_ TELECOM ESCUELA
TÉCNICA VLC SUPERIOR
DE UPV INGENIEROS
DE TELECOMUNICACIÓN

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

VISITA VIRTUAL AL MUSEO DE LAS TELECOMUNICACIONES VICENTE MIRALLES

Luis Pérez Pau

Tutor: Ximo Cerdá Boluda

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2014-15

Valencia, 23 de junio de 2015

Resumen

Esta memoria introduce el uso de las herramientas necesarias para crear desde el principio un entorno virtual con el que se pueda interactuar de forma fluida y sencilla. Se tratará el diseño de la arquitectura buscando la semejanza con el edificio a representar, el modelado del escenario en cuestión así como de los objetos que contenga para dotarlo de realismo, la texturización de todos los objetos modelados así como la programación de la interacción con el usuario.

Durante el proceso se darán consejos útiles y acceso a información que permita ampliar los conocimientos con respecto a todas las tecnologías utilizadas. También se aportaran los conocimientos necesarios para repetir el proceso con cualquier otro espacio o edificio.

Resum

Aquesta memòria introdueix l'ús de les eines necessàries per crear des del principi un entorn virtual amb el que es pugi interactuar de forma fluida i senzilla. Es tractarà el disseny de l'arquitectura buscant la semblança amb l'edifici a representar, el modelat de l'escenari en qüestió així com la dels objectes que contingui per dotar-ho de realisme, la texturització de tots els objectes modelats així com la programació de la interacció amb l'usuari.

Al llarg del procés es donaran consells útils i accés a informació que permeti ampliar els coneixements amb respecte a totes les tecnologies emprades. També s'aportaran els coneixements necessaris per repetir el procés amb qualsevol altre espai o edifici.

Abstract

This memory introduces the usage of the tools needed to develop from the beginning a simply and fluently interactable virtual environment. We will discuss the architecture design trying to get similarity with the building represented, the modeling of the scene and the objects it contains as well as the texturing of the objects created and the scripting of the user interactions.

Along the process useful advices will be given, and information that will improve our knowledge of the software used. It will be given as well the path to follow to create any other scene or building.

Índice

Capítulo 1. Introducción 3
Capítulo 2. Objetivo de este documento 4
Capítulo 3. Metodología 5
3.1 Gestión del proyecto 5
3.2 Distribución de tareas 5
3.3 Diagrama temporal 7
Capítulo 4. Desarrollo 8
4.1 Blender 8
4.1.1 Preferencias, atajos y puesta a punto 8
4.1.2 Addons 10
4.1.2.1 Archimesh 9
4.1.2.2 Window Generator 10
4.1.2.3 Images as Planes 10
4.1.3 Modelado de arquitectura 11
4.1.4 Adquisición de texturas 18
4.1.5 Texturizado de arquitectura 19
4.1.6 Modelado de objetos 20
4.2 Unity 23
4.2.1 Introducción a Unity 23
4.2.2 Importar desde Blender 24
4.2.3 Colocar objetos 25
4.2.4 Iluminación 26
4.2.5 Programación 30
4.2.6 UI 32
4.2.7 Build del proyecto 34
Capítulo 5. Resultados del trabajo 35
5.1 Visita virtual 35
5.2 Interacción con el usuario 35
5.3 Integración de la página web 35

Capítulo 6. Conclusiones y propuesta de trabajo futuro 36
6.1 Plataformas 36
6.2 Optimización 36
6.3 Integración 37
6.4 Crecimiento de la aplicación 37
6.4.1 Autodesk 123D Catch 37
6.4.2 MeshLab 38
Capítulo 7. Referencias y Anexos 39

Capítulo 1. Introducción

Vivimos en un mundo en el que la información es accesible de forma global, los dispositivos para consumir contenido se encuentran prácticamente en todos los hogares, y cada vez más también se dispone de los medios y herramientas para crear dicho contenido desde casa. Únicamente hace falta una buena idea que llevar a cabo.

La Escuela Técnica Superior de Ingenieros de Telecomunicación ha hecho un gran esfuerzo para reunir una gran cantidad de piezas de interés para el sector de las telecomunicaciones y exponerlas en un museo distribuido dentro de la propia escuela.

El Museo de Historia de la Telecomunicación Vicente Miralles pretende ilustrar el avance de la tecnología de comunicación que nació hace apenas 200 años a través de los propios dispositivos responsables de dicho avance.

La telegrafía, la telefonía, la radiocomunicación, los equipos audiovisuales, así como los aparatos para medir e investigar se presentan de una forma didáctica al gran público.

Hasta ahora ha sido necesaria la visita física al museo para poder aprender y descubrir todos los dispositivos que éste contiene. Esto supone que el público tenga que desplazarse en ocasiones largas distancias sin saber lo que encontrará una vez ahí.

Así nace el interés por proporcionar una experiencia previa a la visita, lo más cercana a la realidad, que rompa la barrera geográfica para visitar el museo y permita la visita virtual a todo aquel que esté interesado.

Si bien no se pretende conseguir la sustitución de la visita física al museo, la idea es familiarizar a los usuarios con el entorno y facilitarles una primera impresión del espacio y las dimensiones de la colección que contiene el museo, así como una vista previa de las piezas más significativas e información ampliada a través de un acceso transparente a la web del museo, que contiene una base de datos en constante crecimiento.

Capítulo 2. Objetivo de este documento

El objetivo de este trabajo consiste en la documentación del proceso desde que aparece la necesidad de ofrecer una visita virtual al museo, hasta que se obtiene un prototipo funcional y ampliable que cubre todas las necesidades planteadas.

Al finalizar el mismo se pretende haber conseguido una aplicación viable. El objetivo principal es que cumpla los requisitos para ofrecer información al usuario en forma de visita virtual en primera persona. Primará la sencillez y su uso ha de ser intuitivo.

De cara a ampliaciones posteriores o modificaciones, ya que el museo se encuentra en constante expansión, el diseño del proyecto deberá estar muy estructurado y debe de ser claro, para facilitar su uso y que pueda ser interpretado por personas ajenas a la creación del mismo.

Se tratarán las decisiones que se han tomado a lo largo del proyecto tales como la elección del software a utilizar para la creación de los entornos y objetos, la manera en que se ha abordado el diseño completo a la vista de su complejidad, cómo acceder a recursos que pueden resultar de utilidad o soluciones a los problemas más frecuentes que se van a encontrar durante el proceso.

Capítulo 3. Metodología

A continuación se plantea el diseño del proyecto y se desglosa en tareas concretas que ayuden a la realización del mismo.

3.1 Gestión del proyecto

En primer lugar, se ha de decidir el software principal a utilizar para la creación del proyecto. Existen muchas soluciones para el modelado 3D, por ello, tras analizar diferentes opciones se ha optado por Blender, un software open source disponible para cualquier sistema operativo que permite la creación de contenido 3D en alta calidad con una gran cantidad de herramientas y una comunidad activa que facilita el rápido aprendizaje.

Blender es sin duda una de las mejores opciones para introducir a un nuevo usuario al mundo del modelado 3D. Es completamente gratuito y cuenta con una gran cantidad de tutoriales con los que aprender y dominar su manejo.

Con el uso de Blender se cubre toda la parte de creación de contenido, si bien ahora se ha de decidir el motor gráfico que dará vida a la visita virtual. En este caso también contamos con diversas opciones, Unity 5, Unreal Engine 4, o CryEngine 3 son algunos de ellos.

Se ha decidido utilizar Unity 5 frente al resto por su alta compatibilidad con Blender, ya que el flujo de trabajo entre estas dos aplicaciones es prácticamente transparente. Éste motor gráfico es gratuito para pequeños estudios. Unity, además, ofrece una experiencia de usuario excelente siendo contenida en el consumo de recursos, lo que facilita el acceso a una cantidad mayor de usuarios. Y por último, otro factor importante es el hecho de que Unity es un motor gráfico multiplataforma, lo cual nos permitirá portar la aplicación una vez terminada a todo tipo de dispositivos móviles sin tener que programarla de nuevo para cada dispositivo diferente, así como adaptarla a cualquier sistema operativo con un bajo coste y un simple ajuste en los controles de movimiento e interacción.

3.2 Distribución de tareas

En primera instancia, nos encontramos frente a un proyecto de una envergadura considerable, así pues se debe de abordar dividiéndolo en tareas realizables pensando como si un equipo de diversas personas estuviese trabajando en él.

En primer lugar deberemos recoger la información necesaria para poder plantear la visión global del entorno, esto implica realizar fotografías del edificio a representar desde todos los ángulos y posiciones posibles para posteriormente conseguir una representación fiel del espacio, así como recopilar los planos del edificio que serán de gran utilidad para el modelado.

Con respecto al modelado en Blender, por un lado tenemos la creación de la arquitectura del edificio que contendrá el museo, y por otro lado los objetos situados en su interior, entre los que se encuentran las vitrinas expositoras así como los objetos contenidos en ellas.

Una vez conseguido el modelado, se ha de realizar una tarea intermedia de recopilación de texturas que se aplicarán posteriormente. Las texturas se obtendrán tanto de fotos realizadas dentro de la propia universidad como de páginas web que contengan texturas de acceso gratuito que se puedan adaptar a nuestras necesidades.

El proceso de texturizado se realizará también en Blender de forma independiente para todos los objetos que aparecerán en la escena.

Los objetos finalizados se exportarán a Unity creando una gran biblioteca de objetos, también llamados assets, para su posterior colocación en el entorno. Los objetos se colocarán uno a uno en el entorno de Unity en su lugar correspondiente hasta que se disponga del resultado deseado.

Con la escena terminada, se procederá a ajustar la iluminación global de la misma, colocando un sol en el cielo que ilumine de forma natural el entorno, determinando los objetos que emiten luz, como las lámparas y posicionando puntos de luz discretos en determinadas zonas hasta conseguir un resultado correcto.

A continuación se animarán los objetos móviles de la escena usando el animador integrado en Unity. Esto incluye puertas que deban abrirse, o los indicadores flotantes sobre los expositores para indicar a los usuarios que se puede interactuar con ellos.

Ahora se puede colocar al jugador, que sujeta la cámara a través de la cual el usuario observará el entorno y se desplazará. Se programarán ajustes para controlar parámetros como la velocidad de movimiento, la sensibilidad de visión, o la posibilidad de saltar dentro del entorno.

El siguiente paso consiste en permitir la interacción del usuario con los expositores que componen el museo, de ésta manera se determinarán áreas junto a las vitrinas en las que al entrar se muestre la información deseada y se programará el comportamiento del entorno tomando decisiones en función del estado del jugador.

La información a mostrar se diseña y se estructura para cada expositor de forma personalizada y se basa en información contenida en la base de datos de la que dispone la escuela de los objetos de la colección. Algunos objetos disponen de información adicional en la página web del museo, por lo que se redirigirá al usuario en caso de que desee ampliar el contenido.

Para mejorar la experiencia del usuario se habilita la utilización de menús, paneles de información y controles de la aplicación.

Finalmente hace falta realizar tareas de optimización para mejorar el rendimiento de la aplicación y ya se podrá realizar la construcción de la aplicación propiamente dicha para su utilización por el usuario.

3.3 Diagrama temporal

Tarea	*Tiempo (horas)*	*Descripción*
Planificación del proyecto	20	*Toma de decisiones, planteamiento por tareas del proyecto y aprendizaje de las tecnologías a utilizar*
Toma y edición de fotos	5	*Toma de fotografías y uso de GIMP para poder usarlas en el proyecto*
Modelado de arquitectura	50	*Modelado 3D en Blender de la arquitectura del edificio*
Búsqueda de Texturas	4	*Búsqueda de texturas en páginas web y edición para poder usarlas en el proyecto*
Texturizado de arquitectura	5	*Texturizado de la arquitectura en Blender a base de proyectos UV.*
Modelado de objetos	40	*Búsqueda de ejemplos en internet, modelado y texturizado de objetos en Blender a utilizar en el proyecto*
Migración a Unity	5	*Importación de los modelos de Blender a Unity y edición de los mismos para conseguir el resultado esperado*
Colocación de objetos	12	*Colocación de los objetos modelados en el entorno*
Iluminación	18	*Colocación de puntos de luz y control de la iluminación de la escena*
Scripting	8	*Programación de comportamiento, interacción con el usuario y animaciones*
UI	18	*Elaboración del sistema de datos que se muestra al usuario en la aplicación, así como de menús adicionales*
Optimización	5	*Reducción de peso de la aplicación y aplicación de parámetros de rendimiento dependiendo de la plataforma*

190 horas

Capítulo 4. Desarrollo

En este capítulo se procede con la creación del proyecto, empezando con el uso de Blender para el modelado y la texturización del entorno, y finalizando con el entorno de Unity para la ubicación de los objetos modelados y la programación de los comportamientos esperados resultantes de las acciones del usuario.

4.1 Blender

Blender es un programa informático open source multi plataforma, dedicado especialmente al modelado, iluminación, renderizado, animación y creación de gráficos tridimensionales. Se utilizará para modelar el edificio y la arquitectura interna, así como el mobiliario y todos los objetos a utilizar.

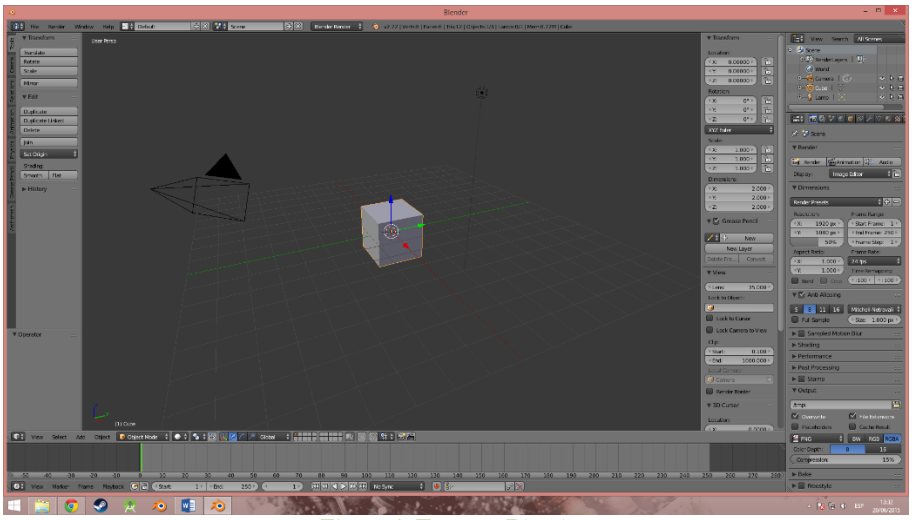


Figura 1. Entorno Blender.

Se puede obtener desde su página web www.blender.org de forma gratuita y dispone de un manual online donde consultar cualquier duda en www.blender.org/manual/, la comunidad ofrece un gran soporte y se puede acceder a todo tipo de tutoriales en Youtube.

El objetivo de este documento no cubre la iniciación al uso del programa, por lo que se recomienda adquirir unos conocimientos básicos antes de comenzar. Se recomienda la visualización de los tutoriales de Youtube del usuario BornCG que cuentan con una curva de aprendizaje perfecta para familiarizarse con el entorno.

4.1.1 Preferencias, atajos y puesta a punto

Antes de empezar a modelar, se recomienda familiarizarse con los atajos que utiliza Blender, ya que agilizarán mucho el trabajo, así como ajustar algunas preferencias de usuario y opciones del programa para facilitar el proceso de creación del proyecto.

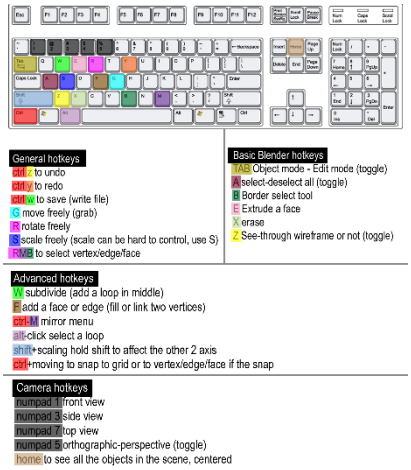


Figura 2. Atajos de teclado Blender.

Se recomienda habilitar un guardado automático de nuestro trabajo, utilizando el siguiente menú.

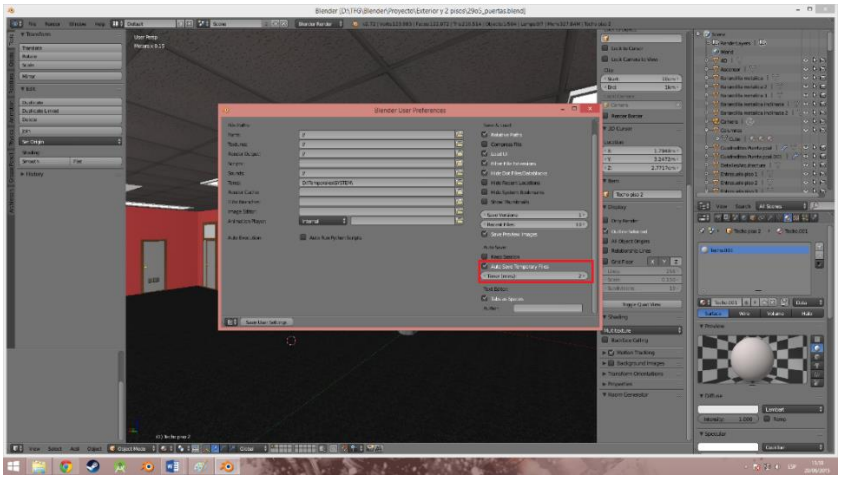


Figura 3. Autoguardado.

También cambiaremos las unidades de medida del programa para que trabaje en el sistema métrico, lo cual nos ayudará a la hora de la representación de objetos.

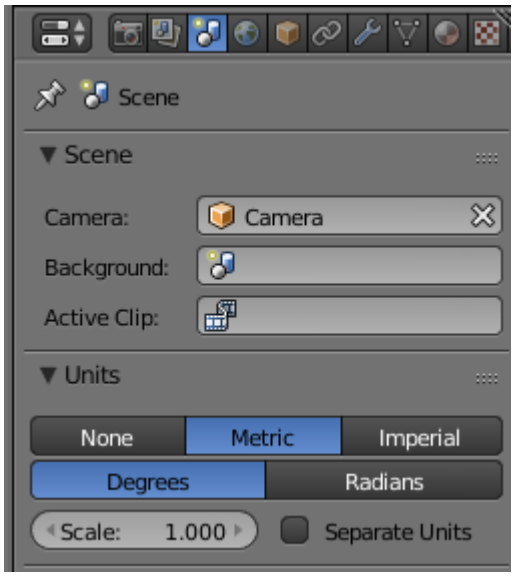


Figura 4. Unidades.

Por último, algo que nos ayudará a ser precisos durante la construcción de la arquitectura es habilitar el botón "snap during transform", que nos permitirá ajustarnos a la parrilla virtual para conseguir un resultado preciso junto con la utilización del SHIFT.

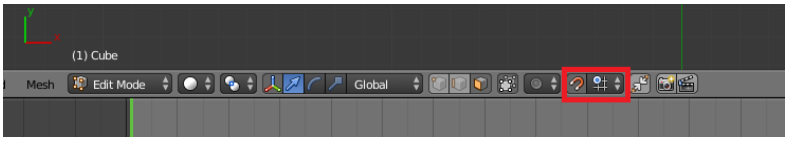


Figura 5. Snap during transform.

Una vez configuradas estas opciones se puede proceder al modelado, pero antes veremos las posibilidades añadidas de las que dispone Blender.

4.1.2 Addons

Los addons son extensiones que se le pueden añadir a Blender para ampliar sus funcionalidades. Existe una gran cantidad de addons, y nos pueden ahorrarnos mucho esfuerzo en determinadas situaciones.

Para añadir addons simplemente hemos de conseguir el script correspondiente (normalmente programado en python) y dirigirnos a la pestaña de Addons dentro de las preferencias de usuario y ahí lo instalaremos y lo habilitaremos para usarlo.

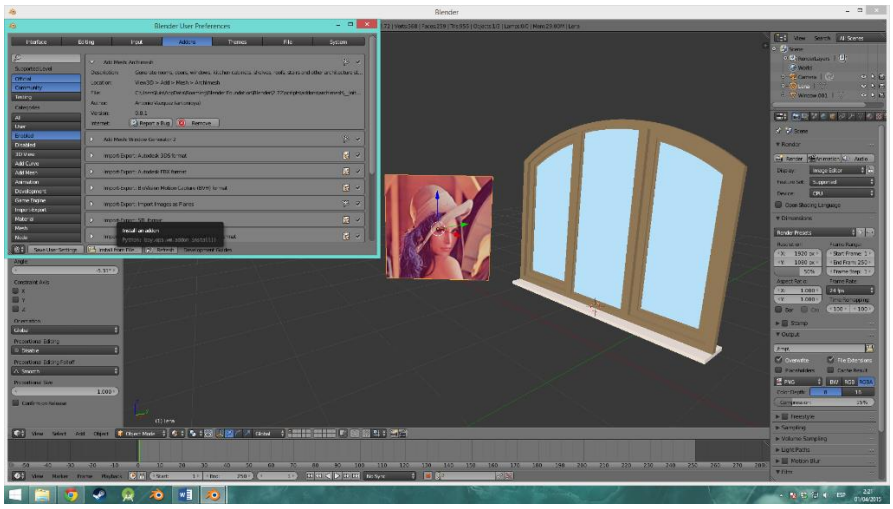


Figura 6. Addons.

4.1.2.1 Archimesh

Archimesh es un addon de arquitectura libre para Blender. Nos permite un acceso rápido a estructuras tales como escaleras, puertas, ventanas o estanterías. Nos proporciona un objeto básico de la clase que seleccionemos y nos permite ajustar parámetros para adecuarlo a nuestras necesidades.

Resulta imposible encontrar ningún modelo que sea como el que queremos, pero son modelos iniciales sobre los que trabajar.

Las ventanas, no obstante, son poco parametrizables y no se ajustan en la mayoría de los casos, así que se ha optado por utilizar Window Generator para ese propósito por ser más completo.

El script de Archimesh se puede obtener desde su página web, junto a las versiones anteriores. Se adjunta como Anexo [3].

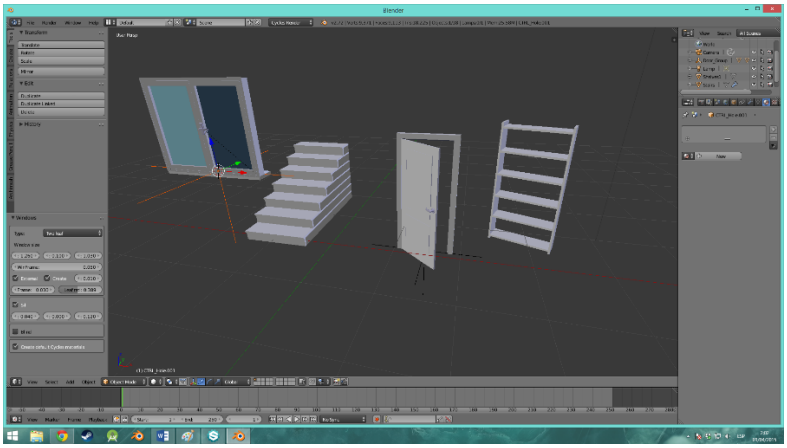


Figura 7. Archimesh.

4.1.2.2 Window Generator

Se trata de un addon arquitectónico también de acceso libre como el Archimesh, pero dedicado exclusivamente a ventanas, por lo que ofrece muchas más opciones que se adapten a nuestras necesidades.

El script se puede descargar desde el anexo correspondiente [4].

4.1.2.3 Images as planes

Es un addon muy útil incluido por defecto en los addons de Blender. Solo es necesario localizarlo en la pestaña de addons y habilitarlo. Nos permite introducir una imagen en nuestra escena como si de un plano se tratara de forma sencilla. Podemos modelarlos conjuntamente con otros objetos para crear objetos realistas.

4.1.3 Modelado de arquitectura

En este apartado se muestra el proceso seguido para levantar la arquitectura. En primer lugar, usaremos una de las opciones de Blender para colocar una imagen de fondo, en nuestro caso el plano de la universidad, que nos ayudará a determinar la forma del edificio.

Habilitamos "Background Images" como se muestra en la Figura 8 y seleccionamos la imagen del plano. Acto seguido ajustamos el tamaño de la imagen para conseguir una escala real, para ello, se ha medido la anchura del recibidor principal de la escuela, que son 9 metros y se ha hecho coincidir con la imagen del plano.

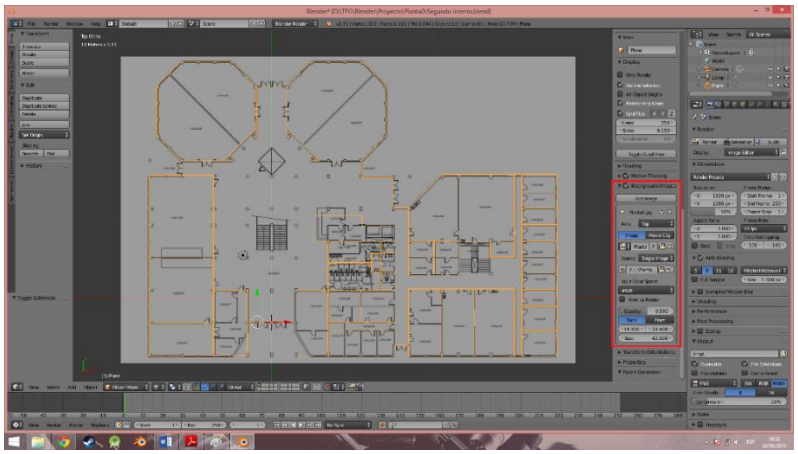


Figura 8. Imagen de fondo.

Se ha tomado como grosor general de las paredes 30cm, es importante dotar de grosor a las paredes, estas no pueden ser simplemente planos porque provocarían errores en las colisiones con los objetos y también nos encontraríamos con errores a la hora de representarlos en Unity, ya que dicho entorno solo interpreta uno de los lados del plano como renderizable, mostrando paredes invisibles por uno de los lados desde que se mirasen.

Partiendo de un plano de 30x30cm colocado al nivel del suelo sobre una de las paredes del plano, cogeremos una de las aristas y la extenderemos a lo largo de la pared. Al llegar a una esquina, se extruirá con la tecla E un nuevo plano de 30cm, consiguiendo el resultado que se muestra en la Figura 9, completando con el mismo proceso todas las habitaciones.

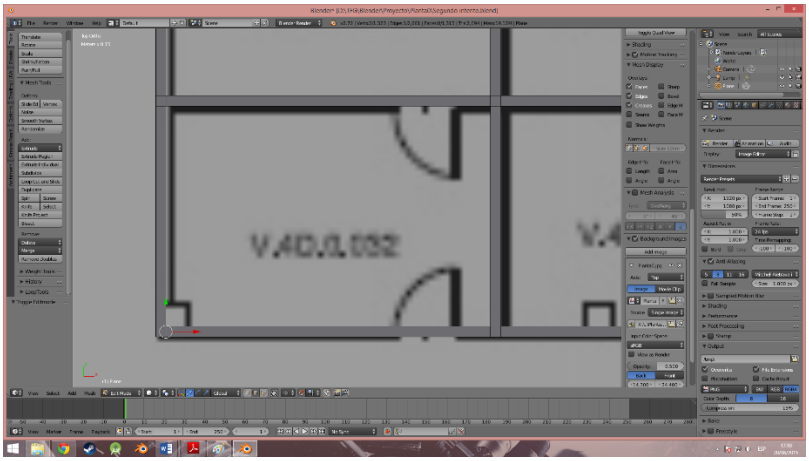


Figura 9. Resultado habitación.

El resultado final es el que se observa en la Figura 8, una vez obtenido, seleccionaremos toda la silueta del edificio con la tecla A y la extruiremos en el eje Z la altura deseada, consiguiendo las paredes.

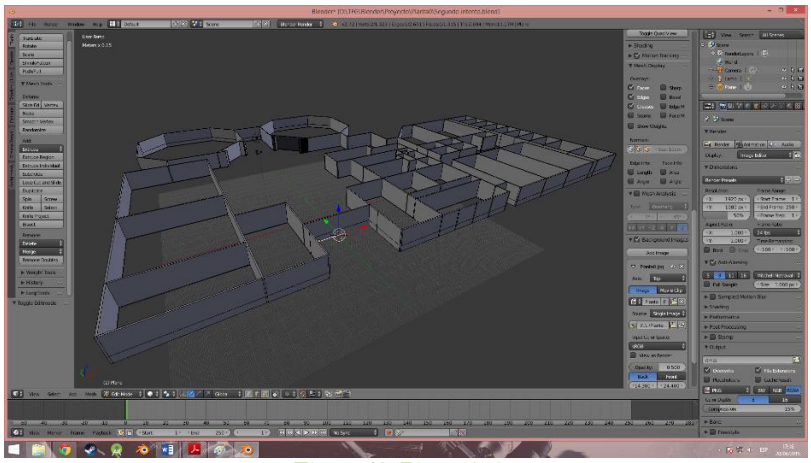


Figura 10. Planta baja.

Para crear el suelo, seleccionaremos las aristas que conforman una habitación y pulsaremos la tecla F, que creará una cara que nos servirá de suelo. Para habitaciones complejas con muchos lados como el vestíbulo, se irán creando triángulos hasta cubrir la superficie por completo.

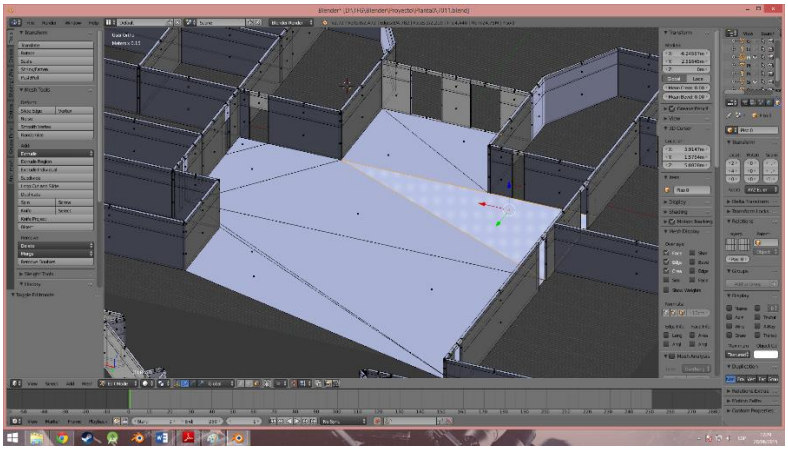


Figura 11. Creación suelo.

Para obtener los huecos donde colocaremos las puertas podemos proceder de dos maneras, una es utilizar la herramienta de "loop subdivide" con el atajo CTRL+R en el modo editar para crear los cortes donde colocaremos la puerta y posteriormente eliminar las caras que queremos que actúen de hueco como se muestra en la Figura 12. Cabe resaltar que primero se han tomado las medidas de los diferentes tipos de puertas para que luego se ajusten a los modelos de puertas que colocaremos.

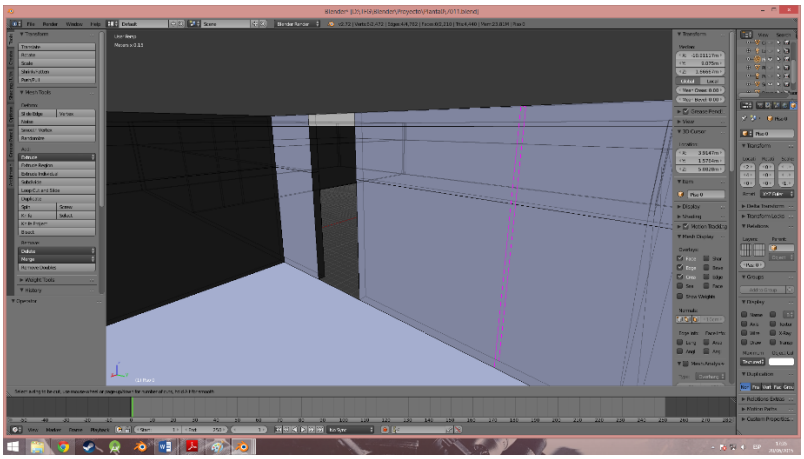


Figura 12. Creación puertas.

La otra forma de proceder consiste en crear un cubo, darle la forma del hueco de la puerta, colocarlo donde debería estar dicho hueco y aplicar sobre la pared el modificador booleano, configurado para que se le reste la forma del cubo que simula el hueco de la puerta. Una vez aplicado el modificador, se elimina el cubo que actuaba de hueco. Para las ventanas se seguirá el mismo proceso.

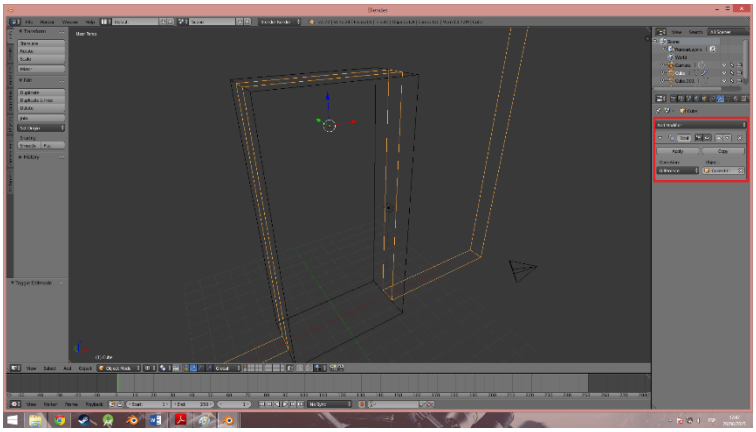


Figura 13. Creación puertas alternativa.

Actuaremos de la misma forma con el resto de plantas, permitiéndonos reutilizar las partes que sean iguales entre plantas seleccionándolas, duplicándolas con SHIFT+D y llevándolas al piso superior. Finalmente obtendremos el resultado mostrado en la Figura 14.

Figura 14. Arquitectura completa.

La universidad ya contaba con un modelado exterior del edificio de Telecomunicaciones, así que podemos aprovecharlo y añadirlo de forma que contenga nuestra arquitectura.

El modelo se proporcionó en formato .fbx, un formato de Autodesk que normalmente Blender soporta sin problema, pero en nuestro caso se produjo un error de compatibilidad.

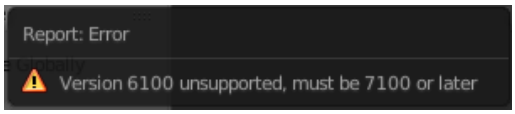


Figura 15. Error de compatibilidad.

Para solucionarlo, debemos convertir el .fbx a una versión más nueva utilizando la herramienta gratuita AutodeskFBX Converter Tool.

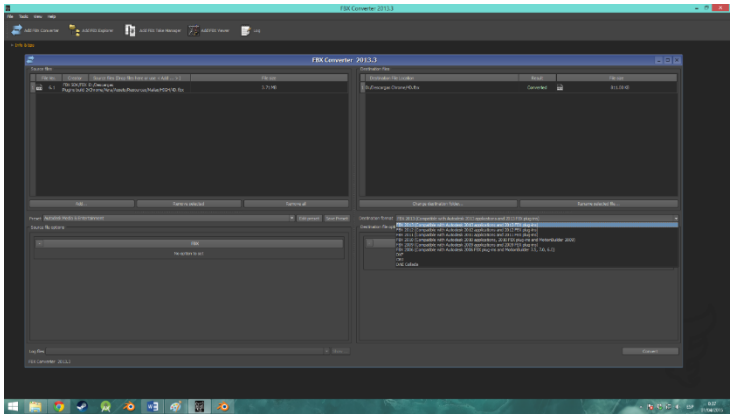


Figura 16. Autodesk FBX converter.

Seleccionamos el fichero .fbx que nos ha dado problemas, nos aseguramos de seleccionar el formato destino FBX 2013 y Blender ya lo podrá abrir. Colocaremos el modelo de forma que contenga la arquitectura creada y lo ajustaremos. Deberemos editar el modelo de la arquitectura para asegurarnos de que todo se ajusta de la forma deseada, obteniendo el resultado que se muestra en la siguiente figura.

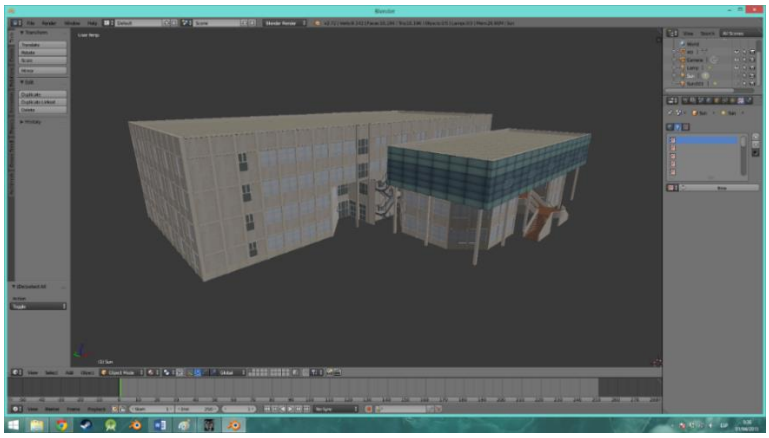


Figura 17. Modelo del edificio 4D.

Ahora procederemos a la construcción de las escaleras, el ascensor, columnas y barandillas. Todo lo que forme parte de la arquitectura propia del edificio.

Las columnas se crearán a partir de cilindros y se posicionarán con la ayuda de los planos de la escuela. El ascensor será también un cilindro simple al que añadiremos una puerta. Para las barandillas crearemos primero un patrón (Figura 18) que podamos repetir.

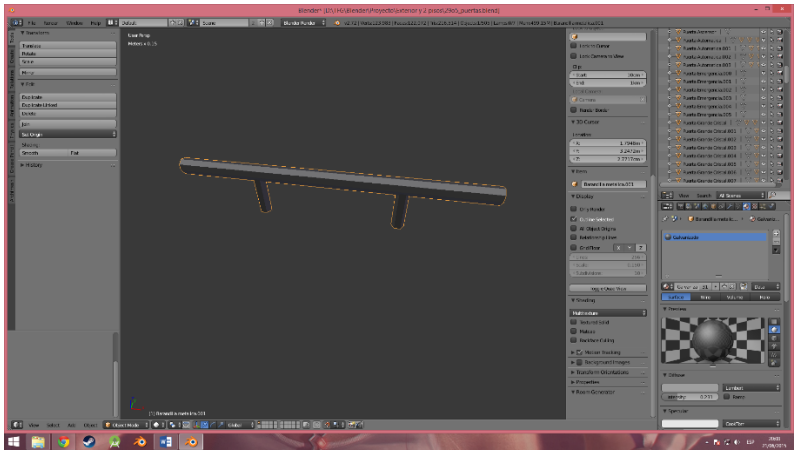


Figura 18. Patrón barandilla.

Para crear las escaleras, utilizaremos el addon Archimesh y ajustaremos los parámetros hasta que encaje en nuestro modelo.

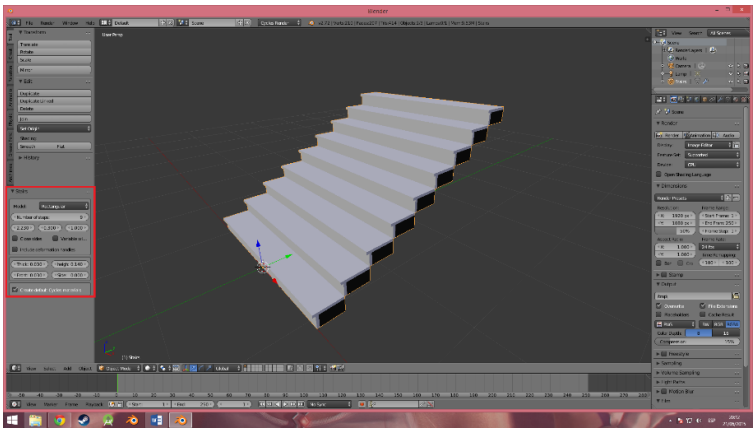


Figura 19. Escaleras.

Finalmente añadiremos detalles de la arquitectura como columnas en las paredes para evitar que las paredes queden completamente planas, dotando de más realismo a la escena.

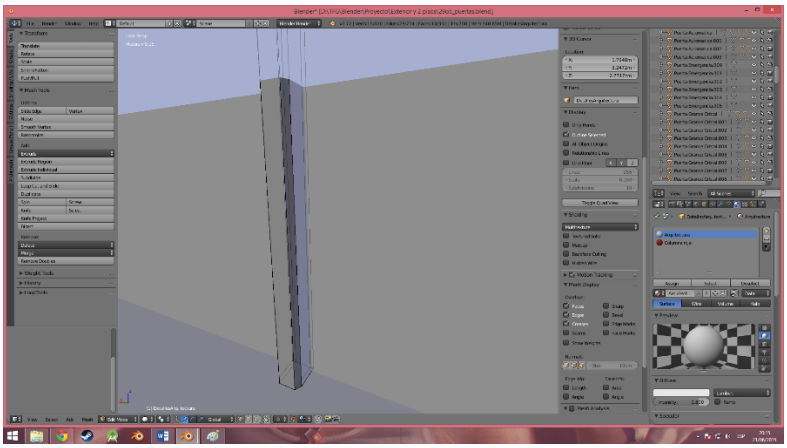


Figura 20. Detalles de la arquitectura.

Habiendo ultimado los detalles, nos centramos en colocar las puertas y ventanas, para ello, comenzaremos un nuevo modelo en Blender. Utilizaremos Archimesh o Window Generator para conseguir el modelo inicial, y lo modificaremos hasta conseguir el resultado deseado. Repetiremos el proceso para cada tipo de puerta o ventana, teniendo en nuestro caso puertas de diferentes tamaños así como con o sin cristal.

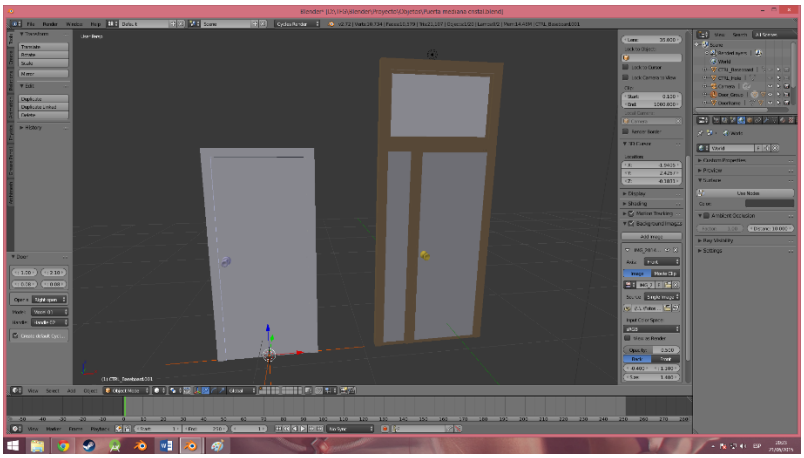


Figura 21. Puertas.

Para añadirla a nuestro proyecto, primero seleccionaremos todas las partes de la puerta y pulsaremos CTRL+G para crear un grupo, que será el que exportaremos. Después en nuestro proyecto seleccionaremos la opción de "Append" bajo el menú "File" y buscaremos el .blend correspondiente a la puerta y añadiremos el grupo que hemos creado anteriormente.

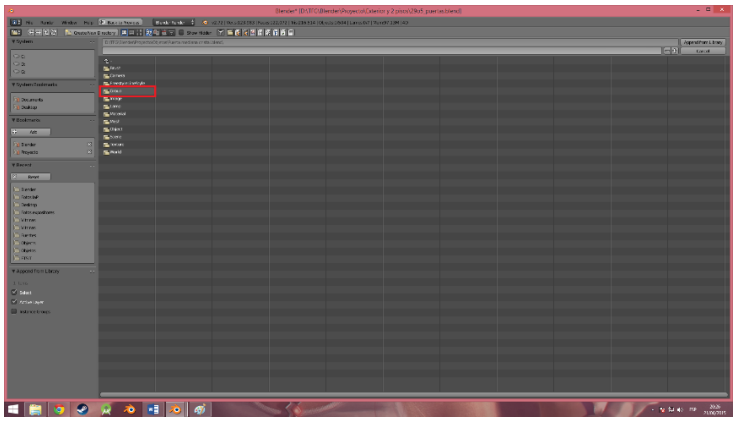


Figura 22. Append.

Una vez introducido el modelo de la puerta en el proyecto, se colocará en su lugar y se duplicará con el uso de SHIFT+D hasta haberlas ubicado todas.

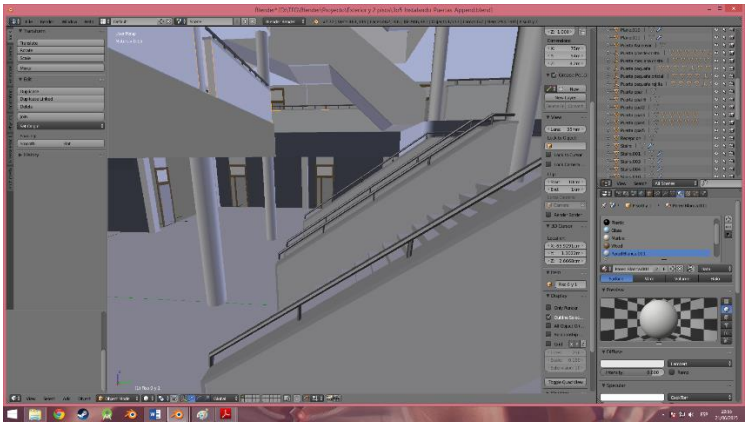


Figura 23. Arquitectura final.

Finalizado todo el proceso, ya podemos empezar a texturizar el modelo de la arquitectura. Para ello es importante que todos los objetos que la componen hayan sido desenvueltas, o en términos del programa, "unwrapped". Esto se hace entrando en el modo editar y abriendo el menú de UV Mapping con la tecla U, seleccionaremos la opción Smart UV Project, esto nos permitirá aplicar texturas a los objetos y modificarlas como deseemos.

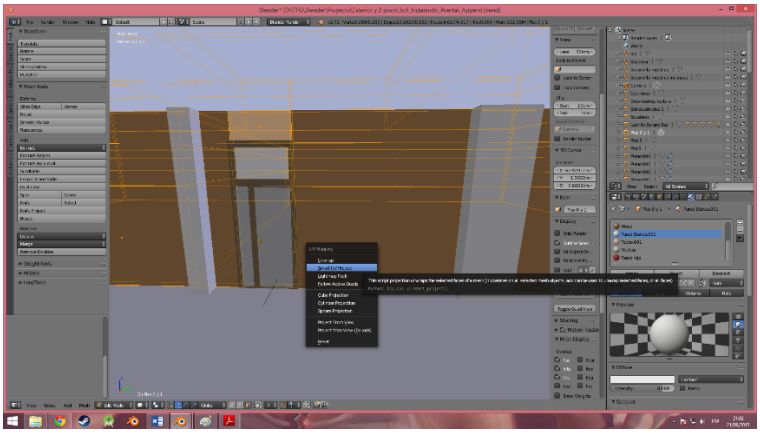


Figura 24. Arquitectura final.

4.1.4 Adquisición de texturas

Antes de proceder con el texturizado, debemos disponer de las texturas que vamos a aplicar. Puesto que estamos intentando representar un espacio real, será difícil encontrar exactamente las texturas que necesitamos en internet, así que realizaremos fotos a las texturas del edificio que queramos representar y las editaremos para poder utilizarlas como texturas dentro de nuestro proyecto.

Para editar las fotografías utilizaremos el programa GIMP2.

GIMP es un programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Es un programa libre y gratuito.

En primer lugar, usaremos las herramientas de transformación como cortar, o la perspectiva, para obtener la imagen que queremos utilizar.

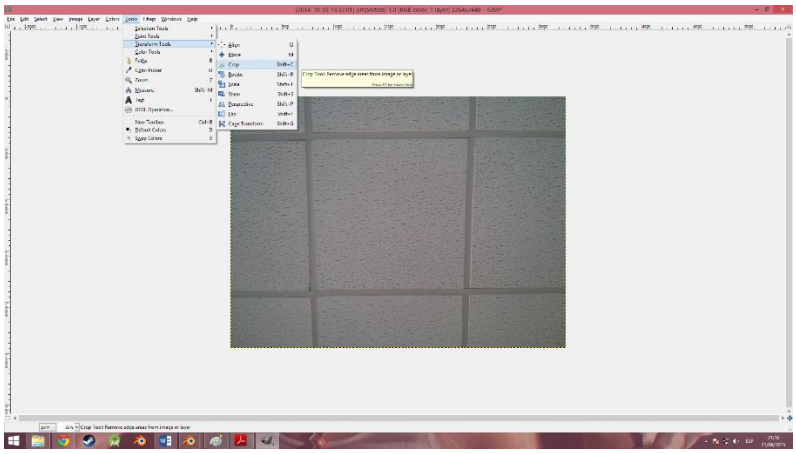


Figura 25. Fotografía del techo.

También nos permitirá evitar las "costuras" que aparecen cuando una imagen se copia de forma repetitiva para cubrir una superficie. Para ello aplicaremos el filtro que se muestra en la siguiente figura.

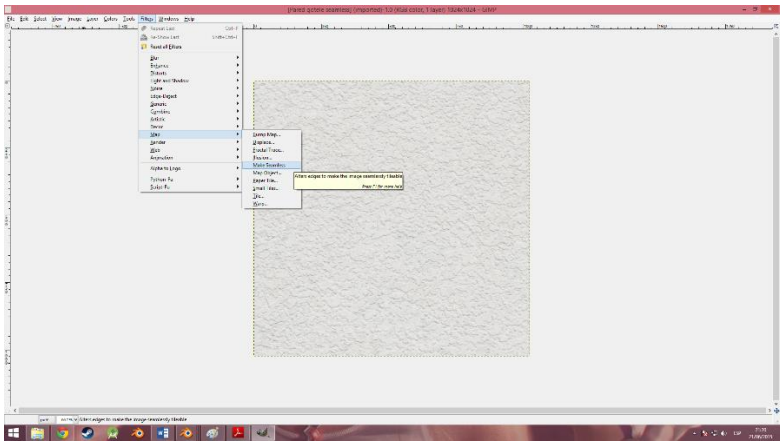


Figura 26. Filtro para evitar costuras.

Para texturas genéricas como las paredes blancas o el suelo de granito sí que podemos obtener en internet texturas de alta resolución que se adapten a nuestras necesidades. Existen páginas web dedicadas a su distribución de forma gratuita. Se adjuntan en los Anexos [5] y [6].

La búsqueda de texturas es una tarea larga y costosa, ya que se necesita mucha similitud con la textura real para poder utilizarla.

4.1.5 Texturizado de arquitectura

Una vez obtenidas las texturas que vamos a utilizar, podemos proceder con el texturizado. Para ello primero crearemos los materiales y ajustaremos sus propiedades hasta que se ajusten al resultado deseado.

En caso de que el material necesite una textura, es decir, que no sea suficiente con aplicar un color, se procederá al menú de texturas, se seleccionará el tipo "Image" y se indicará el directorio donde se encuentra la foto, siendo recomendable que el directorio sea relativo, para facilitar el trabajo y permitir que se pueda trabajar con el proyecto en otros ordenadores.

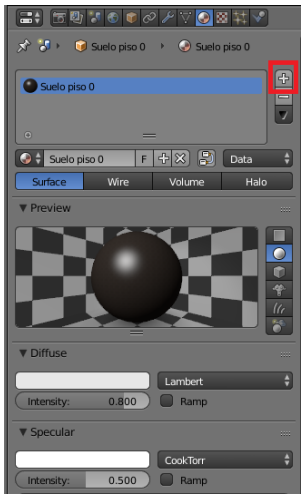


Figura 27. Añadir materiales.

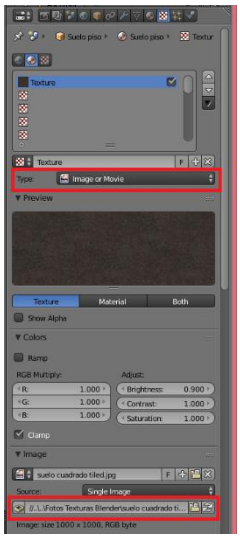


Figura 28. Añadir texturas.

En algunos casos como el techo, nos interesa la posición de la textura. Por ello, podemos abrir dentro de Blender el editor de imágenes, y con la textura seleccionada ajustar su tamaño o su orientación para determinar cómo se visualizará.

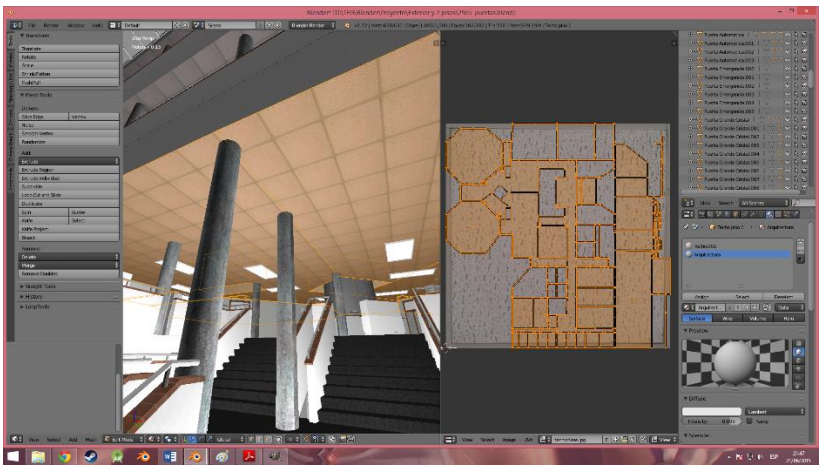


Figura 29. Editor de imágenes de Blender.

4.1.6 Modelado de objetos

Una vez se ha obtenido el modelo de la arquitectura y se ha texturizado, podemos proceder a modelar los objetos que contendrá el edificio.

Se trabajará de forma similar a como se ha hecho con las puertas, se abrirán nuevos proyectos, se modelarán y texturizarán los objetos por separado hasta obtener el resultado final, y posteriormente se añadirán al proyecto principal.

En este caso, el proceso de colocación de los objetos se realizará en el entorno de Unity.

Es necesario un manejo medio del programa y estar familiarizado con el proceso de modelado y el uso de modificadores para conseguir buenos resultados.

A continuación se muestran algunos ejemplos de los objetos modelados.

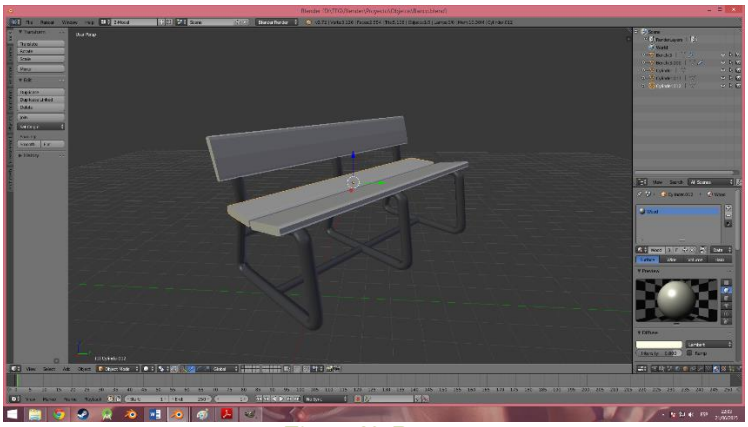


Figura 30. Banco.

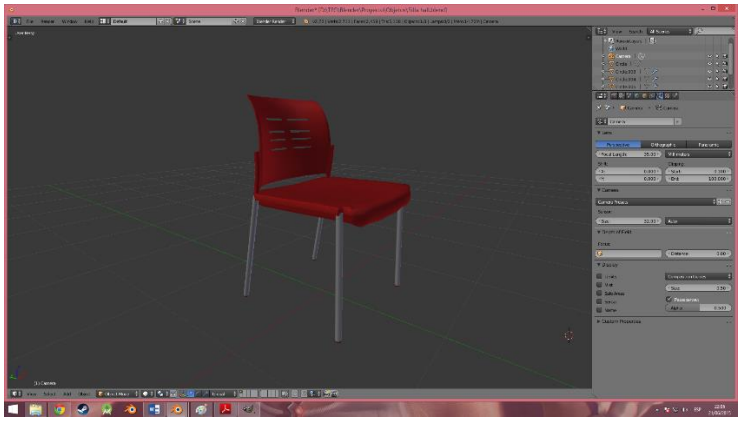


Figura 31. Silla.

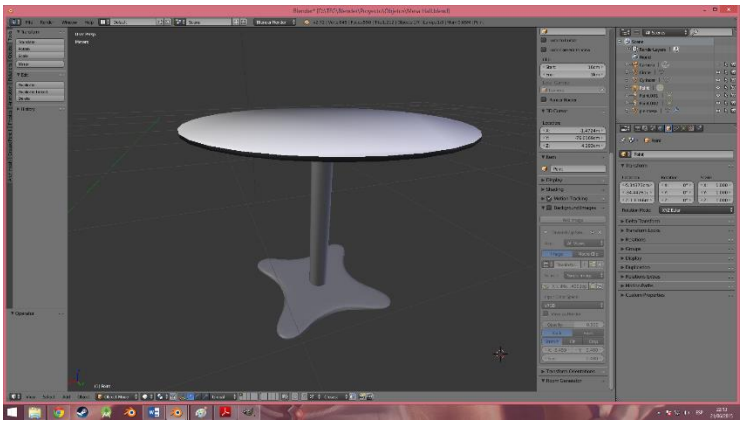


Figura 32. Mesa.

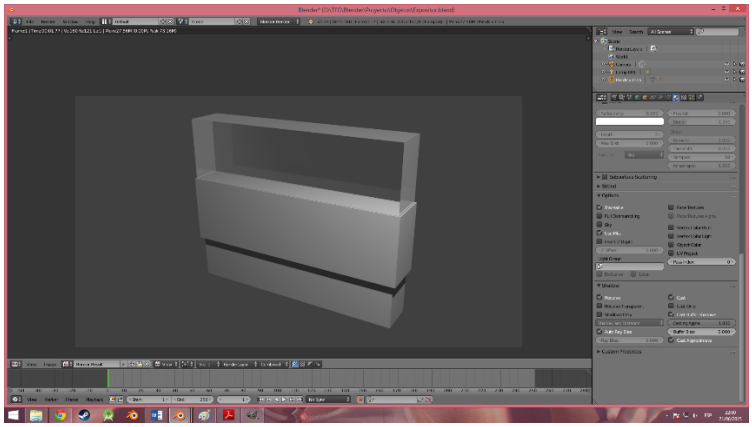


Figura 34. Render de un expositor.

Para la creación de algunos objetos resulta de mucha utilidad el addon "Images as Planes", que nos permite añadir fotos reales y combinarlas con objetos modelados para conseguir un alto nivel de realismo.

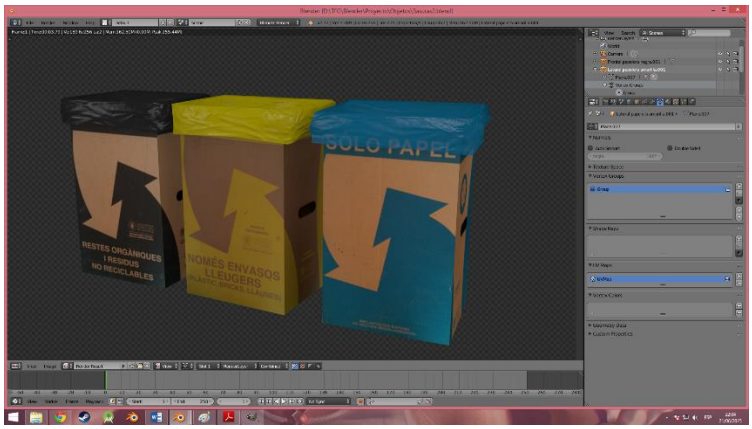


Figura 35. Basuras.

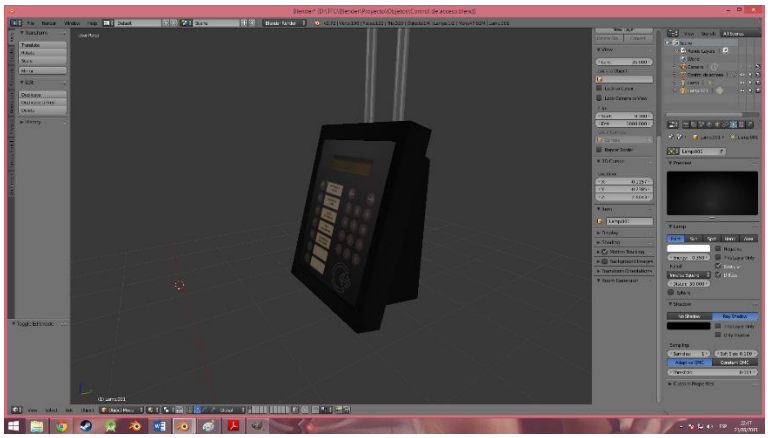


Figura 36. Control de acceso.

Una vez terminado un objeto, es recomendable aplicar su escala, con el atajo CTRL+A, de manera que la escala se mantendrá al ser exportado a Unity.

Este addon también nos da muchas facilidades para la elaboración de los carteles de información, cuadros, o las imágenes que colocaremos dentro de los expositores.

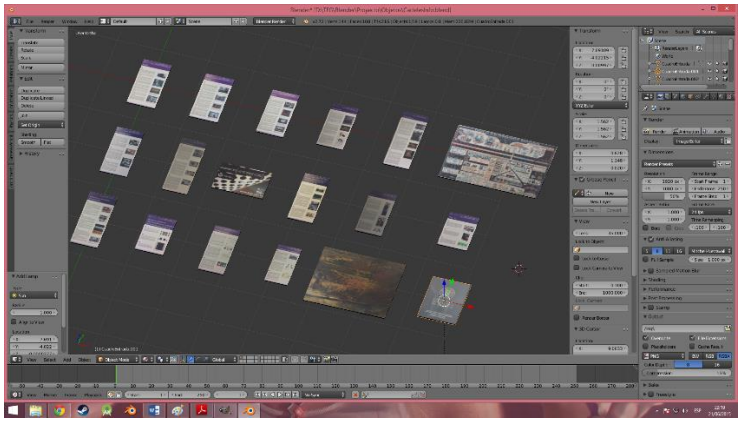


Figura 37. Carteles de información.

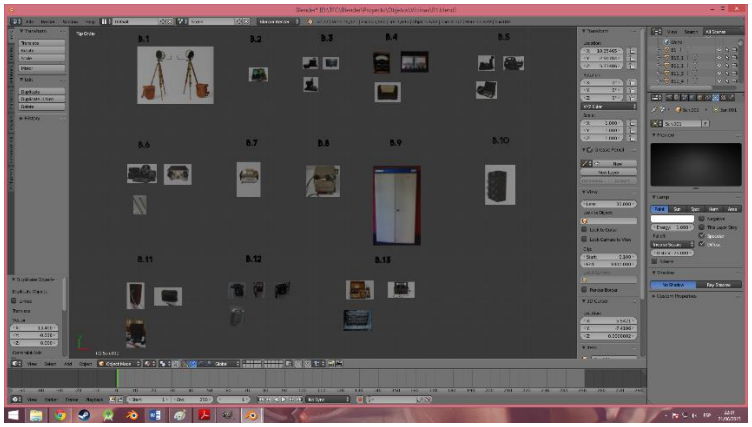


Figura 38. Fotos colección de objetos.

La biblioteca de objetos a crear depende del realismo que pretendamos conseguir y de los recursos de los que se dispone, ya que es un proceso largo y en ocasiones costoso. Siempre se puede ampliar el número de objetos a posteriori.

Antes de dar por acabado el trabajo con Blender, hemos de tener en cuenta que Unity, como motor gráfico, tiene una forma característica de renderizar los modelos, y que solo muestra como visible una de las caras los planos que conforman los objetos, por lo que es necesario asegurarnos de que las normales de dichas caras apuntan en el sentido correcto. Para ello activaremos la opción mostrada en la siguiente figura y cambiaremos el sentido de las normales de las caras que sean necesarias.

Este proceso se ha de repetir con todos los modelos creados, incluida la arquitectura, para evitar que en Unity se produzcan "invisibilidades" indeseadas.

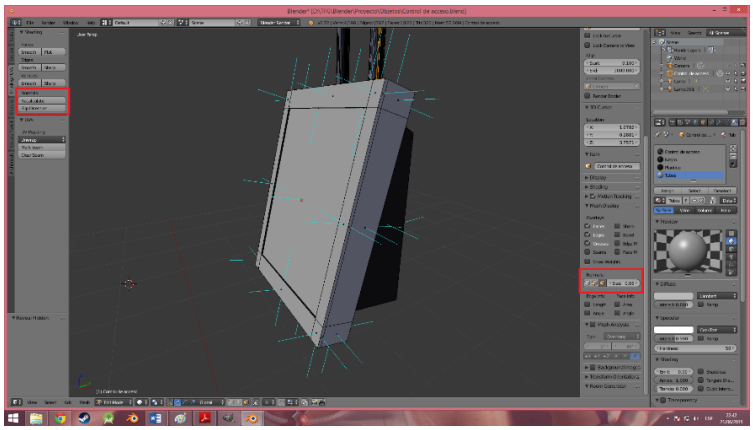


Figura 39. Revisión de normales.

4.2 Unity

Unity es un motor de videojuego gratuito multiplataforma creado por Unity Technologies. Nos servirá para dotar de vida el entorno y permitir la exploración tridimensional del edificio en primera persona.

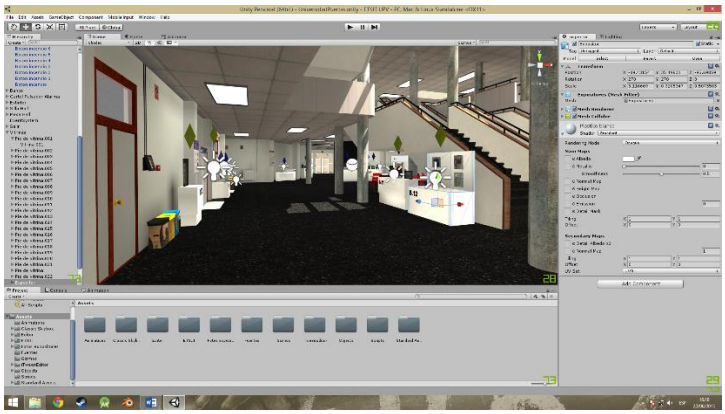
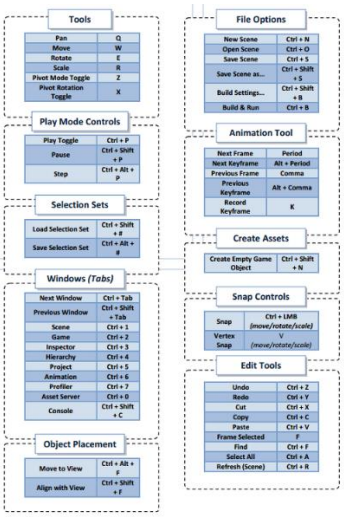


Figura 40. Entorno Unity 5.

Unity dispone de una comunidad enorme y es el motor gráfico más recomendado para iniciarse ya que posee un buen balance entre complejidad de uso y resultados manteniendo un bajo consumo de recursos.

4.2.1 Introducción a Unity

Al igual que en Blender, Unity es un programa que se beneficia mucho del uso de atajos de teclado, y si queremos trabajar de forma ágil y productiva, es muy recomendable que nos familiaricemos con ellos.



Tools

Pan	Q
Move	W
Rotate	E
Scale	R
Pivot Mode Toggle	Z
Pivot Rotation Toggle	X

Play Mode Controls

Play Toggle	Ctrl + P
Pause	Ctrl + Shift + P
Step	Ctrl + Alt + P

Selection Sets

Load Selection Set	Ctrl + Shift + #
Save Selection Set	Ctrl + Alt + #

Windows (Tabs)

Next Window	Ctrl + Tab
Previous Window	Ctrl + Shift + Tab
Scene	Ctrl + 1
Game	Ctrl + 2
Inspector	Ctrl + 3
Hierarchy	Ctrl + 4
Project	Ctrl + 5
Animation	Ctrl + 6
Profiler	Ctrl + 7
Asset Server	Ctrl + 0
Console	Ctrl + Shift + C

Object Placement

Move to View	Ctrl + Alt + #
Align with View	Ctrl + Shift + F

File Options

New Scene	Ctrl + N
Open Scene	Ctrl + O
Save Scene	Ctrl + S
Save Scene as...	Ctrl + Shift + S
Build Settings...	Ctrl + Shift + B
Build & Run	Ctrl + B

Animation Tool

Next Frame	Period
Next Keyframe	Alt + Period
Previous Frame	Comma
Previous Keyframe	Alt + Comma
Record Keyframe	K

Create Assets

Create Empty Game Object	Ctrl + Shift + N

Snap Controls

Snap	Ctrl + LMB (move/rotate/scale)
Vertex Snap	V (move/rotate/scale)

Edit Tools

Undo	Ctrl + Z
Redo	Ctrl + Y
Cut	Ctrl + X
Copy	Ctrl + C
Paste	Ctrl + V
Frame Selected	F
Find	Ctrl + F
Select All	Ctrl + A
Refresh (Scene)	Ctrl + R

Figura 41. Atajos de teclado Unity 5.

Como podemos apreciar en la Figura 40, el entorno es ocupado por diferentes ventanas, la más importante es la vista de la escena, donde obtenemos una visualización previa del proyecto. A la izquierda contamos con la ventana de Jerarquía, equivalente al "Outliner" en Blender, donde veremos organizados todos los objetos que añadamos. A la parte derecha contamos con el Inspector, donde modificaremos los parámetros de los objetos seleccionados. Y por último, en la parte baja, contamos con un explorador del proyecto, donde podemos localizar todos los recursos disponibles, así como scripts o los objetos que importemos desde Blender.

4.2.2 Importar desde Blender

Una de las principales razones para haber elegido los programas Blender y Unity para la creación de nuestro proyecto es la facilidad de flujo de trabajo entre ellas. Para disponer en Unity de los modelos creados en Blender, estudiaremos las opciones de importación disponibles.

Bajo el menú File, encontramos la opción Export, con los diferentes formatos soportados de salida. Los que mejores compatibilidades ofrecen son Autodesk FBX y el formato por defecto Collada. Se pueden utilizar indistintamente sin ningún problema aparente.

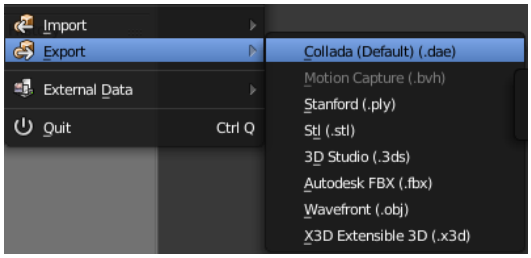


Figura 42. Exportar en Blender.

En las opciones de exportación tendremos que asegurarnos de habilitar el uso de modificadores si los hemos utilizado, así como realizar el copiado de las imágenes que se han utilizado como texturas. Como directorio de guardado seleccionaremos la carpeta dedicada a Assets del proyecto de Unity, y lo organizaremos convenientemente para facilitar su posterior localización.

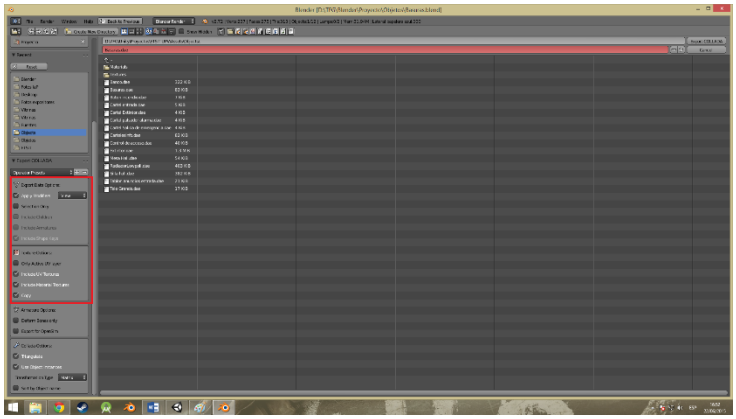


Figura 43. Opciones de exportación.

Una vez guardado, Unity se actualizará con los nuevos assets añadidos y aparecerán en el explorador del proyecto donde se hayan guardado.

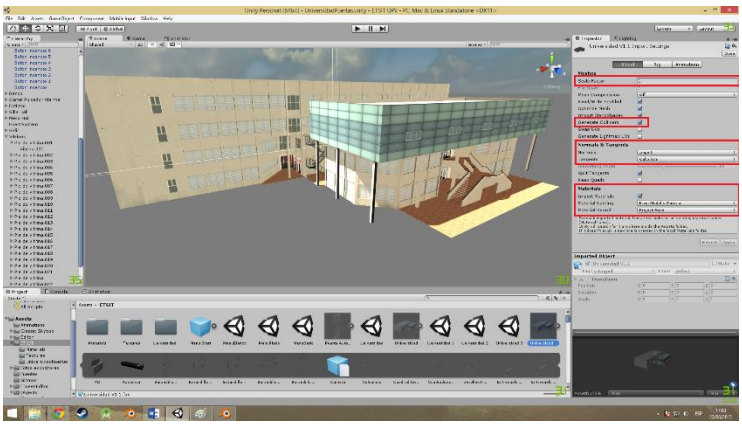


Figura 44. Importación a Unity.

Cuando selecionemos el asset añadido, podemos modificar en el inspector los ajustes de importación. En nuestro caso, si hemos aplicado la escala a los objetos en Blender, podemos dejar el parámetro escala a 1, en cualquier otro caso, lo ajustaremos para obtener el tamaño deseado.

La opción de "Generate Colliders" nos permite que Unity genere de forma automática una malla alrededor de nuestros modelos que funcione como una barrera física, de manera que nuestro jugador no pueda atravesarlos a su antojo.

Continuando con el proceso de importación, si nos hemos asegurado de que las normales de nuestros modelos están situadas en la dirección correcta, podemos importarlas. En caso de que seleccionemos la opción de recalcularlas, Unity aplicará un criterio propio para determinar la dirección de las mismas. Normalmente funciona correctamente, pero cuando tratamos con un edificio que contiene otros modelos en su interior y cuenta con una gran cantidad de habitáculos, los criterios parecen fallar, y obtenemos paredes invisibles en gran parte de nuestro modelado. Para solucionarlo conviene volver atrás, al modelo en Blender, y corregir las paredes que se muestran de forma incorrecta en Unity.

Por último, Unity suele reconocer los materiales aplicados a los objetos y los importa y aplica, pero en caso de que cuenten también con texturas, en ocasiones el proceso falla y no obtenemos los resultados deseados. Podemos jugar con las opciones de importación de materiales hasta conseguir el resultado esperado, o aplicar los materiales que fallen de forma manual.

Figura 45. Organización de carpetas.

Es muy recomendable la organización por carpetas, teniendo por separado los assets junto con las texturas utilizadas para la creación de sus materiales. Al igual que en Blender y en la Jerarquía de Unity, mantener una organización clara y limpia facilita mucho el proceso de trabajo y permitirá que otros programadores puedan entender y trabajar con el proyecto una vez terminado.

4.2.3 Colocar objetos

Una vez importados los objetos, su colocación es muy intuitiva, se puede arrastrar el modelo, también llamado "prefab" a la ventana de la escena y aparecerá de manera inmediata. Ahora utilizaremos las opciones de escala, posición y rotación para colocarlos en su lugar. También serán de mucha utilidad los atajos de teclado para alinear los objetos con la vista actual de la cámara. Una vez colocado un objeto, podemos duplicarlo con CTRL+D para situar todas las copias que sean necesarias.

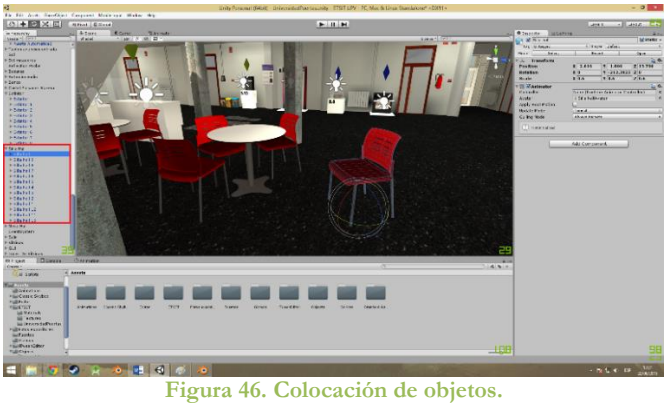


Figura 46. Colocación de objetos.

Cuando hayamos colocado muchos objetos nos daremos cuenta de que nuestra ventana de Jerarquía se ha llenado de copias y ahora es muy difícil navegar por los objetos de la escena, para ello crearemos objetos vacíos bajo la opción "Create Empty" del menú "GameObject". Daremos un nombre representativo a estos objetos vacíos y los utilizaremos para organizar nuestros grupos de objetos.

Figura 47. Organización de jerarquía.

De esta manera mantendremos la claridad y controlaremos la complejidad de la escena.

4.2.4 Iluminación

Una vez colocados todos los objetos podremos apreciar muchos errores en la visualización de las texturas de la escena. Puede que la escena esté completamente a oscuras, que sea demasiado brillante, que la iluminación no sea uniforme, o que las texturas se comporten de maneras inesperadas. Estos errores se deben a la iluminación de la escena.

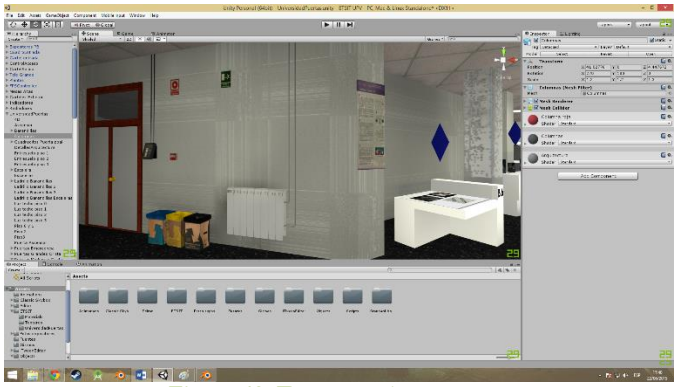


Figura 48. Errores en las texturas.

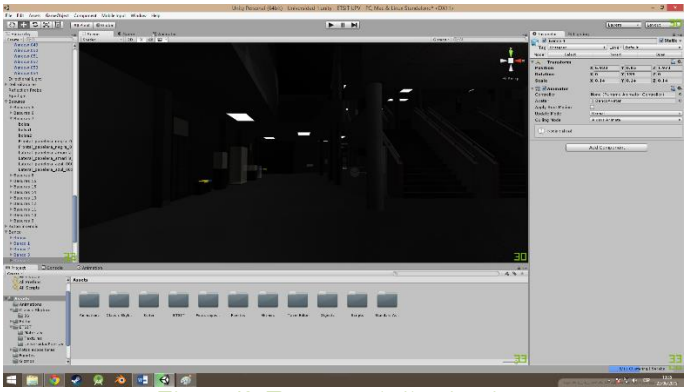


Figura 49. Errores en la iluminación.

Para que el modelado se muestre de la manera esperada tendremos que jugar con muchos de los parámetros con los que cuenta Unity para controlar la iluminación. El proceso es complicado y conseguir los resultados deseados será en ocasiones realmente complicado.

Para empezar colocaremos un "SkyBox" que sustituya el fondo azul que viene por defecto en Unity, para ello accederemos al "Asset Store" y buscaremos un pack de "SkyBoxes" gratuitos.

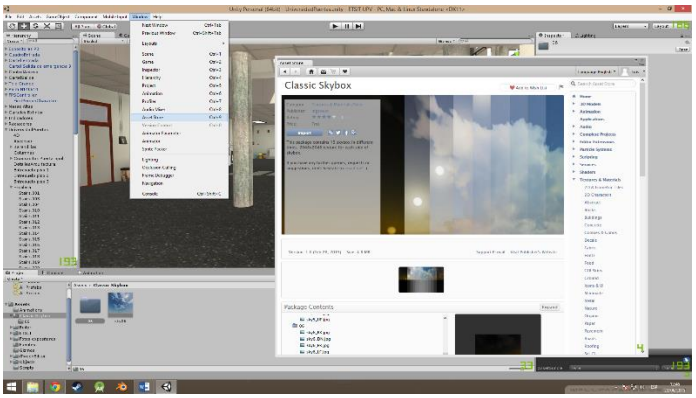


Figura 50. Importar SkyBox.

Una vez importado solo lo tendremos que arrastrar a la escena y quedará como predeterminado. Acto seguido añadiremos un Sol, que será una luz direccional que se encuentra bajo el menú "GameObject". También añadiremos una "Reflection Probe" que actúa como una cámara que nos permitirá capturar la reflexión de la luz de los objetos para recrear su comportamiento físico.

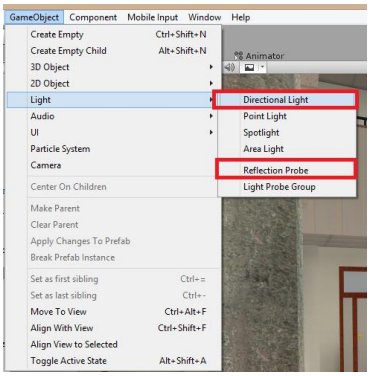


Figura 51. Sol y Reflection Probe.

Por un lado nos aseguraremos de que la rotación del Sol colocado simula la luz que estaría produciendo el Sol representado en nuestro "SkyBox", y por otro lado, ajustaremos sus parámetros hasta conseguir un resultado de luz natural. Unity trabaja con luces en tiempo real, y luces precocinadas, o "baked". Ajustaremos el número de luces de cada tipo para conseguir un rendimiento adecuado sin descuidar el aspecto visual.

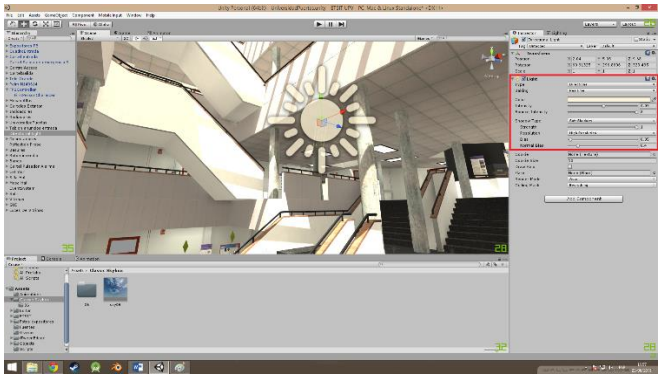


Figura 52.Parámetros Sol.

También podemos utilizar el "SkyBox" para añadir una luz ambiental, para ello, bajo el menú "Window", accedemos a la opción de "Lighting", donde podemos indicar el punto de luz que representa el Sol, la intensidad de la luz ambiente y el trato de la Iluminación Global.

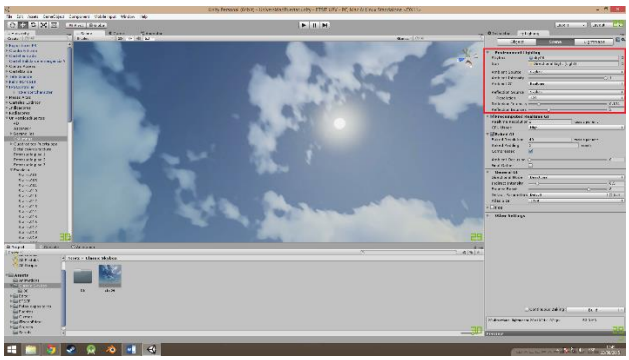


Figura 53.Parámetros de iluminación.

Otra fuente de luz en la escena son las lámparas situadas en el techo, por ello, editaremos su material para que emita luz.

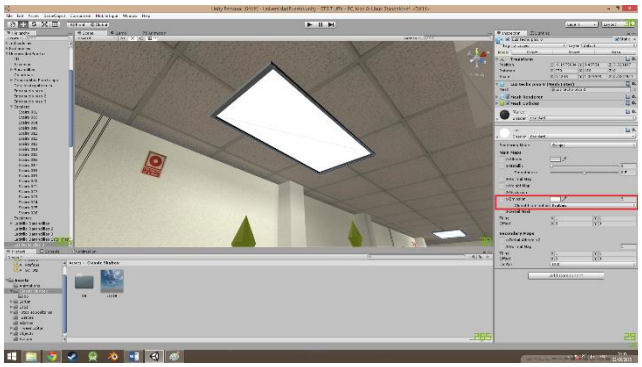


Figura 54.Luces del techo.

Siguiendo con el proceso de iluminación, el objetivo principal de la visita consiste en la visualización de los objetos situados en los expositores, por ello también se han colocado puntos de luz discretos situados frente a los puntos de interés.

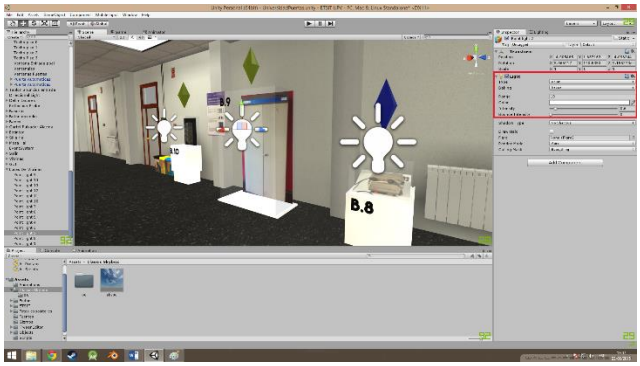


Figura 55.Iluminación de expositores.

Si realizáramos el cálculo de luces en este momento, el resultado no sería el esperado, ya que debemos indicar el comportamiento de los objetos, para que la luz se calcule sobre ellos, de esta manera, tendremos que indicar cuáles son los objetos móviles de la escena y cuales los estáticos.

Esta opción se selecciona desde el inspector, y se habilitará en todos los objetos que no se van a mover en la escena, que son la gran mayoría, incluida la arquitectura del edificio.

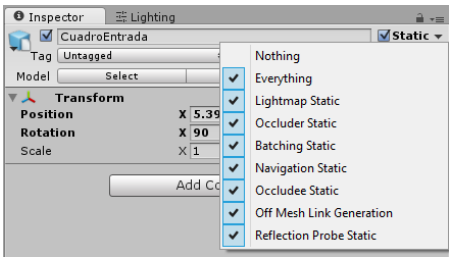


Figura 56.Objetos estáticos.

Dicha opción no se activará para objetos como los indicadores que flotan sobre los expositores, o las puertas automáticas de la entrada, que se tendrán que abrir a nuestro paso y se animarán posteriormente. Es importante que tampoco la activemos para las propias vitrinas de cristal de los expositores, ya que se produciría un efecto de opacidad que nos impediría ver en su interior.

Una vez finalizada la colocación de toda la iluminación y ajustados los comportamientos de los objetos, se puede proceder a "cocinar" las luces que sean de ese tipo, calcular su comportamiento es un proceso costoso para el ordenador, y esto nos permitirá almacenar los resultados de las sombras producidas en unos mapas de luz, o "lightmaps" que se aplicarán como texturas sobre el modelado para conseguir un resultado realista sin mucha carga computacional en tiempo real.

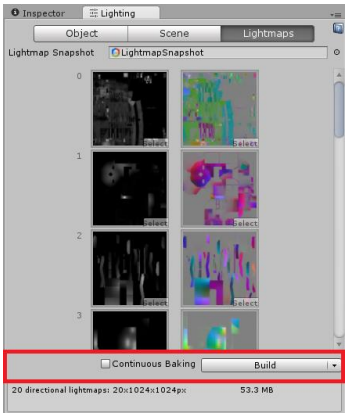


Figura 57.Calcular la iluminación.

La opción de "Continuous Baking" permite que la luz de la escena se actualice y recalcule frente a cualquier cambio, pero puesto que los cambios son constantes, y el consumo de CPU de esta tarea es muy alto, dejaremos la opción desactivada e iniciaremos el proceso cuando queramos obtener el resultado. El proceso de cálculo tarda en torno a 5 horas para una escena de esta complejidad con un ordenador potente (procesador i7-4790), por ello nos convendrá realizar el mínimo número de iteraciones posibles, aunque serán necesarias varias para acercarnos al resultado deseado final.

Todos los ajustes de iluminación han sido testados bajo diferentes condiciones y los parámetros de las luces se han modificado varias veces aproximando cada vez el resultado esperado. Se han necesitado muchas iteraciones para conseguir el resultado final y se puede concluir que la iluminación es igual de importante que el propio modelado para conseguir una experiencia satisfactoria.

Figura 58.Resultado iluminación.

4.2.5 Programación

En este apartado se tratará la introducción del jugador dentro de la escena y de todos los procesos que requieren algún tipo de programación. Lo cual nos permitirá explorar la escena y modificar los comportamientos para determinar la experiencia final del usuario.

En primer lugar colocaremos nuestra cámara. Unity cuenta con un controlador prefabricado para los juegos en primera persona que nos ahorrará mucho trabajo, y que utilizaremos importándolo desde el siguiente menú.

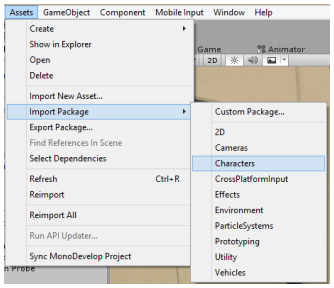


Figura 59.Importar FPSController.

Una vez importado podremos arrastrarlo al lugar de la escena donde queremos que inicie el recorrido. También modificaremos todos los parámetros para conseguir la experiencia de usuario deseada. Desde la velocidad a la que se mueve, hasta la sensibilidad de la cámara.

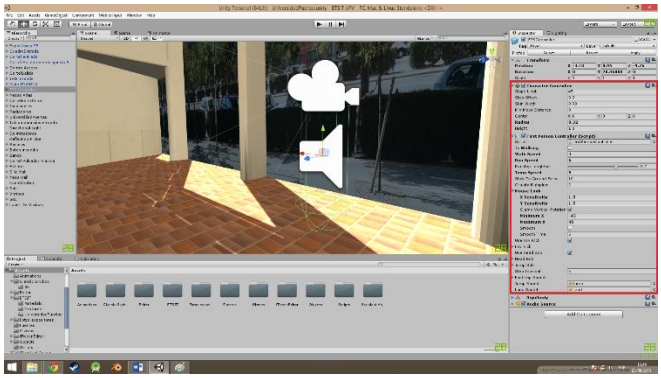


Figura 60.ParámetrosFPSController.

Si queremos modificar el comportamiento de este controlador, podemos abrir el script "FirstPersonController.cs" y ajustar las líneas de código necesarias. En nuestro caso, se ha eliminado la posibilidad de que el personaje salte.

El código utilizado para el movimiento del jugador se encuentra en el Anexo [13].

Cuando se pulse el botón de play, se colocará al jugador en la escena y podrá explorarla.

El primer obstáculo que se encuentra nuestro personaje son las puertas de entrada al edificio. Podemos animarlas de diferentes maneras, mediante el uso de "keyframes" que sería la animación más convencional, a través exclusivamente de código, o mediante iTween, una extensión de Unity dedicada a animación que podemos obtener desde el "Asset Store".

La opción recomendada es el uso de "keyframes", donde crearemos una animación del comportamiento esperado cuando se activen ciertas condiciones, o "triggers".

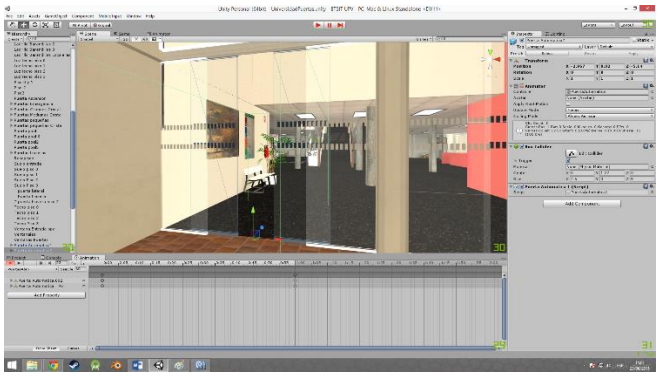


Figura 61.Animación Puertas.

Posteriormente se utilizará el animator integrado en Unity y un poco de código que gestione el comportamiento esperado de las variables para conseguir una animación de apertura de puertas.

El código utilizado para la apertura de puertas se encuentra en el Anexo [13].

Para animar los iconos sobre los expositores bastará con una pequeña animación en iTween, su uso es muy sencillo e intuitivo.

4.2.6 UI

La visita al edificio ya es realizable, pero el objetivo de dicha visita consiste en poder obtener información de los objetos contenidos en los expositores, para ello tendremos que crear una interfaz de usuario que se comporte acorde a unas directrices que se manejarán mediante código y el uso de "triggers" en forma de cubos de colisión que se situarán en las posiciones donde se quiera interactuar con el usuario.

La creación de interfaces de usuario en Unity se ha hecho mucho más intuitiva a raíz de las últimas actualizaciones, y nos permite trabajar de manera gráfica modificando las vistas que se mostrarán al usuario.

Se recomienda adquirir conocimientos previos visualizando una serie de tutoriales recopilados en el Anexo [9].

Para empezar se crearán las cajas que nos servirán para disparar los eventos. Esto se realizará creando cubos corrientes, eliminando su componente "Mesh Renderer" y habilitando la opción "Is Trigger" en sus parámetros. Posteriormente se les añadirá un script con el que determinaremos que lienzo, o "Canvas" mostrará al usuario, y cuál será el comportamiento de los botones mostrados en dicho "Canvas".

El código utilizado para gestionar los "Canvas" se encuentra en el Anexo [13].

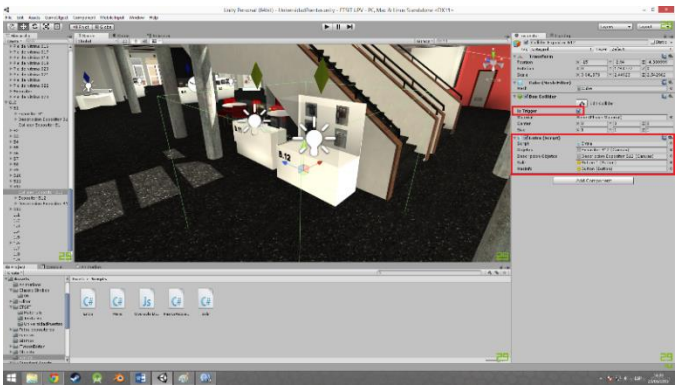


Figura 62.Creación de eventos.

De esta manera, procederemos a la creación de los "Canvas" a mostrar. Utilizando los conocimientos adquiridos tras la visualización será fácil disponer los paneles necesarios y rellenarlos de imágenes y textos. Dicha información ha sido facilitada por María Carmen Bachiller Martín, responsable del museo.

Para que los "Canvas" sean universales y tengan la misma apariencia independientemente de la resolución de la pantalla en la que se están visualizando, tendremos que modificar algunas de las opciones que se muestran a continuación.

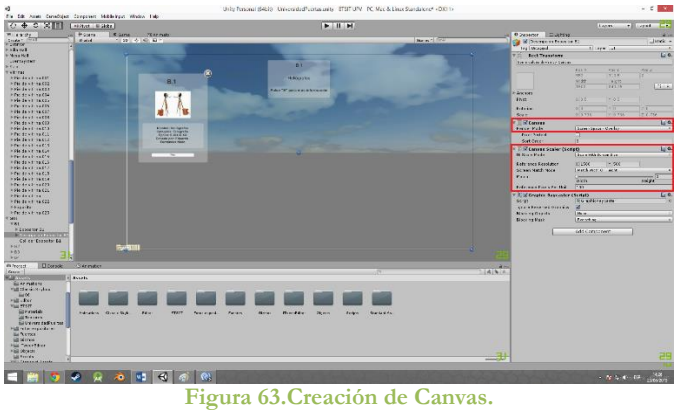


Figura 63.Creación de Canvas.

En caso de que el objeto contenga información adicional en la página web, se implementará una llamada desde la aplicación a la web para poder visualizarla.

En caso de que el expositor contenga varios objetos, ajustaremos la vista del "Canvas" y añadiremos la posibilidad de realizar un "scroll" mediante el componente "Scroll Rect" y una "Mask" para que el resultado sea el esperado.

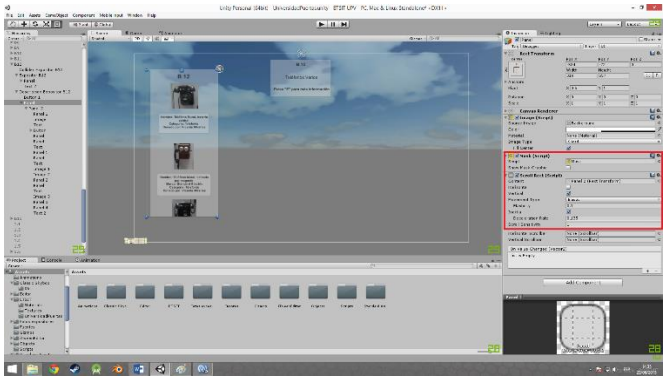


Figura 64.Creación de Canvas con scroll.

Cada caja de colisión llamara al "Canvas" correspondiente, obteniendo una forma organizada de mostrar la información. Y lo más importante, completamente editable tanto en contenido como en aspecto gráfico y ampliable dependiendo de las necesidades.

También se añadirán interfaces gráficos para mostrar los controles de la aplicación, un menú para salir de la aplicación, o el propio menú de inicio, que se creará en una escena aparte.

Figura 65.Menú de inicio.

4.2.7 Build del proyecto

Una vez determinado el aspecto completo de la aplicación podemos proceder a su construcción. Para ello primero determinaremos el nivel de calidad de la aplicación dependiendo de la plataforma para la que lo vayamos a dirigir, ya que como se ha mencionado anteriormente, Unity es un motor multiplataforma.

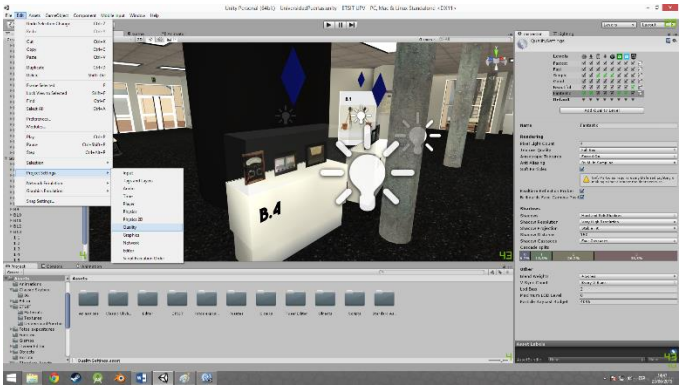


Figura 66.Quality Settings.

Una vez ajustada la calidad deseada podemos proceder a construir la aplicación. Seleccionaremos la plataforma destino y las escenas que debe incluir en la aplicación. La interacción entre escenas se controlara vía código adjunto en el Anexo.

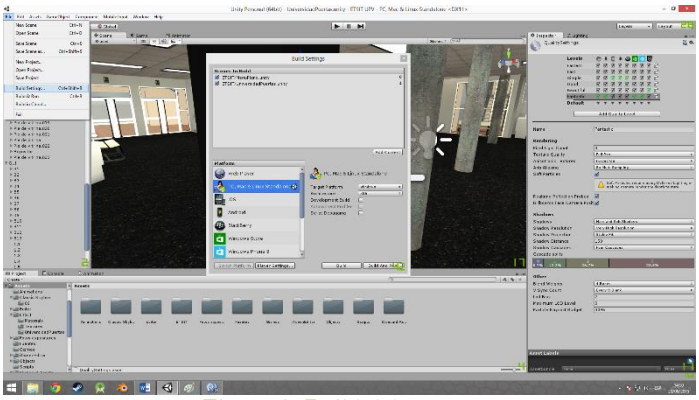


Figura 67.Build del proyecto.

Una vez finalizada la construcción, contaremos con un ejecutable junto con una carpeta con los recursos necesarios para su ejecución. De la misma manera que se ha procedido durante todo el proyecto, se iterará hasta conseguir el resultado deseado.

Capítulo 5. Resultados del trabajo

Se ha obtenido una aplicación completamente funcional que podría distribuirse al público tras algún pequeño retoque. El coste material del proyecto, salvo por el ordenador desde el cual se ha trabajado, ha sido nulo, al igual que el coste del software necesario para llevarlo a cabo. Esto ha sido posible gracias al uso de tecnologías gratuitas que como hemos podido observar, son capaces de obtener grandes resultados.

5.1 Visita virtual

La aplicación final permite la visita virtual del museo de las Telecomunicaciones Vicente Miralles de manera completamente funcional. El uso de mobiliario y objetos adicionales enriquecen la experiencia y ayudan al usuario a sumergirse en el entorno.

La experiencia es fluida e interactiva, y ofrece todo lo que se podría esperar de una visita virtual en primera persona.

5.2 Interacción con el usuario

Los controles con los que se navega por la aplicación son los mismos que se utilizan en todos los videojuegos en primera persona actuales. El diseño de menús se ha realizado con la intención de ser intuitivo y muestra toda la información acerca de los objetos históricos que forman el museo.

5.3 Integración de la página web

Puesto que el museo cuenta con una página web con información ampliada y recursos como imágenes y videos adicionales, se ha integrado una solución para enlazar a la web desde la aplicación de visita virtual facilitando un flujo que no interrumpa la experiencia de la visita.

Capítulo 6. Conclusiones y propuesta de trabajo futuro

Se puede concluir que con tiempo y una buena idea, actualmente se dispone en la red de todas las herramientas necesarias para la creación de este tipo de aplicaciones desde casa, sin necesidad de tratarse de un estudio profesional y haciendo uso de herramientas gratuitas.

Se ha impulsado desde cero una idea cultural para saciar a todos los interesados en la historia de las Telecomunicaciones y se ha facilitado su acceso a cualquier persona que disponga de un ordenador.

Sin embargo, aún hay muchas formas de mejorar la experiencia final.

6.1 Plataformas

Actualmente se ha construido la aplicación como un "Stand Alone" soportado por Windows, Mac y Linux, que si bien no necesita de ningún otro programa para ser ejecutado, es necesario disponer de la aplicación en el almacenamiento del ordenador. Lo cual requeriría una descarga por parte del usuario.

Para solucionarlo podríamos utilizar una nueva funcionalidad incorporada a Unity, la construcción de la aplicación para WebGL. WebGL es un API para javascript que permite usar la implementación nativa de OpenGL ES 2.0 incorporada en los navegadores para permitir la aceleración con el uso de tarjetas gráficas. De este modo la aplicación podría ir enviando al usuario los datos necesarios sobre http para su ejecución y funcionar como si se encontrara instalada en el propio equipo del usuario.

Aprovechando la flexibilidad de Unity como motor multiplataforma, también se podría migrar la aplicación a dispositivos móviles, lo cual ampliaría su repercusión. Con un ajuste en los controles de la aplicación para adaptarlos a pantallas táctiles se podría conseguir una versión para tablets android y para iPads.

Por último, con el auge de las tecnologías de realidad virtual, se podrían usar las últimas opciones de construcción de proyectos incorporadas a Unity en la versión 5.1.1 que añaden el soporte para Oculus Rift, adaptando el ángulo de visión y las características de la cámara utilizada en el proyecto para que se pueda reproducir a través de este dispositivo.

6.2 Optimización

Un factor más a tener en cuenta es la optimización de la aplicación. A lo largo del proceso de creación se han ido añadiendo muchos modelos, incrementando la cantidad de objetos que tiene que procesar el motor del gráfico en tiempo real. Esto no es un problema para ordenadores con tarjetas gráficas dedicadas, pero para ordenadores portátiles con gráficas integradas o para dispositivos móviles, la aplicación puede resultar demasiado exigente.

Para evitar un funcionamiento lento, se tendrán que revisar las opciones de calidad de Unity mencionadas anteriormente y en el caso de los dispositivos móviles, probablemente habría que estudiar eliminar ciertos objetos decorativos como las sillas o el mobiliario ajeno al museo. También se podría reducir la calidad de la Iluminación Global o de las sombras, así como de las reflexiones hasta obtener un rendimiento adecuado para este tipo de dispositivos.

6.3 Integración

De la misma manera que se ha trabajado para integrar la información de la página web del museo de forma transparente con la aplicación. También se podrían integrar futuras aplicaciones que complementaran o ampliaran la experiencia del usuario.

Un ejemplo pueden ser las aplicaciones que están siendo desarrolladas en estos momentos por otros departamentos de la escuela consistentes en la recreación virtual de objetos del museo que se encuentren dañados, o la simulación del funcionamiento de algunos de los objetos más interesantes, como el fonógrafo y el gramófono para los cuales se permite al usuario escuchar una simulación de cómo sonaría una canción reproducida con dichos aparatos.

Estas aplicaciones podrían incorporarse a la visita virtual del museo para ofrecer una experiencia con un valor añadido y un grado de interacción sorprendente.

6.4 Crecimiento de la aplicación

La aplicación en la actualidad satisface los requisitos que se plantearon a su inicio, pero siempre se puede seguir depurando para ofrecer cada vez un mejor resultado. Un ejemplo de ello es que los objetos situados dentro de las vitrinas actualmente se muestran como una foto bidimensional situada en el interior. Se podrían utilizar diversas herramientas para reconstruir modelos en 3D de dichos objetos, para poder mostrarlos con todo lujo de detalle en la visita virtual.

6.4.1 Autodesk 123D Catch

La herramienta gratuita 123D Catch, de Autodesk, nos permite crear un modelo 3D de un objeto a partir de diversas (30+) fotos tomadas alrededor del objeto a diferentes alturas utilizando información del acelerómetro de los dispositivos móviles.

No se capturará únicamente el objeto que deseamos, también la superficie sobre la que se encuentra, pero podemos seleccionar y eliminar todas las partes que no nos interesen y luego lo podemos exportar en formato .obj a Blender.

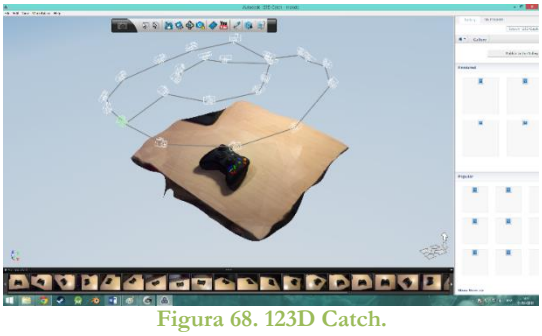


Figura 68. 123D Catch.

El único problema es que los objetos se exportan con un número de caras muy elevado (aunque el programa tiene unas preferencias de calidad), lo cual haría nuestra escena inmanejable. Para solucionarlo podemos recurrir al modificador "Decimate" del que dispone Blender, pero en algunos casos la reducción de caras supone una pérdida enorme de calidad, por lo que es más recomendado utilizar un software específico, como MeshLab.

6.4.2 MeshLab

MeshLab es una herramienta open source para procesado de mallas. Las posibilidades que ofrece son abrumadoras, pero nosotros la utilizaremos solo para reducir el número de caras de nuestros objetos.

Para ello utilizamos la opción bajo la opción Filters > Remeshing, simplification and construction > Quadratic Edge Collapse Detection (with texture). Las opciones determinan la libertad que tendrá el programa para reducir el número de caras.

Podemos observar que el modelo apenas pierde calidad al reducirlo de las 55.000 caras iniciales a 3.000. Podemos seguir reduciendo como creamos conveniente y luego utilizar Blender para corregir algún pequeño fallo que se haya ocasionado.

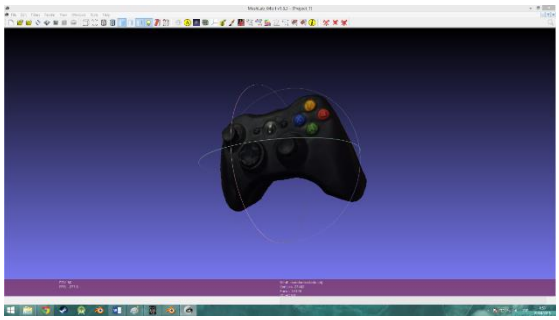


Figura 69. 55.000 caras.

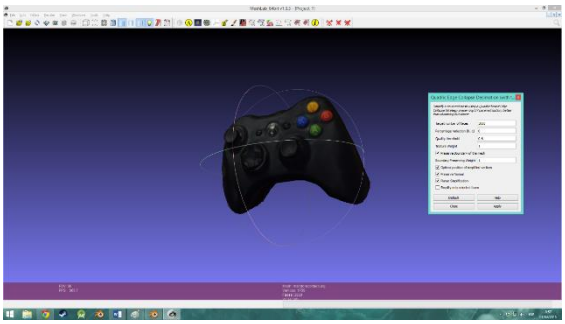


Figura 70. 3.000 Caras.

Una vez exportado a Blender podremos proceder como con cualquier otro objeto que se ha creado con anterioridad y añadirlo a Unity para colocarlo dentro de un expositor.

Si tener todos los objetos a la vez en la escena supusiera una carga gráfica muy grande, se podría optar por un sistema de carga por zonas, mostrando inicialmente todos los objetos como fotos bidimensionales y sustituyéndolos por sus modelos 3D al acercarnos a los diferentes expositores.

De esta forma se crearía una experiencia realmente cercana a la realidad, poniendo el conocimiento al alcance de cualquier usuario de una forma interactiva y visual.

Capítulo 7. Referencias y Anexos

Uso de Blender

[1] Manual de Blender.

[2] Tutoriales de Blender.

Addons de Blender

[3] Archimesh.

[4] Window Generator.

Obtención de texturas

[5] Texturer.

[6] CGTextures.

Uso de Unity

[7] Manual de Unity.

[8] Tutoriales oficiales Unity.

[9] Tutoriales Interfaz de Usuario.

Herramientas utilizadas

[10] GIMP 2.

[11] Autodesk 123D Catch.

[12] MeshLab.

Código utilizado en el proyecto.

[13] Scripts utilizados.

Nota: Todos los enlaces utilizados se encuentran activos a fecha 23/06/2015.