

EL PROYECTO DE ARQUITECTURA EN UNREAL 3D PARA INMERSIÓN VIRTUAL

Caso 1: El Pabellón Mies

Raúl Ferrándiz López

Tutoras: Susana Iñarra Abad
Marina Sender Contell

Curso 2015-2016
Grado en fundamentos de la arquitectura



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCOLA TÈCNICA
SUPERIOR
D'ARQUITECTURA

EL PROYECTO DE ARQUITECTURA EN UNREAL 3D PARA INMERSIÓN VIRTUAL

| Caso 1: El Pabellón Mies

Raúl Ferrándiz López

Tutoras: Susana Iñarra Abad
Marina Sender Contell

Curso 2015-2016
Grado en fundamentos de la arquitectura



Abstract

The present final degree assignment consists on the realization of an immersive virtual model with the game engine "Unreal Engine". It is used with the Barcelona Pavilion, located in this city and designed by the renowned german architect Mies Van der Rohe. The memory has been divided in two sections. In the first one, a brief historical review of the evolution of Virtual Reality had been made. Different software architectural graphic representation that are available today and the technical advances had been revised too. The necessary information to know the tools that we have is obtained. In the second section it has conducted a study of the work in order to obtain a three dimensional model of it. This virtual model is modeled and mapped in 3D Studio Max program and later in the game engine "Unreal Engine" the treatment for real-time visualization is completed. In the latter software the most innovative part for architectural computer graphics is developed. Finally, in the conclusions is verified that it is a tool which open a range of possibilities to the architect with the creation of interactive spaces in real time, improving the understanding of the simulated space.

Key words: Virtual Reality, VR, Unreal Engine, Barcelona Pavilion, architectural computer graphics, real-time renderer.

Resum

El present treball de fi de grau consisteix en la realització d'un model virtual immersiu amb el motor de videojocs Unreal Engine. Es treballa amb el Pavelló de Barcelona, situat en aquesta mateixa ciutat i projectat per l'arquitecte alemany de renom Mies Van der Rohe. S'ha dividit la memòria en dos apartats. En el primer s'ha fet una breu revisió històrica sobre l'evolució dels sistemes de Realitat Virtual, els diferents softwares de representació gràfica arquitectònica que es disposen avui dia i els avenços tècnics que han patit els videojocs. S'obté la informació necessària per conèixer les eines de les que disposem. En el segon apartat s'ha realitzat un estudi de l'obra amb la finalitat d'obtenir un model tridimensional de la mateixa. Aquest model virtual es modela i mapeja en el programa 3D Studio Max i posteriorment, en el motor de videojocs Unreal Engine, es completa el seu tractament per a la visualització en temps real. En aquest últim software, es desenvolupa la part més innovadora per a la infografia arquitectònica. Finalment en les conclusions es verifica que es tracta d'una eina que obri un ventall de possibilitats a l'arquitecte amb la creació d'espais interactius en temps real, millorant la comprensió de l'espai simulat.
Palares clau: Realitat Virtual, VR, Unreal Engine, Pavelló de Barcelona, Infografia arquitectònica, renderitzat en temps real.

Palares clau: Realitat Virtual, VR, Unreal Engine, Pavelló de Barcelona, Infografia arquitectònica, renderitzat en temps real.

Resumen

El presente trabajo de fin de grado consiste en la realización de un modelo virtual inmersivo con el motor de videojuegos Unreal Engine. Se trabaja con el Pabellón de Barcelona, situado en esta misma ciudad y proyectado por el arquitecto alemán de renombre Mies Van der Rohe. Se ha dividido la memoria en dos apartados. En el primero se ha realizado una breve revisión histórica sobre la evolución de los sistemas de Realidad Virtual, los diferentes softwares de representación gráfica arquitectónica que se disponen hoy en día y los avances técnicos que han sufrido los videojuegos. Se obtiene la información necesaria para conocer las herramientas de las que disponemos. En el segundo apartado se ha realizado un estudio de la obra con la finalidad de obtener un modelo tridimensional de la misma. Este modelo virtual se modela y mapea en el programa 3D Studio Max y posteriormente, en el motor de videojuegos Unreal Engine, se completa su tratamiento para la visualización en tiempo real. En este último software, se desarrolla la parte más innovadora para la infografía arquitectónica. Por último en las conclusiones se verifica que se trata de una herramienta que abre un abanico de posibilidades al arquitecto con la creación de espacios interactivos en tiempo real, mejorando la comprensión del espacio simulado.

Palabras clave: Realidad Virtual, Unreal Engine, Pabellón de Barcelona, Infografía arquitectónica, renderizado en tiempo real.

Agradecimientos

Agradecer en primer lugar a todo el profesorado de la Escuela Técnica Superior de Arquitectura y en especial a los profesores de expresión gráfica por haberme inculcado esta pasión por el dibujo y todo lo que conlleva.

Y a mis compañeros, que siempre me apoyan en lo que hago. En especial a Miguel, que siempre que lo he necesitado me ha prestado su equipo para la realización de este trabajo.

Por último gracias a mi familia y amigos por siempre estar a

Índice

1 · INTRODUCCIÓN	10
2 · INFORMÁTICA GRÁFICA	13
2.1 · Evolución histórica	14
2.2 · Aplicada a la arquitectura	18
3 · REALIDAD VIRTUAL	25
3.1 · Qué es la RV	26
3.2 · Breve introducción histórica	28
4 · VIDEOJUEGOS	31
4.1 · Evolución gráfica	32
4.2 · Games Engines	35
5 · CASO PRÁCTICO	37
5.1 · Pabellón de Barcelona	38
5.2 · Conceptos de renderizado en tiempo real	42
5.3 · Caso práctico. Modelo: Pabellón de Barcelona	45
6 · CONCLUSIONES	79
Bibliografía	84
Enlaces Web	85
Índice de imágenes	86

1 | INTRODUCCIÓN

INTRODUCCIÓN

A lo largo de la historia de la arquitectura el modo de comunicarse de los arquitectos siempre ha sido a través del dibujo. Esta técnica de representación ha ido variando y evolucionando a lo largo del tiempo, desde los dibujos en diédrico hasta las vistas cónicas, que son las más comúnmente utilizadas para expresar y comunicar un espacio.

El desarrollo tecnológico, la aparición de los ordenadores y los softwares de CAD, produjeron que la herramienta más utilizada fuese, y siga siendo, el ordenador. Estas técnicas permiten ahorrar tiempo en el delineado de planos, hacer volumetrías en tres dimensiones e incluso realizar vistas foto realísticas de un espacio.

Con los avances en el campo de los gráficos por ordenador, sobre todo gracias a la mejora técnica de los videojuegos, se ha conseguido crear nuevas formas de representación de espacios con visualización en tiempo real.

Hasta hace poco nuestros equipos sólo permitían visualizar imágenes estáticas, ahora es posible recrear espacios con movimiento y poder interactuar con ellos con gran calidad gráfica. Estas tecnologías pueden ser usadas por ordenadores personales, dispositivos móviles (tablets, smartphones...), videoconsolas o incluso desde páginas web, abriendo el campo de información.

En el siguiente trabajo se estudia la informática gráfica, la realidad virtual y los videojuegos, para comprender en qué momento se está. Y posteriormente, una vez comprendido qué relación tienen estas materias, se realiza un caso práctico.

El caso práctico consiste en la creación de un modelo digital arquitectónico para su visualización en tiempo real con un motor de videojuegos.

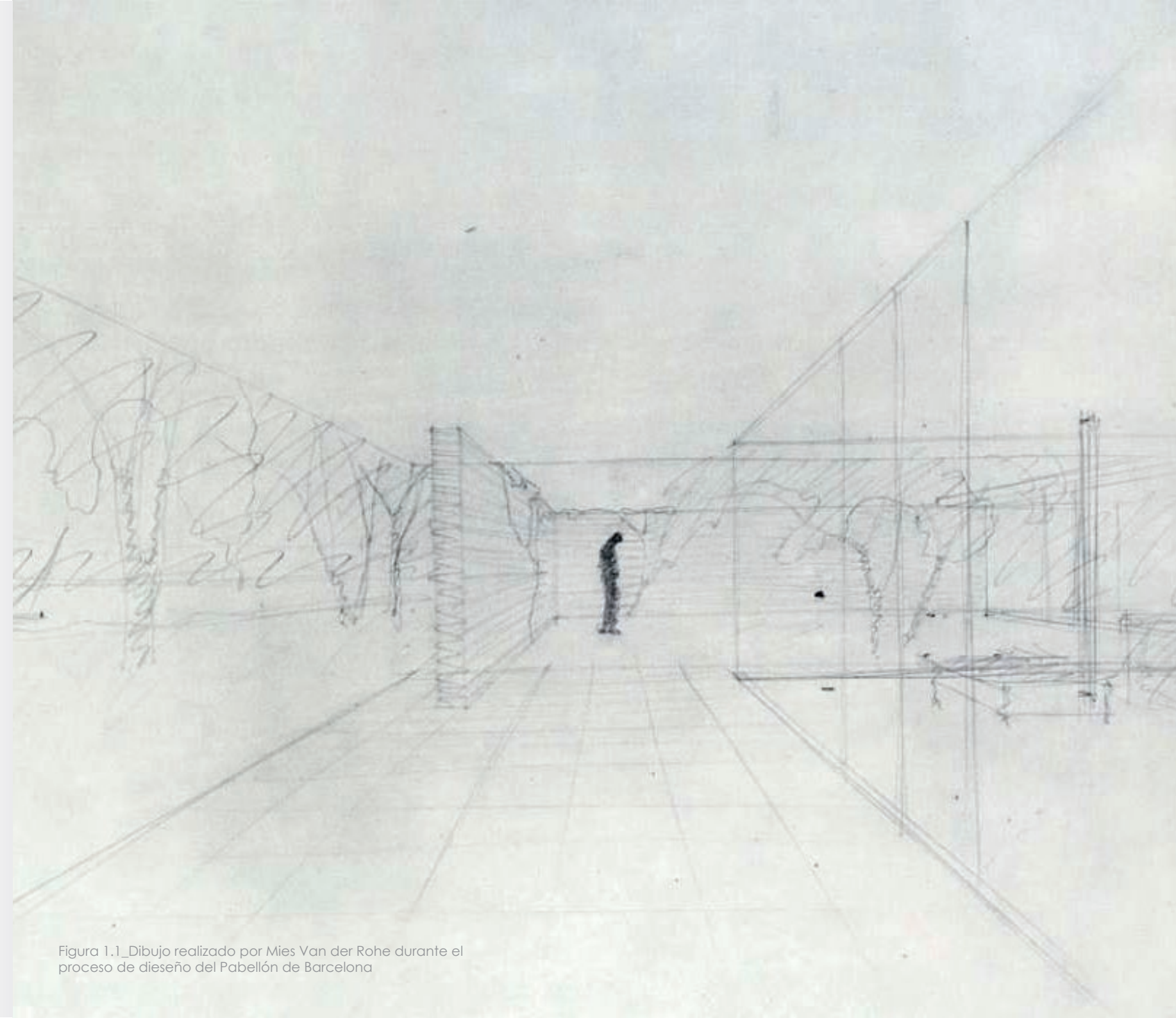


Figura 1.1_Dibujo realizado por Mies Van der Rohe durante el proceso de diseño del Pabellón de Barcelona

2.1 · INFORMÁTICA GRÁFICA

Evolución histórica

Se podría definir a la informática gráfica como el conjunto de técnicas que permiten crear una imagen por ordenador. Hoy en día es un campo presente en multitud de sectores como pueden ser el cine, el diseño, la medicina, la arquitectura, los videojuegos... Pero nació con la intención de transmitir la información al usuario de una manera comprensible, dejando a un lado los datos numéricos. El aumento de la capacidad de procesamiento de los ordenadores ha permitido que actualmente disfrutemos de imágenes mucho más sofisticadas, realistas e impactantes. Siendo este un campo que está en constante evolución.

El término "gráficos por ordenador" (en inglés "computer graphics" o "CG" es un término habitual) fue acuñado en los años 60 por William Fetter, diseñador gráfico de Boeing, para definir su trabajo (Carlson, W 2003). Aunque la historia comienza antes.

A finales de los ochenta se había inventado una tecnología que permitía visualizar imágenes mediante un haz de rayos catódicos, llamada CRT (cathodic rays tube). El invento permitía generar luz mediante la colisión de un electrón lanzado desde el cátodo a la superficie fosforescente del ánodo. Este tubo, desarrollado por William Crookes, fue la base para los primeros televisores y monitores.

El primer ordenador en trabajar con representación gráfica de datos en tiempo real fue la computadora *Whirlwind* que fue desarrollada en el MIT (Massachusetts Institute of Technology) en 1944 para las Fuerzas Armadas estadounidenses. El objetivo de su creación era crear una máquina capaz de simular un vuelo de entrenamiento para los pilotos de bombarderos.



Figura 2.1_ Jay Forrester con los desarrolladores y programadores del Whirlwind

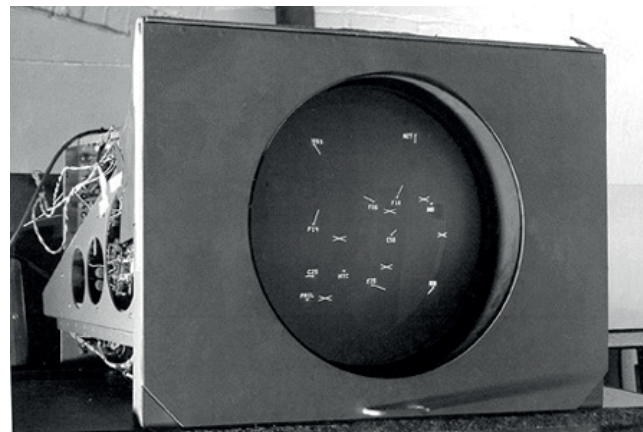


Figura 2.2_ Monitor del Whirlwind

La importancia de este hecho es que hasta la fecha ninguna computadora había trabajado en tiempo real, es decir trabajaba con entrada de datos y salida gráfica instantáneamente, sin tener que esperar un tiempo para ver el resultado. Para el simulador se diseñó también un dispositivo de introducción de datos en tiempo real, un puntero, que asemejaríamos a lo que hoy en día es un ratón de ordenador.

El que es considerado el padre de los gráficos interactivos por ordenador, Ivan E. Sutherland, diseñaría para la TX-2, una máquina con pantalla gráfica y un lápiz óptico incorporado capaz de interactuar con ella, financiada por las Fuerzas Aéreas Estadounidenses, el Sketchpad en 1961. Se trataba de un programa que permitía realizar dibujos técnicos directamente sobre la pantalla con el lightpen, teniendo algunas herramientas innovadoras para la época como hacer zoom, trabajar a escala, borrar, trazar rectas... que hoy en día son muy comunes en los programas de CAD. Siendo así el primer programa de dibujo por ordenador.

A finales de los 50 IBM y General Motors unirían fuerzas para crear un sistema de diseño gráfico para sus automóviles con el fin de optimizar el tiempo. Así desarrollarían el DAC-1 (*Design Automated by Computer*) en 1959 que no se terminaría hasta principios de los sesenta. Se podría considerar que el primer CAD comercial fue DAC-1 (Krull, F.N. 1994). Se trataba de un Software de sistema de representación de 3D, que permitía dibujar y diseñar en tiempo real los



Figura 2.3_ Ivan E. Sutherland



Figura 2.4_ Ivan E. Sutherland mostrando el SketchPad



Figura 2.5_ Diseñador de General Motors trabajando con el software DAC-1

16

automóviles de la compañía. Esto generó un gran impacto en el campo y se comenzó a investigar sobre ello.

En las siguientes décadas se realizan multitud de avances en el campo gráfico e informático. Así por ejemplo en 1963 se realizan los primeros cortos de animación para la simulación de efectos físicos, Lawrence Roberts del MIT desarrolla los primeros algoritmos para ocultar líneas en representaciones tridimensionales, a finales de los sesenta un grupo de investigación de la universidad de Utah realiza el primer algoritmo de superficies ocultas y se interesan por el realismo mediante sombreado de superficies con color. Ivan Sutherland propone el primer algoritmo para eliminar el cálculo de las partes que no se ven en la imagen, optimizando el tiempo de cálculo, desarrollando en paralelo sistemas de realidad virtual y aumentada.

En los años setenta, tras los avances obtenidos en el campo por medio de diversas universidades, el gobierno de los Estados Unidos y las empresas de este país toman conciencia y ven en este sector un potencial en el que investigar. Es por ello que durante esta década se establecen fondos para impulsar los avances en la informática gráfica.

De forma paralela la industria empieza a tomar papel en el sector comercializándose el microprocesador en 1971, se funda Atari en el mismo año, empresa que impulsaría los videojuegos, en el 75 Bill Gates funda Microsoft, en 1979 Lucasfilm crea la división de gráficos por computador para la elaboración de películas, en el 81 esta misma empresa desarrolla el primer motor de renderizado, el REYES, en los ochenta Autodesk empieza a desarrollar un programa de CAD que funcione sobre un PC, Autocad.

Ya en los 90 se estandarizó la plataforma PC (*Personal Computer*) y se empieza a comercializar las tarjetas gráficas de bajo coste y altas prestaciones. Suponiendo esto una oportunidad para que más personas investiguen y creen gráficos en 3D.

A día de hoy, cualquier persona puede disfrutar de equipos con altísimas prestaciones gráficas. Todo ello permite diseñar complejos modelados tridimensionales, renderizar imágenes fotorealistas con mayor agilidad o realizar visualizaciones en tiempo real con un alto grado de definición.

17

2.2 · INFORMÁTICA GRÁFICA

Aplicada a la arquitectura

La incorporación de la informática aplicada a la arquitectura ha venido principalmente de la mano del CAD, debido a su componente gráfica y a sus multitud de opciones que permiten optimizar el proceso de dibujo. Se entiende por CAD (*Computer Aided Design*) o Diseño Asistido por Ordenador los softwares para la creación, modificación, análisis y optimización de diseños. Principalmente para diseños técnicos.

Los CAD son descendientes directos del *Sketchpad* de Sutherland, visto en el apartado anterior, que en un inicio era un programa de delineación permitiendo dibujar línea a línea como cuando se elaboraba un diseño a mano.

18

Los programas CAD trabajan en gráficos vectoriales, a diferencia de otros programas que trabajan en gráficos de mapa de bits (*bits-map*). Esta manera de trabajar va definida por expresiones matemáticas lo cual otorga un alto grado de precisión y definición, dando la posibilidad de realizar grandes zooms, cambiar la escala del dibujo sin perder resolución, construcción de líneas, curvas, círculos o rectángulos perfectamente definidos, etc.

Dependiendo de las características y posibilidades cada programa va dirigido a un sector de usuarios u otro, teniendo programas de carácter general, como podría ser AutoCAD de Autodesk que se trata de un programa genérico, o programas de carácter más específicos.

En consecuencia han ido apareciendo diferentes subgéneros del CAD dependiendo del uso que se le ha ido asignando.

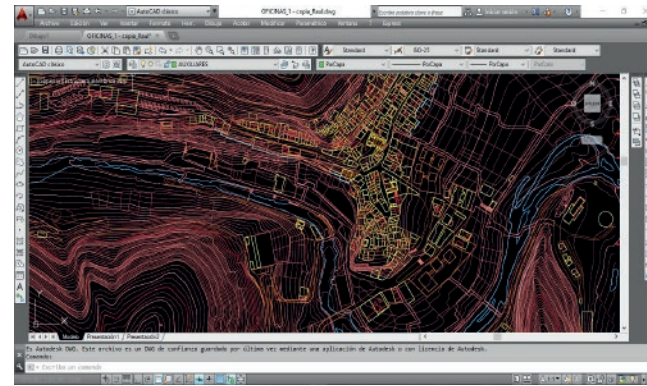


Figura 2.6_ Entorno de trabajo de AutoCAD 2015

CAAD

Sería la herramienta CAD adaptada a la arquitectura, significando *Computer Aided Architectural Design*, traducido como Diseño Arquitectónico Asistido por Ordenador. Los programas de CAAD se diferencian de los genéricos en que incorporan la creación de elementos arquitectónicos tales como puertas, muros, ventanas o detalles constructivos, e incorporan herramientas que facilitan el trabajo del arquitecto. Un ejemplo de este tipo de software sería AutoCAD Architecture de Autodesk.

CAE

Los CAE o *Computer Aided Engineering*, en castellano Ingeniería Asistida por Ordenador, son los programas que se basan en entornos CAD pero además permiten analizar el modelo

de diversas formas, por ejemplo cinemática y dinámicamente, desde el punto de vista de cálculo de fluidos, cálculo térmico o cálculo de presiones. Ejemplos de estos programas serían el ANSYS de ANSYS inc., el ABAQUS de Dassault Systèmes o el NASTRAN de MSC Software.

CAM

Los programas CAM o *Computer Aided Manufacturing*, traducido como Fabricación Asistida por Ordenador son los softwares detrás de las máquinas que manufacturan los productos, las máquinas de control numérico (CNC, por sus siglas en inglés). Estas máquinas son herramientas controladas por un ordenador que se pueden mover libremente en el espacio en dos o tres direcciones y girar. Ejemplos de estas máquinas pueden ser las fresadoras, los tornos, las grabadoras, brazos de soldadura, cortadoras láser...

Gracias a los avances en los programas CAM la introducción de datos en las CNC no es tan laboriosa. Anterior a estos softwares se introducían los datos con código, sin embargo ahora se pueden generar complejos diseños realizados con programas CAD de una forma gráfica y precisa.

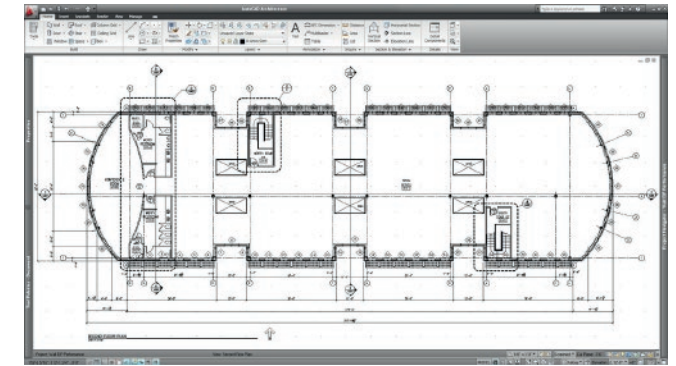


Figura 2.7_ Entorno de trabajo de AutoCAD Architecture

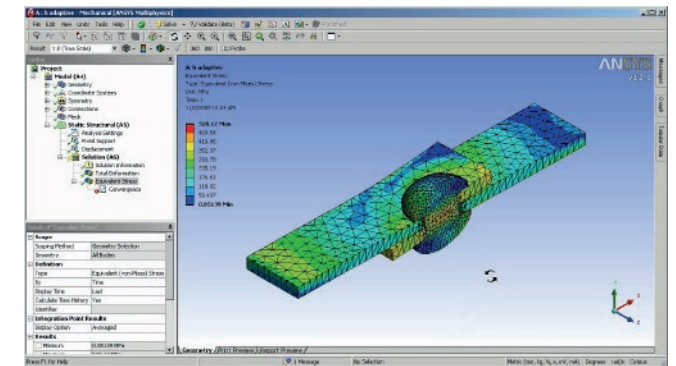


Figura 2.8_ Entorno de trabajo de ANSYS

19

Una vez vista la herramienta más utilizada en las últimas décadas se hace un breve repaso por otros softwares que se utilizan hoy en día en los estudios de arquitectura. Centrándose en los programas que tienen relación con la informática gráfica, dejando a un lado otros programas que se utilizan en un estudio de arquitectura como son programas de ofimáticas, de cálculo de presupuestos u organización de archivos.

Programas de modelado digital

Son Softwares de modelado tridimensional aquellos que permiten la creación y modificación de gráficos 3D por ordenador. Entendiendo modelado tridimensional como la creación de sólidos y superficies y su posterior modificación. A partir de formas sencillas, mediante el uso de herramientas (extrusión, booleana, revolución, deformación...) podremos crear modelos más complejos.

Los programas de CAD en sus inicios no trabajaban con gráficos en 3D, pero a medida que los softwares fueron evolucionando implementaron herramientas de modelado 3D en sus interfaces. Al principio eran herramientas que permitían la creación de modelos sencillos sin complejidades, por ello era frecuente el uso de programas específicos de modelado digital. Con el tiempo los programas CAD han ido mejorando su forma de trabajar en 3D permitiendo crear modelados con cierta complejidad y detalle. Sin embargo para la realización de modelos muy complejos y orgánicos se sigue demandando los programas de modelado digital, ya que permiten una mayor

variedad de acciones a la hora de realizar el modelo.

Hay una gran cantidad de programas de modelado 3D hoy en día, ofreciendo una gama amplia de plataformas con las que trabajar. Algunos de los programas de modelado más populares son:

3D Studio Max, de Autodesk, sea posiblemente el programa de modelado digital y visualización más estandarizado hoy en día en el mundo de la arquitectura. Se utiliza en múltiples campos como puede ser en el mundo del diseño industrial, cine o video juegos. Se trata de un programa muy completo que permite no solo modelar si no renderizar y animar un modelo.

Blender, de Blender Foundation, programa de características similares a 3D Studio Max que ha

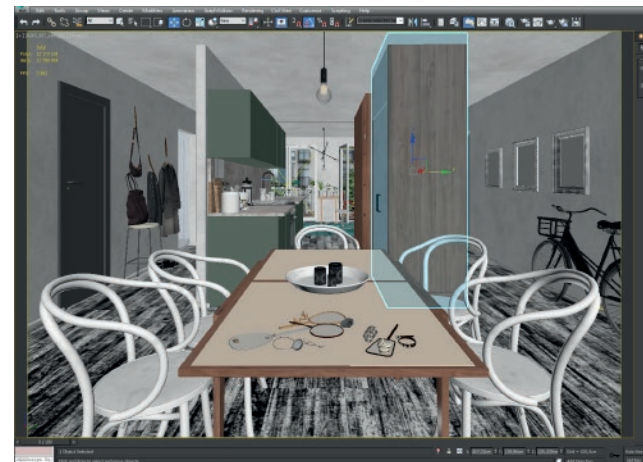


Figura 2.9_ Entorno de trabajo de 3D studio Max

ganado muchísima popularidad debido a que es gratuito y de código abierto, lo cual permite a cualquier usuario realizar una mejora en el programa.

Rhinoceros, de Robert McNeel & Associates, es un software de modelado que trabaja con curvas NURBS, lo cual lo hace una herramienta idónea para el trabajo con superficies curvas u objetos con formas redondeadas. Su modo de trabajo lo ha hecho popular en el diseño industrial, diseño automotriz y en cierto tipo de arquitectura.

Sketchup, de Trimble desde 2012, anteriormente de Google, es un programa gratuito y fácil de usar, lo que lo ha hecho atractivo para los diseñadores, aumentando así su popularidad. Otro punto fuerte del software es que dispone de una gran librería de modelos gratuitos, el 3D Warehouse, aportados por los usuarios.

Zbrush, de Pixologic, es un software de modelado digital que trabaja los modelos esculpiéndolos, a partir de una especie de arcilla digital se va esculpiendo la forma deseada con distintas herramientas. Es muy utilizado para el modelado de personajes u objetos que requieren una faceta artística.

Motores de renderizado

Como dice Juan López-Tarruella Maldonado, el proceso de renderizar consiste en generar una imagen o vídeo a partir de un modelo tridimensional.

Estos motores trabajarían en conjunto con los programas nombrados anteriormente, los programas de modelado. Para que a partir del modelo realizado se pueda obtener una imagen. La forma de interactuar de estos programas puede ser de diversas formas.

Que el programa de modelado no disponga de motor de render, lo cual obligaría a exportar la geometría a otro programa de renderizado u otro software de modelado que disponga de motor de renderizado.

Otros modeladores disponen de motores de render, aunque en algunas ocasiones este puede no ser demasiado potente, por lo que se suele hacer el mismo proceso que en el caso anterior exportar a un software que tenga un motor de render más potente. Serviría de ejemplo el caso de AutoCAD que incorpora el motor de render MentalRay con prestaciones básicas. Con lo que para imágenes más trabajadas se suele exportar a 3ds Max que contiene herramientas más potentes de renderizado.

Por último podemos encontrar motores de render que funcionan como plug-ins. El ejemplo más común es el Vray, un motor de render multiplataforma que funciona en diversos programas como por ejemplo 3ds Max, Sketchup, Blender, Cinema 4D o Rhinoceros.

Por lo tanto, con estos motores, mediante un proceso de cálculo, obtenemos imágenes realistas a partir de una serie de componentes. Estos

componentes son básicamente la iluminación (directa e indirecta), los materiales, parámetros de la cámara y definición de los parámetros de cálculo (*presets*).

Por último enumeramos algunos de los motores de render más populares.

El ya nombrado Vray, de Chaos Grup, el motor de renderizado más popular en la infoarquitectura. Ganó popularidad por la manera innovadora que tuvo de calcular la luz indirecta, el algoritmo del Irradiance Environment Map, conocido como Irradiance Map.

Mental Ray, de Nvidia, es conocido por ser el renderizador que viene incorporado en gran parte de aplicaciones, como por ejemplo AutoCAD, 3ds Max o Maya. Trabaja con iluminación directa (por *Raytracing*) e indirecta (*photon mapping* y *final gathering*).

22



Figura 2.10_ Render realizado con V-ray

Maxwell Render, de Next Limit Technologies, es un motor de renderizado desarrollado por un grupo español. Multiplataforma al igual que Vray. Este software estaría en la categoría de motor de render unbiased, que se basa en los cálculos físicos de la luz sin crear aproximaciones, siendo lo más fieles posibles a la física de la luz real.

Corona Render, de Render Legion, se trata de un plug-in para 3ds Max y próximamente para Cinema 4D y Maya. Surgió como un proyecto universitario de Ondřej Karlík en la Universidad Técnica de Czech. Al igual que el anterior, estaría en la categoría de motor de render unbiased.

BIM

BIM es el acrónimo de Building Information Modeling, traducido al castellano Modelado de Información de Construcción. Este tipo de softwares va orientado directamente al sector de la arquitectura, ingeniería y construcción.

Los programas BIM son generadores y gestores de información de elementos de un proyecto. Cada elemento generado tiene las propiedades físicas y lógicas de sus componentes reales e interacciona con otros elementos que tendrá cada uno sus propias características. Todo ello conformará un modelo del que se podrá obtener cualquier tipo de información de forma visual o numérica.

Estos elementos son la "versión digital" de los elementos que conforman un edificio como los muros, las ventanas, pilares, puertas, etc. De este modo el programa identifica cada elemento

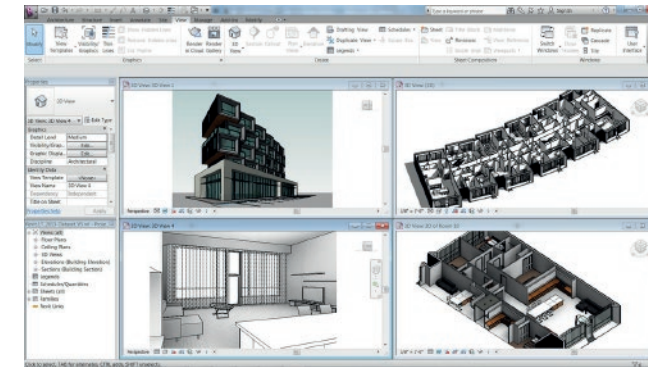


Figura 2.11_ Entorno de trabajo de Revit

y sabe qué es, cuáles son sus atributos y cómo debe de interactuar con los demás elementos del modelo. Además todos estos objetos son tridimensionales, con lo cual cuando "dibujamos" en un programa BIM en realidad estamos introduciendo datos tridimensionales de los objetos, obteniendo un modelo tridimensional íntegro del proyecto.

Esto trae consigo una serie de ventajas respecto al CAD, por ejemplo se puede obtener cualquier tipo de vista del modelo indicando simplemente el plano de corte, si se producen algún cambio en el diseño se cambiará automáticamente en cualquier vista, si se modifica un elemento automáticamente se modifica la manera de interactuar con los distintos objetos, podemos obtener listados de datos a partir de la información del modelo, etc.

Todas estas posibilidades que proporcionan los programas BIM ayudan a optimizar el tiempo de

diseño de un proyecto y a minimizar los errores. Todo esto está provocando que se esté realizando un cambio de CAD a BIM, siendo considerado el sucesor el uno del otro. En algunos países de Europa y en Estados Unidos es común el uso de programas BIM.

Algunos ejemplos más conocidos de programas comerciales BIM son el Revit de Autodesk, Allplan de Nemetschek y ArchiCAD de Graphisoft, este último considerado el primer software BIM.

Programas de edición digital de imágenes

En los estudios de arquitectura es frecuente el uso de estos programas para realizar la postproducción de imágenes y maquetación de paneles. Una imagen digital no se considera acabada tal y como está al finalizar el proceso de renderizado. Necesitan las últimas correcciones y acabados finales antes de obtener el resultado deseado. Estos retoques pueden ir desde la corrección de valores de temperatura de color, iluminación, brillo y contraste, etc. Hasta crear efectos de cámara o añadir personas y vegetación. Aunque el trabajo de postproducción sea el último paso antes de obtener la imagen final no se debe menospreciarlo, ya que pueden cambiar notoriamente la calidad de la imagen.

Debido a la capacidad de edición de estos programas existe la tendencia de dar mayor importancia a la fase de postproducción que a la de renderizado. Es decir, se trabaja con imágenes renderizadas básicas, que solo proporcionan la geometría e iluminación, añadiendo texturas,

23

efectos atmosféricos y de más en el software de edición de imagen.

Es posible incluso, con las herramientas que ofrecen estos programas, crear una imagen virtual a partir de fragmentos de diferentes imágenes, sin necesitar una imagen renderizada de base.

Decir también que existen programas de postproducción de vídeo y animaciones digitales, que ofrecerían herramientas similares a los de edición de imagen: edición de iluminación, efectos de lente, corrección de temperatura de color, etc.

Algunos ejemplos de softwares de edición de imagen serían el conocido Photoshop, de Adobe Systems, programa estandarizado y utilizado por excelencia, Gimp, de Gimp.org, programa gratuito o el Photo-Paint de Corel.

Ejemplos de softwares de edición de vídeo serían After Effects de Adobe o Final Cut de Apple.



Figura 2.12_ Imagen renderizada antes de realizar la post-producción



Figura 2.13_ Imagen renderizada tras realizar la post-producción

3.1 · REALIDAD VIRTUAL

Qué es la RV

Realidad Virtual es por lo general entornos tridimensionales generados por ordenador o sistemas informáticos en el que el usuario puede sentir presencia física. Dependiendo del nivel de inmersión se puede interactuar en mayor o menor grado con el mundo virtual y los objetos que se encuentran en él.

Estos entornos visuales pueden ser réplicas de entornos reales o mundos imaginados, pero deben tener una apariencia realística para que el usuario se sienta inmerso, por ejemplo no se debería ver el entorno en perspectiva axonométrica.

26

Los mundos virtuales deben poder interactuar con el usuario, o por lo menos ser experimentados de forma dinámica y en tiempo real. No nos valdrían imágenes estáticas o recorridos predefinidos.

La interacción se suele realizar por medio de los sentidos de la vista, oído y tacto. Aunque a lo largo de la historia, como se verá en el próximo apartado, se han estudiado sistemas que aprovechaban los cinco sentidos para una sensación de realidad plena. El sentido del tacto y cinestésico es parcialmente simulable mediante dispositivos hápticos que dan sensación de desplazamiento o movimiento. Un ejemplo serían los suelos deslizantes uni u omnidireccionales, utilizados por la Armada de Estados Unidos para el entrenamiento de soldados.

La inmersión total sería aquella que permitiese una interacción total sin límites con el mundo virtual, además de aportar los mismos sentidos que se tienen en el mundo real. Sin embargo los sistemas actuales se centran habitualmente en dos sentidos, la vista y el oído, por ser los más sencillos de incorporar en un entorno virtual.

Un factor importante en el campo de la Realidad Virtual es el de la inmersión. Los sistemas de Realidad Virtual pueden ser más o menos inmersivos según el tipo de dispositivos de comunicación hombre-máquina que se utilicen.

Podemos hablar de Realidad Virtual semiinmersiva, que interactuamos con el mundo tridimensional sin sumergirnos en el mismo, por ejemplo a través de un monitor y un mando de videoconsola, como ocurre con un videojuego común.

Se distinguen diferentes grados de Realidad Virtual inmersiva, en los que se usarían periféricos como cascos de realidad virtual, gafas 3D o sensores de movimiento entre otros. A mayor nivel de inmersión se requiere una tecnología más sofisticada y por lo tanto más costosa.

En los últimos años se ha aprovechado esta tecnología para diferentes usos. A continuación se enumeran algunos de ellos.

En el campo de ciencias de la salud se ha beneficiado enormemente de este campo, surgiendo aplicaciones que permiten el estudio de anatomía, el entrenamiento quirúrgico y la rehabilitación física y psicológica de pacientes.

Muchas investigaciones sobre esta herramienta se han realizado en el campo de la educación. Los entornos simulados e interactivos permiten enfocar la atención del usuario, creando una experiencia de aprendizaje más efectiva frente a métodos docentes tradicionales. Aplicaciones que han surgido en este campo pueden ser visitas de arquitectura, pudiendo visitar edificios lejanos, de otra época o que no llegaron ni siquiera a existir físicamente.

En el ámbito que se está desarrollando esta herramienta con mayor potencia sería en el ámbito lúdico. Las compañías de videojuegos están desarrollando nuevas formas de jugar y creando sus propios dispositivos de Realidad Virtual. Ejemplos de esta nueva era de los videojuegos podrían ser las Oculus Rift que posee multitud de video juegos para PC o la apuesta de Sony, unas gafas de realidad virtual que podremos adquirir para jugar con la consola PS4.

“La pantalla es una ventana a través de la cual uno ve un mundo virtual. El desafío es hacer que ese mundo se vea real, actúe real, suene real, se sienta real”

Ivan E. Sutherland



27

Figura 3.1

3.2 · REALIDAD VIRTUAL

Breve introducción histórica

En primer lugar, se comenzará por describir los primeros sistemas visuales inmersivos desarrollados, sin remontarse a los murales panorámicos del pintor irlandés Robert Barker de finales del S.XVIII.

En 1844 Charlse Wheatstone crearía el estereoscopio, consiste en unas gafas en las que se posiciona un par de imágenes casi idénticas, cambiando únicamente el punto de toma de la imagen ligeramente. Cada imagen es observada por un ojo de manera separada y el cerebro las mezcla en una sola imagen creando cierto grado de tridimensionalidad. Este sistema serviría de base para los siguientes visores de realidad virtual. Louis Ducos du Hauron patenta en 1891 el Anaglifo. Consiste en provocar efecto tridimensional a partir de imágenes en dos dimensiones cuando se ven con unas lentes especiales, lentes con color rojo y cian en cada ojo.

En 1961 los empleados de Philco Corporation, Corneau y Bryan, construyeron un casco de realidad virtual, que podríamos decir que es el primer casco de realidad virtual de la historia. Este dispositivo permitía al usuario ver imágenes en movimiento y gracias a un sensor magnético que tenía incorporado podía determinar la orientación de la cabeza.

Más tarde, en 1962, se desarrolla el que es considerado el primer sistema inmersivo de la historia, el Sensorama. Este aparato diseñado



Figura 3.2_ Casco de Corneau y Bryan



Figura 3.3_ Sesorama

por Morton Heiling pretendía que el cine fuese percibido por todos los sentidos. Para ello constaba de una cabina en la que se proyectaba una película estereoscópica con sonido estéreo, un sistema de ventiladores que simulaban viento y emitía aromas en ciertos puntos de la proyección. Heiling patentaría posteriormente un casco con visión y audio estereoscópicos y emisión de olores, pudiéndose considerar este como el primer Head-Mounted Display (HMD).

En 1965 Ivan E. Sutherland publicaría su artículo "The Ultimate Display" en el que describiría y asentaría el concepto de realidad virtual. En el año 1968 Sutherland junto a Bob Sproull presentaría su artículo "A head-mounted Three-dimensional Display" en el que presentan el primer casco 3D llamado "La espada de Damocles". El sistema utiliza un par de tubos de rayos catódicos (CRTs), uno para cada ojo, capaces de mostrar sencillas imágenes y un dispositivo mecánico de seguimiento para conocer la dirección en la que se está mirando.

Un año más tarde, en 1969, Myron Krueger creó el denominado "Artificial Reality", un sistema que permitía la interacción con sombras creadas virtualmente, pudiendo así dibujar o mover siluetas de un lugar a otro.

A finales de los setenta un grupo de investigadores del Media Lab del MIT, crearon un espacio virtual de viaje llamado el "Aspen Movie Map". El objetivo era permitir que un usuario condujera a través de la ciudad d Aspen, Colorado. Lo

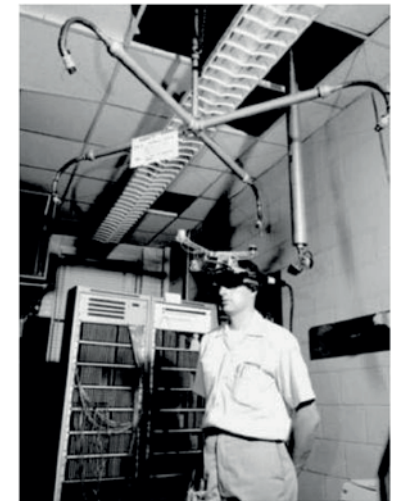


Figura 3.4_ Espada de Damocles

realizaron mediante la grabación de las calles de la ciudad con cuatro cámaras montadas en un vehículo, similar al trabajo realizado por Google Streetview.

En 1983, el Dr. Gary Grimes, de los Bell Labs de AT&T, patentó el primer guante que reconocía las posiciones de la mano con la intención de crear caracteres alfabéticos y numéricos y sustituir el teclado por estos guantes.

Mike Mc Greevy y Jim Humphries junto con la NASA desarrollaron en 1985 el sistema "Vived" (Visual Environment Display system). Siendo estas las primeras estaciones de bajo coste dotadas de un campo de visión amplio, estéreo, con sensores de posición. Este sistema lo usaría la Nasa para sus entrenamientos. A finales de los ochenta se hace

común el término introduciéndose en la cultura popular y apareciendo así las primeras películas que trataban el tema.

En las décadas siguientes el avance de la tecnología gráfica e informática en general hace que se le dé un mayor impulso a estos sistemas, empezando a comercializarlos y refinarlos con el tiempo. Ejemplos destacables de esta evolución serían la aparición en los salones recreativos de Virtuality en 1991, la aparición de la primera consola de realidad virtual creada por la compañía Nintendo llamada "Virtual Boy" en el año 1995, o el anuncio en 2012 del proyecto Oculus Rift, casco de realidad virtual que daría paso a una nueva forma de jugar.

30

En los últimos años han ido surgiendo gran variedad de plataformas comerciales para el disfrute de la realidad virtual. Gracias al gran avance técnicos en los smartphones se puede disfrutar de visores de realidad virtual *low cost* como podrían ser las Google Card Board, un soporte de cartón plegado que es capaz de transformar cualquier móvil en una plataforma de realidad virtual. Podemos encontrar modelos más sofisticados como las VR Box, Vitroba X5 o las Samsung Gear VR, todas ellas de plástico, con sujeción a la cabeza y ajuste de la posición de las lentes.

No se deben olvidar los modelos más avanzados como las ya nombradas Oculus Rift que han ido mejorando el modelo e incorporando nuevos juegos de RV en plataformas como Steam, o las

HTC Vive que permiten moverse dentro de un espacio acotado e incorporan un par de mandos con los que podemos interactuar con el mundo virtual, mejorando la experiencia.

Estos soportes surgidos en los últimos años proporcionan una gran variedad de plataformas al alcance de todos, para experimentar entornos virtuales. Diferentes disciplinas como la medicina, arquitectura o educación se han, y siguen, beneficiándose de estas herramientas.



Figura 3.5_ Virtual Boy



Figura 3.6_ Oculus Rift

4 | VIDEOJUEGOS

4.1 · VIDEOJUEGOS

Evolución gráfica

Gran parte de la evolución de la informática gráfica es debida a los videojuegos ya que estos han conformado un motor económico potente en la sociedad. Los primeros jugadores podían disfrutar de juegos con gráficos monocromo sencillos, un sonido en 8 bits e ideas sencillas, pero que abrían todo una nueva manera de entretenimiento. Esta industria ha avanzado a pasos agigantados en estas algo más de cuatro décadas, ofreciendo videojuegos con capacidades gráficas excelentes y experiencias de juego cada vez mejores.

Paradigma de juegos pioneros tenemos al famoso Pong, juego lanzado por Atari en 1972 junto a la consola Odyssey. Representaba un partido de tenis de la forma más sencilla posible, un punto móvil blanco y dos barras verticales, también blancas, que se podían desplazar en vertical sobre su eje, todo ello sobre un fondo negro. Esto marcaría el pistoletazo de salida a la industria de los videojuegos.

A principios de los ochenta surgieron consolas de segunda generación con una mejora gráfica y de sonido importante. Se introdujo el color y mejor definición de personajes. Aparecieron consolas como la Atari 2600, con 128 bytes de RAM, que trajo consigo títulos como Mario Bros, Pac-Man o Donkey Kong; Commodore 64, que disponía de unos mejores chips de sonido del mercado o la MSX.

Los gráficos vectoriales ayudaron a crear los primeros entornos con sensación de tridimensionalidad cuando el hardware de la industria no disponía de la suficiente capacidad para crear entornos de otra manera. Prueba de su eficacia fue el éxito de Battlezone, un juego de tanques lanzado en 1983.

En 1985 llegaría al mercado la primera consola de Nintendo, la NES (*Nintendo Entertainment System*), vendiendo algo más de 60 millones de unidades. Esto supuso un enorme impulso para la industria de los videojuegos, marcando un antes y un después en la manera de jugar y entender una consola. Los gráficos mejoraron respecto a sus predecesoras ofreciendo mayor paleta de colores y mejor definición de los personajes, aunque seguía teniendo gráficos 8 bits.



Figura 4.1_ Videojuego Battlezone

La siguiente generación de consolas llegaría a finales de los ochenta y principios de los noventa con una mejora de gráficos, sonido y velocidad considerable ya que incorporaban CPU de 16 bits, dejando atrás la era de los 8 bits. Esta mejora gráfica se puede apreciar en juegos como Sonic the Hedgehog con escenarios coloridos, bien definidos y con movimiento, simulación de tridimensionalidad en los objetos y personajes mejor definidos, o Star Fox que simulaba escenarios y naves en 3D con polígonos sencillos gracias al chip Super FX, que iba incorporado en el cartucho. Consolas como la SNES (*Super Nintendo Entertainment System*), conocida como Súper Nintendo o Sega Mega Drive pertenecería a esta generación de consolas.

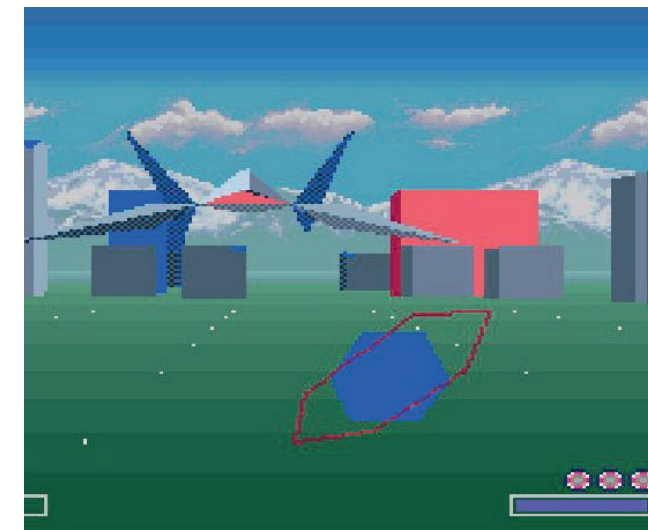


Figura 4.2_ StarFox para Super Nintendo

La llegada de la siguiente generación de consolas traería consigo un salto en la calidad gráfica dejando atrás las 2D para introducir los entornos 3D. Esto supondrá una nueva forma de jugar y de interactuar en los videojuegos, suponiendo un antes y un después en la industria.

Las primeras consolas de esta generación funcionaban a 32 o 64 bits, introduciendo los primeros modelados tridimensionales funcionando en tiempo real, texturas, mapas y animaciones. Al introducir una dimensión más la jugabilidad cambió por completo, reinventando la manera de jugar e interactuar con el espacio digital.

Títulos como The Legend of Zelda: Ocarina of Time y Super Mario 64, Metal Gear Solid o Final Fantasy marcaron esta generación de consolas.

Atari Jaguar, lanzada al mercado en 1993 y considerada la primera consola en dar el salto al 3D, PlayStation de Sony lanzada en 1994, que introdujo el soporte CD para video consolas, o Nintendo 64 lanzada dos años más tarde, son algunos ejemplos de consolas de esta generación.

Con la entrada del nuevo siglo llegaría una nueva generación de consolas que se caracterizó por el uso de memoria externas, el uso de soportes nuevos como el DVD y, la novedad más importante, la conexión a internet y red para jugar online.

Gráficamente estas máquinas funcionan a 128 bits, mejorando las animaciones, texturizado y cinemáticas de los títulos disponibles de

la época. Shenmue, juego para la consola Dreamcast de Sega, ofreció unos gráficos por encima de la media, recreando escenarios y personajes realísticos y fue una de las primeras representaciones con éxito de un mundo abierto.

En la década siguiente la mejora de hardware en las consolas, la influencia económica del sector y la importancia que tienen los gráficos frente a la demanda del público, hace que se consigan cada vez mejores gráficos, en resumen mayor realismo en los videojuegos. En las consolas siguientes el avance técnico se producirá de manera exponencial mejorando mucho la calidad gráfica en poco tiempo. Así pues consolas como Xbox 360 o Play Station 3 apostarán por una mejora gráfica, mientras que consolas como Wii apostarán por innovadoras formas de jugar.

34



Figura 4.3_Shenmue

Las consolas de última generación disponen de capacidades gráficas muy superiores a lo que podríamos imaginar hace unos años. Los videojuegos de estos últimos años son capaces de representar ciudades enteras a todo detalle, tener un nivel de realismo altísimo en el modelado de los personajes y disponer de efectos de luz dignos de películas. Todo ello hace que la visualización de estos videojuegos nos sorprenda cada vez más y se asemejen a imágenes fotorrealistas que podríamos generar mediante la elaboración de un render, pero con la diferencia que los gráficos de un videojuego son en tiempo real.

Prueba de la calidad gráfica que podemos disfrutar hoy día se refleja en títulos como Forza Motorsport 6, Assassin's Creed Unity, Uncharted 4.



Figura 4.4_Uncharted 4

4.2 · VIDEOJUEGOS

Los Game Engine

Como ya se ha visto el mundo de los videojuegos ha favorecido mucho a la evolución de la informática gráfica. Todo esto ha llevado a la creación de motores, cada vez más potentes, de creación y ejecución de videojuegos, los *Game Engine* o motor de videojuego.

La funcionalidad básica de un motor de videojuego es servir de motor de renderizado para los gráficos 2D y 3D, motor físico, de sonido y de animación entre otros. Estos programas permiten el desarrollo de software con casi cualquier tipo de plataforma, PC, MAC, iOS, Android, PS4, XBOX, Steam...

Comenzaron siendo desarrollados por cada compañía para evitar el trabajo de empezar un nuevo proyecto desde cero teniendo que volver a programar todos los campos. Poco a poco se han ido comercializando llegando a terceros para poder realizar sus propios videojuegos. Estos programas permitían crear juegos a partir de una base común, pudiendo crear escenarios, personajes y cualquier otro tipo de contenido, dejando la coordinación del conjunto al motor.

En estos últimos años se han desarrollado algunos motores de videojuegos con la finalidad de poder crear de manera sencilla y visual aplicaciones, sin necesidad de tener formación avanzada en programación.

La facilidad de acceso a estas herramientas, su sencillez a la hora de crear una aplicación y su bajo coste, han hecho que distintas disciplinas ajenas al mundo de los videojuegos hayan desarrollado aplicaciones con muy diversos propósitos. Los principales usos han sido la creación de simuladores y los llamados *serious games*, juegos destinados al aprendizaje, mejora de habilidades o tratamiento de problemas.

En el campo de la visualización arquitectónica estos programas están aportando enormes beneficios. Algunos de los motivos por los que estos programas están cada vez más generalizados en los estudios de arquitectura son la posibilidad de moverse libremente por un modelo digital, entendiendo mejor el espacio. Los motores trabajan bien con entornos arquitectónicos ya que se usan en videojuegos. La visualización en primera persona, los materiales y la iluminación son elementos comunes en los videojuegos y en la visualización arquitectónica. Se puede potenciar más la experiencia introduciendo movimiento, interacción o sonido.

Han aparecido empresas especializadas en este tipo de visualización que ofrecen sus servicios para mostrar espacios ficticios, proyectos en fase de diseño o tours virtuales por edificios icónicos. Un ejemplo sería la empresa CGcloud, de la que podemos descargar gratuitamente su demo *Loft in London*. Es un tour virtual por un pequeño apartamento ubicado en Londres para ver las posibilidades que nos pueden dar estas herramientas.

35

En el siguiente apartado se describen algunos Games Engine que podemos encontrar en el mercado.

Unity 3D es uno de los motores de videojuegos más estandarizados hoy en día, debido a que no necesita grandes requerimientos del sistema para funcionar y resulta relativamente más sencillo a la hora de aprenderlo.

Es un editor multiplataforma que permite trabajar el desarrollo de aplicaciones en una muy amplia gama de dispositivos; iOS, Android, PC, Mac, Windows phone, Linux, PS4, WiiU o Xbox One entre otros.

Otro gran atractivo de este motor, y una de las causas de su frecuente uso, es que dispone de una versión gratuita con la que se pueden crear y comercializar aplicaciones. Aunque también dispone de diferentes tarifas según la necesidad del usuario que van ofreciendo mejores prestaciones, por un precio razonable mensualmente.

Monument Valley, de Ustwo, juego de puzles en axonométrica con una estética cartoon para móvil, Ori and the Blind Forest, de Microsoft, juego de plataformas en 2D con un resaltable trabajo de diseño de escenarios o el shooter Contract Wars son juegos desarrollados por el motor Unity 3D.

Unreal Engine otro motor que cada vez más se utiliza en diferentes sectores.

Tiene una capacidad gráfica superior a Unity 3D, pero por contra consume más recursos del sistema y tiene más opciones con lo que su aprendizaje es más complejo.

Es un motor de videojuego que se está difundiendo mucho entre los estudios de arquitectura debido a la calidad gráfica de materiales que ofrece y al poder trabajar de una forma adecuada la iluminación. Estudios como UE4, CGcloud o la firma americana HKS, desarrollan proyectos de visualización arquitectónica con este motor.

Al igual que el software anterior, Unreal Engine dispone de una versión gratuita para poder realizar todo tipo de aplicaciones, además de las versiones de pago.

Juegos como Planets³ de Cubical Drift, juego con gráficos lowpoly de mundo abierto, Paragon, de Epic Games, juego de estrategia con gráficos muy trabajados o Kingdom Hearts III de Square Enix, tercera entrega de la saga que fusiona personajes de Final Fantasy con personajes de Disney, son algunos de los títulos desarrollados con el motor Unreal Engine.

En definitiva se podría decir que en estos softwares es el lugar donde se cruzan los caminos de la arquitectura y los videojuegos, ofreciendo nuevas posibilidades a ambos sectores.

5.1 · PABELLÓN DE BARCELONA

Mies van der Rohe desarrolló el proyecto del pabellón alemán para la exposición internacional de 1929, encargado por la República de Weimar y en el que participó Lily Reich durante el proceso de diseño del mismo. Este proyecto llega a manos de Mies en junio de 1928, ya que tras el éxito de la exposición en Weissenhof, el gobierno del Reich alemán encomendó a Mies la dirección artística de todos los departamentos alemanes y la construcción del pabellón (Claire Zimmerman, 2014). El proyecto debía realizarse en un plazo corto de tiempo, lo que hizo que todo el proceso de proyecto y diseño, los montajes para industrias varias y un pabellón para suministro de electricidad, se llevara a cabo en menos de un año.

En un primer momento, la parcela asignada para el pabellón por parte de los burócratas de la Exposición fue un solar cercano a la Fuente Mágica, eje central de la exposición construida por Puig i Cadafalch. Mies la rechazó e insistió en cambiar a un nuevo emplazamiento en pendant con el Pabellón de Francia, al pie de las escalinatas que ascendían hacia el Palacio Nacional. En este nuevo emplazamiento junto a las ocho columnas jónicas se desarrollará el proyecto que hoy en día se conoce.

El diseño del Pabellón tenía una función representativa sin más. No exhibe más de lo que es (Moisés Puente, 2000). Representaba una



Figura 5.1_Visita del Rey de España, Alfonso XIII, al Pabellón. Mayo de 1929



Figura 5.2_Visita del Rey de España, Alfonso XIII, al Pabellón. Mayo de 1929

nueva Alemania, una Alemania que tras una guerra, disturbios y pobreza económica, iba a ser democrática y progresista en el plano cultural, próspera y pacífica en el social. El Pabellón sirvió para actos representativos y ceremoniales. Fue el lugar en el que se llevó a cabo la recepción de los reyes de España durante la ceremonia inaugural.

Una vez finalizada la Exposición Universal se planteó qué se debía hacer con el Pabellón. Muchas fueron las voces que pedían su conservación, pero tras un intento fallido de venta a un empresario Barcelonés interesado en reutilizarlo como restaurante y sumado a las dificultades económicas que atravesaba el estado alemán; en Enero de 1930 se decidió que el pabellón fuese desmontado finalmente tras la Exposición. Los mármoles fueron devueltos a Alemania, gestionados por la casa suministradora para su posible reutilización. Las piezas cromadas se devolvieron con el objetivo de venderlas



Figura 5.3_Vista exterior con el Palacio de Victoria Eugenia al fondo

para subsanar el déficit ocasionado. Los perfiles metálicos estructurales fueron desmantelados y vendidos en Barcelona como chatarra.

Años después esta obra iría ganando reconocimiento en el mundo de la arquitectura como un referente del movimiento moderno y de la arquitectura del siglo XX. Así pues, en 1959 comienzan las conversaciones con Mies para la reconstrucción del Pabellón por parte del arquitecto Oriol Bohigas, secretario del Grup R. Esta iniciativa de construir el Pabellón en su emplazamiento original va tomando fuerza hasta su materialización en la década de 1980. Tras un estudio detallado y previa recopilación de todos los datos disponibles, el pabellón fue edificado nuevamente entre los años 1983 y 1986, a cargo de los arquitectos Ignasi de Solà-Morales, Cristian Cirici, Fernando Ramos y Ana Vila. La reconstrucción se basó íntegramente en el primer pabellón construido siguiendo el diseño original y con los mismos materiales. Tras la finalización de las obras, el pabellón se inauguró en junio de 1986 en su emplazamiento original.

El Pabellón forma parte de la historia de la arquitectura moderna, ya que el paso decisivo logrado por Mies por primera vez en su totalidad en el Pabellón de Barcelona que consistió en la realización de la <<planta libre>> y el <<espacio fluido>> (Claire Zimmerman). La obra se inserta en su emplazamiento asentándose sobre un zócalo de travertino, cuyo ámbito se delimita al sur por un muro del mismo material con forma de U, que finaliza en una pequeña estancia de servicios.

A esta composición se añade un estanque de bienvenida y cuyas placas base se extienden para crear la sensación de que el agua continúa bajo el zócalo. El volumen es delimitado al norte por un muro, nuevamente en forma de U, de mármol verde que delimita la superficie de un pequeño estanque interior de menor tamaño. Sobre este estanque, en posición asimétrica, descansa la estatua de bronce El mañana, realizada por el escultor Georg Kolbe. Una losa cubre el espacio interior sostenida por unos pilares cruciformes cromados, que reflejan todo su alrededor y otorgan una sensación de ligereza al conjunto. El resto de cerramientos de vidrio, ónice y mármol, se componen de tal manera que generan vistas controladas y crean un espacio fluido que diluye interior-exterior.

40

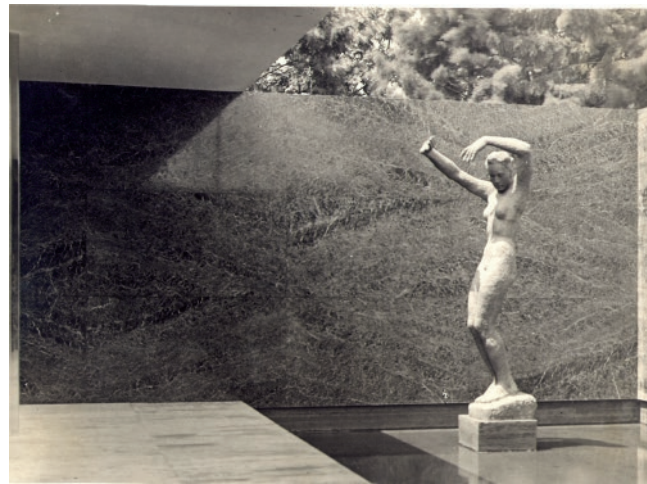
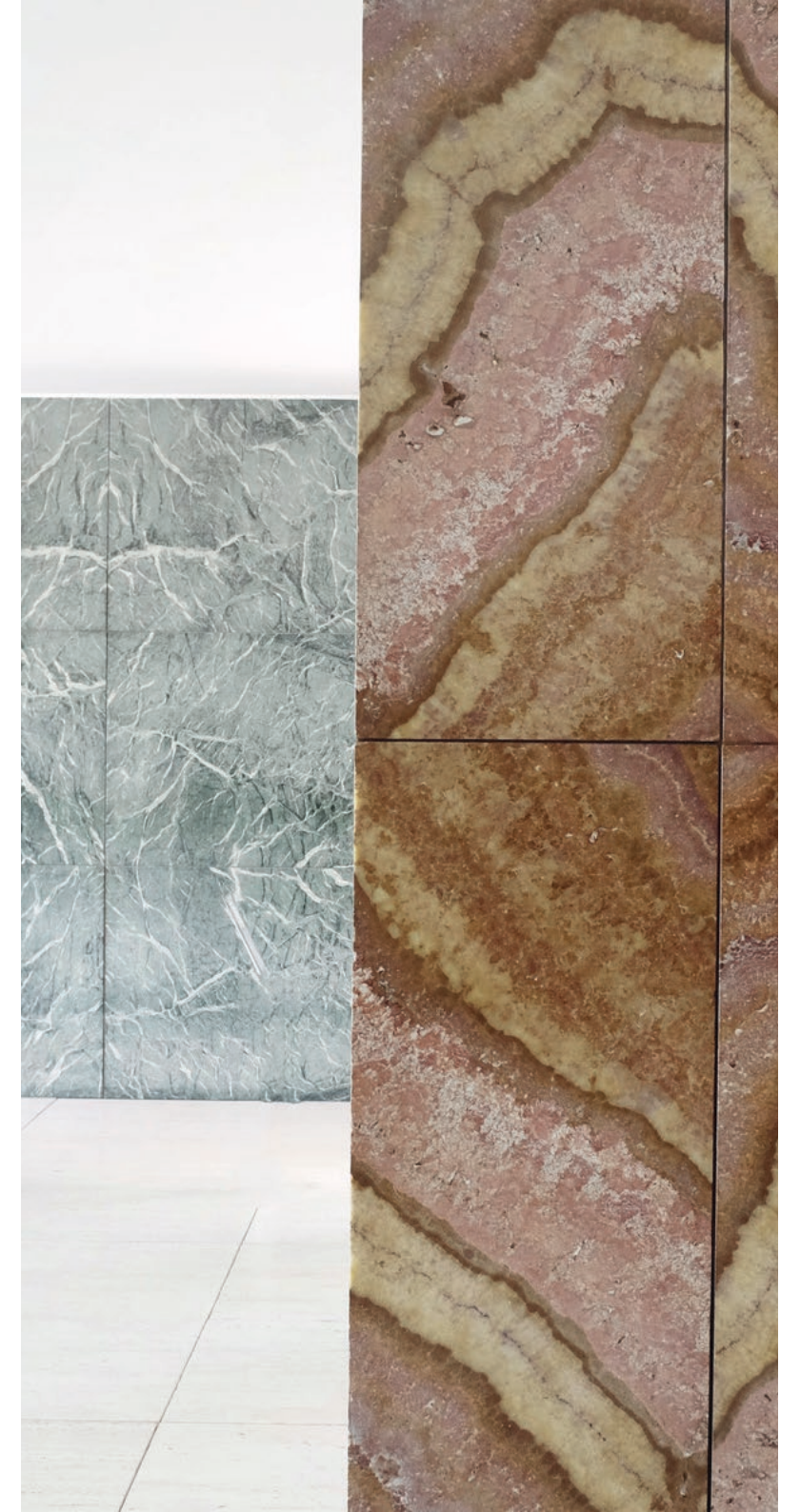


Figura 5.4_Escultura de G.Kolbe en el Pabellón

El Pabellón de Barcelona es un edificio representativo y de referencia de la arquitectura del siglo XX. Es un modelo muy interesante desde diferentes puntos de vista: espacial, material y perceptivo. El interés por realizar el modelo sobre este edificio reside en el objetivo de entenderlo como un edificio didáctico para la representación digital. Sus materiales son muy característicos, con una gran riqueza y variedad de texturas. Por otro lado, existe abundante documentación gráfica de la que se puede extraer información. El modelo da la oportunidad de trabajar espacios interiores, exteriores y la relación entre ambos, potenciando de esta manera el aprendizaje de la herramienta. Su ubicación en Barcelona, es de especial interés para el trabajo propuesto ya que se pretende realizar el estudio de un edificio representativo con facilidad de ser visitado. De esta manera, se podría realizar un análisis comparativo entre la percepción digital y la percepción real del edificio, con la finalidad de extraer unas conclusiones que reflejen la capacidad representativa-perceptual que el programa ofrece.

Figura 5.5_Fotografía del Pabellón de Barcelona desde el interior resaltando los materiales



5.2 · CONCEPTOS DE RENDERIZADO

En tiempo real

En este apartado se explicarán y definirán conceptos que manejamos en la realización de imágenes digitales y el proceso de renderizado tridimensional. Estos conceptos se manejarán durante el caso práctico.

Renderizado en tiempo real

Renderizar en tiempo real implica que los cálculos para generar la imagen que estamos viendo en pantalla, se realizan a una velocidad suficiente para crear sensación de movimiento continuo. Al menos deben de ser veinticuatro imágenes por segundo para tener una correcta visualización.

Este tipo de renderización, contraria al de las imágenes pre renderizadas, se utiliza en aplicaciones interactivas, como videojuegos o visualizaciones inmersivas.

Para conseguir una correcta visualización y obtener el máximo fotorrealismo, sin sufrir bajada de frames por segundo, es necesario optimizar el modelo. Para ello se debe hacer uso de simplificaciones o artificios que puedan ofrecernos un resultado correcto. El control de polígonos en los modelos, el uso de normal maps o lightmaps, son algunos de los artificios que usamos.

Back-Face Culling

Los modelos tridimensionales están compuestos por polígonos bidimensionales, es decir planos, que tienen dos caras, una principal que está definida por el vector normal al plano y su opuesta.

Back-Face Culling es el algoritmo que evita que se calculen las caras posteriores de cada polígono, renderizando sólo la cara principal, para reducir tiempos de cálculo.

Físicas

Por norma general, los programas de renderizado, animación y motores de videojuegos permiten simular propiedades físicas de los objetos, por ejemplo la gravedad, la masa, aceleración o inercias.

Los elementos principales son el generador de físicas, encargado de la parte que simula la física del movimiento, y el detector de colisiones, que es el encargado de la interacción entre los objetos.

Shading

Este concepto hace referencia en la manera de interactuar que tiene la luz con una superficie y como varía esta a lo largo de las superficies. Este proceso altera el color de las superficies en la escena, basándose en el ángulo de la luz y la distancia a esta para conseguir un efecto lo más real posible. El shader sería el algoritmo encargado de calcular este efecto durante el proceso de renderizado.

Reflexión

Reflexión es la propiedad física que describe cómo rebotan los rayos de luz en una superficie dependiendo de la rugosidad entre otros factores. Además estas propiedades permiten que las superficies reflejen su entorno.

Refracción

El fenómeno de refracción es el que sufre la luz cuando pasa de un medio transparente a otro, produciéndose un cambio en su trayectoria debido al índice de refracción de los materiales. Éste último se debe a la distinta velocidad de propagación que tiene la luz en los diferentes medios materiales. La distorsión del rayo depende de los distintos índices de refracción de los materiales y el ángulo de incidencia entre en la interfaz entre medios.

En los motores de render podemos simular distintos materiales transparentes dependiendo de su índice de refracción (IOR) a partir de valores estandarizados.

Mapeado de texturas

El mapeado de texturas consiste en aplicar una imagen bitmap a una superficie, como si se pegara una fotografía, aunque en realidad se trata de una proyección de la imagen sobre el objeto

En los programas de visualización se puede modificar el tamaño de la imagen y la posición, teniendo en cuenta que esta imagen se va a repetir hasta el infinito. Por lo tanto es habitual que se corrijan los bordes para que no se acusen las uniones entre una imagen y otra. También se puede elegir el ángulo de proyección de la imagen sobre la superficie, dependiendo de las necesidades.

UVW Mapping

El modificador encargado de realizar las modificaciones antes citadas, y algunas más, es el UVW Mapping. Permite definir las coordenadas de aplicación de la textura. (posición de inicio, orientación o inclinación de la imagen y el tamaño del mapa).

Los ejes de coordenadas usados para las texturas vienen definidos como UVW para distinguirlos de los ejes de coordenadas XYZ utilizados para los objetos.

Este modificador nos permite también variar el modo de proyección de la imagen sobre el objeto, pudiendo ser proyección plana, cilíndrica, cúbica o esférica.

Normal Map

Los Normal Maps son la evolución de los Bump Maps. Mientras que estos sólo aportan información de la altura de cada punto de la superficie, por tratarse de imágenes en blanco y negro, los Normal Maps son imágenes a color que nos proporcionan más información.

El color es un espacio tridimensional, es decir que está definido por tres combinaciones de primarios. Tenemos dos modelos. El modelo sustractivo (CMYK) y el modelo aditivo (RGB).

Podemos asociar los vectores R, G, B de una imagen a los vectores espaciales X, Y, Z y así conseguir aportar información tridimensional a cada punto de la imagen.

Estas imágenes se usan normalmente para generar modelos de gran realismo con pocos polígonos, lo cual beneficia a los tiempos de cálculo.

5.3 · CASO PRÁCTICO

Modelo: Pabellón de Barcelona

En el siguiente apartado de investigación, se testea un motor de videojuegos como soporte para realizar una visualización arquitectónica interactiva en tiempo real.

El modelo escogido es el Pabellón de Barcelona, como ya se ha detallado antes, y los programas utilizados para la realización de la visualización son, por un lado 3D studio Max para el modelado del pabellón y Unreal Engine como motor de videojuego.

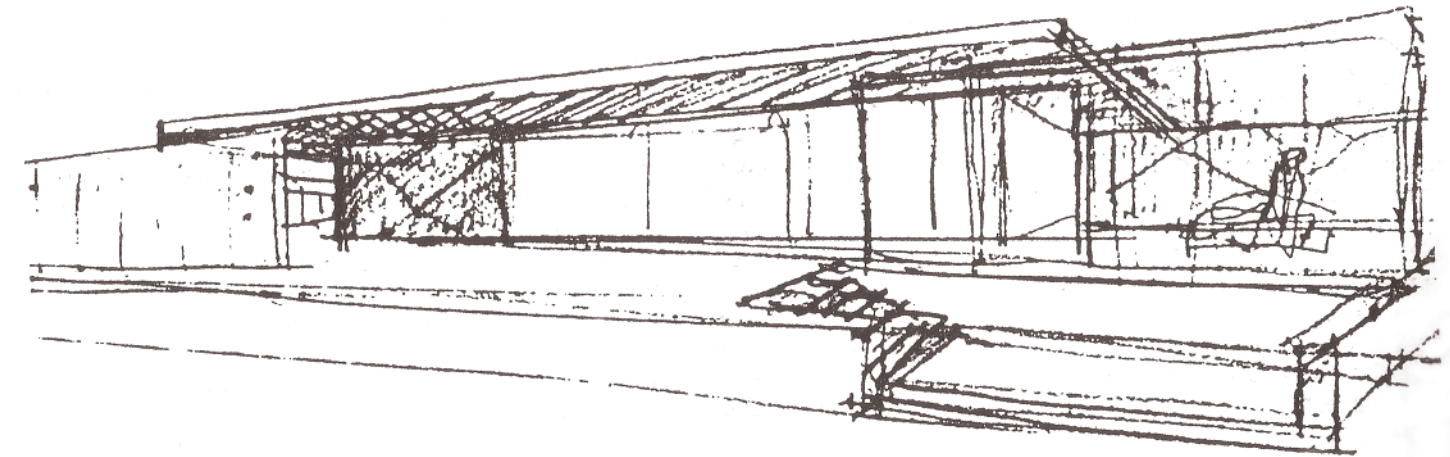


Figura 5.6_Dibujo realizado por Mies durante el proceso de diseño del Pabellón

1 · Estudio del modelo

Antes de comenzar a realizar trabajo de modelaje o texturizado se hace un estudio del edificio seleccionado para obtener imágenes de referencia, planimetría y detalles de encuentros.

Esta información se obtiene principalmente de fuentes como la Fundación Mies Van der Rohe Barcelona (<http://miesbcn.com/es/el-pabellon/>), que dispone de una galería de imágenes del pabellón y planimetrías en pdf, o el libro "Mies Van der Rohe. El Pabellón de Barcelona" de Ignasi de Solá-Morales, Cristian Cirici y Fernando Ramos, que contiene planimetrías a varias escalas y una amplia galería de imágenes.

A partir de la documentación extraída se entienden las características del proyecto, su materialidad y modulación. Con estas premisas se está en disposición de comenzar el modelado.

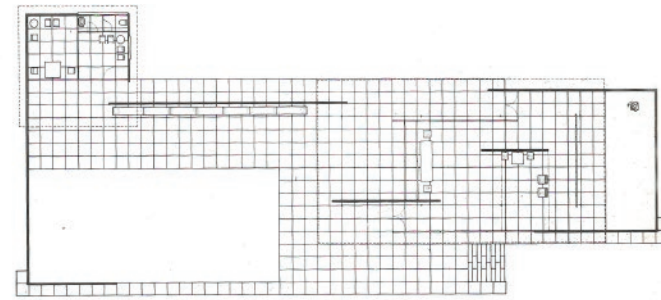


Figura 5.7_ Planta principal

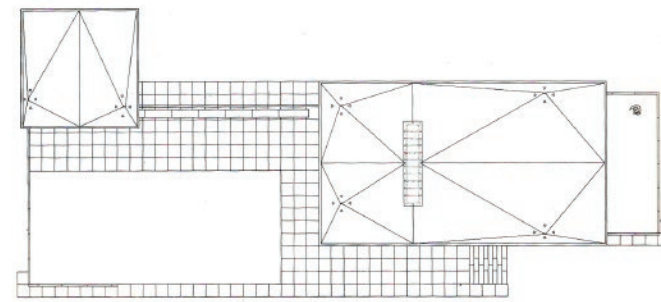


Figura 5.8_ Planta de cubierta

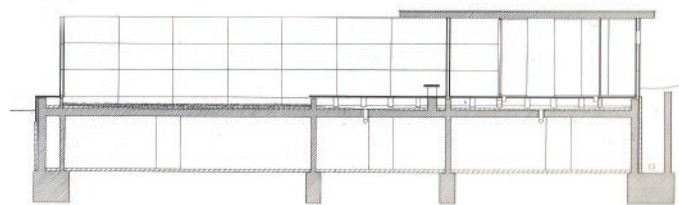


Figura 5.9_ Sección transversal



Fig.5.10



Fig.5.11



Fig.5.13

Fig.5.12

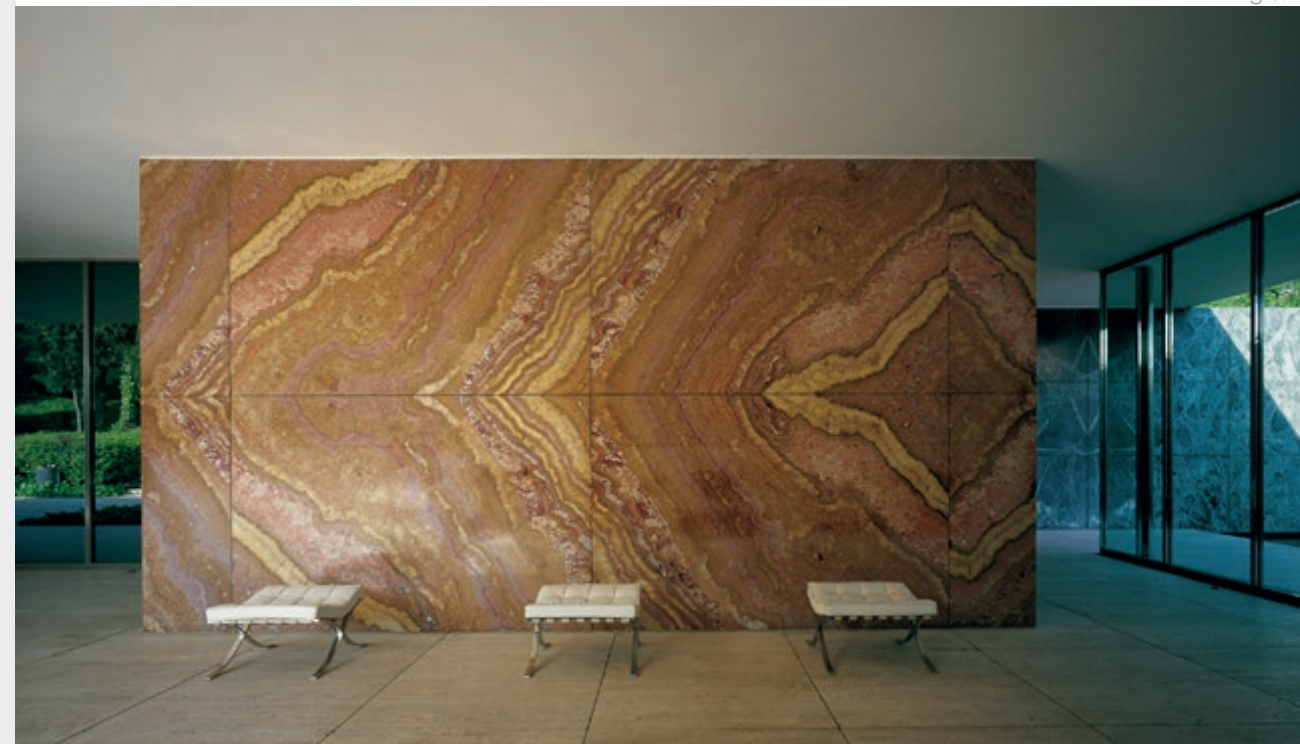


Fig.5.15

Fig.5.14

2 · Modelado del edificio

Para la realización del modelo tridimensional del Pabellón de Barcelona se escoge el programa 3d Studio Max. La elección de este software se debe a la familiarización del autor del trabajo con el programa, al mayor control que permite sobre la geometría gracias a los modificadores, a las herramientas y a la posibilidad que proporciona el software para aplicar materiales a un modelo.

El número de polígonos que contiene la escena se controla en todo momento para conseguir un modelado efectivo con el menor número de geometrías y de esta forma permitir que el motor de videojuegos no consuma recursos excesivos.

Se comienza modelando el pabellón a partir del módulo principal del proyecto, 1,1 x 1,1 metros. Para comenzar a trabajar se crea el volumen del zócalo, que define las proporciones del proyecto, mediante la agrupación de *boxes* con las medidas del módulo.

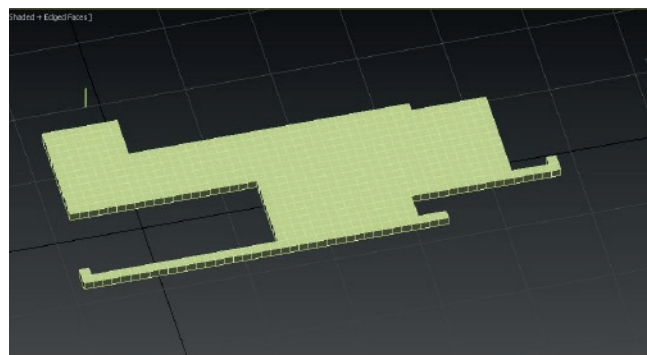


Figura 5.16_ Modelado tridimensional del zócalo generado a partir de agrupación de "boxes"

A continuación se procede al modelado de los pilares en cruz. Para ello se crea una *box* y se eliminan partes de sus caras para disminuir el número de polígonos. A partir de este elemento, mediante un duplicado y una rotación de 90° obtenemos la forma básica del pilar completo (figura 5.17). Por último, para dotar de mayor detalle al modelado de los soportes se redondean las aristas exteriores.

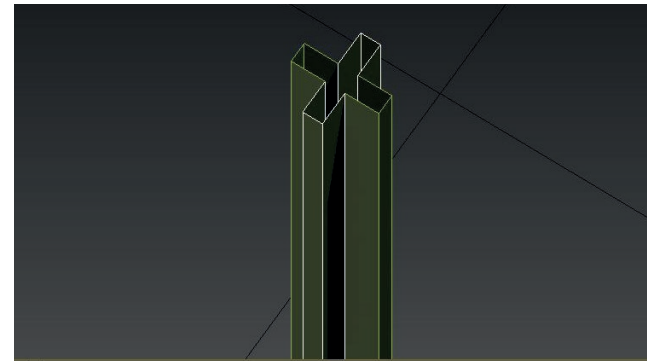


Figura 5.17_ Detalle de modelado de pilar

A partir de la geometría del zócalo se define la posición y altura de los demás paramentos verticales que componen la volumetría del pabellón, eliminando la cara superior e inferior en los paramentos que acometen contra el techo. Además se añaden los volúmenes que definen los suelos de los estanques, carpinterías, techos y vidrios.

Para completar la volumetría del pabellón se modela el jardín trasero (desnivel y escalera) y se importa la volumetría de la estatua de Kolbe. Este modelo se obtiene de la página web rumlig.blogspot.com.es.

Con el modelo tridimensional definido, para reducir número de polígonos, se sustituye la malla de módulos cúbicos que define el zócalo, por una geometría más sencilla y de menor número de caras. Se genera el zócalo con dos volúmenes para definir el encuentro entre el pavimento y las láminas de agua y de esta forma obtener un mayor realismo en la visualización digital.

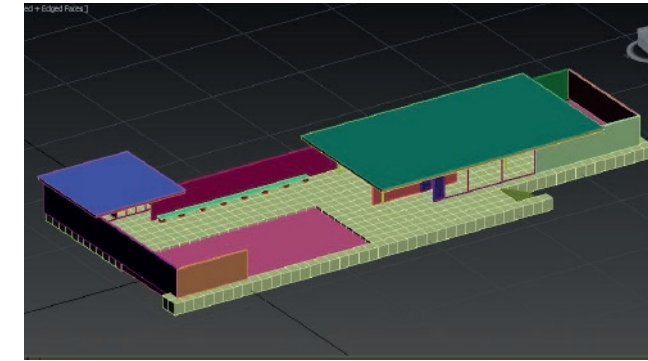


Figura 5.18_ Modelado tridimensional del Pabellón de Barcelona

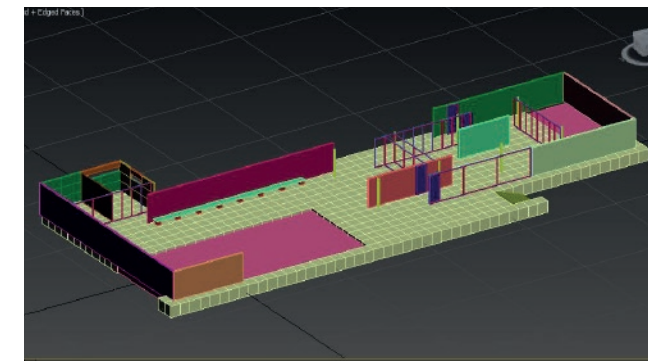


Figura 5.19_ Modelado tridimensional del Pabellón de Barcelona sin cubierta

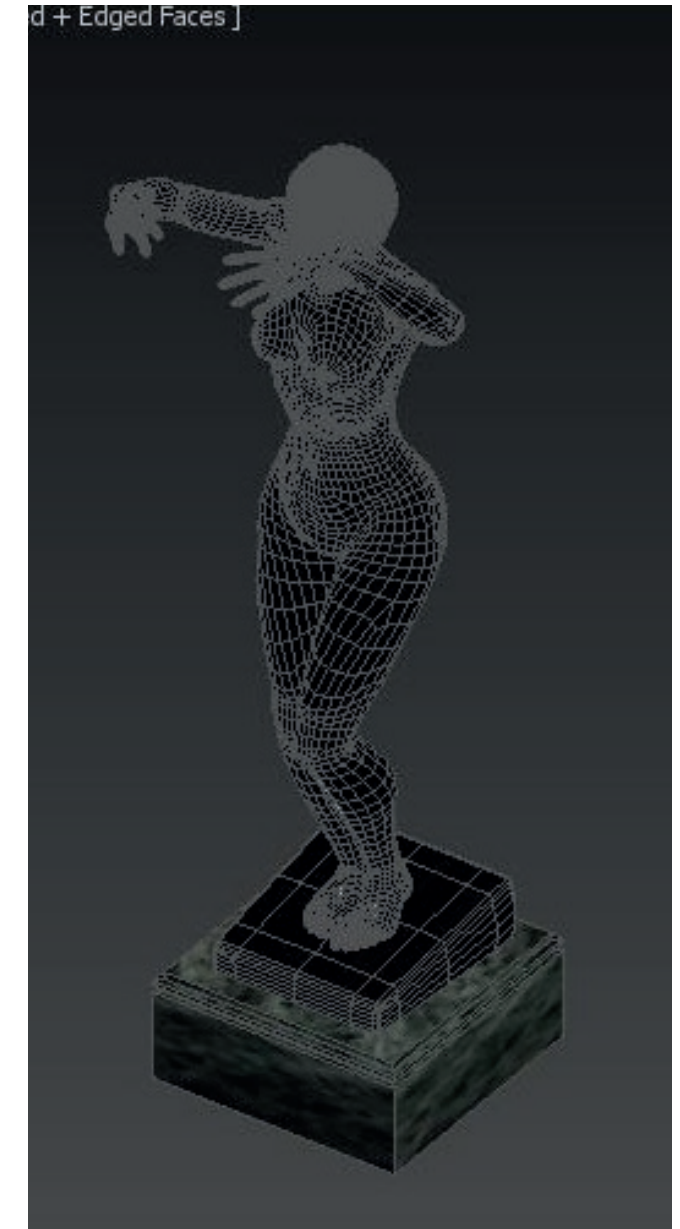


Figura 5.20_ Modelado descargado de la escultura de Kolbe

3· Mapeado en 3D Max

El mapeado y texturizado del entorno digital se lleva a cabo en el software 3d Studio Max en vez de realizarlo directamente en el motor de videojuegos Unreal Engine, ya que es más cómodo y sencillo para el autor de este trabajo. Además al importar a Unreal los mapeados se conservarán en posición y escala, con lo cual no supone ningún tipo de problema y ayuda a optimizar los tiempos empleados en el desarrollo del trabajo.

Se trabaja con las texturas facilitadas en la asignatura optativa Formalización Visual Avanzada de Arquitectura impartida por el profesor Alfredo Santonja Llabata, impartida en esta escuela, ya que durante el curso lectivo 2014-2015 se trabajó con este mismo modelo arquitectónico.

Para la creación de mapas normales (Normal maps) se recurre al plug-in de NVIDIA para Photoshop, recomendado por el tutorial online de Evermotion (www.evermotion.org/tutorials), *NVIDIA DDS plug-in*. Este software que se puede descargar gratuitamente desde la página web de NVIDIA (www.developer.nvidia.com), permite crear mapas normales en RGB a partir de cualquier tipo de imagen insertada en Photoshop.

Se editan las texturas en Photoshop para obtener la modulación deseada de los mapas, ajustar colores y crear los “normal maps” correspondientes.

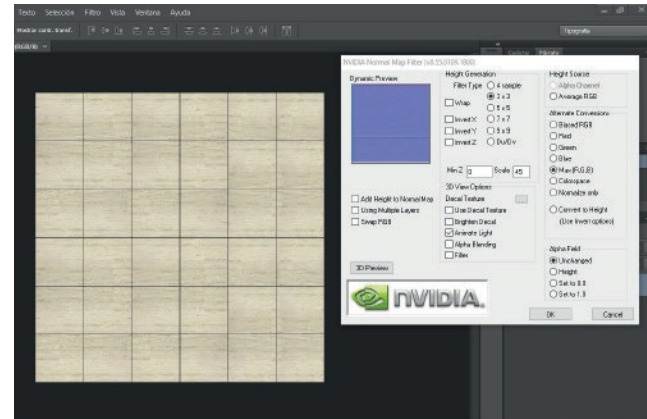


Figura 5.21_ Interfaz del programa Photoshop con el plug-in de NVIDIA, NVIDIA DDS

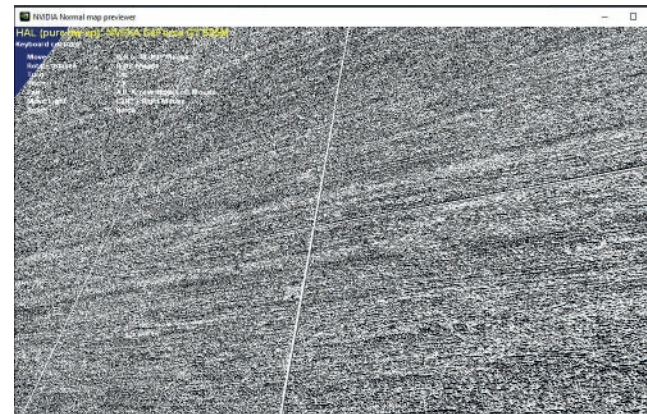


Figura 5.22_ Previsualización tridimensional del Normal Map en plug-in NVIDIA DDS

Una vez definidas todas las texturas que se van a usar en el modelo se procede al mapeado. Para ello se usan materiales *standard*, ya que otros tipos de materiales no los reconocería el motor de videojuegos y nos produciría un error a la hora de importar el modelo. A la hora de crear un material sólo se introduce la imagen que proporciona la apariencia deseada, que va asociada al *diffuse color*, y el *normal map*, que va asociado al *bump*.

Los materiales que no contienen imagen de textura, como puede ser el acero cromado de los pilares o el vidrio, simplemente se les asigna un color para diferenciarlos. Se realizará su posterior edición en Unreal Engine, como veremos más adelante.

Una vez definido el número y aspecto de materiales que van a ser usados en la escena, se aplican a la geometría ajustando en tamaño y posición de las imágenes para que queden lo más fieles posibles al modelo real. En el texturizado del zócalo se ajusta en escala y posición de la textura previamente compuesta, junto al mapa normal correspondiente, para que simule las piezas de pavimento sin ser necesario el modelado de todas las piezas.

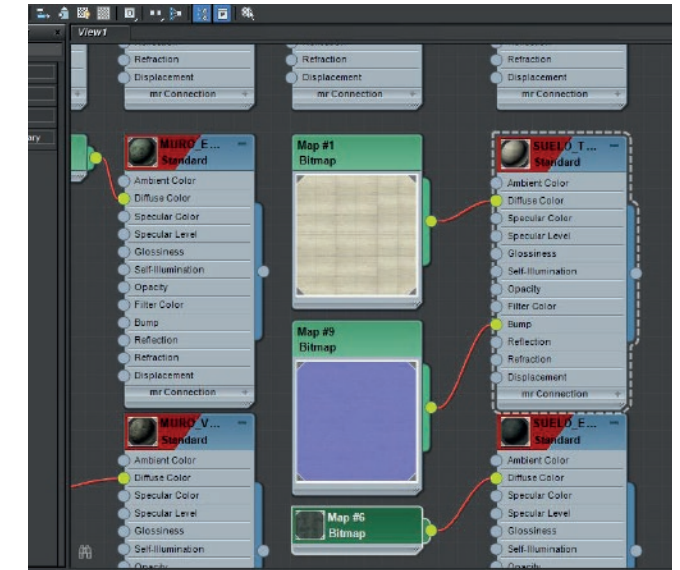


Figura 5.23_ Interfaz del Slide Material del software 3D Max con algunos de los materiales que se usan en el modelo

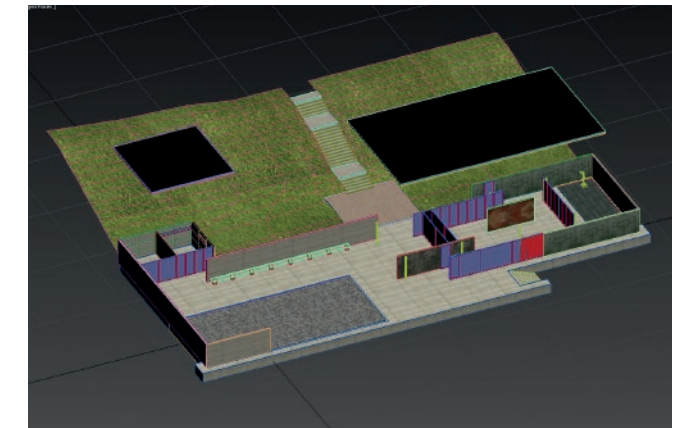


Figura 5.24_ Modelo tridimensional mapeado

4· Comprobación de Normales y generación de Lightmap

Como se ha comentado en apartados anteriores de este trabajo, los modelos tridimensionales están compuestos en por polígonos planos definidos por dos caras, una principal y otra opuesta. Se debe de comprobar que los vectores normales de estos polígonos están correctamente orientados según el modelo, para evitar visualizaciones incorrectas en el programa de videojuegos.

Se activa para ello en el visor de 3d Max la opción *Face Direction*. Corrigiendo las direcciones de los polígonos erróneas con el modificador *Normal*.

Como se explica en los tutoriales online de Unreal (docs.unrealengine.com), todos los modelos tridimensionales necesitan un mapeado en el canal 1 para su correcta iluminación, el canal cero iría asociado al texturizado visual del modelo. Este mapeado se denomina *lightmap*.

Con la referencia de los tutoriales online, la generación del *lightmap* se realiza de la siguiente forma. Se genera un nuevo mapeado con la herramienta *Unwrap UVW* que crea una disposición de todos los polígonos de nuestro modelo a modo de desplegable. Este mapeado debe de estar en el canal dos, o *Map Channel* dos, para que en Unreal se asocie al *lightmap* que sería el canal uno en Unreal. El canal uno en 3ds Max correspondería al canal cero de Unreal.

Para finalizar, se ajusta el mapeado en al cuadro que nos delimita la herramienta, sin superponer polígonos uno encima de otro.

A continuación se exponen algunas imágenes para ver los diferentes mapas de *Unwrapping* que se generan dependiendo de la geometría (figuras 5.26 y 5.27).

52

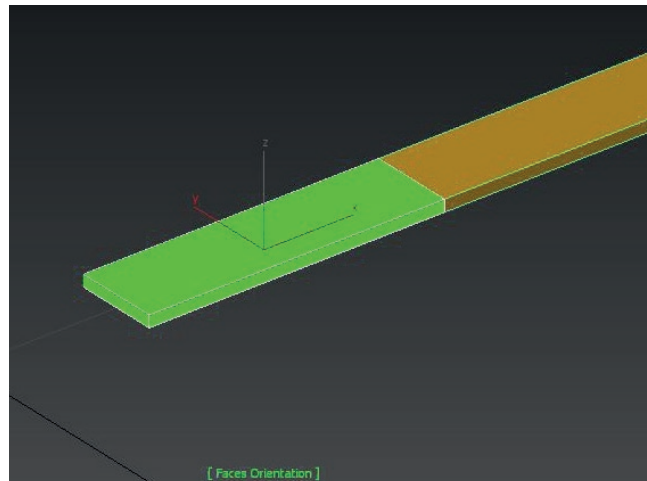


Figura 5.25_Visor Face Orientation en 3D Max

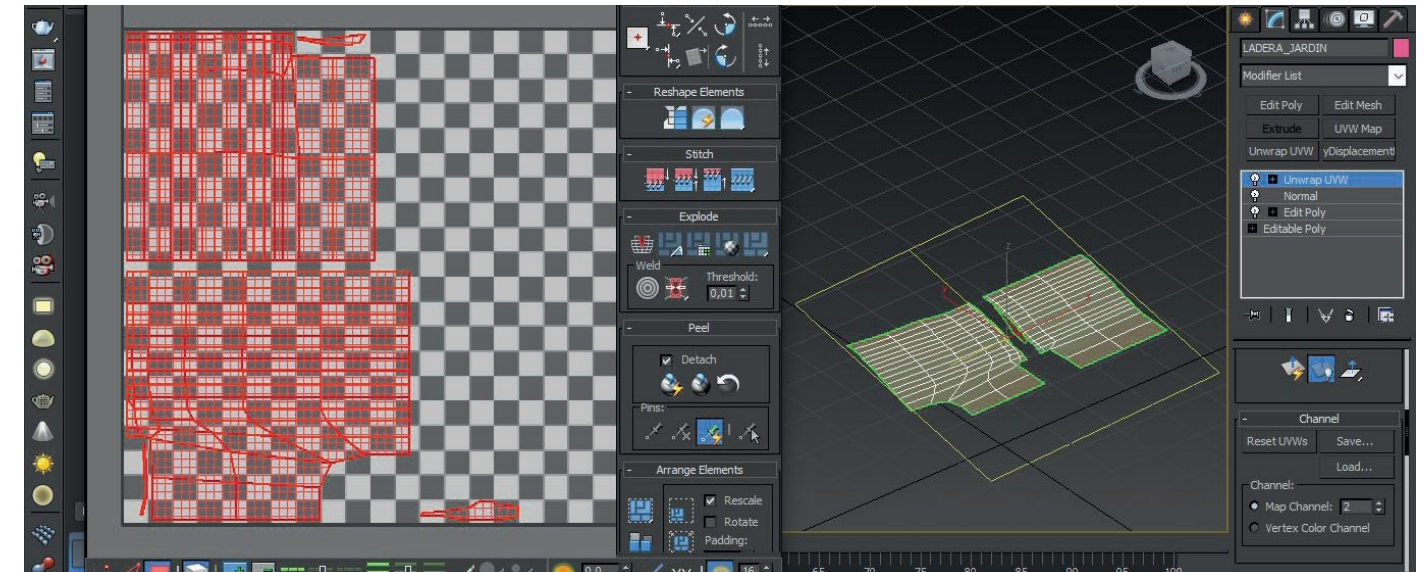


Figura 5.26_Mapeado Unwrapping de parte de la geometría del modelado

53

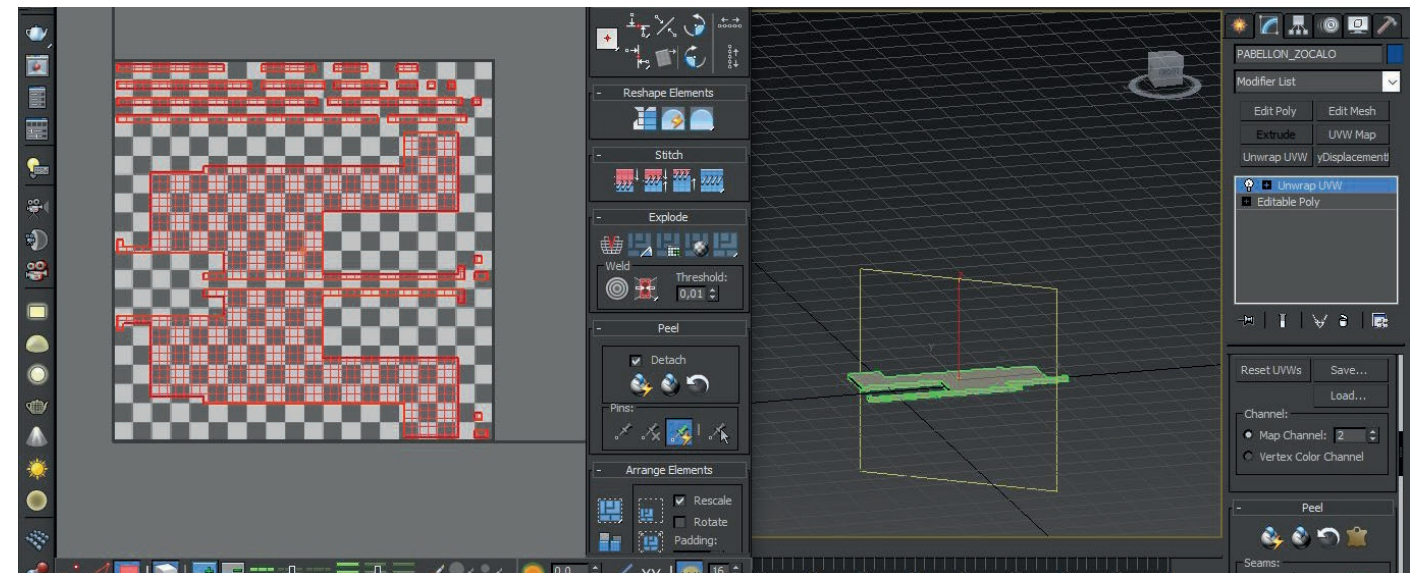


Figura 5.27_Mapeado Unwrapping de parte de la geometría del modelado

Comenzamos a trabajar en Unreal Engine

A partir de este punto se empieza a describir los diferentes aspectos tratados en el programa durante la realización del modelo, desde la edición de materiales, ajustes de luces o definición de colisiones, hasta la incorporación de sonido ambiente en la escena.

Se trabaja con el editor gratuito que se puede descargar desde la página oficial de Epic Games (www.unrealengine.com), con previa creación de cuenta. El modelo se ha realizado principalmente con la versión 4.10.4 del motor de videojuegos, aunque en poco tiempo se ha actualizado, gratuitamente, hasta la versión 4.13, la última actualización publicada recientemente el 1 de Septiembre.

5 · Crear un nuevo proyecto

Al ejecutar el programa descargado lo primero que se verá es un menú global del software de Epic Games. Aquí, además de ejecutar el editor propio Unreal Engine, se puede encontrar foros, enlaces a tutoriales, últimas noticias, juegos para descargar de la comunidad Unreal y más opciones en las que no se va a profundizar ya que no es objetivo del trabajo.

Al presionar el botón de “iniciar” se ejecutará el editor de aplicaciones Unreal Engine. La primera ventana que aparece permite elegir la plantilla que se desee para iniciar un nuevo proyecto o empezar sin contenido predefinido, como muestra la figura 5.29. También aparecen las opciones de dispositivo de salida, pantalla (*Desktop*) o móvil y *Tablet*, y la selección de calidad, así como la ruta de guardado del proyecto.

Para el motivo de investigación que ocupa este trabajo se comienza a partir de la plantilla de juego en primera persona (*First Person*), para ordenador, con máxima calidad y sin contenido inicial.

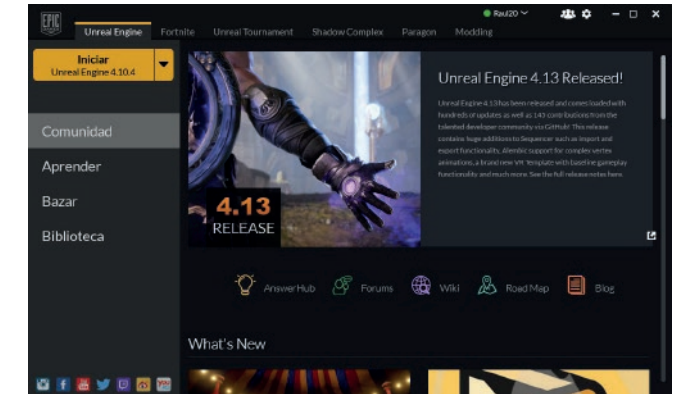


Figura 5.28_Ventana de inicio de Epic Games



Figura 5.29_Ventana de Nuevo proyecto de Unreal Engine

6 · Interfaz del programa

Al seleccionar los ajustes deseados y crear un proyecto nuevo lo primero que se ve es una pantalla dividida en varios sectores como se puede ver en la figura 5.30. Esta pantalla corresponde al *Level Editor*. A continuación se van a describir brevemente las diferentes partes.

Ventanas (Viewports)

En la parte central se encuentra el visor de la escena, que permite ver en todo momento el espacio virtual sobre el que se está trabajando. Tiene diferentes tipos de vistas, la que se encuentra predefinida es la vista en perspectiva, y diferentes tipos de visualización del espacio.

56

Para moverse por este visualizador se utiliza el teclado y el ratón para desplazarse, acercarse/alejarse y rotar la visión.

Barra de herramientas (Toolbar)

Esta barra, como en la mayoría de aplicaciones, contiene un grupo de comandos que permiten el acceso a herramientas u operaciones que se realizan con frecuencia en el programa.

Modos (Modes)

En la parte superior del lateral izquierdo se encuentra la ventana de modelos. En ella se encuentran diferentes tipos de elementos que podemos incorporar al modelo, como por ejemplo geometrías básicas, actores o luces.

Buscador de contenido (Content Browser)

En el mismo lateral en la parte inferior está el buscador de contenido. Permite navegar entre los distintos elementos generados y ordenarlos en forma de carpetas.

Elementos de la escena (World Outliner)

Se trata de un gestor de elementos de la escena. Permite ver un listado de todos los elementos que se encuentran insertos en el espacio virtual seleccionarlos, ordenarlos por carpetas y desactivar su visión en el viewport de manera similar que podemos hacer en Photoshop. Además proporciona información de que tipo es cada elemento.

Detalles (Details)

En este apartado se pueden modificar los ajustes de los diferentes elementos seleccionados de la escena o los ajustes globales, llamados, Ajustes del mundo.

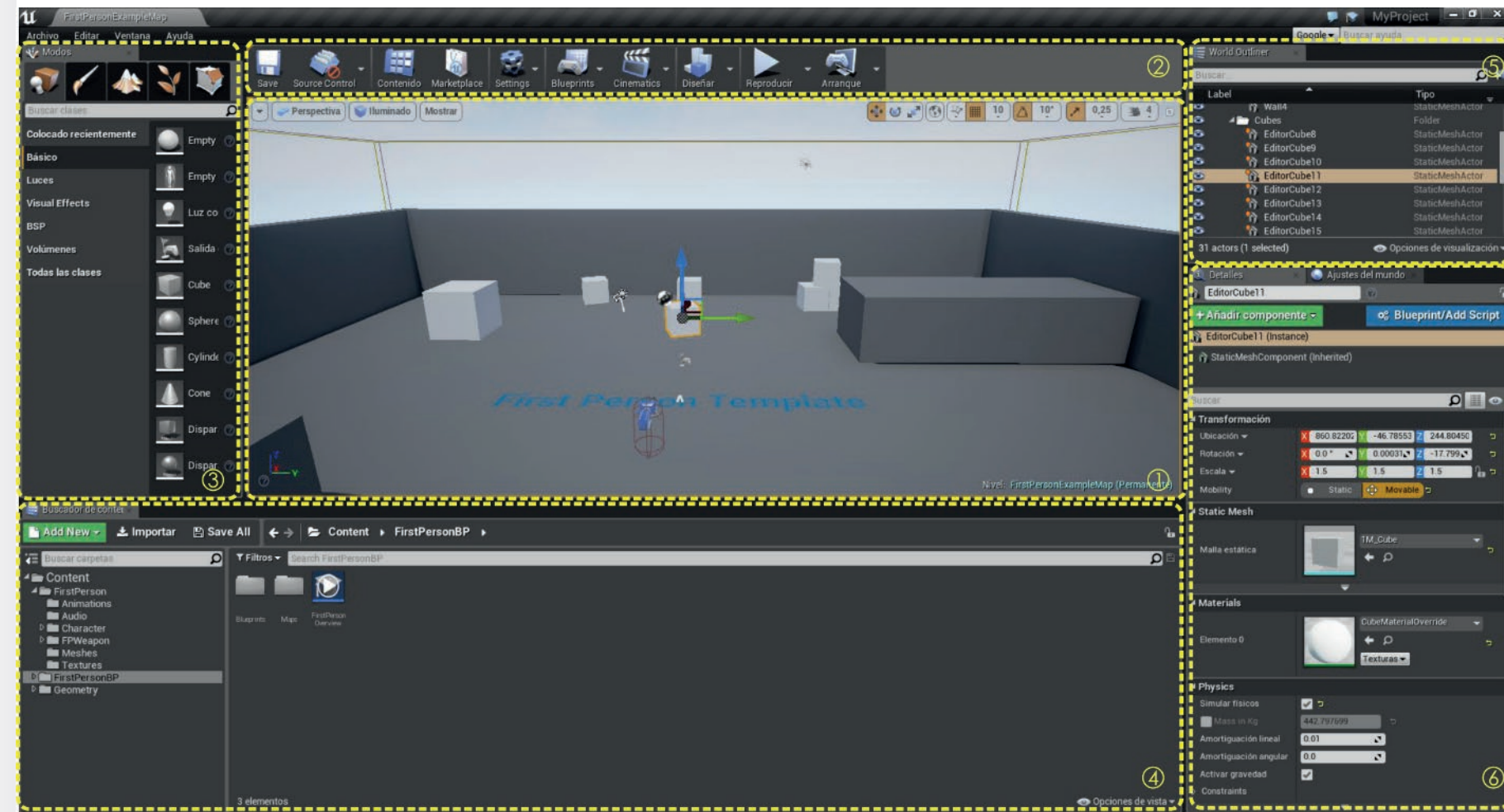


Figura 5.30_ Interfaz del software Unreal Engine

- 1 - Ventanas (Viewports)
- 2 - Barra de herramientas (Toolbar)
- 3 - Modos (Modes)
- 4 - Buscador de contenido (Content Browser)
- 5 - Elementos de la escena (World Outliner)
- 6 - Detalles (Details)

7 · Edición de la plantilla

Al abrir el archivo con las opciones citadas anteriormente, se puede ver un espacio delimitado por cuatro paredes, una cámara y unos cuantos prismas sueltos en el escenario.

Si se presiona el botón reproducir en la barra de herramientas se iniciará una simulación del juego. Los controles son sencillos, con las teclas W,A,S,D se desplaza por el escenario, con el ratón se rota la cámara, con espacio se salta y con el botón izquierdo disparamos. Lo interesante de esta plantilla, para realizar una visualización arquitectónica, es que dispone de una cámara con unos controles ya programados y en primera persona, por lo que evita el tener que realizar ese apartado.

58



Figura 5.31_Juego lanzado sin modificar la plantilla

Se va a trabajar con esta plantilla de base pero se van a realizar modificaciones para ajustarlas mejor al resultado que se quiere conseguir.

En primer lugar se eliminan los prismas de la escena y el letrero "First Person Template", dejando el suelo para tener una referencia.

En segundo lugar se va a modificar aspectos de la cámara. Se eliminan las mallas del arma y de los brazos, ya que no son necesarios para el modelo que buscamos realizar, y se borran las programaciones del comando disparo en los "Blueprints", para que al pulsar el botón izquierdo en la visualización no ocurra nada.

De esta forma se tiene una plantilla preparada para insertar el modelo arquitectónico y comenzar a trabajar, como se muestra en la fig 5.33.

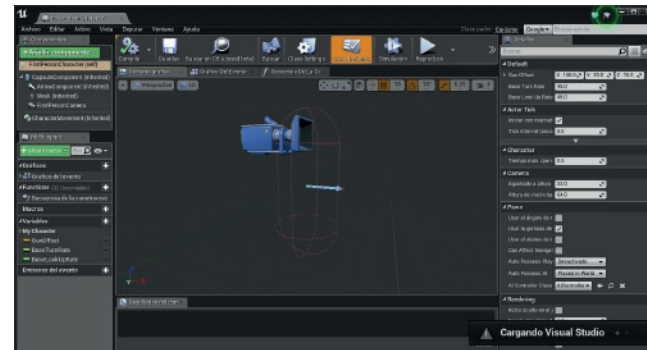


Figura 5.32_Interfaz de modificador de Blueprints de la malla de la cámara ya modificada

59



Figura 5.33_Visión general de la plantilla modificada

8 · Importar un modelo 3D

Se vuelve al programa de 3D Studio Max para ver en qué tipo de archivo y cómo exportamos la malla que se ha modelado anteriormente. Se seguirán los pasos recomendados por los tutoriales online de Unreal Engine y el curso virtual “3Dmax+Unreal”, de la web Udemy. (www.udemy.com).

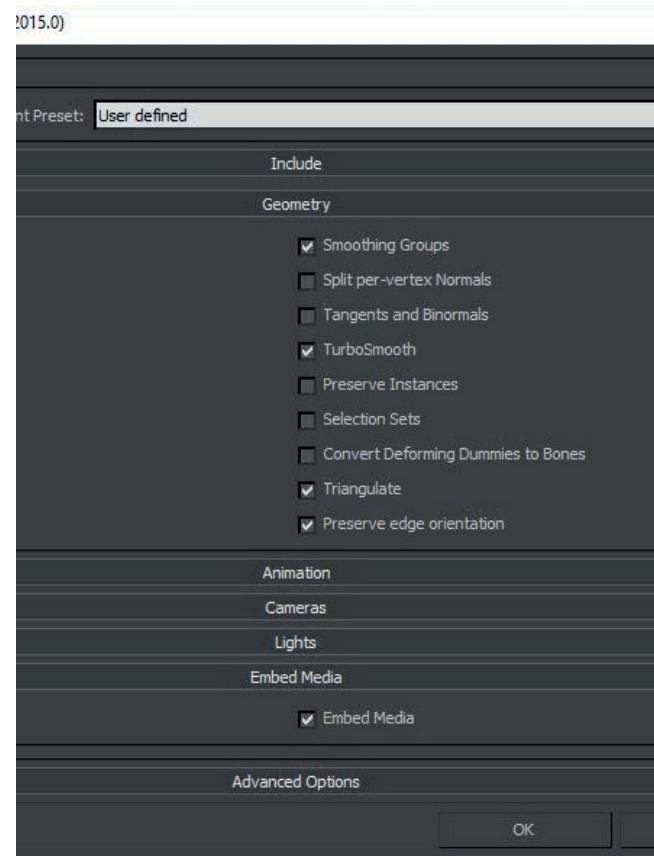


Figura 5.34_Parámetros seleccionados para exportar modelado a archivo .FBX

Con el modelo entero seleccionado se exporta en el formato .FBX, formato recomendado en el tutorial online de Unreal Engine. Los parámetros seleccionados a la hora de exportar la malla son los siguientes: Smoothing groups, para conservar cualquier tipo de suavizado que contenga el modelo, TurboSmooth, para suavizar superficies, Triangulate, esta opción triangulará de forma automática los polígonos de nuestro modelado y Preserve edge orientation, para conservar la orientación de los vectores normales. (Figura 5.34)

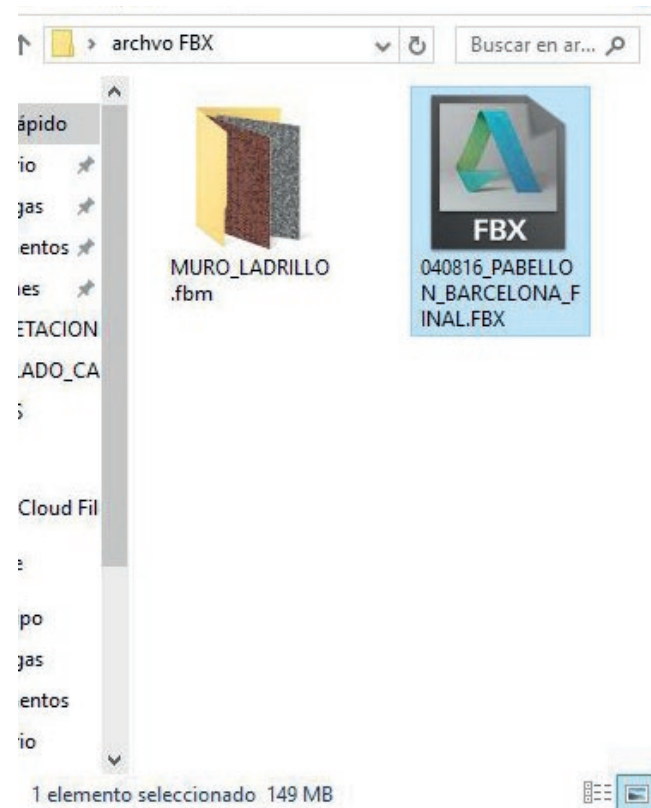


Figura 5.35_Archivos generados al exportar

Una vez generado el archivo FBX se regresa a Unreal. Para importar un modelo en Unreal clicamos la opción de importar que se encuentra en la zona del buscador de contenido y seleccionamos el archivo deseado.

Aparecerá una ventana para definir las opciones de importación. Se tratarán solo dos de estos parámetros. Por un lado, la pestaña de combinar mallas permite, si está activada, juntar todas las mallas del modelo FBX haciéndolo un único elemento. En el caso contrario conservará los elementos independientes tal y como vienen del modelo original de 3D Max, en este caso. Por otro lado, en el apartado de transformaciones se debe de tener en cuenta en que unidades se ha trabajado el modelo en 3D studio, ya que Unreal trabaja por defecto en centímetros.

Para el modelo de estudio y teniendo en cuenta que se ha creado con unidades en metros, se define las siguientes opciones de importación. Ver figura 5.36.

Importado el Pabellón, aparecerá en el buscador de contenido la geometría, los materiales y las texturas que tiene el modelo tridimensional.

Para insertar el modelo tridimensional en la escena, basta con arrastrar el modelo desde el buscador de contenidos al viewport.

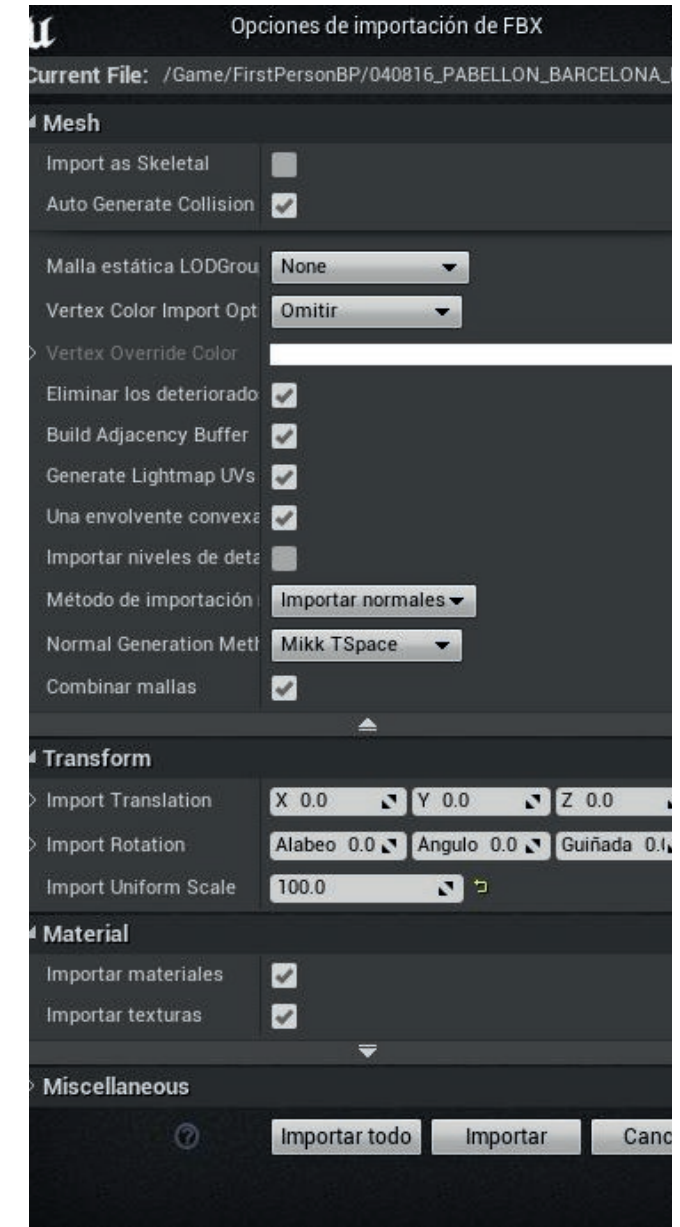


Figura 5.36_Parámetros para importar modelado en Unreal Engine

9 · Colisiones

Una vez importado el modelo se trabaja sus colisiones para que el programa detecte la geometría tridimensional como un elemento físico y se pueda interactuar con él.

Se selecciona la geometría del pabellón y en el apartado de detalles se vuelve a dar doble *click* a la malla estática. Se abrirá una nueva ventana de edición de geometrías. Aquí se establecerán los volúmenes de colisiones o el comportamiento de la malla estática.

Como se ha combinado la malla para trabajar más cómodamente en la edición del modelo, bastará con cambiar en "Ajustes de la malla estática", en el apartado "Complejidad de la colisión" a la opción "Usar la colisión compleja como colisión simple". Con esta acción la malla será físicamente correcta y ya no será posible atravesarla, lo cual permitirá interactuar y caminar por ella.

En el espacio de los estanques se añadirán más adelante, junto a los planos que definen la lámina de agua, geometrías de colisión para evitar que durante la visualización digital se pueda entrar en dichos estanques.

Si el modelo no hubiese estado combinado se debería hacer esto con cada elemento que compone la geometría global del edificio. Una buena opción sería el uso de cajas simplificadas de colisión (*Box simplified collision*).

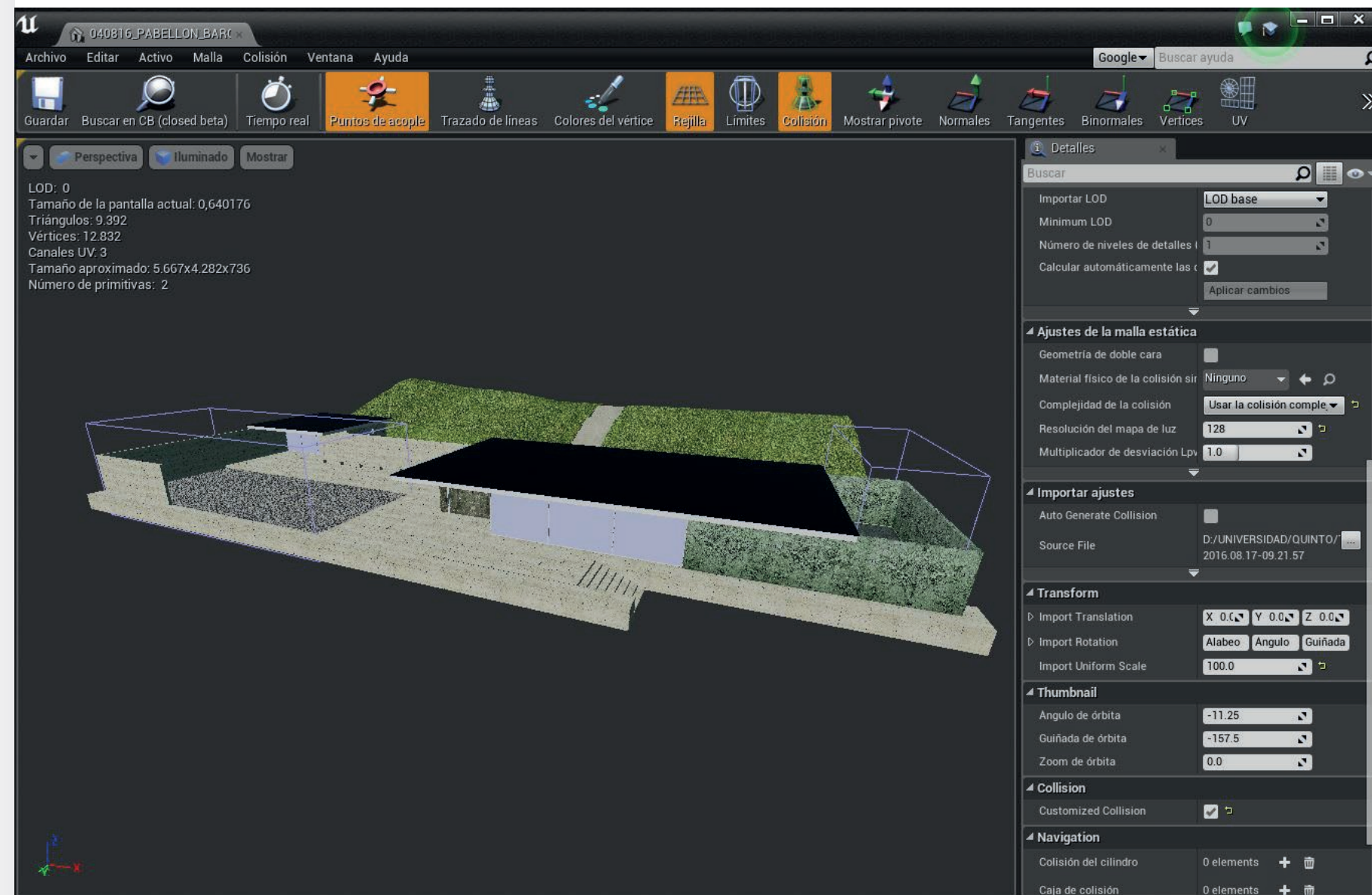


Figura 5.37_ Interfaz de edición de malla estática con parámetros de colisión

10 · Materiales

En este apartado se va a tratar un tema importante en las visualizaciones tridimensionales, la edición de materiales. Se verán de forma breve las características básicas de los materiales en Unreal, como trabajan y se mostrará cómo se han realizado algunos de los materiales que aparecen en el modelo de estudio.

Para comenzar a trabajar con los materiales en Unreal, se puede partir desde cero con la creación de un material nuevo, pulsando el botón derecho en la ventana de buscador de contenidos y seleccionando material. Por otro lado se pueden editar los materiales que se han obtenido al importar el modelo. En este caso se van a editar los materiales del modelo ya que son los que tiene el modelo y están ajustados previamente su tamaño y escala en 3D Studio.

Para editar un material se hace doble clic sobre el icono de éste en el buscador de contenido. Al hacer esto se abrirá la ventana Editor de Materiales (*Material Editor*). Se van a describir las diferentes zonas del Editor de Materiales que se muestra en la figura 5.39.

Por un lado se tiene la barra de herramientas en la parte superior, que funciona de manera similar a la del *Level Editor*, en el lateral izquierdo está la previsualización del material con los detalles que definen la mismo. En la zona central se encuentra el espacio de editor de material, que funciona con

conectores, similar a la interfaz del *Slide Material* de 3D Max o Grasshopper. Por último en el lateral derecho se encuentra la paleta de herramientas que se disponen para editar el material.

Las principales entradas que se le pueden hacer al material son las siguientes. Color base, que sirve para introducir la textura o color que deseamos que tenga el material, en el caso de estudio se ve como esta textura es la que se aplicó en 3D Studio Max, sería la equivalencia al *Difusse*. Metálico, con esta entrada se determina si el material tiene un aspecto metálico en mayor o menor medida. Especular, controla la cantidad de reflejo que tiene el material, sería el equivalente a *Reflection*. Desigualdad, parámetro que controla si el reflejo es más o menos nítido, equivaldría a la inversa de *Glossiness* en V-ray. Por último *Normal*, en esta entrada se introducen los mapas de relieve.

A continuación se van a explicar brevemente como se han realizado algunos de los materiales del modelo. Se escogen materiales de diferentes condiciones para ver las posibilidades que puede ofrecer el programa Unreal Engine.



Figura 5.38_Iconos de materiales en el buscador de contenido

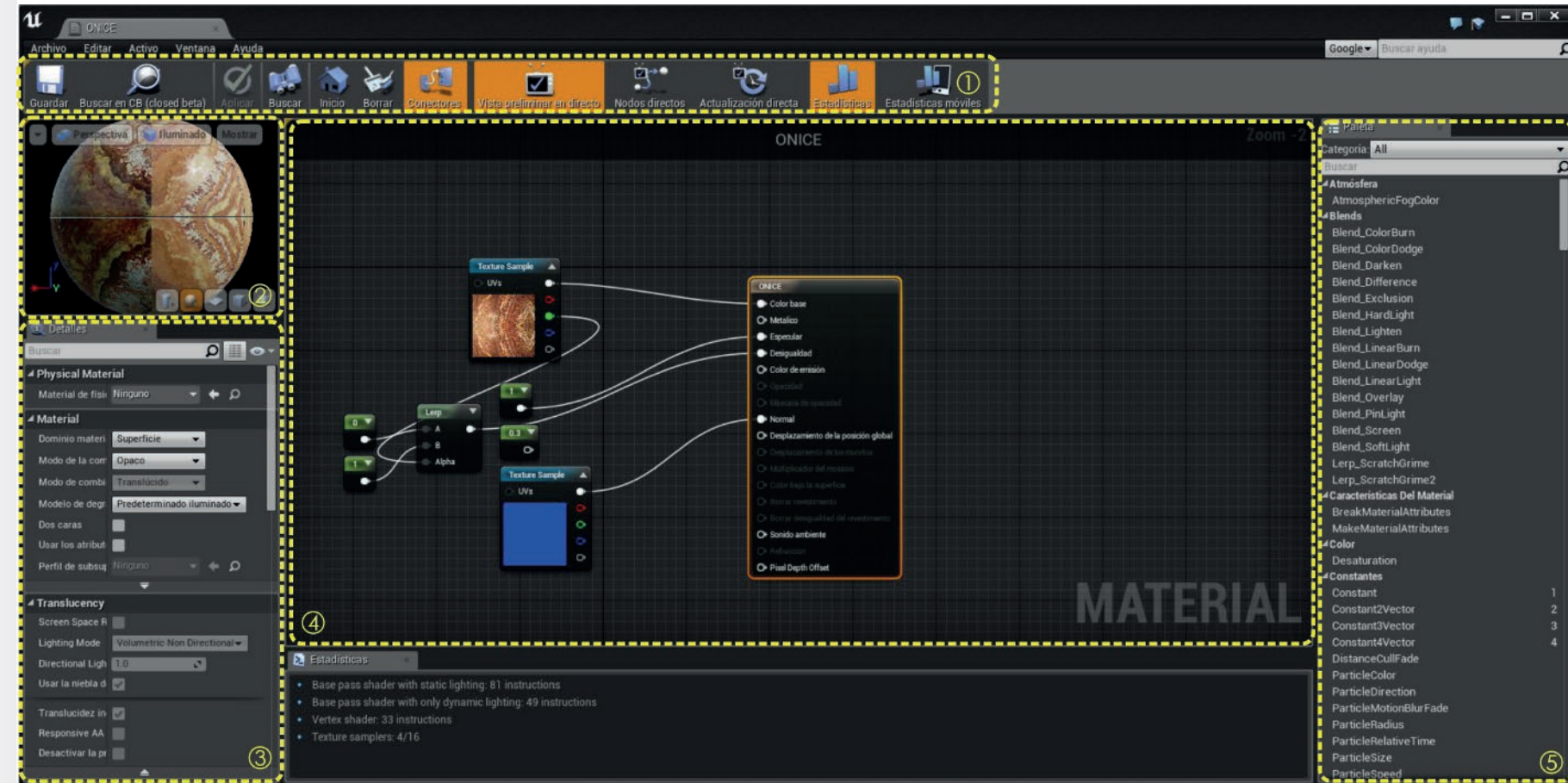


Figura 5.39_Interfaz de Material Editor

- 1 - Barra de herramientas (*Toolbar*)
- 2 - Previsualizador del material (*Viewport*)
- 3 - Detalles (*Details*)
- 4 - Ventana (*Graph*)
- 5 - Paleta (*Palette*)

Se comienza por el material más sencillo, material cromado. Para realizar este material se introduce de color base un gris claro para determinar el color de fondo del metal. Para la edición de los demás aspectos del material se introducen constantes de un solo vector (*Constant*) de la paleta. También es posible introducir estos valores pulsando el botón derecho en el espacio de edición. Estos modificadores contienen valores de una sola cifra, con ellos vamos a definir las características de nuestro material. En la entrada "Metálico" y "Especular" definimos el valor "1", ya que es un material totalmente metalizado y se quiere que refleje con totalidad. En desigualdad se introduce el valor "0.1" para que el reflejo no sea totalmente nítido.

En segundo lugar se detalla el material ónice. Se trabaja con el material que viene definido de 3D Max que tiene definida la imagen de "color base" y el mapa normal para realizar el efecto de juntas en la geometría. El parámetro metálico se deja vacío ya que no influye en este material y en especular introducimos el valor "1" para tener un material reflectante. Para el tratamiento de desigualdad se va trabajar con el modificador Lerp (*Linear Interpolate*). Permite introducir una imagen en el canal alfa, en este caso el canal verde de la textura principal e interpolar dos valores que van asociados a blanco y a negro. De este modo se consiguen unos reflejos más reales y detallados.

Para el material vidrio se trabaja con el mismo razonamiento que en los casos anteriores añadiendo dos nuevos conceptos. Por un lado se trabaja con el modificador *Fresnel* para obtener un reflejo menos pronunciado cuando el vidrio se ve de frente y obtener un comportamiento más realista. Por otro lado se debe modificar el "Modo de la combinación" a translúcido en el apartado de detalles. Así el material permitirá ver a través de él y dejar pasar la luz.

Siguiendo este mismo razonamiento se realizan todos los materiales que componen la escena, trabajando su edición a partir de los materiales base importados del programa de modelado.

Para concluir el apartado de materiales, decir que se pueden sustituir los materiales del modelo de una forma sencilla. Simplemente con seleccionar la geometría a editar y sustituir arrastrando el material nuevo por el material a cambiar en la ventana de "Detalles" se verá como el material cambia.



Figura 5.40_Parámetros de material cromado

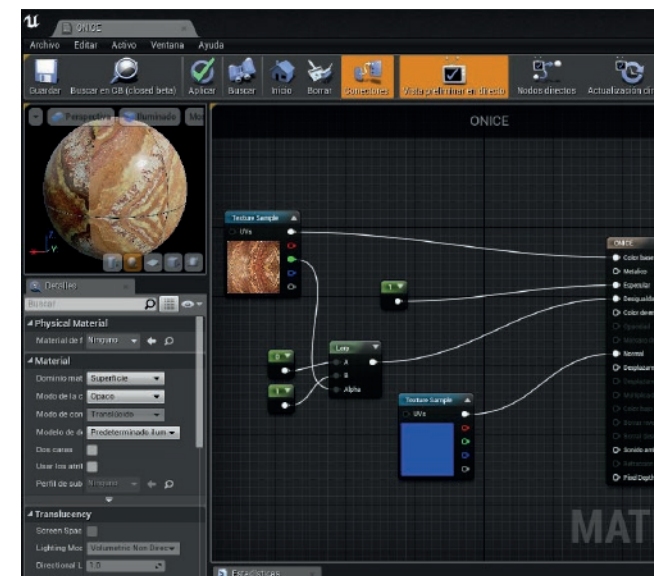


Figura 5.41_Parámetros de material ónice

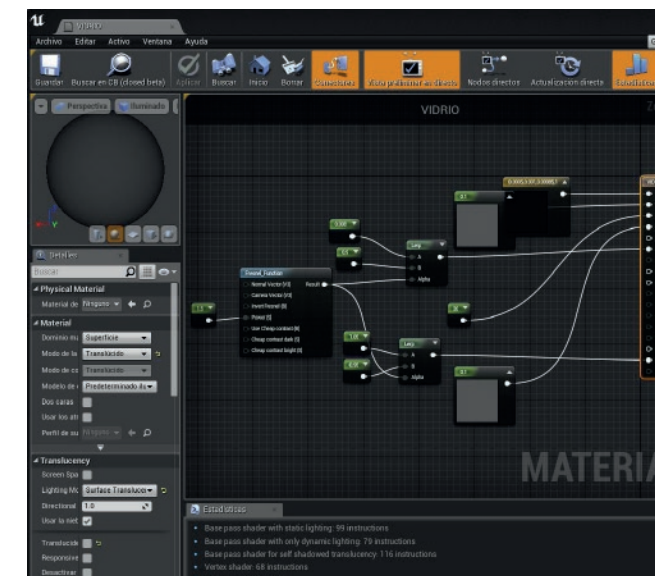


Figura 5.42_Parámetros de material vidrio

11 · Luces y reflejos

El apartado de iluminación en el modelo realizado ha sido sencilla, pues se ha iluminado toda la escena únicamente con la luz solar (*Light Source*). Este punto de luz viene definido con la plantilla de la escena.

Se han modificado algunos aspectos para conseguir una mejor iluminación y una escena más cuidada. Se ajusta la dirección y posición de los rayos de iluminación para obtener las sombras arrojadas deseadas en el modelo.

Además se modifica en el apartado de detalles la intensidad de la luz, para conseguir una mejor iluminación y algunos parámetros de tratamiento de sombras. Se muestra en la figuras 5.43 y 5.44 el resultado de las sombras arrojadas.

El tratamiento de reflejos funciona de forma diferente en Unreal a lo que se puede estar acostumbrado por otros programas de edición de infografías.

El material por sí mismo no refleja el entorno, refleja un mundo estándar y vacío, como si un HDRI se tratase. Para que los objetos reflejen el entorno y los elementos que se encuentran a su alrededor se deben introducir Esferas de captura de Reflexión (*SphereReflectionCapture*).

Estos elementos capturan de forma esférica la imagen que se encuentra a su alrededor,

como si unas esferas de cromo se tratasen, para proyectar esta captura, a modo de reflejo, en los demás objetos. Por supuesto estos elementos son invisibles cuando se ejecuta la aplicación.

Se introducen varios de estos elementos en la escena, en diferentes posiciones y con diferentes radios de influencia, para conseguir un reflejo adecuado y lograr un alto grado de realismo. En la figura 5.45 podemos ver varias Esferas de Captura de Reflexión en el interior del modelo con sus radios de influencia.

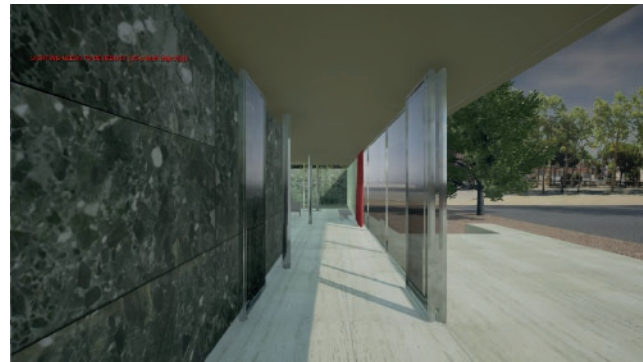


Figura 5.43_Resultado de las sombras arrojadas con la iluminación aplicada



Figura 5.44_Resultado de las sombras arrojadas con la iluminación aplicada

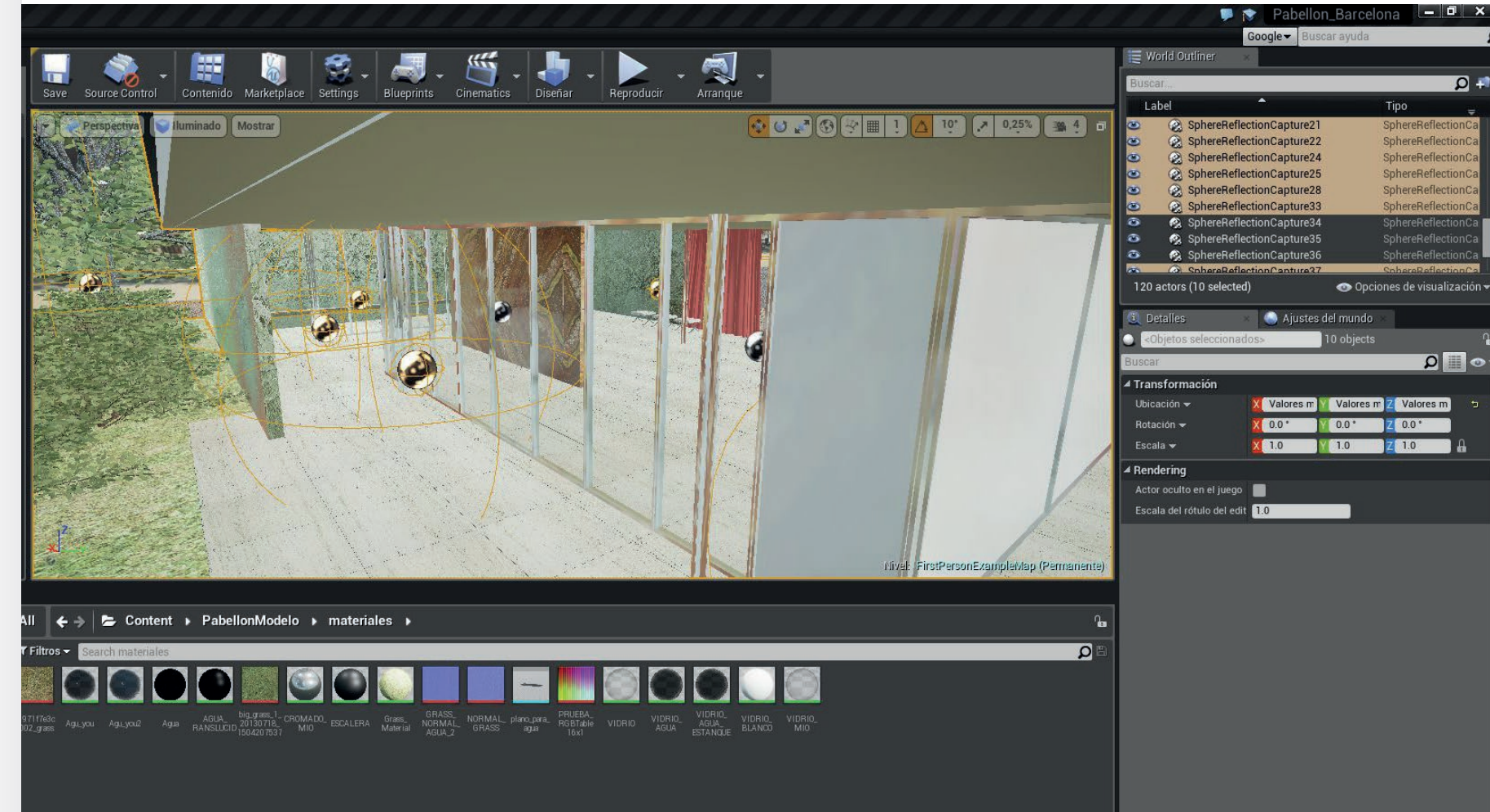


Figura 5.45_Esferas de Captura de Reflexión en el modelo

12 · Post Process Volume

Los efectos de imagen que permiten tratar esta herramienta sería el equivalente a los efectos que se pueden realizar en postproducción a una imagen renderizada pero en una presentación en tiempo real.

Estos efectos son visibles dentro del volumen definido por el *Post Process Volume*. Con lo cual se pueden tener diversos volúmenes de efectos y determinar distintos tratamientos de imagen según las áreas. Sería útil para dar diferentes tratamientos al interior de un edificio que tiene poca luz, un bosque que se necesita mayor grado de contraste o incluso cuando se sumerge una cámara y se quiere simular un efecto acuático. Los efectos de imagen afectan a como se visualiza toda la imagen mostrada.

Algunos de los efectos más destacados en el campo de visualización arquitectónica son el *Bloom*, que crea aureolas alrededor de las fuentes lumínicas, *Color Grading*, que permite ajustar valores de la imagen como Saturación contraste o el *Gamma* o *Depth Of Field* permite ajustar la profundidad de campo o ajustar la distancia para tener zonas enfocadas y desenfocadas.

Destacar la herramienta *Scene Color* que mediante el manejo de imágenes LUT permite realizar un ajuste completo de los efectos de imagen en Photoshop e implementarlos en el

programa. Como se puede observar en las figuras 5.46, 5.47, y 5.48.

La palabra LUT significa *Look Up Table*. Unos archivos con una tabla con valores que modifican los colores de entrada para corregir los colores de salida, que es lo que finalmente se ve en pantalla. Estas imágenes consiguen obtener resultados más cinematográficos.

En el modelo se modifican algunos parámetros para obtener una temperatura de color más adecuada, iluminaciones mejor definidas y mayor resolución de texturas.



Figura 5.46_Imagen inicial sin modificar

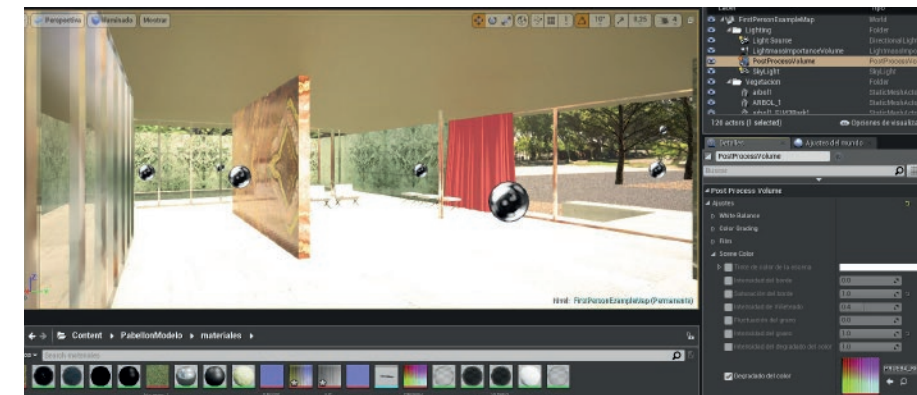


Figura 5.47_Imagen modificada, cambiando temperatura de color, subiendo la iluminación y contrastes, aplicando imagen LUT.



Figura 5.48_Imagen modificada, desaturada, y con menos iluminación, aplicando imagen LUT.

13 · Mobiliario, vegetación y entorno

Una vez definidos todos los aspectos anteriores, se procede al trabajo de incorporación de mobiliario, vegetación y entorno.

Este trabajo se realiza tras haber hecho diferentes pruebas de ejecución de la aplicación y detectar carencia de realismo. El modelo estaba en un entorno abstracto sin definir, lo cual no beneficiaba a la inmersión.

Para la incorporación de mobiliario se trabaja con los modelados del sillón de Barcelona, realizado para este mismo edificio, y el taburete Barcelona. Estos modelos se trabajaron en la asignatura optativa citada anteriormente.

Los pasos a seguir para la importación del modelo en el software Unreal serían los mismos que se han descrito anteriormente para la importación del modelado del pabellón. En las siguientes figuras vemos como se texturiza y se crea el *lightmap* de las geometrías.

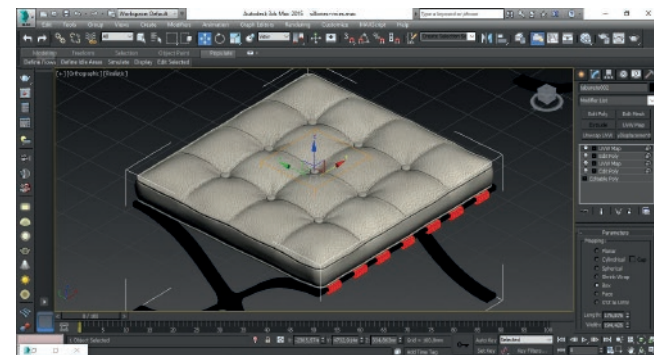


Figura 5.49_Texturizado y mapeado del taburete

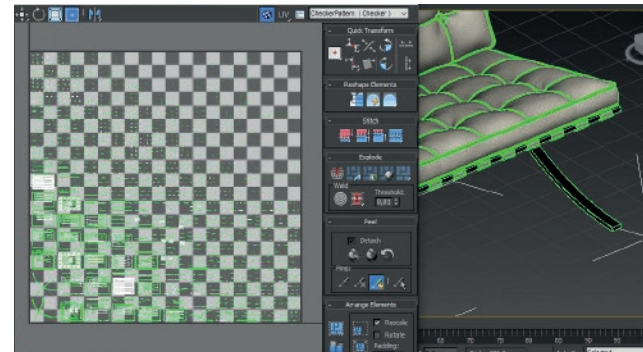


Figura 5.50_Unwrapping del sillón Barcelona

Una vez importados al programa Unreal se ajustan los materiales que componen al sillón y se distribuyen por el edificio basándose en diferentes imágenes de referencia.

Los elementos vegetales se trabajan a partir de modelos gratuitos descargados desde la web de la empresa Xfrog (www.xfrog.com).

Los modelados se modifican, cambiando principalmente sus materiales de *V-ray Material* a *Standard Material* y simplificando algunas geometrías. Un trabajo óptimo sería haber reducido el número de polígonos del modelo para tener menos elementos en escena y no consumir excesivos recursos del sistema. Siempre que se lleve a cabo esta práctica hay que tener especial cuidado para que la visualización de este elemento no resulte irreal.

Debido a que el sistema en que se realizan los lanzamientos soporta bien la escena con la geometría sin modificar se conserva el número de polígonos de los árboles.

Se trabaja con tres modelos para generar distintos tipos de árboles y arbustos. Al igual que con el mobiliario se organiza la vegetación basándose en imágenes de referencia.

Para conseguir un mayor realismo se crea un material para las hojas con modo de combinación enmascarado, que nos permite tener un material similar al *2Slide Material*, dejando pasar parte de luz a través de él. Además se activa la pestaña de dos caras, para evitar en este material el *Back-Face Culling* y se añade movimiento al material, para simular el viento con el modificador *Simple Grass Wind*.



Figura 5.51_Material de hojas para la vegetación. Se trabaja con material de tipo enmascarado y se añade movimiento con la herramienta *Simple Grass Wind*

Por último para incorporar entorno se modela y texturiza de forma muy sencilla la plaza que se encuentra en la parte delantera del pabellón.

Además, se añade un anillo con forma oval que cubre toda la escena para que sirva de imagen de fondo. La imagen que se utiliza para texturizar el anillo se realiza a partir de diferentes fragmentos de imágenes extraídas de *Google Street View*. Las figuras 5.52, 5.53 y 5.54 muestran la composición final.



Figura 5.52_Previsualización en entorno de trabajo

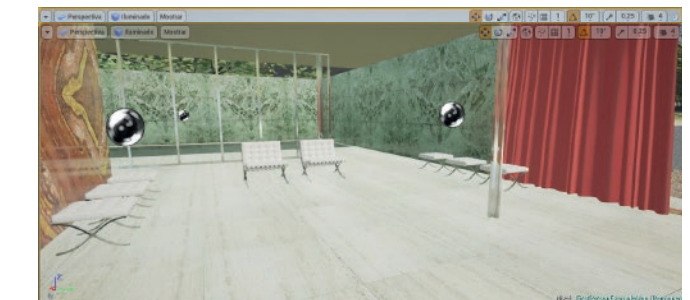


Figura 5.53_Previsualización en entorno de trabajo



Figura 5.54_Previsualización en entorno de trabajo

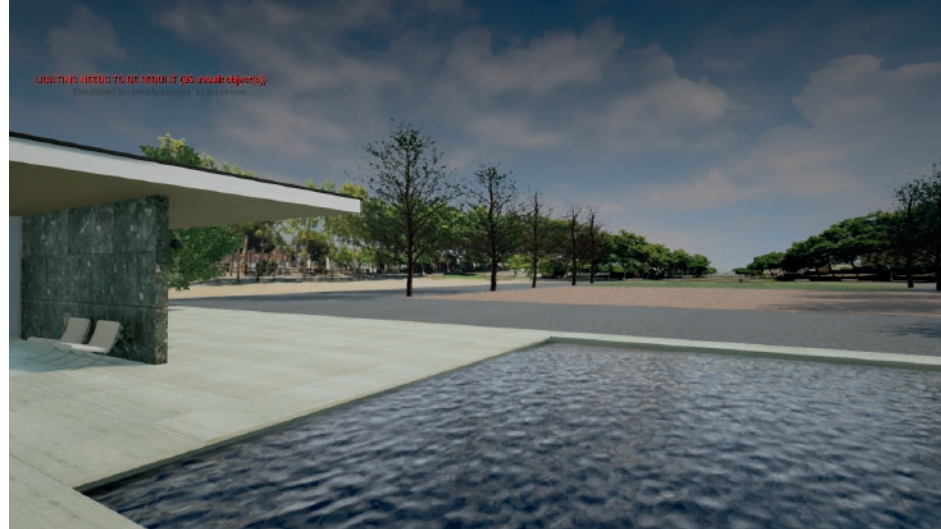


Figura 5.55_Visualización final de la aplicación
 Figura 5.56_Visualización final de la aplicación
 Figura 5.57_Visualización final de la aplicación

14 · Sonido

Se incorpora sonido a la escena para generar un ambiente más realista y aprovechar las posibilidades que nos ofrece este tipo de visualización.

Como se puede leer en el tutorial online de Unreal, los archivos que admite el programa son los archivos .WAW con una serie de especificaciones. En el buscador de contenido se podría importar los archivos de audio que se necesitasen en la escena.

Para el modelo del Pabellón se va a utilizar un archivo de sonido que viene con el contenido inicial de Unreal (*Starter Content*). Se utiliza el audio "Starter_Birds01", que reproduce sonido ambiente con cantos de pájaros.

Se arrastra el punto de emisión sonora al jardín trasero y se ajusta el radio de atenuación. De esta manera se consigue que suene con mayor intensidad la fuente sonora cuando el jugador se encuentra en el jardín que cuando se está moviendo por dentro del edificio.



Figura 5.58_Punto de emisión sonora situado en la escena

15 · Edición de menú y empaquetado

Por último, antes de generar la aplicación, se crea un menú principal que nos permitirá comenzar el juego, cambiar de resolución o salir de la aplicación.

Para la realización del menú se introduce un nuevo nivel, para que se convierta en el primer nivel al activar la aplicación y un *Blueprint Widget*. Este último nos permite realizar de forma visual la composición del menú introduciendo imágenes y botones. En la figura 5.60 se muestra la interfaz del *Widget*.

Además se programan las diferentes acciones que debe realizar cada botón. Por ejemplo, se programa el botón “OPCIONES”, para que deje de ser visible el menú principal y pase a ser visible el menú de resoluciones, de forma inversa se programa el botón de “MENU”.

76

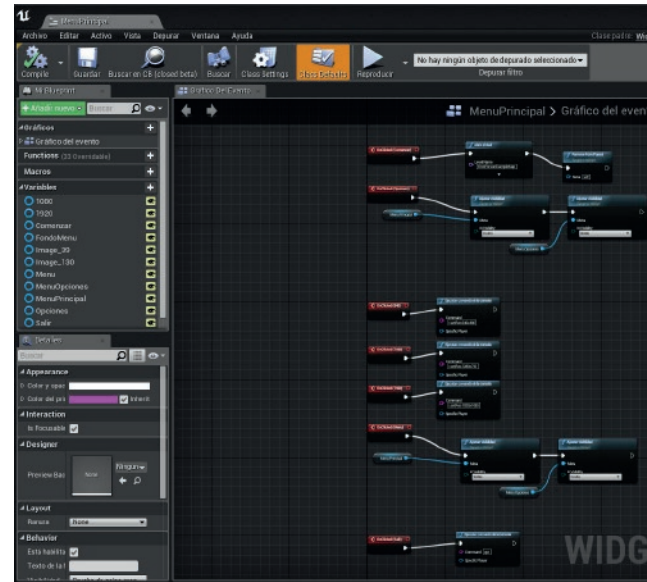


Figura 5.60_Blueprints creados para la programación del menú

Una vez obtenido un resultado satisfactorio se procede al empaquetado de la aplicación. Esta acción permite exportar el juego para su visualización en otros dispositivos.

El menú Ajustes del proyecto, que se encuentra en el apartado de Editar, permite modificar la información del proyecto. Por ejemplo nos permite introducir el nombre de la compañía, la miniatura del proyecto que deseemos ver en el editor o seleccionar las plataformas con las que se desee que sea compatible la aplicación.

Se tratan algunos de Ajustes del proyecto para el correcto empaquetado del mismo. En primer lugar se selecciona, en el apartado de Mapas y modos,

el mapa predeterminado del juego, siendo este el mapa del menú principal. Si no se realiza este cambio no aparece el menú al ejecutar el juego. Por otra parte se añaden imágenes diseñadas para la ocasión en la “Plantilla de bienvenida del juego”, que aparecerá mientras la aplicación cargue, y en “Icono del juego”, que será el icono que representará a la aplicación. Se deben realizar las imágenes en el formato y tamaño que especifica el programa (información disponible en el tutorial online).

Con los últimos retoques realizados se puede proceder al empaquetado del proyecto. Para ello se escoge el sistema operativo al que se desee exportar la aplicación, en este caso Windows (64-bits) o Mac, y una ruta de guardado.

Tras finalizar el proceso de empaquetado se puede observar que el programa realiza una carpeta con la aplicación .EXE.

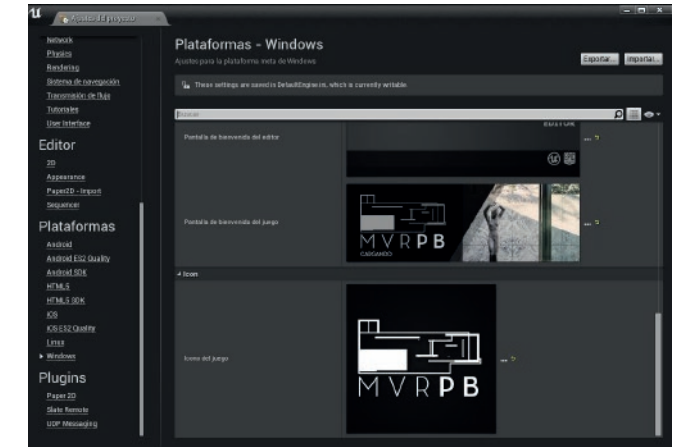


Figura 5.61_Imágenes diseñadas para el arranque del juego e icono de representación de la aplicación

77

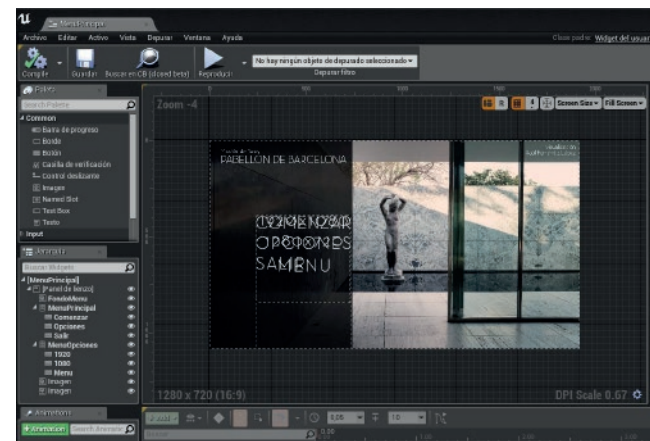


Figura 5.59_Interfaz de Widget con el menú diseñado para la aplicación

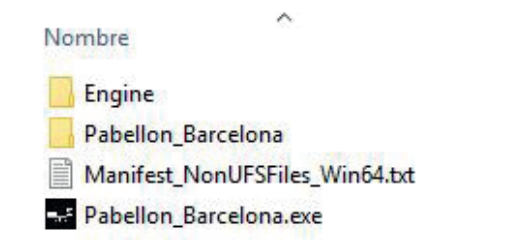


Figura 5.62_Archivos generados al empaquetar el proyecto

Visualización del resultado

Tras la realización de todo el proceso de edición del modelo con el software Unreal Engine, se realiza un video demostración para poder ser visualizado por cualquier persona que consulte este trabajo. De esta manera se puede apreciar los aspectos tratados durante la ejecución del caso práctico.



Enlace al video

<https://www.youtube.com/watch?v=RPI8yG4TjcU&feature=youtu.be>

En el presente trabajo se ha realizado un análisis de los diferentes sistemas de representación que están disponibles hoy día para la realización de infografías arquitectónicas. Con el objetivo de dar un paso más allá e investigar las posibilidades que ofrecen los motores de creación de videojuegos. Tras la experimentación anterior con técnicas de visualización arquitectónica más habituales (render, fotomontaje, render 360°...), el autor de este trabajo se ha introducido en este campo con la voluntad de iniciarse en él. Con el objetivo de acercarse al mundo de la Realidad Virtual mediante la visualización arquitectónica interactiva y en tiempo real.

En primer lugar se realiza un recorrido histórico por el proceso evolutivo de la informática gráfica. En este apartado se ha observado como en un período de tiempo relativamente corto, tanto los hardwares como los softwares, han experimentado una gran mejora de sus capacidades. Gracias a los avances tecnológicos de las últimas décadas cualquier persona con un ordenador doméstico puede realizar gráficos fotorrealistas de alta calidad.

Tras el estudio y análisis de los softwares utilizados en arquitectura se ha podido observar que los programas CAD supusieron un salto cualitativo en cuanto a productividad. Gracias a estas herramientas se minimizaron tiempos dedicados a dibujar, obteniendo además mayor calidad en los dibujos. De esta misma manera existen otros tipos de software (cálculo de estructuras, instalaciones, plazos de ejecución...) que aumentan la

productividad en la realización de un proyecto. Cada vez más los nuevos softwares apuntan a unificar tareas en una sola aplicación. Un ejemplo de ello pueden ser los programas BIM, que día tras día se utilizan con más frecuencia. Esta constante reinvención y perfeccionamiento de los softwares obliga al arquitecto a estar actualizado para ser competitivo. Estos programas amplían las posibilidades proyectuales y de diseño, pudiendo realizar formas y estudios más complejos.

La Realidad Virtual implica la creación de entornos interactivos y funcionamiento en tiempo real. Los sistemas de Realidad Virtual pueden ser más o menos inmersivos según el tipo de dispositivos de comunicación hombre-máquina que utilicemos, partiendo de lo más básico (sonido envolvente, pantalla y joystick o teclado) hasta sistemas más sofisticados (casco de realidad virtual, sistemas de control por movimiento o suelos deslizantes). Esta herramienta se está expandiendo a diferentes campos de aplicación. Se ha utilizado en ámbitos médicos, psicológicos y educativos. En el sector de la arquitectura ofrece un gran número de posibilidades. Incluso con los dispositivos más básicos se consigue una mejor comprensión del espacio arquitectónico.

Gracias a los avances tecnológicos que el sector ha experimentado podemos trabajar modelos tridimensionales con mayor número de polígonos y mejor edición de materiales, luces y efectos de imagen. Esto beneficia a la visualización arquitectónica en tiempo real ya que puede lograrse un mayor realismo. Además el avance

en telefonía móvil ha permitido poder tener sistemas *low cost* de Realidad virtual, haciendo más común su uso.

En cuanto al sector de los videojuegos, tras el análisis evolutivo histórico, se puede destacar varios aspectos. Por un lado ha sido una industria que en apenas cuatro décadas ha sufrido una evolución enorme. Partiendo de las primeras consolas domésticas que tenían una capacidad reducida, hasta las consolas de última generación que tienen una capacidad muchísimo mayor y nos ofrecen diferentes formas de interactuar. La industria de los videojuegos ha supuesto un impulso enorme al desarrollo de los gráficos digitales. Este impulso se debe al importante motor económico que supone la industria y a la demanda por parte de los usuarios de tener mejores gráficos realistas en sus videojuegos.

La aparición de los *Games Engines* ha supuesto una alternativa económica y sencilla de crear entornos de Realidad Virtual interactivos. Los motores de videojuegos disponibles trabajan con una alta definición gráfica y además permiten crear diversos tipos de aplicaciones. Por ello, estos softwares se están haciendo un hueco entre profesionales fuera del campo de la programación informática y diseño de videojuegos. Este hecho ha provocado que el arquitecto valore las opciones que ofrecen los softwares que no están específicamente orientados para la profesión, abriendo así un nuevo abanico de posibilidades. Desaparecen, de esta forma, las ideas preconcebidas de que

un programa sólo puede aplicarse al sector para el que fue definido.

En último lugar se prueba un motor de videojuegos como soporte para realizar una visualización arquitectónica interactiva en tiempo real. El modelo escogido es un edificio representativo de la arquitectura del S.XX, el Pabellón de Barcelona del arquitecto alemán Mies Van der Rohe. Los programas utilizados para la realización de la visualización son, por un lado 3D studio Max para el modelado del pabellón y Unreal Engine como motor de videojuego. Se escoge este *Game Engine* ya que dispone de una versión gratuita de descarga y las prestaciones gráficas que ofrece son de muy buena calidad.

En el proceso de elaboración del modelo se han llegado a las siguientes conclusiones. En el modelado para visualización arquitectónica en tiempo real, es recomendable simplificar la geometría para tener menor número de polígonos y conseguir el realismo y detalle mediante el uso de texturas y mapas normales. De esta manera se consigue mayor fluidez de movimientos ya que no se abusa de los recursos del sistema. La mayoría de los programas de modelado en 3D tienen la posibilidad de exportar sus archivos en formato .FBX. Esto es un punto a favor a la hora de usar Unreal Engine, ya que importa correctamente esta extensión.

Para no tener problemas con la visualización del modelo en el motor de renderizado se debe comprobar la dirección del vector normal de los

polígonos que forman la escena. De esta manera evitamos visualizaciones incorrectas producidas por el algoritmo *Back-Face Culling*, que sólo calcula las caras principales de los polígonos. Se podría desactivar este algoritmo pero esto haría que se consumiesen más recursos. Para conseguir una correcta iluminación de las mallas tridimensionales es recomendable realizar un *lightmap*. De esta manera el motor de renderizado interpretará correctamente cómo afecta la luz al objeto.

La edición de materiales es fundamental en la visualización arquitectónica. La posibilidad de crear materiales con reflejo, mapas de relieve, movimiento o de tipo enmascarado es ideal para la visualización arquitectónica. La forma de trabajar los materiales en Unreal es sencilla e intuitiva. Al ser la interfaz similar a otros programas más habituales en la profesión, como el editor Slate material de 3D Studio Max o Grasshopper, resulta rápida la comprensión del funcionamiento. Este programa permite la edición visual de la imagen en tiempo real. Estos efectos son muy útiles a la hora de aportar realismo a la visualización con un relativo bajo consumo de recursos. Además, aportan una mayor calidad estética.

En conclusión y tras haber tratado todos los aspectos anteriores se puede decir que Unreal Engine es una herramienta fácil de aprender. Nos permite realizar visualizaciones en tiempo real de una forma sencilla que hasta la fecha requería conocimientos de programación. Trabajar en tiempo real facilita y agiliza el proceso de edición

del modelo, ya que se ven los cambios realizados al instante, sin necesidad de realizar renderizados para comprobar el resultado como se hace con otros softwares de visualización arquitectónica. Se logran visualizaciones arquitectónicas más completas. El motor de videojuegos permite moverse libremente por el espacio tridimensional, mejorando la comprensión del espacio arquitectónico recreado. Además, es posible añadir movimiento a ciertos objetos, crear sonidos en tres dimensiones que se atenúan con la distancia o interactuar con elementos de la escena, proporcionando un resultado más realista e impactante.

Por otra parte, se han localizado ciertas desventajas que es necesario mencionar. Para el trabajo con escenas con un cierto grado de complejidad en cuanto a materiales y polígonos, un ordenador con una gráfica y procesador corrientes puede no ser suficiente. Es recomendable trabajar con equipos con mayores prestaciones gráficas para visualizar con total fluidez las representaciones arquitectónicas. Esto supone una mayor inversión económica para el desarrollador. Los equipos que permiten aprovechar al máximo esta herramienta (casco de realidad virtual o dispositivos hápticos) son, hoy en día, de un coste elevado.

Añadir, que durante el proceso de realización de este trabajo, se hicieron pruebas para exportar el modelo a dispositivos móviles, con el fin de poder testear el entorno tridimensional en un medio con mayor grado de inmersión. Debido a que es necesario acotar el trabajo y la falta de recursos,

no se pudo finalizar la tarea. En un futuro el autor de este trabajo quiere proseguir con esta tarea y seguir trabajando sobre ello. Se anima a cualquier persona interesada que continúe con esta tarea.

En definitiva, las ventajas que presenta respecto a los sistemas de visualización más usuales y las bondades que se han ido exponiendo a lo largo del trabajo, hace pensar que se convertirá en una herramienta habitual en el campo de la arquitectura en los próximos años.

BIBLIOGRAFÍA

KRULL, F. "The Origin of Computer Graphics within General Motors"

SHUTHERLAND, I. (1963) "Computer Reads Design Sketches"

SHUTHERLAND, I. (1965) "The Ultimate Display"

SHUTHERLAND, I., (1968) "A head-mounted three dimensional display".

KRUEGER, M (1991). *Realidad artificial 2*, Addison-Wesley Profesional.

RHEINGOLD, H. (1992). *La realidad virtual*, Simon & Schuster, New York, NY

HOWELL, I., BATCHELER, B. "Building Information Modeling Two Years Later- Huge Potential, Some Success and Several Limitations"

SHUTHERLAND, I. (2003) *Sketchpad: A man-machine graphical communication system*. University of Cambridge. Computer Laboratory.

DE FREITAS, S.(2013) *Learning in Immersive Worlds: A Review of Game-Based Learning*.

Van Dam, A., D. Foley, F., Feiner, S., Hughes, J., *Computer Graphics: Principles and Practice (2nd edition in C)*. Reino Unido: Addison and Wesley. ISBN: 978-0201848403

PÉREZ, D.C. (2009) *Desarrollo de sistemas de Realidad Virtual y Aumentada para la visualización de entornos acrofóbicos. Estudios comparativos entre ellos*. Trabajo final de Máster. Valencia: Universidad Politécnica de Valencia.

LÓPEZ-TARRUELLA, J. (2013) *Realidad Virtual aplicada a la Arquitectura: Estudio de aplicación de tecnología para el desarrollo de videojuegos en la visualización arquitectónica*. Trabajo final de Máster. Valencia: Universidad Politécnica de Valencia.

BALDASANO, J.M. (1989) *Influencia de la Informática en el Proceso de Proyectar*. V Encuentro Nacional de Proyectos de Ingeniería, UPV. Valencia,

GARCÍA, J.A., *La evolución gráfica de los videojuegos. Desde la aparición de consolas de sobremesa*. Universidad Rey Juan Carlos.

SOLÁ-MORALES, I., CIRICI, C., RAMOS, F., (2002) *Mies Van der Rohe. El Pabellon de Barcelona*, Barcelona: Gustavo Gili. ISBN: 84-252-1601-X

ZIMMERMANN, C.(2014). *Mies Van der Rohe*, Alemania: TASCHEN. ISBN:978-3-8228-2858-8

PUENTE, M. (2000). *100 años Pabellones de exposición*, Barcelona: Gustavo Gili. ISBN:84-252-1791-1

ENLACES WEB

Navarro, E. (2005) "El CAD.", Documenta. <http://catedu.es/documenta>.

Proyecto Whirlwind_ <http://histinf.blogs.upv.es/2011/11/21/proyecto-whirlwind/>

RTVE_ <http://blog.rtve.es/webdocs/2014/12/la-realidad-virtual-como-afectar%20a-los-webdocs-parte-3.html>

Informática gráfica. UJI_ http://repositori.uji.es/xmlui/bitstream/handle/10234/5980/1_Introduccion.pdf?sequence=1

Introducción a la realidad Virtual_ <http://www.dtic.upf.edu/~gvirtual/master/rv/seccio1/seccio1.htm>

Historia de la realidad virtual_ <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/3D/Realidad%20Virtual/web/historia.html>

Panorama de Barker_ <http://tectonicablog.com/?p=42191>

Baixauli arquitectura_ <http://www.baixauli.es/es/el-pabellon-barcelona-de-mies/>

Unreal Engine_ <https://www.unrealengine.com/>

Unity 3D_ <https://unity3d.com/es>

Autodesk_ <http://www.autodesk.es/>

Índice de imágenes

Figura 1.1_ Dibujo realizado por Mies Van der Rohe durante el proceso de diseño del Pabellón de Barcelona
<http://i1.wp.com/architecturalogy.com/wp-content/uploads/2015/02/MIES-VAN-DER-ROHE-BARCELONA-PAVILLION-SKETCH.jpg>

Figura 2.1_ Jay Forrester con los desarrolladores y programadores del Whirlwind.
<https://libraries.mit.edu/news/exhibit-thanks-memory/13994/>

Figura 2.2_ Monitor del Whirlwind
http://design.osu.edu/carlson/history/images/sage_f1693.jpg

Figura 2.3_ Ivan E. Sutherland
<http://blog.rive.es/a/6a014e6089cbd5970c01b8d0a91b12970c-pi>

Figura 2.4_ Ivan E. Sutherland mostrando el SketchPad
https://i.ytimg.com/vi/USyoT_Ha_bA/hqdefault.jpg

Figura 2.5_ Diseñador de General Motors trabajando con el software DAC-1
<http://excelsior.biosci.ohio-state.edu/~carlson/history/tree/images/dac.JPG>

Figura 2.6_ Entorno de trabajo de AutoCAD 2015
Elaboración propia

Figura 2.7_ Entorno de trabajo de AutoCAD Architecture
<http://static-dc.autodesk.net/content/dam/autodesk/www/products/autodesk-autocad-architecture/responsive-center/images/features/2016/wall-dimensioning-large-1152x648.jpg>

Figura 2.8_ Entorno de trabajo de ANSYS
<https://i.ytimg.com/vi/kfFpl2uK6rQ/maxresdefault.jpg>

Figura 2.9_ Entorno de trabajo de 3D studio Max
http://www.evermotion.org/files/tutorials_content/uploads/AI45S07__0036_Layer_1_evermotion.jpg

Figura 2.10_ Render realizado con V-ray
Elaboración propia

Figura 2.11_ Entorno de trabajo de Revit
<http://static-dc.autodesk.net/content/dam/autodesk/www/products/autodesk-revit-It/responsive-center/images/overview/3d-design-visualization-large-1152x648.jpg>

Figura 2.12_ Imagen renderizada antes de realizar la post-producción
Elaboración propia

Figura 2.13_ Imagen renderizada tras realizar la pos-producción
Elaboración propia

Figura 3.1_ Sketch sobre la Realidad Virtual
Elaboración propia

Figura 3.2_ Casco de Corneau y Bryan, llamado Headsight
<https://edu.hstry.co/timeline/la-realite-virtuelle-au-20e-siecle>

Figura 3.3_ Sesorama
<https://edu.hstry.co/timeline/la-realite-virtuelle-au-20e-siecle>

Figura 3.4_ Espada de Damocles, con el sensor de posición en uso
SHUTHERLAND, I., (1968) "A head-mounted three dimensional display". *Figure 3—The mechanical head position sensor in use.*

Figura 3.5_ Virutal Boy
<https://edu.hstry.co/timeline/la-realite-virtuelle-au-20e-siecle>

Figura 3.6_ Oculus Rift
<http://blogs-images.forbes.com/davidewalt/files/2016/03/oculus-rift-vr-headset-1200x698.jpg>

Figura 4.1_ Video juego Battlezone, pionero en sus gráficos vectoriales que simulan a las 3D
<http://www.abadiadigital.com/las-consolas-mas-vendidas-de-la-historia/>

Figura 4.2_ StarFox para Super Nintendo
<http://www.obsolete-tears.com/photos/snin-starwing-jeu1.jpg>

Figura 4.3_ StarFox para Super Nintendo
<http://www.obsolete-tears.com/photos/snin-starwing-jeu1.jpg>

Figura 4.3_ Shenmue, videojuego con altísimas prestaciones gráficas
<http://www.dc4you.ru/download/file.php?id=2060>

Figura 4.4_ Uncharted 4
<http://www.clubic.com/mag/jeux-video/article-801592-1-preview-uncharted-4-drake-preuve-ouverture.html>

Figura 5.1_ Visita del rey de España, Alfonso XIII, al Pabellón. Mayo de 1929
SOLÁ-MORALES, I., CIRICI, C., RAMOS, F., (2002) *Mies Van der Rohe. El Pabellon de Barcelona, Barcelona: Gustavo Gili.* ISBN: 84-252-1601-X

Figura 5.2_ Visita del Rey de España, Alfonso XIII, al Pabellón. Mayo de 1929, en el estanque pequeño, junto a la estatua de G.Kolbe
<https:// analisisdeformas.files.wordpress.com/2015/02/1307376579434.jpg>

Figura 5.3_ Vista exterior con el Palacio de Victoria Eugenia al fondo
SOLÁ-MORALES, I., CIRICI, C., RAMOS, F., (2002) *Mies Van der Rohe. El Pabellon de Barcelona, Barcelona: Gustavo Gili.* ISBN: 84-252-1601-X

El proyecto de arquitectura en Unreal 3D para inmersión virtual.
Caso 1: El Pabellón Mies

Figura 5.4_ Escultura de G.Kolbe en el Pabellón
<http://www.georg-kolbe-museum.de/wp-content/uploads/2014/04/barcelonapavillon.jpg>

Figura 5.5_ Fotografía del Pabellón de Barcelona desde el interior resaltando los materiales
http://images.adsttc.com/media/images/54c6/a11d/e58e/cefd/7100/0004/large_jpg/Mies.jpg?1422303491

Figura 5.6_ Dibujo realizado por Mies durante el proceso de diseño del Pabellón
SOLÁ-MORALES, I., CIRICI, C., RAMOS, F., (2002) *Mies Van der Rohe. El Pabellon de Barcelona, Barcelona: Gustavo Gili.*
ISBN: 84-252-1601-X

Figura 5.7_ Planta principal
SOLÁ-MORALES, I., CIRICI, C., RAMOS, F., (2002) *Mies Van der Rohe. El Pabellon de Barcelona, Barcelona: Gustavo Gili.*
ISBN: 84-252-1601-X

Figura 5.8_ Planta de cubierta
SOLÁ-MORALES, I., CIRICI, C., RAMOS, F., (2002) *Mies Van der Rohe. El Pabellon de Barcelona, Barcelona: Gustavo Gili.*
ISBN: 84-252-1601-X

Figura 5.9_ Sección transversal
SOLÁ-MORALES, I., CIRICI, C., RAMOS, F., (2002) *Mies Van der Rohe. El Pabellon de Barcelona, Barcelona: Gustavo Gili.*
ISBN: 84-252-1601-X

88

Figura 5.10_ Imagen desde la plaza exterior hacia el interior del Pabellón
<http://miesbcn.com/es/el-pabellon/>

Figura 5.11_ Fotografía interior con estanque pequeño al fondo
<http://miesbcn.com/es/el-pabellon/>

Figura 5.12_ Fotografía exterior con el sillón Barcelona
<http://miesbcn.com/es/el-pabellon/>

Figura 5.13_ Ónice del pabellón con taburetes Barcelona
<http://miesbcn.com/es/el-pabellon/>

Figura 5.14_ Fotografía interior del pabellón
<http://miesbcn.com/es/el-pabellon/>

Figura 5.15_ Fotografía exterior con el pabellón de servicios al fondo
<http://miesbcn.com/es/el-pabellon/>

- A partir de la figura 5.16 en adelante todas son de elaboración propia -

89