG tiene como intención promover virtualmente el patrimonio de la ciudad de La Laguna y en esta misma línea podemos encontrar otros Trabajos de Fin de Grado relacionados:

- *Realidad virtual en San Cristóbal de La Laguna Patrimonio Histórico* [2]: El objetivo de este TFG fue obtener una aplicación móvil de Realidad Virtual que contribuyera a la difusión de la historia y el patrimonio cultural de la ciudad de San Cristóbal de La Laguna. Mediante ésta, el usuario final podría verse inmerso en un entorno 3D que simulara la Plaza del Adelantado, pudiendo además moverse libremente por la misma. Para esto se usó la extensión de ProBuilder de Unity para crear el modelo inicial de cada edificio y para terminar cada uno se usó Blender. La plataforma para la que se desarrolló fue Samsung VR Gear.
- *Diseño y modelado de personajes para la reconstrucción histórica virtual de La Laguna en el Siglo XVI* [3]: Se tuvo como objetivo enriquecer el proyecto anterior añadiendo diferentes personajes ataviados con vestimentas de época, con la finalidad de proveer una mejor experiencia. Para el desarrollo de dichos personajes se usaron programas como Adobe Fuse, Mixamo, Blender y Marvelous Designer. En este TFG se aborda el modelado de personajes y vestimenta. Diferentes alternativas fueron barajadas para su desarrollo, como pudieron ser Blender, 3D Studio Max, ZBrush, etc. El problema de estos es que son programas de modelado en 3D genérico y hace que sea complicado generar dichos modelos e implementar las animaciones

de las vestimentas. Se decidieron por Marvelous Designer que es un programa específico usado para la industria del diseño y del entretenimiento. Este programa crea casi automáticamente la ropa a partir de los dibujos de los patrones de la ropa y además también anima las vestimentas ya que cada tela viene con características predefinidas que pueden ser editadas como el peso, elasticidad, etc. Además se ayudaron de Blender para exportar todos estos modelos a Unity. La plataforma utilizada en este caso fue aplicación de escritorio.

- *Aplicación Web de una casa lagunera del s.XVI* [4]: El objetivo de este TFG fue producir una aplicación que simulara de forma realista como era una casa doméstica lagunera del s.XVI. Se realizó un estudio de los estilos constructivos de la época en La Laguna en dicho siglo y se reprodujo una vivienda típica con el software Blender. El soporte elegido para su difusión fue una aplicación web utilizando WebGL obteniéndose un espacio navegable, permitiendo así al usuario moverse libremente por dentro de la casa, siendo capaz de observar las diferentes habitaciones y costumbres de la época.

La experiencia adquirida en los trabajos mencionados en los puntos anteriores fueron desarrollados posteriormente en la ponencia **AN INTERACTIVE 3D APPLICATION OF A HOUSE FROM THE XVI CENTURY IN SAN CRISTÓBAL DE LAGUNA AS A CASE STUDY FOR THE DISSEMINATION OF CULTURAL HERITAGE** [5] presentado en el congreso Arqueológica 2.0. En este escenario también se incluyeron modelos de personajes ataviados según los usos del momento.

En cuanto a museos de vestimenta en RV, existen muy pocos actualmente, como ejemplos tenemos:

- **Valentino Garavani Museum** [6]: Valentino Garavani es un diseñador de ropa el cual ha creado su propio museo virtual en el cual

te puedes mover libremente y admirar sus diseños desde el propio navegador o descargando una pequeña aplicación (Ver Figura 1.2).



Figura 1.2: Vista del Museo de Valentino Garavani desde el interior

- **Museo de Brooklyn [7]:** Este museo ofrece una visita en primera persona sobre los trajes y las vestimentas usadas en las conocidas series Gambito de dama y La Corona de Netflix, además de mostrarnos objetos relacionados con las temáticas de estas series. Todo ha sido seleccionado, organizado y presentado por Mathew Yokobsky (Ver Figura 1.3).



Figura 1.3: Vista del Museo Virtual de Brooklyn desde el interior

1.2. Objetivos

Este TFG trata de integrar en Unity los modelos de vestimenta del s.XVI comunes en San Cristóbal de La Laguna con sus correspondientes animaciones, haciendo uso de Blender para el modelo de los personajes y de Marvelous Designer para las vestimentas y animaciones.

El objetivo final es desarrollar una aplicación en donde se una el modelo digital del Museo Cayetano Gómez Felipe y dichos modelos de personajes para así poder ofrecer la experiencia de una visita en un museo en Realidad Virtual de las vestimentas del s.XVI. Para realizar este proyecto se han establecido los siguientes objetivos:

- Crear una aplicación en Unity y aprender a configurar los paquetes necesarios para la Realidad Virtual con Oculus Quest 2
- Establecer las interacciones y movimiento del usuario en el entorno
- Añadir los modelos de personajes con sus ropas y animaciones
- Establecer la interacción entre el usuario y los personajes
- Elegir dónde y cómo se reproducirán las animaciones
- Añadir la información de sus vestimentas
- Añadir el modelo del museo
- Instalar la iluminación necesaria en dicho museo
- Preparar una guía para la visita del museo

Capítulo 2

Herramientas y Tecnologías

2.1. Oculus Quest 2

Para el desarrollo de este proyecto se han usado las gafas de realidad virtual Oculus Quest 2, en este caso la versión de 256 GB de memoria. Este nuevo modelo de gafas posee 6 GB de RAM y el nuevo procesador Qualcomm® Snapdragon™ XR2 que aseguran un rendimiento óptimo y gracias a su compatibilidad con los ordenadores, puede ser conectado mediante USB o Wi-Fi y ser usado para desarrollo con paquetes de software de realidad virtual compatibles con Oculus. Adicionalmente este modelo tiene seguimiento de manos gracias a las cámaras delanteras, permitiendo controlar el menú sin los controladores (Ver Figura 2.1) e incluso ser usado en aplicaciones compatibles con esta funcionalidad.

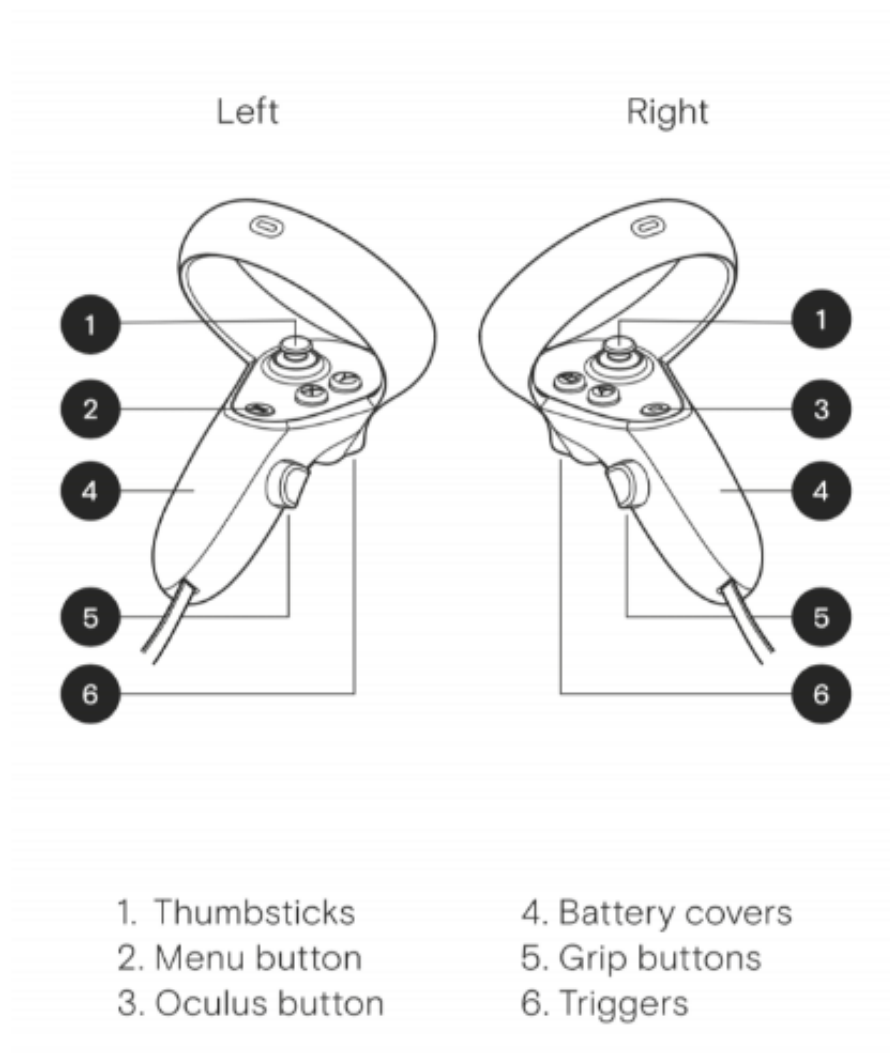


Figura 2.1: Controladores de las gafas Oculus Quest 2 [8]

2.2. Unity (v2019.4.9f1)

Unity es uno de los motores de videojuegos más conocido en el mundo y gracias a su reputación, a la sencillez para crear aplicaciones multiplataformas y a su compatibilidad con Oculus, ha sido el motor elegido para desarrollar este proyecto. Este motor posee una gran cantidad de paquetes auxiliares los cuales simplifican el proceso de desarrollo de un proyecto y en este caso, se han usado varios paquetes que implementan funcionalidades como la configuración básica necesaria de la aplicación para conectarte con las gafas de realidad virtual y usarlas o la importación

e implementación de las animaciones de los modelos de los personajes.

2.2.1. C#

C# fue creado como resultado de una mezcla de C y C++. Es un lenguaje de programación multiparadigma desarrollado y estandarizado por Microsoft y es el usado para programar en Unity.

2.2.2. Paquetes de Realidad Virtual

XR Plugin Management (v3.2.16)

Es un paquete que proporciona un manejo simple de plug-ins de Realidad Extendida (RE). Maneja y proporciona ayuda con la carga, inicialización, opciones y compilación de plug-ins de RE.

Oculus XR Plugin (v1.4.3)

Este paquete ofrece compatibilidad con gafas y controladores de dispositivos Oculus. Permite construir aplicaciones para las gafas Rift, Rift S, Quest, y Quest 2.

No es necesario usar este paquete con el XR Plugin Management pero integrarlo simplifica el uso de este y otros paquetes pertenecientes a Oculus.

XR Interaction Toolkit (v0.10.0)

El paquete XR Interaction Toolkit es un sistema de interacción de alto nivel basado en componentes que facilita el desarrollo de la interacción en aplicaciones RE en Unity. Proporciona un marco que hace que las interacciones 3D y UI estén disponibles desde los eventos de entrada de

Unity. El núcleo de este sistema es un conjunto de componentes base Interactuador e Interaccionable y un organizador de interacciones que une estos dos tipos de componentes. También contiene componentes auxiliares que se pueden utilizar para ampliar la funcionalidad, dibujar elementos visuales y enganchar sus propios eventos de interacción.

2.2.3. Vertex Animation Tools (v3.41.0)

Vertex Animation Tools (VAT) 3.0 [9] es un robusto conjunto de herramientas multiplataforma para importar y reproducir rápidamente cualquier tipo de animación de vértice. Este paquete incluye exportadores de animación de vértices personalizados de 3ds Max y Cinema4d, dos pares independientes de reproductores en tiempo de ejecución y su activo correspondiente que almacena datos de animación. Permite importar animaciones y reducir la carga que sufriría el dispositivo al hacerlo sin la ayuda de este paquete.

En los trabajos que precedieron a este se usó el formato alembic para incluir las animaciones de las vestimentas. Unity no proporciona soporte para este formato en dispositivos de RV, por lo que se hace necesaria el uso de esta herramienta.

Las animaciones generadas con Marvelous Designer no se pueden exportar directamente a un proyecto de RV sin recibir un tratamiento previo, es por eso que haciendo uso de Blender, es posible importar mediante VAT las animaciones a Unity.

A través de una secuencia de objetos, VAT es capaz de crear una animación, esto se debe a que genera de manera automática el movimiento

de un objeto desde un vértice de posición hasta el siguiente, además configurando algunos parámetros es capaz de crear animaciones fluidas, esto lo que permite es reducir la cantidad de objetos introducidos en Unity y en escena, optimizando así la ejecución.

2.3. Marvelous Designer 9 (v5.1.477.28705)

Marvelous Designer es un software 3D popular utilizado por muchos artistas para crear ropa 3D dinámica para videojuegos, arte 3D, animaciones 3D, experiencias de realidad virtual, etc.

Este programa se especializa en el diseño de ropa a través de patrones y gracias a una interfaz sencilla, permite coser, ajustar las ropas a los avatares y hacer simulaciones realistas del movimiento de los tejidos. Además, estas simulaciones pueden ser exportadas a otros softwares.

Se ha decidido usar debido a su gran utilidad, a su sencilla interfaz, a su compatibilidad con otros softwares y al gran repertorio de documentación oficial y creada por la comunidad. Todas estas características han permitido que este programa tenga curva de aprendizaje allanada en comparación con softwares similares, permitiendo optimizar nuestro tiempo de desarrollo.

2.4. Blender (v2.92.0)

Blender se puede utilizar para crear visualizaciones en 3D, como imágenes fijas, animaciones en 3D, tomas VFX y edición de video. Es una aplicación multiplataforma que se ejecuta en sistemas Linux, macOS y Windows. También tiene requisitos de memoria y almacenamiento relati-

vamente pequeños en comparación con otras suites de creación 3D. Su interfaz utiliza OpenGL y proporciona una experiencia consistente en todos los equipos y plataformas compatibles.

Como se menciona anteriormente, es necesario un tratamiento previo para importar las animaciones a Unity. Se optó por usar este software debido a la gran cantidad de funcionalidades que posee, facilitando este proceso sobremanera, además, al tener Blender tanto recorrido en el mundo del modelado 3D, la cantidad de documentación disponible era más que suficiente para los procesos que realizaríamos con este.

Capítulo 3

Desarrollo

3.1. Preparación inicial

En mi formación inicial, para los elementos básicos de Unity ya había hecho un curso en Udemy previamente y por otro lado para la Realidad Virtual, la asignatura de Interfaces Inteligentes fue suficiente como para ponerme en contacto y aprender lo necesario para empezar la investigación de los diferentes paquetes de RV para las Oculus Quest 2 desde el primer día.

Respecto a Blender, Marvelous Designer y el asset Vertex Animation Tools, para poder importar los modelos a Unity, seguí una guía que fue proporcionada por el grupo de proyecto de investigación "Reconstrucción Histórica Virtual de San Cristóbal de La Laguna". El documento recoge todos los avances que se han ido alcanzando en la evolución del proyecto respecto al uso de vestimentas animadas en una aplicación de RV. El proceso describe el flujo de trabajo desde el principio en Marvelous Designer pasando por Blender y terminando en Unity con el asset de VAT. Más allá de buscar información sobre estos softwares y ver algunos tutoriales para comprender su funcionamiento más básico, no me fue necesaria una

preparación inicial exhaustiva.

Finalizando con la preparación de tecnologías del proyecto cabe mencionar la configuración de las gafas Oculus Quest 2. Para poder desarrollar con ellas es necesarias conectarlas a una aplicación de móvil de Oculus y descargar los drivers oficiales, una vez hecho, ya estaba listo para empezar.

3.2. Paquetes de Realidad Virtual

En un principio, la idea era trabajar con el paquete de Updated Controller Interaction [10], este paquete fue actualizado por última vez en 2018 y estaba preparado para usarse principalmente con las gafas Gear VR, además era necesario usar también el paquete de Oculus Utilities For Unity el cual está obsoleto, es por eso que las primeras semanas fueron dedicadas a intentar configurar la Realidad Virtual con las gafas Oculus Quest 2 con estos paquetes. Sin embargo no llegó a ser posible y se optó por empezar a investigar otros paquetes de RV para las gafas.

3.2.1. Oculus Integration Package

Buscando paquetes oficiales de Oculus, se encontró este. Oculus Integration Package [11] ofrece soporte avanzado de renderizado, redes sociales, plataformas, audio y desarrollo de Avatares para dispositivos Oculus VR y algunos dispositivos compatibles con Open VR.

De todas las funcionalidades ofrece la integración de Oculus, las más necesarias para el inicio del proyecto fueron estas:

- **Sample Framework:** esta carpeta contiene ejemplos que demuestran los conceptos básicos de la integración de Unity. Se utilizó el Unity

Sample Framework para obtener información sobre los ejemplos y los conceptos que demuestran.

- VR: esta carpeta contiene las utilidades de Oculus VR, un conjunto de scripts y prefabricados para permitir el desarrollo de VR. Las Utilidades de Oculus para Unity se usaron para obtener más información.

Esta vía de trabajo se descartó debido a la imposibilidad de compatibilizarlo con las gafas RV con las que se trabaja. A esta conclusión se llegó después de realizar pruebas con diferentes versiones de Unity, paquetes auxiliares, tutoriales realizados por la comunidad y guías oficiales.

3.2.2. XR Interaction Toolkit (v0.10.0)

El análisis de las herramientas de RV llevó al XR Interaction Toolkit [12]. Este kit proporciona un conjunto de componentes que permiten a los usuarios crear experiencias interactivas e inmersivas de forma rápida y sencilla. El objetivo de este paquete es permitir interacciones comunes de realidad aumentada (RA) y realidad virtual (RV) sin la necesidad de escribir código, y al mismo tiempo hacer que el sistema sea extensible para los desarrolladores que desean crear sus propias interacciones.

Este paquete fue finalmente con el que se pudo iniciar y avanzar en el desarrollo del proyecto. La configuración fue muy sencilla y existe bastante documentación y tutoriales sobre como preparar y configurar la aplicación de Unity para poner en funcionamiento la aplicación de RV en las gafas.

Jugador

El jugador (Ver Figura 3.1) es un conjunto de objetos y scripts que ya vienen creados por el paquete de XR Interaction Toolkit y a los cuales

hay que configurar algunas propiedades para adaptarlo a los objetivos del proyecto. De manera predeterminada el jugador solo será capaz de mirar a través de las gafas el mundo virtual, pudiendo rotar sobre si mismo, es por esto que el movimiento y la interacción deben ser implementadas por el desarrollador.

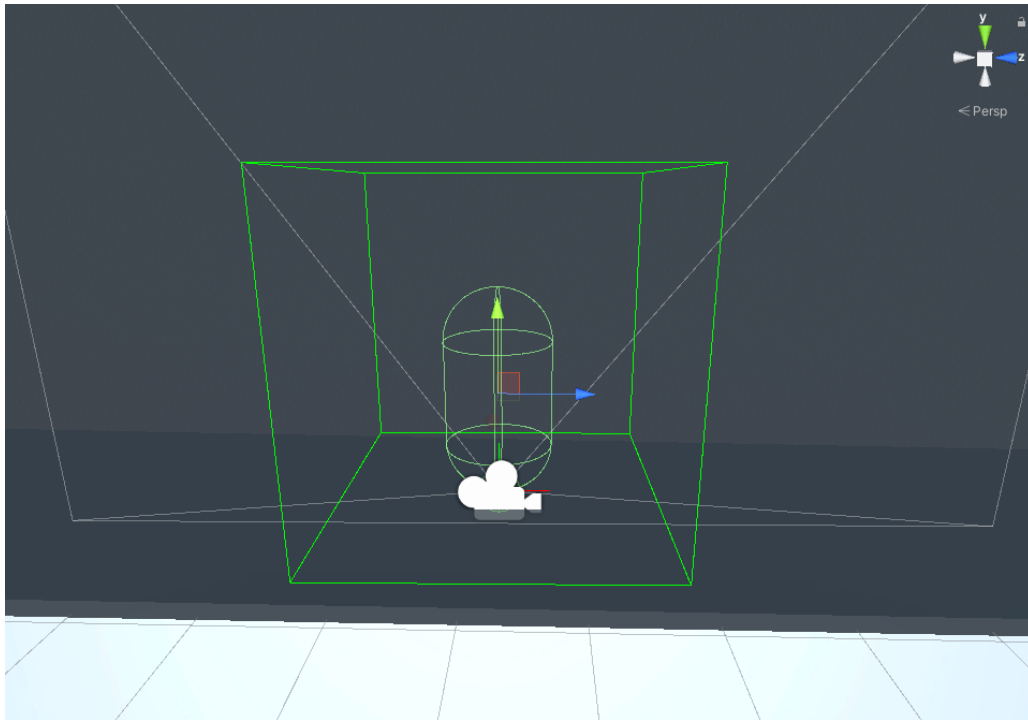


Figura 3.1: Modelo del jugador

Movimiento

Para moverse, el jugador debe usar el botón número uno de la Figura 2.1 del mando derecho. Este movimiento podía ser causante de mareos o desestabilizaciones, para evitar ambos se han realizado varias pruebas.

Se ha intentado establecer una velocidad de movimiento relativamente baja, intentado imitar la velocidad a la que caminamos en la vida real. En las pruebas realizadas, si la velocidad era demasiado alta, la sensación de estabilidad se reducía de manera casi proporcional a la rapidez. Es por esto, que a menos velocidad, el jugador siente mayor control sobre

su estabilización, por lo tanto, para elegir la velocidad que finalmente se ha usado, se ha decidido usar un punto medio, en donde el jugador no se sintiera incómodo ni en donde la velocidad de movimiento fuera demasiado lenta.

En lo que concierne al mareo, al no existir una animación de caminar en donde la cámara tenga un vaivén de movimiento, no existen balanceos de la visión, es por esto que la altura de nuestro personaje se ajusta a la altura del jugador en la vida real constantemente, esto permite que no haya movimientos indeseados de la vista del jugador, reduciendo así la posibilidad de mareo.

Por último se tuvo que hacer un pequeño ajuste al jugador. Cuando en el mundo real existen paredes, es físicamente imposible atravesarlas, sin embargo en el mundo virtual sucede lo contrario, si nos acercamos a una pared y no tenemos un colisionador, seremos capaces de pasar a través de ella como si no existiera.

Por esto se ha añadido el componente RigidBody al jugador y un Mesh Collider a las paredes o cualquier estructura que no deba ser atravesada, permitiendo así al jugador moverse en este mundo virtual de la manera más realista posible. Estos componentes también nos ayudan en caso de que si el jugador se acercara a una pared y físicamente intentara atravesarla caminando a través de ella, la aplicación desplazaría de manera automática su posición dentro del juego y le dejaría cerca de esa pared.

3.2.3. Interacción

La interacción es una funcionalidad integrada en el paquete de RV y solo es necesario añadir ajustes como distancia, grosor del rayo de interacción, color, etc. El resultado de todas las configuraciones nos deja con la Figura 3.2 y 3.3

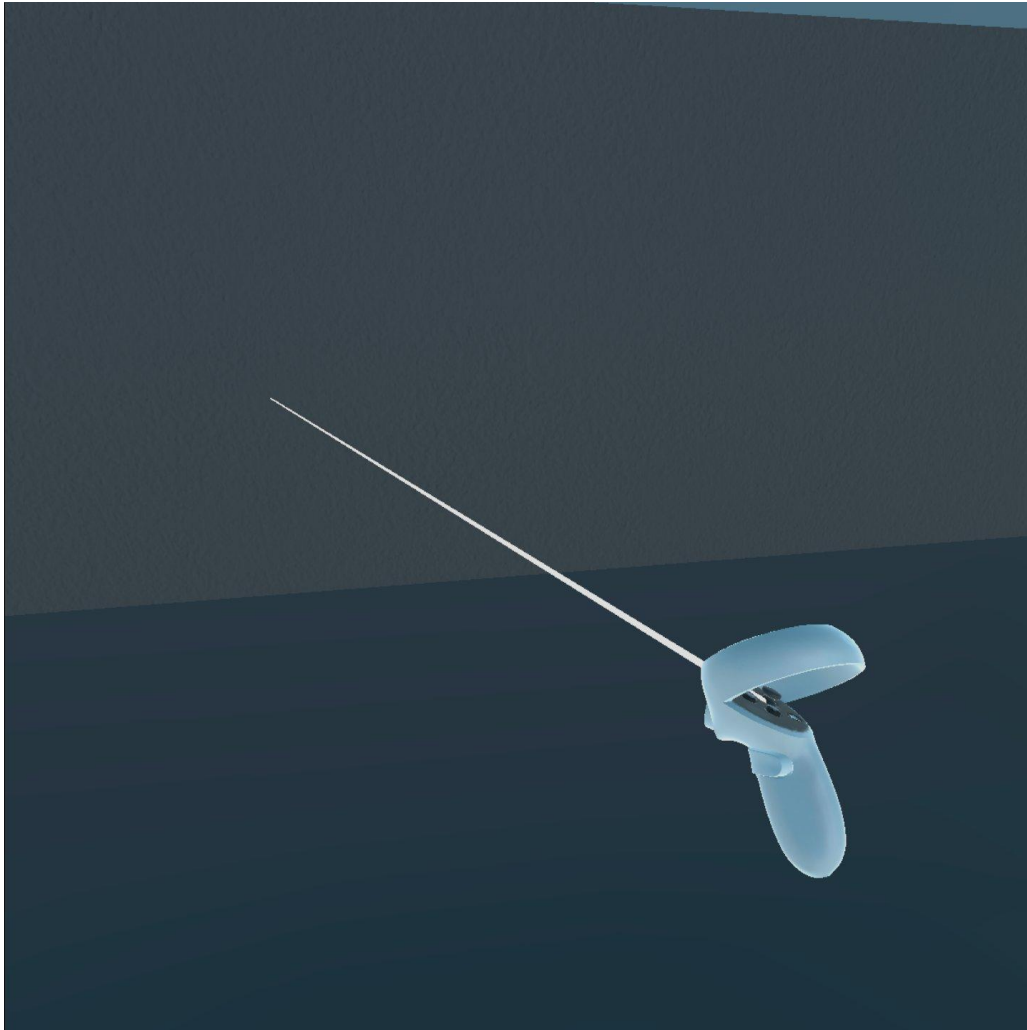


Figura 3.2: Rayo interactuador predeterminado



Figura 3.3: Rayo interactuador haciendo contacto con un botón de un canvas

3.3. Personajes

Este TFG usa modelos de otros proyectos, en este caso los personajes fueron proporcionados por el equipo del proyecto Reconstrucción Histórica Virtual de San Cristóbal de La Laguna. Cada personaje representa un tipo social en el contexto en el que se desarrolla la reconstrucción histórica de San Cristóbal de La Laguna en el s. XVI. Los modelos se diseñaron siguiendo las descripciones de la moda española de la época [13] y han sido elaborados con las herramientas de Marvelous Designer y Blender por Cecile Meier ya que cada modelo necesitaba un tratamiento previo

antes de poder ser importado a Unity mediante VAT.

El objetivo era tener una exposición de dichos modelos en donde cada uno tuviera una animación propia de movimiento como por ejemplo hablando, caminando o simplemente estando sentado, además de esto cada uno tendría un panel cerca en donde se mostraría información sobre su vestimenta, permitiendo así al usuario aprender poco a poco, modelo a modelo sobre los usos en la moda de la época en San Cristóbal de La Laguna. Para conseguirlo cada modelo debía pasar por Marvelous Designer, Blender y finalmente ser importado mediante el asset Vertex Animation Tools a Unity.

En el primer paso, haciendo uso de Marvelous Designer se simulaba el movimiento de la ropa del personaje con la animación del modelo (Ver Figura 3.4) y una vez calculado se exporta la animación de la ropa a Blender.



Figura 3.4: Modelo de un cortesano de San Cristóbal de La Laguna del siglo XVI en Marvelous Designer

Una vez en Blender (Ver Figura 3.5) se une el modelo del personaje y la animación de la ropa obtenida de Marvelous Designer exportándose el conjunto vestimenta-avatar animado. Por último, gracias a VAT se importa el personaje con su secuencia de modelos obj correspondientes a la animación a Unity. Este paquete además facilita la reproducción eficiente de las animaciones en aplicaciones de RV, WebGL, etc, salvando así las carencias de Unity con el formato alembic para estas plataformas.



Figura 3.5: Resultado del modelo del cortesano de San Cristóbal de La Laguna del siglo XVI en Unity

Por último, a cada personaje se le ha añadido un colisionador para que el jugador no sea capaz de atravesarlo, aportando así mayor realismo a la aplicación.

3.3.1. Modelos femeninos

A continuación describiremos los modelos femeninos utilizados en la aplicación

Cortesananas

Las cortesananas (Ver Figura 3.6) usaban vestidos lujosos, en donde podíamos encontrar 3 tipos:

- Saya
- Vasquiña, jubón, saboyana
- Vasquiña, jubón, ropa

Usaban cartón pecho en el cuerpo y verdugado debajo de la falda. Las mangas podían ser redondas o de puntas. Además no sacan el brazo por el puño sino por las cortadas.



Figura 3.6: Modelo de una cortesana en Unity

Mujer común

La mujer común (Ver Figura 3.7) usaba vasquiña sin verdugado debajo, y un jubón o un cuerpo. Por otro lado sus tejidos y adornos no eran tan lujosos. Estas usaban medias mangas que sujetaban con cintas.



Figura 3.7: Diferentes modelos de la mujer común

Mujer principal

Se visten al estilo cortesano con menos lujo en las telas, menos adornos, etc. Esto se debía a que su condición social y económica no les permitía adquirir estos elementos (Ver Figura 3.8).



Figura 3.8: Modelo de una mujer principal en Unity

Mujeres humildes

Las mujeres humildes (Ver Figura 3.9) eran artesanas, labradoras, sirvientas... Usaban un cuerpo muy escotado sin mangas, camisa y vasquiña, con una toca a modo de pañuelo sobre los hombros y delantal. Usaban jirones de tela de colores superpuestos para decorar.

No era raro que fuesen descalzas o usando unos zapatos llamados chinelas, podían sobreponer botines o podían usar zapatillas de badana.



Figura 3.9: Modelo de una mujer humilde en Unity

3.3.2. Modelos masculinos

A continuación describiremos los modelos masculinos utilizados en la aplicación

Cortesianos

Un vestido se entiende por un conjunto de prendas que generalmente se hacían de la misma tela y color, aunque no necesariamente. Los cortesianos (Ver Figura 3.10) podían usar:

- Ropilla, calzas o gregüescos, capa o herreruelo, el jubón es opcional. Si se pone jubón sólo muestra las mangas.
- Jubón, cuera o colete, calzas o gregüescos, capa o herreruelo. El jubón sólo muestra las mangas.



Figura 3.10: Modelo de un cortesano en Unity

Soldados

Los soldados podían usar diferentes estilos:

- Gregüesco: Calzón muy ancho que se usaba en los siglos XVI y XVII.
- Valones: Zaragüelles o gregüescos al uso de los valones, que los introdujeron en España.

- Cuera: Especie de chaqueta de piel, que se usaba antiguamente sobre el jubón.

En los soldados podíamos ver dos vestimentas, ambas representadas en las Figuras 3.11 y 3.12.



Figura 3.11: Modelo A de un soldado en Unity
Figura 3.12: Modelo B de un soldado en Unity

Médicos

Los médicos usaban borlas y loras. Son trajes holgados y austeros. También usaban sotana, manto, bonete. Bajo esta sotana vestían con jubón y calzón. (Ver Figura 3.13)



Figura 3.13: Modelo médico en Unity

Monjes Franciscanos

Los monjes franciscanos vestían con túnica larga, capucho, cuerda y calzones. El gris fue el color oficial para todos los franciscanos hasta mediados del siglo XVIII. (Ver Figura 3.14)



Figura 3.14: Modelo de un monje franciscano en Unity

Monjes Dominicanos

Los monjes dominicos vestían con túnica blanca, capilla con capucha o esclavina blanca y en el invierno capilla con capucha negra. (Ver Figura 3.15)



Figura 3.15: Modelo de un monje dominico en Unity

Clérigos

Los clérigos usaban sotana, manteo, bonete. Las sotanas eran talares y más holgadas que las de los hombres de estudios. Los altos dignatarios usaban loba sin manga y con cola. (Ver Figuras 3.16 y 3.17)



Figura 3.16: Modelo A de un clérigo en Unity



Figura 3.17: Modelo B de un clérigo en Unity

3.4. Museo

Otro de los componentes externos que usa este TFG es el modelo del Museo Cayetano Gómez Felipe realizado por Joshua García Expósito para su Trabajo de Fin de Grado. El modelo fue elaborado con el software de modelado de información de construcción Revit. La versión que se nos facilitó para usarlo en la aplicación de RV y poder ser importado en Unity está en formato obj. Para contar con las texturas de paredes, puertas, barandas, etc, era necesario extraerlas de dicho modelo y aplicárselas. Este proceso es automático, sin embargo, había ciertas zonas en donde

se producían fallos, por lo que se optó por aplicárselas a mano. Una vez terminado este proceso el museo se veía como en la Figura 3.18

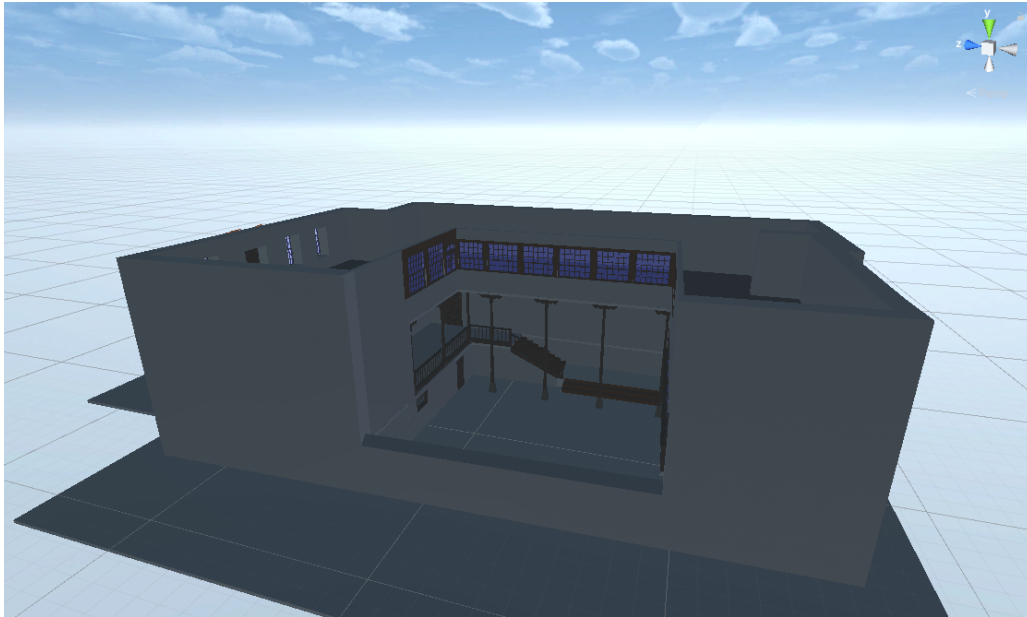


Figura 3.18: Réplica en 3D del Museo Cayetano Gómez Felipe realizada por Joshua García Expósito

En el caso de que se intentara usar el museo sin ninguna modificación adicional, el jugador no sería capaz de interactuar con este, ni siquiera sería capaz de caminar sobre él. Sin colisionadores no existen colisiones ni interacciones. Por esto el rayo de interacción del controlador no sería capaz de interactuar con nada.

Para conseguir estas interacciones se han valorado los Mesh Colliders y los Box Colliders. La desventaja del primero es que su coste computacional es mayor, sin embargo se ajusta de manera automática a la forma del objeto. Por otro lado, el Box Collider crea un colisionador en forma de caja que contiene al objeto y si se busca tener como objetivo el realismo, es peor, ya que no se ajusta a su forma aunque es más eficiente. Por esto se decidieron usar ambos colliders:

- Mesh colliders para zonas irregulares y extensas.
- Box Colliders para estructuras pequeñas y con contornos simples.

Por último fue necesario añadir algunos Colliders extras para limitar que el usuario tome caminos no deseados, por ejemplo, el modelo del museo tiene las puertas principales abiertas, lo que permitiría al usuario salir de dicho museo y caer infinitamente. Debido a esta razón se decidió poner un Collider invisible en medio de la puerta, impidiendo al jugador atravesarla y manteniéndolo encerrado en la zona deseada.

3.5. Iluminación

Unity ofrece varios tipos de iluminación [14], Real Time Lighting, Baked Lighting y Precomputed Realtime GI Lighting.

- Real Time Lighting: este tipo aporta luz directa a la escena y se actualiza en cada fotograma. A medida que las luces y los GameObjects se muevan dentro de la escena, la iluminación se actualizará de inmediato. Esto se puede observar tanto en la vista de la escena como en la del juego. La desventaja es que la luz en tiempo real no rebota, por lo que las sombras son muy oscuras y a no ser que pongas muchas luces en la escena, te quedará un ambiente muy oscuro. (Ver Figura 3.19)

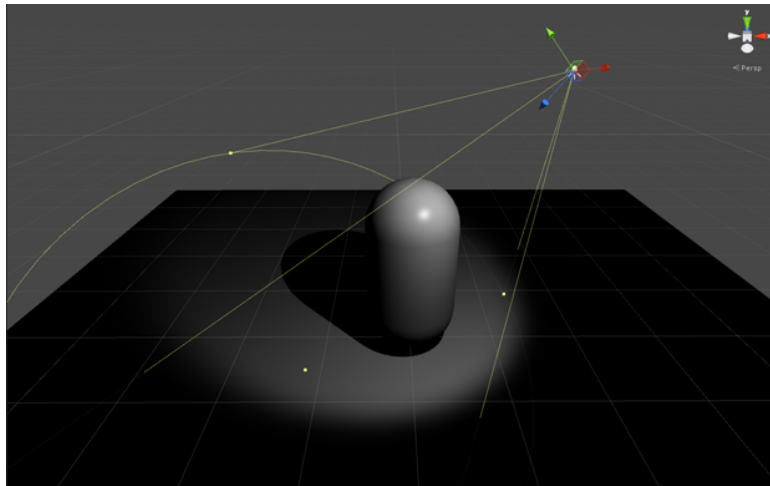


Figura 3.19: Ejemplo de iluminación en tiempo real [14]

- Baked Lighting, al 'hornear' un 'mapa de luz', se calculan los efectos de la luz sobre los objetos estáticos de la escena y los resultados se escriben en texturas que se superponen sobre la geometría de la escena para crear el efecto de iluminación. Estos "mapas de luz" pueden incluir tanto la luz directa que incide en una superficie como la luz "indirecta" que rebota en otros objetos o superficies dentro de la escena. En este tipo de configuración las sombras reciben la luz indirecta que rebota de objetos de su alrededor, por lo que en lo que respecta a iluminación, ofrece resultados mucho más realistas que el Real Time Lighting. (Ver Figura 3.20)

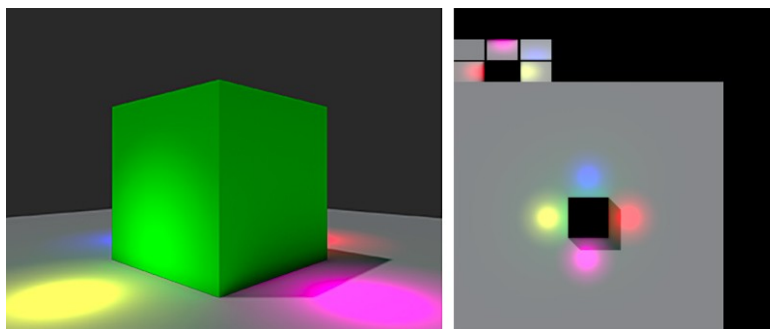


Figura 3.20: Ejemplo de iluminación horneada [14]

- Precomputed Realtime GI Lighting, si bien los mapas de luz estáticos tradicionales no pueden reaccionar a los cambios en las condiciones

de iluminación dentro de la escena, la GI en tiempo real precomputada nos ofrece una técnica para actualizar la iluminación de escenas complejas de forma interactiva. Con este enfoque, es posible crear entornos iluminados con una iluminación global rica con luz rebotada que responde, en tiempo real, a los cambios de iluminación. Un buen ejemplo de esto sería un sistema de hora del día, donde la posición y el color de la fuente de luz cambian con el tiempo. Con la iluminación tradicional horneada, esto no es posible. (Ver Figuras 3.21 y 3.22)

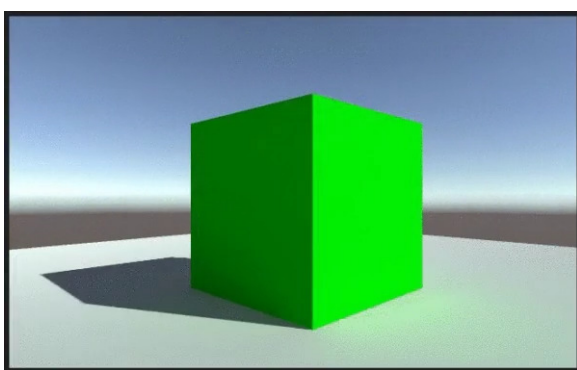


Figura 3.21: Ejemplo de luz diurna con luz precomputada [14]

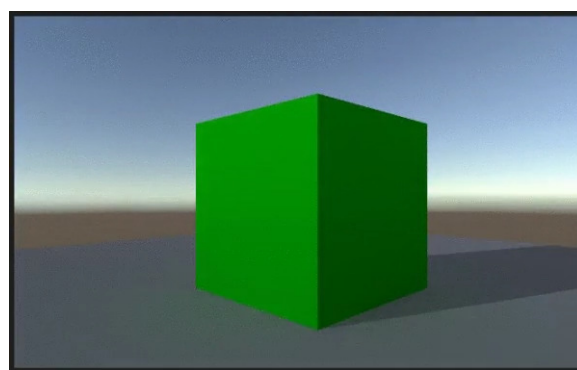


Figura 3.22: Ejemplo de luz nocturna con luz precomputada [14]

Teniendo todo esto en cuenta y que uno de los grandes problemas actuales de los dispositivos de RV son la capacidad de procesamiento que tienen, la opción a usar más lógica de iluminación debería ser Baked Lightning; en un principio se optó por usar Real Time Lightning y aunque en principio no afectaba al procesamiento, es cierto que el museo se veía bastante oscuro con este metodo, por eso con el tiempo se acabó usando el método Baked Lightning, mejorando mucho la calidad de las sombras, la percepción del ambiente en general y mejorando incluso el rendimiento de la aplicación.

Una vez establecida la configuración de iluminación, se decidió cambiar el skybox (Ver Figura 3.23) que viene como predeterminado en Unity,

mejorando también así la sensación de claridad. Dicho skybox fue obtenido del asset Skybox Series Free [15].



Figura 3.23: Skybox usado en el proyecto

3.6. Clases implementadas

En este proyecto se ha decidido añadir el sufijo *Controller* a cada clase implementada.

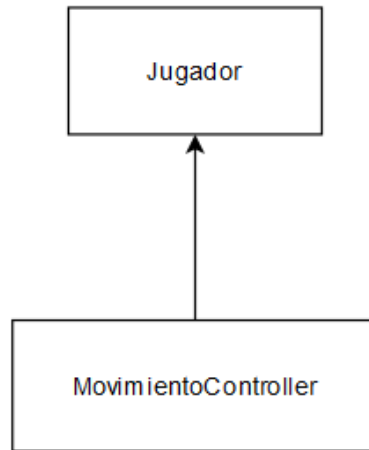


Figura 3.24: Jerarquía de clases de la entidad jugador

En la Figura 3.24 se muestra como el jugador solo ha necesitado que se cree una clase encargada de su movimiento, debido a que el resto de funcionalidades son conseguidas a través de los assets de RV.

Esta clase se encarga de recibir el input del controlador para que el jugador pueda moverse. También maneja la velocidad de movimiento y se encarga de que el jugador no sea capaz de atravesar paredes caminando en la vida real. Como ya se ha mencionado en la sección 3.2.2, el jugador es teletransportado al punto más cercano a la pared que está intentando atravesar.

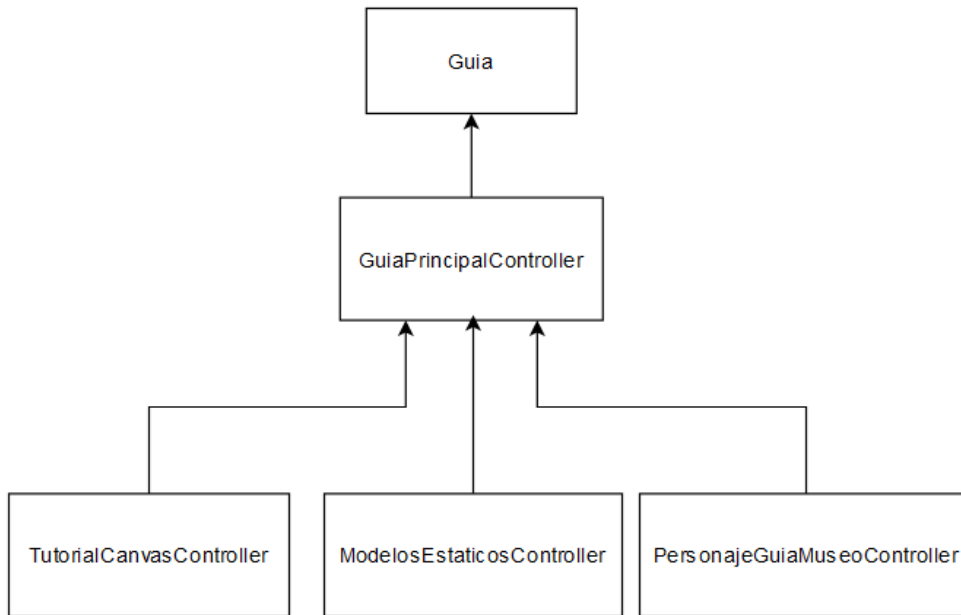


Figura 3.25: Jerarquía de clases de la entidad guía

Como vemos en la Figura 3.25 la guía ha necesitado de varias clases:

- **TutorialCanvasController:** Al comienzo de la aplicación hay una explicación escrita en paneles. Para que el jugador sea capaz de moverse entre estos, es necesario esta clase. La clase maneja el intercambio entre paneles.
- **ModelosEstaticosController:** Los modelos sin animación están divididos en grupos, para poder avanzar a través de estos, se usa esta clase.
- **PersonajeGuiaMuseoController:** El modelo que tiene animación se controla a través de esta clase. Esta maneja cuando debe avanzar, que animación debe reproducir y cuando mostrar su panel de información en cada parada.

Por último todas estas clases, son manejadas de manera externa a través de la clase **GuiaPrincipalController**. Esta se encarga de coordinar

las funciones del resto, permitiendo crear el servicio resultante de la guía.

3.7. Aplicación

3.7.1. Ejecución

Una vez comienza la ejecución lo primero que vemos es la Figura 3.26.

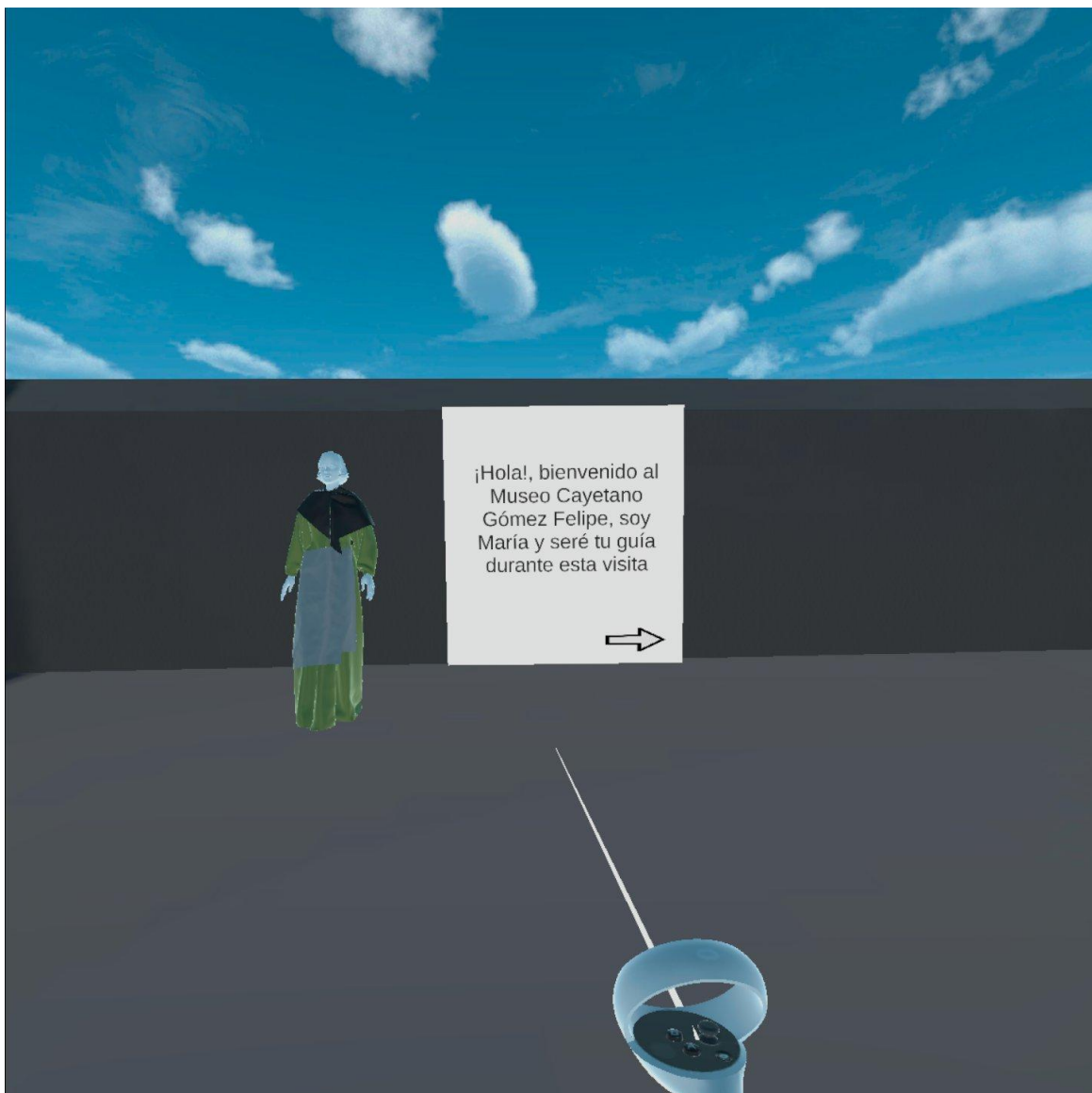


Figura 3.26: Primera vista obtenida al comenzar la aplicación

En este momento gracias a la clase TutorialCanvasController el jugador podrá ir avanzando y retrocediendo entre los paneles de información

disponibles usando las flechas, las cuales podemos ver en las Figuras 3.26 y 3.27.

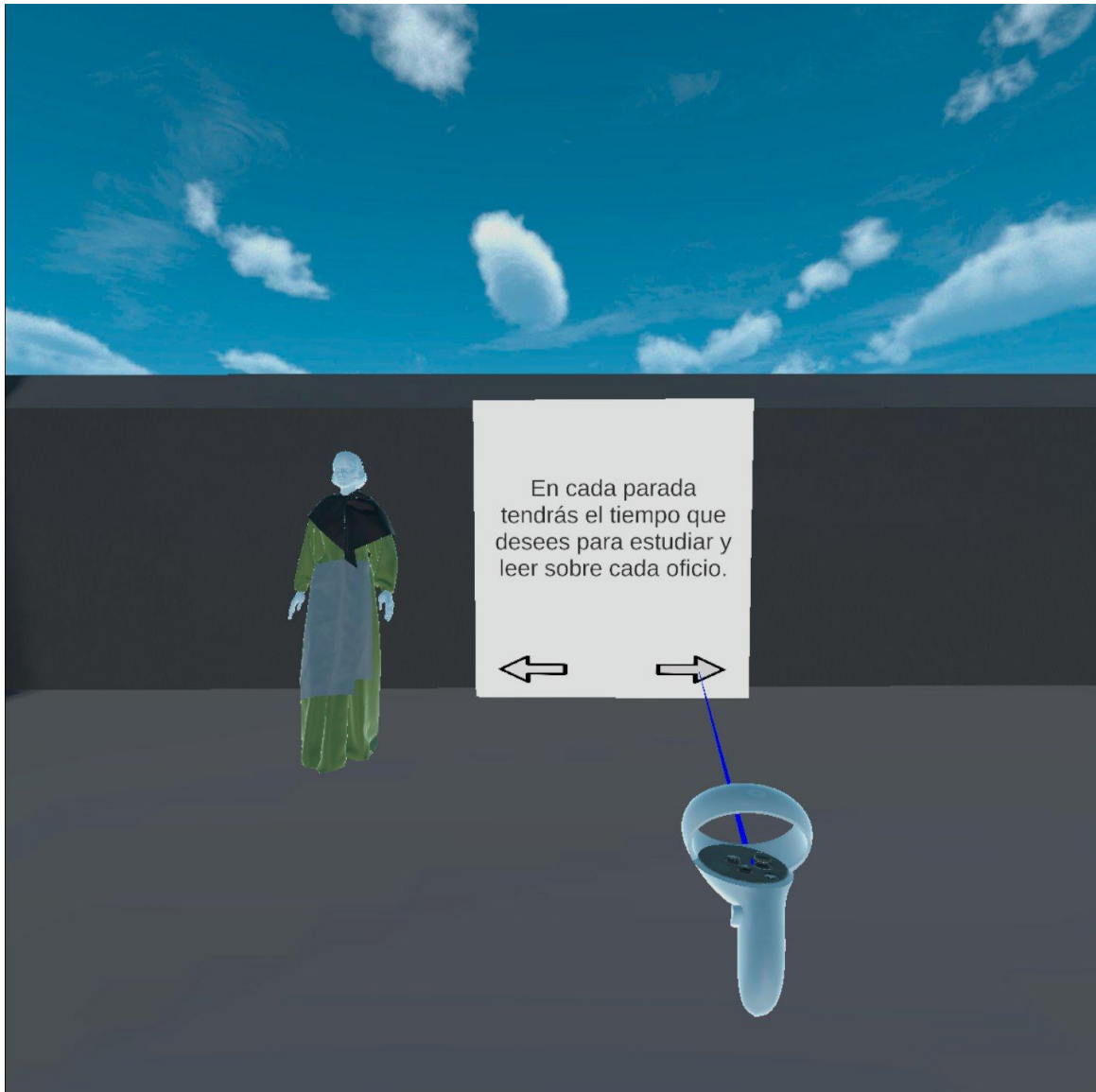


Figura 3.27: Cartel de información

Una vez llegado al último panel, se da la opción para comenzar la visita. Al seleccionarlo, la clase `GuiaPrincipalController` empezará a organizar todos los elementos a través de las diferentes clases. Los paneles de información inicial desaparecerán, nuestra guía empezará a dirigirse al primer grupo de modelos y estos serán mostrados. Dejándonos en una siguiente escena que podemos comprobar en la Figura 3.28



Figura 3.28: Primer grupo de modelos

Como vemos en la Figura 3.28, la guía posee una opción para continuar con la visita y de esta manera avanzaremos a través de todos los grupos (Ver Figura 3.29). Una vez se seleccione la opción de continuar, la clase `ModelosEstaticosController` se encargará de desactivar el grupo actual y de activar el siguiente, la clase `PersonajeGuiaMuseoController` se encargará de reproducir la animación de nuestra guía en donde avance hasta el siguiente grupo. Durante la animación, la misma clase se encargará de esconder el menú y una vez se llegue al destino, este menú será mostrado de nuevo.



Figura 3.29: Guía avanzando de un grupo de modelos al siguiente

Una vez llegado al final, el jugador se encontrará con la escena de la Figura 3.30. En este momento se da la opción de repetir la visita o de volver a ver a todos los modelos en la misma plaza sin tener que moverse.



Figura 3.30: Escena final

Estos modelos también son controlados por otra instancia de la clase `ModelosEstaticosController`, permitiendo cambiar de grupo usando las flechas que se encuentran debajo de ellos. (Ver Figuras 3.31 y 3.32)

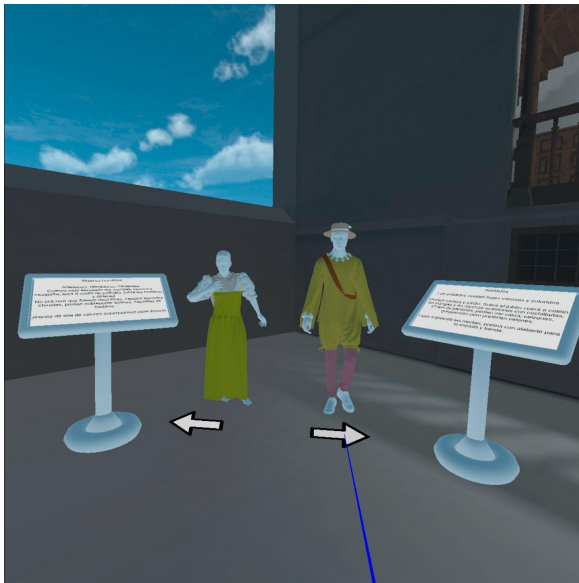


Figura 3.31: Grupo de modelos antes de cambiar

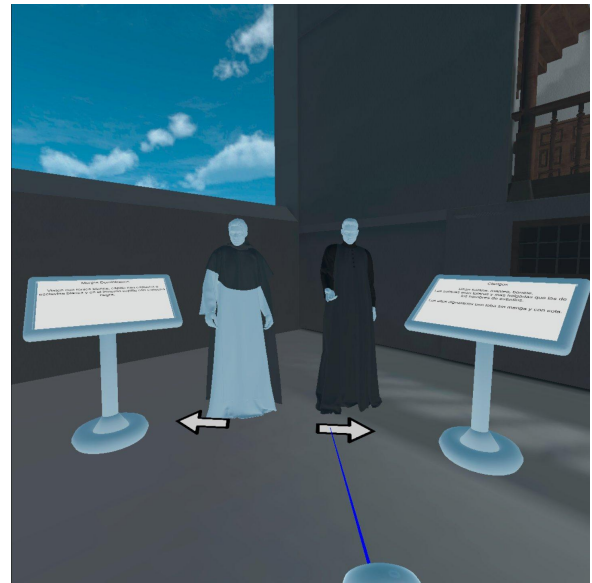


Figura 3.32: Grupo de modelos después de cambiar

3.7.2. Información

Para mostrar la información se tuvieron en cuenta diferentes formatos. Finalmente se optó por una estructura similar a un podio. Esta se desarrolló en Blender y posteriormente fue importada a Unity. En la parte superior se halla la información.

Para no verse limitado por la cantidad de espacio, la información que se encuentra en los podios puede ser desplazada (Ver Figuras 3.33 y 3.34) usando el botón trigger del controlador derecho y arrastrando de manera similar a como se haría en un dispositivo Android. Además a cada podio se le ha añadido un colisionador, evitando así que el jugador sea capaz de atravesarlos.



Figura 3.33: Podio antes de desplazar a través de la información

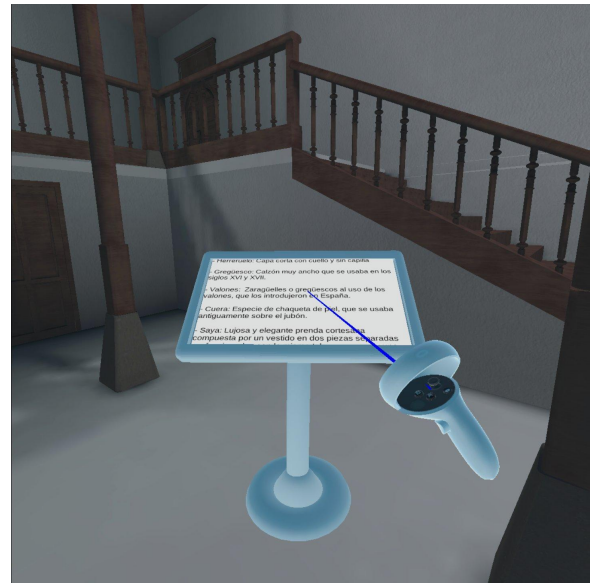


Figura 3.34: Podio después de haber desplazado

3.7.3. Audio

Sonido ambiente

Cuando nos encontramos en el mundo real, de manera constante escuchamos ruido de fondo, ya sean coches, animales, personas, etc. Si tuviéramos los ojos cerrados, esto nos indicaría que alrededor de nosotros existe vida, que el mundo no está vacío.

Cuando nos paramos a analizar diferentes productos audiovisuales, ya puedan ser películas, series, animaciones o videojuegos entre muchos otros. Todos comparten un aspecto y es que siempre existe ruido de fondo a no ser que la situación requiera de silencio. Esto permite que el mundo creado se asemeje a la realidad.

Por esto, en pos del realismo, se ha decidido añadir sonido ambiente [16]. La longitud de este son 5 minutos los cuales se repiten de manera infinita y el reinicio es imperceptible. Esta funcionalidad simplemente

ayuda a crear una mejor inmersión.

Voz

Como pudimos ver en las Figuras 3.26 y 3.27, la aplicación consta de una pequeña guía. Para hacerla más amena y darle una personalidad más realista , con la ayuda de Carolina Méndez Hernández se han interpretado los textos informativos mediante voz. Con esto se ha logrado que cuando se avance por los paneles, se escuche a la guía hablar.

Capítulo 4

Dificultades e investigación

4.1. Paquetes de Realidad Virtual

En el capítulo 3.2 hablo de como la intención inicial en este proyecto era usar el paquete Updated Controller Interaction para manejar la Realidad Virtual, su configuración dentro del proyecto, la programación y puesta a punto del jugador.

Este asset está preparado para ser usado con la versión de Unity que se ha usado finalmente en este proyecto y con las gafas Gear VR, sin embargo en la guía menciona explícitamente *"If no Gear VR controller is connected, this system will fall back to using a gaze pointer."* dándonos a entender que debería ser capaz de poder configurarse y usarse con cualquier dispositivo RV. Sin embargo, una vez configurado el proyecto, no obtenemos respuesta a través de las Oculus Quest 2, indicando que la RV no ha sido configurada correctamente.

En versiones antiguas de Unity, era necesario seguir una configuración diferente, por esto se intentó probar con este procedimiento, sin embargo, el resultado fue el mismo. En este punto del desarrollo se decidió

empezar a experimentar con otros paquetes de Unity/Oculus de RV, de nuevo siguiendo una guía oficial de Oculus [17] de como configurar el proyecto para poder trabajar con dispositivos Oculus.

Tras haberse realizado diferentes pruebas y experimentar con diversos paquetes el único resultado satisfactorio se consiguió con el paquete XR Interaction Toolkit, un paquete que actualmente está en estado de Preview, es decir, que se puede usar para producción pero bajo tu propio riesgo porque puede contener bugs. Por esto se siguió experimentando con los paquetes oficiales, sin embargo al no haber avance, se decidió usar el asset XR Interaction Toolkit.

4.2. Modelos de los personajes

En el capítulo 3.3 se habla de los personajes y de como se han tenido que tratar para poder usar en Unity, sin embargo no menciono que el mayor desafío al que se ha enfrentado este proyecto ha sido a causa de dichos modelos.

Cuando se empezaron a importar los modelos a Unity y a ser testeados desde el propio editor, sin RV, los modelos ejecutaban su animación correctamente. Debido a mi inexperiencia, asumí que si funcionaban correctamente desde un ordenador de sobremesa, funcionarían correctamente en la RV porque el paquete de VAT no interactúa internamente de ninguna manera con la RV. Sin embargo, una vez teniendo una cantidad notable de modelos dentro del proyecto con los que pudiera trabajar y avanzar en el desarrollo, decidí probar la aplicación dentro de las gafas.

Al ejecutar la aplicación en las gafas de RV, solo se obtuvo una pan-

talla en negro. Después de haber realizado varias pruebas, se llegó a la conclusión de que la RV dejaba de funcionar en cuanto renderizaba un modelo en la visión del jugador. El resultado que se obtenía si un modelo estaba a la derecha, el jugador podía mirar hacia delante y a la izquierda, pero en cuanto miraba al modelo, la aplicación entera se caía.

A partir de aquí se decidió contactar con el soporte de VAT y fue durante este intercambio en donde nos dimos cuenta de que las mallas de los modelos tenían demasiados vértices, creando una cantidad abrumadora de polígonos en el modelo de Unity. (Ver Figura 4.1)



Figura 4.1: Ejemplo del número de vértices de un modelo

VAT trabaja de tal manera que carga las animaciones en caché para que sean menos costosas, pero al tener los modelos tantos polígonos y vértices, la memoria caché de las gafas no lo soporta ocurriendo un desbordamiento de memoria, haciendo que la aplicación dejara de funcionar. En este punto se decidió empezar a investigar como reducir los vértices de los personajes, recurriendo así a la retopología.

En un principio se intentó experimentar con la retopología desde Blender [18], ya que en este se juntaban el modelo del personaje y su ropa, el problema es que ambos contenían su esqueleto con sus animaciones y al hacer la retopología, ese esqueleto se perdía y por lo tanto la animación también. Aparte, al ser modelos tan complicados, el retopologizar consumía mucho tiempo, por lo que la opción de coger cada frame y retopologizarlo no era viable debido a la fecha de entrega.

Nuestra siguiente idea fue intentar retopologizar en Marvelous Designer. Este contiene herramientas que lo hacen de manera automática al igual que Blender, sin embargo el resultado apenas tenía diferencia y no era suficiente para poder ejecutar la aplicación de RV. También se intentó hacerlo manualmente, sin embargo al hacerlo, Marvelous Designer cambia las costuras y con mi falta de experiencia con este software y a la falta de tiempo, esta solución tampoco era viable.

Finalmente se optó por usar el único modelo que es capaz de reproducir su animación sin generar problemas en la aplicación. Este corresponde a la mujer humilde y cuenta con una cantidad pequeña de vértices, permiti-

tiendo así no sobrecargar la caché cuando se ejecuta. Esto afectó al diseño que se tenía pensado hacer para la aplicación, decidiendo así mostrar la mayoría de modelos de manera estática y usando el que es capaz de caminar gracias a su animación como guía del museo.

4.3. Colliders

En el capítulo 3.4 hablo del museo y sus colliders. Antes de tener el modelo del museo oficial, estaba usando una versión inicial sin muchos detalles. En concreto las barandillas de pasillos y escaleras no estaban. Sin embargo una vez se recibió el modelo oficial completo y se probó, se experimentó que cuando se pasaba por ciertas zonas, la aplicación se ralentizaba demasiado, habiendo un retraso de hasta 2 segundos entre una acción en la vida real y su replicación dentro de la aplicación.

A simple vista la ralentización solo afectaba a la zona techada del museo pero al ser una zona que necesitaba ser transitada varias veces a lo largo de la ejecución se hizo inviable la opción de impedir el paso o limitar dicha zona.

Después de varias pruebas e intentos, se descubre que el problema lo causan las barandillas, concretamente, el Mesh Collider que se autogeneró cuando se le aplicó los colliders al museo entero.

Investigando, se averigua que cuando un modelo es complicado y se genera un collider automático sobre este, el collider consume muchos recursos para calcular las colisiones, coincidiendo esta información con la experiencia previa con dicho problema, ya que cuando el jugador se movía cerca de las barandillas, la aplicación se ralentizaba y cuando ibas por las

paredes no había problemas de rendimiento.

En este punto se decide eliminar el Mesh Collider generado automáticamente en todo modelo complejo y se les asignan a estos los Box Colliders manualmente, creando así colliders compuestos con formas geométricas mucho más sencillas, mejorando el rendimiento y eliminando el problema de los Mesh Colliders.

4.4. Rendimiento

En el capítulo 3.5 hablo de la iluminación y de los métodos elegidos. Existe un pequeño inconveniente y es la aplicación resultante se ejecuta dentro de un dispositivo portable, lo cual limita mucho la capacidad de rendimiento que puede tener.

El ser un dispositivo portable afecta a las opciones de calidad del proyecto, debido a que cuando la aplicación se ejecuta en las gafas, la configuración de calidad tiene su valor predefinido en mediano.

El inconveniente que este ofrece más allá de la calidad de imagen, es que la iluminación de puntos de luz no se cargan a cierta distancia, haciendo que si miras a lo lejos, parezca que no haya luces en el techo, sin embargo al acercarte, esta distancia se reduce y se carga dicha luz, iluminando la zona. Esto realmente no supone ningún problema real, pero rompe un poco la inmersión.

Por esto se intentó aumentar la distancia a la que se cargan las luces, pero al hacerlo, las gafas no son capaces de ejecutar la aplicación de manera fluida debido a la sobrecarga, por lo que se decidió dejar su

configuración de calidad en su valor predeterminando aun sabiendo el inconveniente existente.

Capítulo 5

Conclusiones y líneas futuras

5.1. Conclusiones

Para concluir repasaremos los objetivos más importantes, de esta manera recordaremos y daremos por finalizado dichos apartados.

El primer gran reto al que se enfrentó este proyecto como ya se dijo, fueron los paquetes de realidad virtual. Lo cierto es que hay poca documentación oficial de como preparar un proyecto y todos los ensayos y pruebas de configuración de paquetes de RV, se hicieron siguiendo diversas guías, siendo muy pocas de estas oficiales.

Otro apartado negativo es que si se necesita consultar alguna duda, la información existente en foros también es escasa. Esto se debe a que al ser una tecnología tan reciente, no posee unas bases sólidas. La RV carece del suficiente apoyo como para crear recursos públicos bien diseñados y completos. Además al tener una comunidad tan pequeña, es difícil que genere recursos online. A esta conclusión se llegó debido a que muchas veces se necesitó información sobre errores relacionados con la configuración de la RV y no encontrar nada relacionado.

En lo que respecta al movimiento del jugador, se tomó la decisión de que solo se pudiera mover en los ejes X y Z, haciendo que solamente sea capaz de moverse en el eje Y si sube escaleras.

Los movimientos solo se pueden realizar caminando dentro del mundo virtual. Se ha evitado usar la teletransportación debido a que el escenario es pequeño y nuestro objetivo final es que el jugador disfrute lo más posible del entorno, si se añadiera dicha función se estaría favoreciendo a que de manera inconsciente, el jugador se saltara zonas teletransportándose, quitándole así la importancia que se le ha querido dar al museo.

Debido a la falta de experiencia con el modelado 3D y la creación de aplicaciones en Unity, se consumió bastante tiempo investigando cuál era el problema de los personajes introducidos con VAT en el proyecto. Esto ocasionó que una vez encontrada la solución, la retopología, no se tuviera el tiempo suficiente para investigarla.

Aunque un modelo estático no posea animaciones, sigue teniendo cierto peso en la aplicación debido a su cantidad de vértices. Por esto al renderizar 3 o 4 modelos a la vez, la aplicación empieza a ralentizarse. Debido a este inconveniente se tomó la decisión de mostrarlos en grupos separados, de forma que cada vez que se avanza de grupo, desaparece el anterior.

Esto muestra que el diseño 3D requería más tiempo y afinación para poder ser usados con la RV satisfactoriamente, ya que en un ordenador de sobremesa no se daban estos problemas de rendimiento. Desafortunadamente la situación no permitió dedicarle más horas.

Analizando todos los problemas encontrados, vemos que la mayoría de problemas con los que lidió este proyecto estaban relacionados de una manera u otra con la RV y con la capacidad de procesamiento de las Oculus Quest 2, es decir, su rendimiento. Es un claro indicio de que de cara a próximos proyectos, debe ser tenido en cuenta.

Las gafas de RV al ser un dispositivo portable, no es capaz de rendir igual que un ordenador de sobremesa. Esto causa que cada decisión tomada en el proyecto afecte de una manera u otra al rendimiento. Se debe evaluar cada opción y ver como afectará, ya que sin un buen rendimiento, no se puede obtener una buena aplicación.

5.2. Lineas de trabajo futuro

De cara al futuro, este proyecto puede tener varias mejoras:

- Aunque es cierto que se ha encontrado un paquete de RV que funciona, este está en estado de Preview, así que no asegura que algún usuario no pueda tener ningún problema, es por esto que considero que usar un paquete oficial sería mejor, ya que por un lado aseguran un soporte y por otro, se puede encontrar más documentación.
- Los modelos podrían ser mejorados en cuestión de vértices, para esto se requeriría investigar/aprender más sobre la retopología en Blender y/o Marvelous Designer. Si se consiguiera se podrían tener a todos los modelos con sus animaciones, permitiendo conseguir el que era el objetivo inicial de este proyecto.
- El movimiento del jugador, podría ser mejorado añadiéndole un movimiento más realista al caminar, en este proyecto se evitó añadir

vaivenes, sin embargo, con más tiempo se podría diseñar un movimiento realista que permita una mejor inmersión.

- Aunque es cierto que el modelo del museo es bastante fiel al real, hay puertas cerradas que dan a salas, se podría añadir un sistema para interactuar con las puertas y que estas se abran y cierren, dándonos más libertad así como más espacio donde colocar modelos, información, etc .

Capítulo 6

Summary and Conclusions

6.1. Conclusions

To conclude, we will review the most important objectives, this way will allow us to remember and end these sections.

The first great challenge that this project faced as already mentioned, were the virtual reality packages. The truth is that there is little official documentation on how to prepare a project and all VR package configuration tests and trials were done following various guidelines, only being a few of these official.

Another negative section is that if any question is needed to be consulted, the information in forums is also scarce. This is due to the fact that being such a recent technology, it does not have a solid foundation. VR lacks sufficient support to create well-designed and comprehensive public resources. In addition, with such a small community, it is difficult for it to generate online resources. This conclusion was reached due to the fact that when searching for errors related to the configuration of the VR, no related information was founded.

Regarding to the player movement, the decision was made that the player could only move in the X and Z axes, being only able to move in the Y axis if you climb stairs.

The movements can only be performed by walking inside the virtual world. The use of teleportation has been avoided because the stage is small and our ultimate goal is for the player to enjoy the environment as much as possible, if this function were added, it would be favoring that unconsciously, the player would skip areas when teleporting, removing then the importance that has been wanted to give to the museum.

Due to the lack of experience with 3D modeling and creating applications in Unity, a lot of time was spent investigating what was the problem with the characters introduced with VAT in the project. This meant that once the solution, the retopology, was found, there was not enough time to investigate it.

It is a fact that static models have the same number of vertices as animated ones, but as they do not contain animations, they occupy much less. However, when rendering 3 or 4 models at the same time, the application begins to slow down, which is why the decision was made to show them in separate groups, being that each time you advance through the group, the previous one disappears.

This shows that the 3D design required more time and tuning to be able to be used with VR successfully, since these performance problems did not occur on a desktop computer. Unfortunately the situation did not allow more hours to be spent on it.

Analyzing all the problems found, we see that most of the problems that this project dealt with were related in one way or another to VR and the processing capacity of the Oculus Quest 2, that is, its performance. It is a clear indication that regarding future projects, it must be taken into account.

VR glasses, being a portable device, are not capable of performing the same as a desktop computer. Since each decision made in the project affects performance in one way or another each option must be evaluated and see how it will affect, since without a good performance, a good application cannot be obtained.

6.2. Future Lines

Looking ahead, this project may have several improvements:

- Although it is true that a working VR package has been found, it is in Preview status, so it does not ensure that some user may not have problems at all, that is why I consider that using an official package would be better, since on the one hand they ensure a support and for another, you can find more documentation.
- The models could be improved in terms of vertices, this would require researching / learning more about retopology in Blender and / or Marvelous Designer. If it were achieved, all the models could have their animations, allowing to achieve what was the initial objective of this project.
- The movement of the player could be improved by adding a more realistic movement when walking, in this project adding swings was

avoided, however, with more time, a realistic movement could be designed which would allow for a better immersion.

- Although it is true that the museum model is quite faithful to the real one, there are closed doors that lead to empty rooms, a system could be added to interact with them and let them be opened and closed, thus giving us more freedom and more space where to place models, information, etc

Capítulo 7

Presupuesto

7.1. Presupuesto

Tipos	Descripción	Horas	Precio
Hardware	Ordenador sobremesa	-	650€
Hardware	Oculus Quest 2	-	449€
Software	Software usado(Unity, Blender, Marvelous Designer)	-	133 €
Assets	Assets usados en unity(Vertex Animation Tools)	-	40.20 €
Investigacion	Estudio de las diferentes tecnologías	84 h	1,381.80 €
Diseño	Diseño del jugador, de la escena y jugabilidad	15 h	246.75 €
Desarrollo	Programación en Unity	120 h	1,974 €
Modelado	Modelado de los personajes	184 h	2,208 €
Información histórica	Busqueda de la información histórica	20h	285.40 €
Documentación	Desarrollo de la documentción	25 h	411.25 €
Total			5,807.37 € €

Tabla 7.1: Resumen de tipos

7.2. Justificación

Todos los precios han sido calculados usando el salario promedio en España del oficio correspondiente:

- Investigación, diseño, desarrollo y documentación: el salario de diseñador de videojuegos es de 16.45 €/h [19].
- Modelado: el salario de un diseñador 3D es de 12 €/h [20]
- Información histórica: el salario de un experto en historia es de 14.27 €/h [21]

Capítulo 8

Glosario

- **Ropilla:** Vestidura corta con mangas y brahones, de los cuales pendían regularmente otras mangas sueltas o peridas.
- **Jubón:** Vestidura que cubría desde los hombros hasta la cintura, ceñida y ajustada al cuerpo.
- **Calzas:** Prenda de vestir que cubría en forma holgada solo el muslo o la mayor parte de él.
- **Herreruelo:** Capa corta con cuello y sin capilla
- **Gregüesco:** Calzón muy ancho que se usaba en los siglos XVI y XVII.
- **Valones:** Zaragüelles o gregüescos al uso de los valones, que los introdujeron en España.
- **Cuera:** Especie de chaqueta de piel, que se usaba antiguamente sobre el jubón.
- **Saya:** Lujosa y elegante prenda cortesana compuesta por un vestido en dos piezas separadas confeccionadas con la misma tela: un cuerpo y una falda con cola.

- **Saboyana:** Similar a la saya pero no lleva cola, ligeramente más corta que la vasquiña y con una V invertida que deja ver la vasquiña.
- **Ropa:** Prenda holgada, abierta por delante, se vestía sobre la saya o sobre el conjunto jubón + vasquiña.
- **Vasquiña:** Falda exterior, puede estar tapada por las anteriores.
- **Verdugado:** Vestidura que las mujeres usaban debajo de las basquiñas para ahuecarlas.
- **Sotana:** Ropa holgada de un sólo cuerpo sin costura a la cintura.
- **Sotanilla:** Más corta que la sotana, los estudiantes la usan abierta por delante y se abrochan con botones. Se lleva con un cinturón.
- **Ropa de levantar:** Ropa larga pero no talar, abierta por delante y holgada con mangas tubulares colgantes.
- **Manteo:** Capa de forma circular similar al ferreruelo, con cuello levantado y estrecho.
- **Loba:** Prenda talar holgada con o sin mangas.
- **Garnacha:** Ropa larga con mangas muy abultadas en la parte superior y que no se abrocha por delante.

Bibliografía

- [1] Yazmina Rozas. “Los primeros mapas auténticos de las Islas. (Español)”. En: (2018). url: <https://diariodeavisos.elespanol.com/2018/01/los-primeros-mapas-autenticos-las-islas/>.
- [2] Kevin Estévez Expósito. “Realidad virtual en San Cristóbal de La Laguna Patrimonio Histórico. (Español)”. En: (2018). url: <https://riull.ull.es/xmlui/handle/915/10415>.
- [3] Miguel Aurelio García González. “Diseño y modelado de personajes para la reconstrucción histórica virtual de La Laguna en el Siglo XV. (Español)”. En: (2020). url: <https://riull.ull.es/xmlui/handle/915/19771>.
- [4] Alejandro González González. “Aplicación Web de una casa lagunera dels.XVI . (Español)”. En: (2020). url: <http://riull.ull.es/xmlui/handle/915/21323>.
- [5] Isabel Sánchez-Berriela y col. “AN INTERACTIVE 3D APPLICATION OF A HOUSE FROM THE XVI CENTURY IN SAN CRISTÓBAL DE LAGUNA AS A CASE STUDY FOR THE DISSEMINATION OF CULTURAL HERITAGE . (English)”. En: (2021). url: https://www.researchgate.net/publication/351164848_AN_INTERACTIVE_3D_APPLICATION_OF_A_HOUSE_FROM_THE_XVI_CENTURY_IN_SAN_CRISTOBAL_DE_LAGUNA_AS_A_CASE_STUDY_FOR_THE_DISSEMINATION_OF_CULTURAL_HERITAGE.

- [6] Valentino Clemente Ludovico Garavani. *Valentino Garavani Museum* . (English). url: <http://www.valentinogaravanimuseum.com/>.
- [7] Netflix, Brooklyn Museum y Matthew Yokobosky. *The Queen and the Crown*. (English). url: <https://www.thequeenandthecrown.com/>.
- [8] VR TeamSpace. "Getting ready for your VR TeamSpace™ meeting . (English)". En: (). url: <https://www.vrteamspace.co.uk/setup/>.
- [9] Andrey Volkov. "Vertex Animation Tools . (English)". En: (). url: <https://polyflow.xyz/content/vertex-animation-tools/documentation/index.html>.
- [10] Gabor Szauer. "Easy Controller Selection . (English)". En: (2017). url: <https://developer.oculus.com/blog/easy-controller-selection/>.
- [11] Oculus. "Oculus Integration . (English)". En: (2021). url: <https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022>.
- [12] Matt Fuad y Matt Dalby. "XR Interaction Toolkit Preview Package is here . (English)". En: (2019). url: <https://blog.unity.com/technology/xr-interaction-toolkit-preview-package-is-here>.
- [13] Carmen Bernis. *El traje y los tipos sociales en El Quijote*. Madrid, España: El Viso, 2001.
- [14] Unity. "Introduction to Lighting and Rendering . (English)". En: (2020). url: <https://learn.unity.com/tutorial/introduction-to-lighting-and-rendering#5c7f8528edbc2a002053b529>.
- [15] Avionx. "Skybox Series Free . (English)". En: (2018). url: <https://assetstore.unity.com/packages/2d/textures-materials/sky/skybox-series-free-103633?aid=1100l355n&gclid=CjwKCAjwmeiIBhA6Ei>

uaeFc6EYHMcQTYFwlkPU8_4qMSHtqgothwo-NH2v30gFxl1yz_4u9XMWRoCLZAQA
BwE&pubref=UnityAssets%2ADyn02%2A1723478829%2A67594162255%
2A336277505224%2Ag%2A%2A%2Ab%2Ac%2Agclid%3DCjwKCAjwmeiIBhA6EiwA-
uaeFc6EYHMcQTYFwlkPU8_4qMSHtqgothwo-NH2v30gFxl1yz_4u9XMWRoCLZAQA
BwE&utm_source=aff.

- [16] Infinity Soundscapes. “In the Park - Relaxing binaural audio ASMR no talking sounds for sleep . (English)”. En: (). url: <https://www.youtube.com/watch?v=AAdQWF5Dg78>.
- [17] Oculus. “Configure Unity Settings . (English)”. En: (). url: <https://developer.oculus.com/documentation/unity/unity-conf-settings/>.
- [18] Blender. “Retopology . (English)”. En: (). url: <https://docs.blender.org/manual/en/latest/modeling/meshes/retopology.html>.
- [19] Trabajar por el Mundo. “Trabajar como Diseñador de videojuegos . (Español)”. En: (). url: <https://trabajarporelmundo.org/trabajo-como-disenador-de-videojuegos/>.
- [20] indeed. “Desarrolla tu carrera profesional . (Español)”. En: (). url: <https://es.indeed.com/career/dise%C3%B1ador-3d/salaries>.
- [21] Tusalarario.es. “Filósofos, historiadores y especialistas en ciencias políticas . (Español)”. En: (). url: <https://tusalarario.es/carrera/funcion-y-sueldo/filosofos-historiadores-y-cientificos-politicos>.