
Modelado 3D de sitios históricos

Fortaleza Protectora Argentina

Montagna, Augusto (LU: 77586)

Pachiana, Gabriel (LU: 78703)

Directora: Dra. Castro, Silvia

Co-Director: Dr. Larrea, Martin

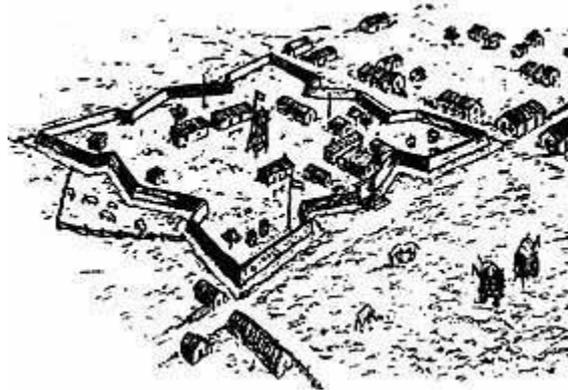
UNIVERSIDAD NACIONAL DEL SUR

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN



Proyecto Final

Modelado 3D de sitios históricos



Fortaleza Protectora Argentina

Montagna, Augusto (LU: 77586)

Pachiana, Gabriel (LU: 78703)

Índice

1. Introducción	6
2. Antecedentes	11
2.1 CAD (Computer-Aided Design)	13
2.2 Modelado basado en imágenes	15
2.3 Modelado basado en distancia (range-based modeling)	18
2.4 Renderización	22
2.5 Interactividad	23
2.5.1 Realidad Virtual	23
2.5.1.1 - Realidad Virtual Inmersiva	23
2.5.2 Realidad Aumentada	24
2.6 Antecedentes en CAD	26
2.6.1 Coliseo Romano	26
2.6.2 Foro de Augusto	29
2.6.3 The Palace	31
2.6.4 Tjuvholmen F1	32
2.6.5 Lone House	33
2.6.6 Castillo Medieval	34
2.6.7 Casa histórica de la Independencia Argentina	35
2.6.8 Iglesia De Los Capuchinos - Córdoba Argentina	36
2.6.9 Congreso de la Nación Argentina	38
2.6.10 Fortaleza Protectora Argentina	41
2.7 Herramientas de Modelado y Renderizado	43
2.7.1 Google SketchUp	43
2.7.2 Autodesk 3ds Max	44
2.7.3 Blender	45
2.7.4 Unity 3	47
2.7.5 Autodesk AutoCAD 2012	49
2.7.6 Autodesk Maya	51
2.7.7 Cinema 4D	52
2.7.8 Rhinoceros 3D	54
2.7.9 TrueSpace	55
3. Propuesta	57

3.1 Descripción de la Propuesta	60
3.2 Modelado	61
3.3 Rendering	63
3.4 Interactividad	64
3.5 Licenciamiento	64
3.6 Recopilación de Datos	65
4. Implementación	67
4.1 Modelado de la Fortaleza Protectora Argentina	69
4.1.1 Estructura	70
4.1.2 Moldeado	72
4.1.3 Suavizado y textura del terreno	78
4.2 Implementación de las escenas	81
4.2.1 Implementación del recorrido predefinido	81
4.2.2 Implementación del recorrido libre	83
4.2.3 Implementación del menú inicio	85
Conclusiones	88
Trabajo a futuro	90
Agradecimientos.....	92
Bibliografía	93
Anexo I - Manual de instalación	97
Anexo II - Manual de usuario	102

Capítulo 1

Introducción

1. Introducción

Los sitios históricos son aquellos que marcaron, y todavía marcan, una diferencia en la historia de la humanidad, ya sea por hechos que acontecieron en ellos o simplemente por su magnificencia arquitectónica. Muchos de estos sitios ya no existen o no pueden ser accedidos debido a que se pondría en riesgo la arquitectura o la vida de los visitantes.

Un ejemplo de esto es el cierre de la tumba de Tutankamón ([1]). En Enero de este año se anunció que se cerraran las puertas de la tumba del antiguo faraón, junto con otras dos, para poder así preservarlas: "Estas tumbas se destruirán totalmente dentro de 200 años debido a la respiración de los turistas que las visitan,...., la única manera de proteger estas antigüedades es cerrarlas y hacer réplicas de ellas que podrán visitar los turistas", señaló en una entrevista el secretario general del Consejo Supremo de Antigüedades, Zahi Hawas.

La desaparición de los sitios históricos (ya sea catástrofes naturales, destrucción por parte de los humanos o simplemente por paso del tiempo), la fragilidad de algunos de ellos, o solamente debido a la distancia que nos separa de ellos nos arrebató la posibilidad de visitar, recorrer y conocer estos lugares tan importantes.

La aparición de nuevas tecnologías nos permite repensar la forma como visitamos y conocemos un lugar o sitio. Las nuevas tecnologías pueden ser utilizadas para reconstruir

virtualmente los sitios históricos, permitiendo no sólo sólo tener una imagen de cómo era en realidad, sino también poder recorrerlos y así entender mejor los sucesos que en ellos ocurrieron. Esto permite crear un panorama realista sobre estos lugares que, en gran cantidad, están parcial o completamente destruidos.

Un ejemplo de sitio histórico que ya ha desaparecido es el de la Fortaleza Protectora Argentina. La Fortaleza Protectora Argentina (ó Fuerte Argentino), fundada en 1828 por el coronel Estomba, es el origen de la ciudad de Bahía Blanca. En diciembre de 1823 el gobierno de Buenos Aires envió a José Valentín García a la bahía Blanca en busca del lugar adecuado para un puerto. En 1824 el gobernador Martín Rodríguez realizó una campaña militar contra los indígenas para llevar a cabo la fundación, pero la consideró impracticable.

Posteriormente, el proyecto fundacional se vio impulsado por Bernardino Rivadavia durante la Guerra del Brasil, cuando la flota brasileña intentó ocupar Carmen de Patagones en marzo de 1827. En Diciembre de ese mismo año, el gobernador coronel Manuel Dorrego dispuso una nueva línea de fronteras en el sur y ordenó al comandante general Juan Manuel de Rosas dirigirse al Fuerte Independencia (actual Tandil) y de allí a la bahía Blanca para fundar un fuerte. Pero la orden se postergó brevemente por el pedido de Rosas sobre el envío de técnicos. El 9 de abril de 1828 llegó Estomba y se procedió a levantar un acta de fundación. La construcción del fuerte demandó cuatro meses. ([2], [3], [4], [5], [6])

De esta fortaleza se cuenta únicamente con planos, relatos y dibujos. Sin embargo, no existen imágenes certeras, como fotografías, que nos permitan ver cómo era en realidad. Una representación virtual de esta fortaleza, obtenida a partir de los documentos existentes, nos permitiría tanto tener una imagen clara de cómo era, como también usar esta reconstrucción en la enseñanza y educación sobre la historia de la ciudad.

Este trabajo tiene como objetivo construir un modelo 3D, virtual, de la Fortaleza Protectora Argentina, a partir de la bibliografía disponible en la actualidad. Es también un

objetivo de este trabajo el diseño y el desarrollo de un software que permita utilizar el modelo creado con fines educativos y turísticos. Dicho software servirá como visor del modelo virtual y será utilizado en la Municipalidad de Bahía Blanca, junto con el modelo 3D.

Este informe consta de 6 capítulos. En este primer capítulo se realiza una breve introducción en la temática de sitios históricos virtuales. En el siguiente capítulo se muestran y comparan los antecedentes a esta propuesta. Se realiza una introducción a la reconstrucción virtual en general y su aplicación en sitios históricos. A continuación se discuten ejemplos de sitios históricos reconstruidos virtualmente, sus implementaciones y las herramientas utilizadas para tal fin. En el capítulo 3 se presenta nuestra propuesta de trabajo, detallando tanto el alcance como la funcionalidad que tendrá el modelo y el software que se utilizará. También se discuten en este capítulo las posibles herramientas y el software a utilizar para alcanzar la propuesta realizada. El capítulo 4 contiene los detalles de implementación de la propuesta, haciendo mención a los obstáculos presentados durante la construcción del modelo, así como también el trabajo de relevamiento realizado para recopilar la información asociada a la Fortaleza Protectora Argentina. Finalmente se detallan las conclusiones y luego se describen las posibles extensiones que se podrán realizar a partir de este proyecto.

Capítulo 2

Antecedentes

Antecedentes

El uso de digitalización y modelado de los sitios de patrimonio histórico ha aumentado significativamente en los últimos años. Esto ocurre principalmente debido a los avances en las técnicas de escaneo láser, software de modelado 3D, técnicas de imagen basadas en el modelo y la potencia actual de las computadoras. Hay muchos métodos disponibles en la actualidad. Los métodos más usados siguen siendo las herramientas CAD y / o fotogrametría tradicional con puntos de control y un operador humano.

A continuación se detallan algunos métodos para el modelado 3D.

2.1- CAD (Computer-Aided Design)

Como su nombre lo indica, este tipo de modelado se basa en diseñar mediante la ayuda de una computadora. Puede ser desarrollado desde un boceto, o desde figuras primitivas. Algunos estudios de datos, o mediciones de dibujos y mapas pueden ser requeridos. Es uno de los métodos más lentos, pero a su vez uno de los más baratos. Este método es el único que puede ser utilizado para reconstruir estructuras inexistentes.

Para crear un modelo 3D, en sus comienzos, se necesita un programa de modelado, denominado "modelador". La mayoría de los modeladores comerciales le dan al usuario el control total de la creación del modelo.

Otra posibilidad es proveer un modelador general con la habilidad de permitir al usuario especificar una secuencia de operaciones de modelado específicas para el caso de objetos o sitios históricos, que pueden ser aplicadas automáticamente al modelo. Si se pudiese definir una secuencia de operaciones creadas interactivamente para modelar un objeto, podrían obtenerse varios tipos de objetos con un pequeño esfuerzo. Sin embargo, la definición de secuencias de operaciones de modelado parametrizadas no es una tarea trivial, y sólo podrá ser dominada por usuarios entrenados.

Un método utilizado para modelar objetos o partes de sitios históricos es *Combined BRep*. Éste permite, de manera versátil, modelar objetos o partes de sitios históricos. Este método está basado en la representación de sólidos por medio de su frontera (BRep, boundary representation [7]).

Así, se podrán modelar objetos complejos de la arquitectura histórica. Un ejemplo de modelado lo constituyen las ventanas de las construcciones clásicas y neo-clásicas de la ciudad de Braunschweig, Alemania de los siglos XVIII y XIX. Ver figura 2.1.1 y figura 2.1.2.

La reconstrucción 3D de sitios históricos es un proceso tedioso, que aún hoy involucra mucho modelado 3D a mano. Con "Combined BRep" este proceso podría acelerarse considerablemente.



Figura 2.1.1. Ejemplo de reconstrucción de una cornisa de la ciudad de Braunschweig

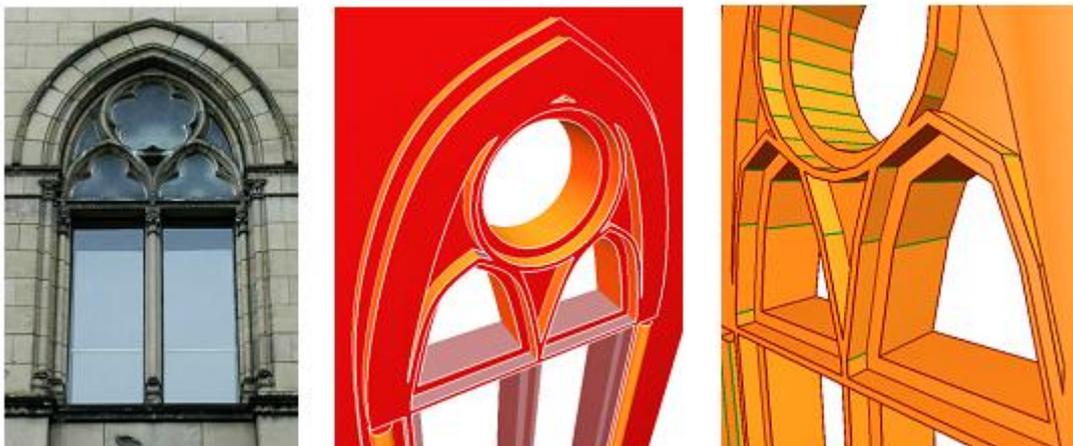


Figura 2.1.2. Ejemplo de reconstrucción de ventanal de la ciudad de Braunschweig

2.2- Modelado basado en imágenes

Los métodos de modelado basados en imágenes se apoyan en un conjunto de imágenes en dos dimensiones de una escena para generar un modelo en tres dimensiones. Implican hardware ampliamente disponible (cámara de fotos, videos) y el mismo sistema se puede utilizar para una amplia gama de modelos de objetos de arquitectura (figura 2.2.1) ([8]).



Figura 2.2.1. Uso de imágenes para la representación 3D.

Por otro lado, la medición 3D de imágenes requiere de puntos de interés o tener ciertas características visibles. Es decir, se requiere una buena imagen, que muestre bien el objeto a modelar. Esto no siempre es posible, ya sea porque una región está oculta o se obstruye, o porque no hay ninguna marca, borde, o una característica visual para extraer ([9]).

En ([10]) se proponen 2 métodos para la extracción de puntos característicos: uno interactivo y uno completamente automático.

El modo automático consta de dos etapas: la primera consiste en la técnica de detección de puntos de interés descrita por Harris and Stephens (Harris and Stephens, 1988). La segunda es un procedimiento automático de matching propuesto por (Brown et al., 2003).

En el modo interactivo, un operador humano agrega puntos correspondientes a dos imágenes, mientras que el sistema calcula los puntos de modelado 3D y proyecta la característica extraída en todas las imágenes de la secuencia de imágenes.

Este método se aplicó, por ejemplo, en la construcción del modelo virtual de Josefsplatz, en Vienna. Con cámaras de alta definición se obtuvieron varias imágenes (ver figura 2.2.2), y luego por medio de algoritmos se fueron obteniendo los puntos y las líneas (Figura 2.2.3 y 2.2.4). El resultado final se puede observar en la figura 2.2.5.

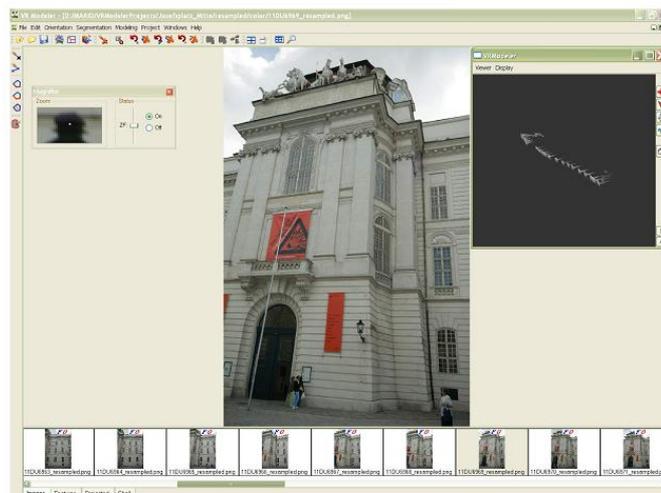


Figura 2.2.2 – Distintas imágenes tomadas al Josefsplatz

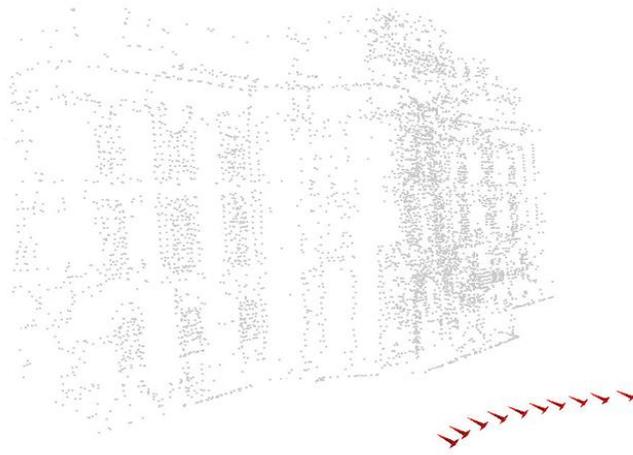


Figura 2.2.3. Puntos obtenidos desde las imágenes

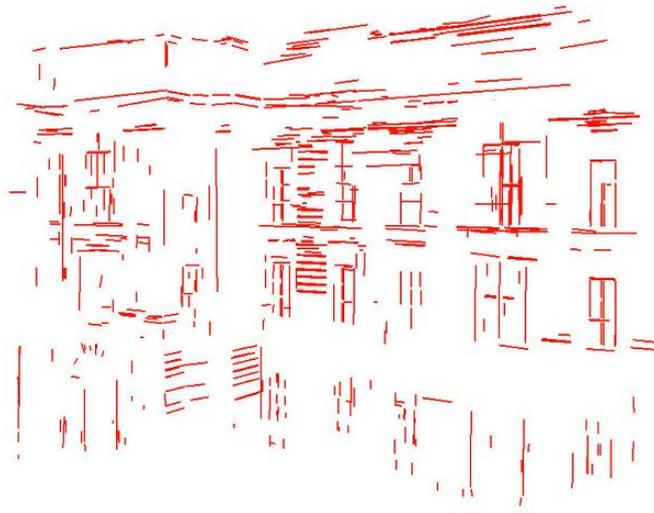


Figura 2.2.4 – Líneas obtenidas desde las imágenes



Figura 2.2.5. Modelo final

2.3 - Modelado basado en distancia (range-based modeling)

Como se mencionó anteriormente, el modelado basado en imágenes requiere de puntos de interés o características visibles y se ve afectado por problemas de iluminación. Los sensores evitan estas limitaciones mediante la creación de características en la superficie de proyección de luz controlada ([9]).

Hay dos tipos principales de sensores de rango. La primera está basada en triangulación, proyectando rayos en una dirección conocida desde una posición conocida, y se medía la dirección del rayo que retorna. La exactitud de las mediciones depende de la base del triángulo en relación a su altura. El segundo tipo se basa en el principio tiempo de vuelo. Es decir, se mide el retardo entre la emisión y la detección de la luz reflejada, y por lo tanto la exactitud no se deteriora rápidamente al aumentar la intensidad del rayo. Esto permite realizar mediciones en rango de kilómetros.

Es común que, al momento de modelar un sitio histórico, no se aplique una sola técnica sino una combinación de ellas. Esto se puede ver en la reconstrucción del atlante del templo de Zeus en Agrigento ([11]). Aquí se puede observar cómo se utilizan tanto los datos adquiridos por láser (figura 2.3.1) como los obtenidos por fotografía (ver figura 2.3.2) para reconstruir el atlante. Primero se crea una nube de puntos mediante láser (ver figura 2.3.3) y luego se interpola con los resultados obtenidos mediante las imágenes.

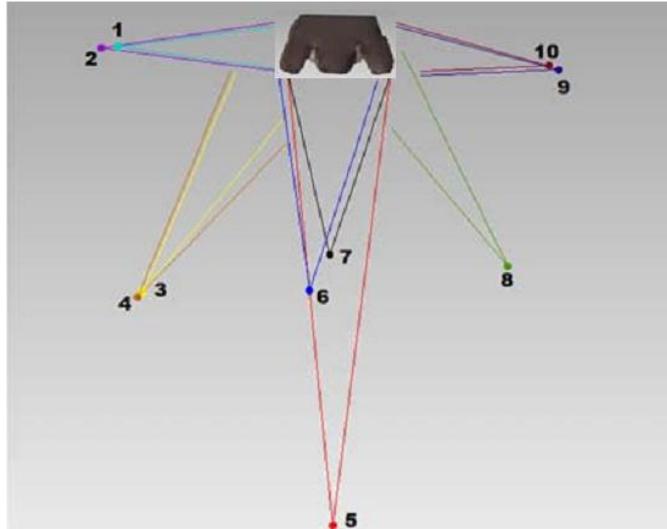


Figura 2.3.1. Posiciones del láser en los distintos escaneos

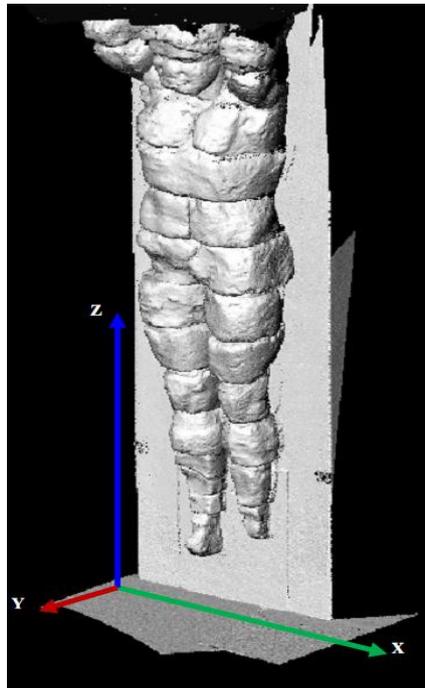


Figura 2.3.2. Nube de puntos generada en una escaneada del láser



Figura 2.3.3. Tomado de imágenes del atlante

Otro ejemplo es la reconstrucción a través del escáner láser de la plaza de La Habana Vieja, Cuba [12]. Se utilizó un escáner laser RIEGL Z420i combinado con una cámara réflex digital calibrada Nikon D100 para el levantamiento. La captura de los datos se realizó en 6 días, para lo que se pudieron tomar datos desde 76 posiciones. (Ver figura 2.3.4). Luego, con un programa para la extracción de puntos 2d y 3d se puede generar una imagen con color de la nube de puntos obtenida. (ver figura 2.3.5)



Figura 2.3.4. Nube de puntos con datos no deseados resaltados



Figura 2.3.5. Nube de puntos con color

2.4 – Renderización

Renderizado (*render* en inglés) se refiere al proceso de generar una imagen desde un modelo. La renderización es un proceso de cálculo destinado a generar una imagen 2D a partir de una escena 3D. Las técnicas de renderización existentes son:

- *Scanline rendering and rasterisation*: consiste en un escaneo línea por línea y cálculo del color del pixel que tendrá la imagen final.
- *Ray Casting*: permite indicar que superficies están visibles calculando la intersección del rayo de vista con los objetos y luego calcular el color del pixel.
- *Ray Tracing*: es similar al *Ray Casting*, sólo que *Ray Tracing* calcula el color dependiendo también de los rayos de luz que inciden de forma secundaria en el objeto (reflejos).

2.5- Interactividad

Una vez creados los modelos es necesario proveer una forma de interactuar con ellos, ya sea desde recorrerlos hasta poder modificarlos durante la interacción entre el usuario y el modelo. Nombraremos 2 grandes disciplinas relacionadas con el tema.

2.5.1– Realidad virtual

El termino Realidad Virtual (VR) es utilizado por muchas personas con distintos significados. Para algunas personas VR es una colección específica de las tecnologías, como un casco virtual, un guante de dispositivos de entrada y de audio. Otras personas estiran el plazo para incluir a los libros convencionales, películas o pura fantasía y la imaginación. Nosotros tomaremos VR como una concepción virtual computarizada de la realidad.

El concepto mismo de virtualización implica tanto visualización en 3 dimensiones, como sistemas interactivos con el propósito de crear ambientes generados por computadoras en tiempo real [13]. Si bien existen varios tipos de VR, hay 2 que engloban a la mayoría:

2.5.1.1. – Realidad virtual Inmersiva

Los métodos inmersivos de realidad virtual con frecuencia se ligan a un ambiente tridimensional creado por un ordenador, el cual se manipula a través de cascos, guantes u otros dispositivos que capturan la posición y rotación de diferentes partes del cuerpo humano

(ver figura 2.5.1.1.1). Este hardware es costoso, y ésta es una de las principales razones por la cual la VR inmersiva no está tan insertada en la sociedad.

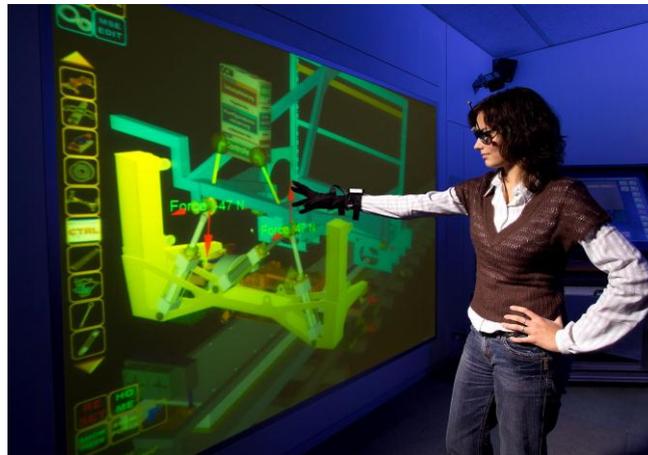


Figura 2.5.1.1.1. Realidad Virtual Inmersiva

La realidad virtual no inmersiva es aquella que se crea cuando el participante explora diversos ambientes haciendo uso de los dispositivos de hardware comunes: Mouse, monitor, tarjeta de sonido, etc. Sin duda este tipo de VR es la más utilizada, debido al bajo costo y alta disponibilidad del hardware requerido.

2.5.2– Realidad Aumentada

La realidad aumentada (RA) es el término que se usa para definir una visión directa o indirecta de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real. Es decir, consiste en un conjunto de dispositivos que añaden información virtual a la información física ya existente.

Hay distintas formas de integrar información virtual y real. Sólo se citaran 2 opciones generales. Un método puede ser HMD (*Head-Mounted Display*), que es básicamente un casco que permite mostrar tanto imágenes reales como virtuales. El otro método es conocido como

Realidad Aumentada Espacial (SAR) y hace uso de proyectores digitales para mostrar la información virtual sobre objetos físicos reales.

2.6 – Antecedentes en CAD

Se continuará con una descripción general de la problemática del modelado 3d de sitios históricos y luego describirá diferentes casos reales en donde se utilizó CAD como modelado.

Entre la gran cantidad de modelos disponibles, se han seleccionado aquellos que permiten ilustrar las herramientas que luego serán utilizadas en el proyecto, o aquellos que son de nuestro interés por ser reconstrucciones históricas de sitios pertenecientes a la República Argentina. Primero se mostraran modelos que ilustran algunas herramientas y luego modelos de reconstrucciones de patrimonios argentinos.

Para cada caso real se analizarán los detalles y la complejidad de los modelos, las facilidades que proveen para su recorrido. En la siguiente sección se detallarán las herramientas utilizadas para la creación de los modelos.

2.6.1 - Coliseo Romano

Reconstrucción del Coliseo, en el centro de Roma. Se trata del anfiteatro más grande jamás construido en el Imperio Romano. Su construcción se inició entre el 70 y el 72 dC bajo el emperador Vespasiano y fue terminado en el año 80 dC al mando de Tito, con modificaciones posteriores que se hicieron durante el reinado de Domiciano (81-96). Capaz de acoger a 50.000 espectadores, el Coliseo fue utilizado para combates de gladiadores y espectáculos públicos, tales como simulacros de batallas navales, caza de animales, ejecuciones, recreaciones de famosas batallas, y obras de teatro basadas en la mitología clásica. Se trata de un modelo

detallado más alto. El interior se reproduce en su totalidad. Se han hecho todas las galerías internas, pasillos, escaleras y pasillos [21].

En la figura 2.6.1.1 podemos observar cómo se encuentra el Coliseo Romano actualmente.



Figura 2.6.1.1. Foto del Coliseo Romano en la actualidad

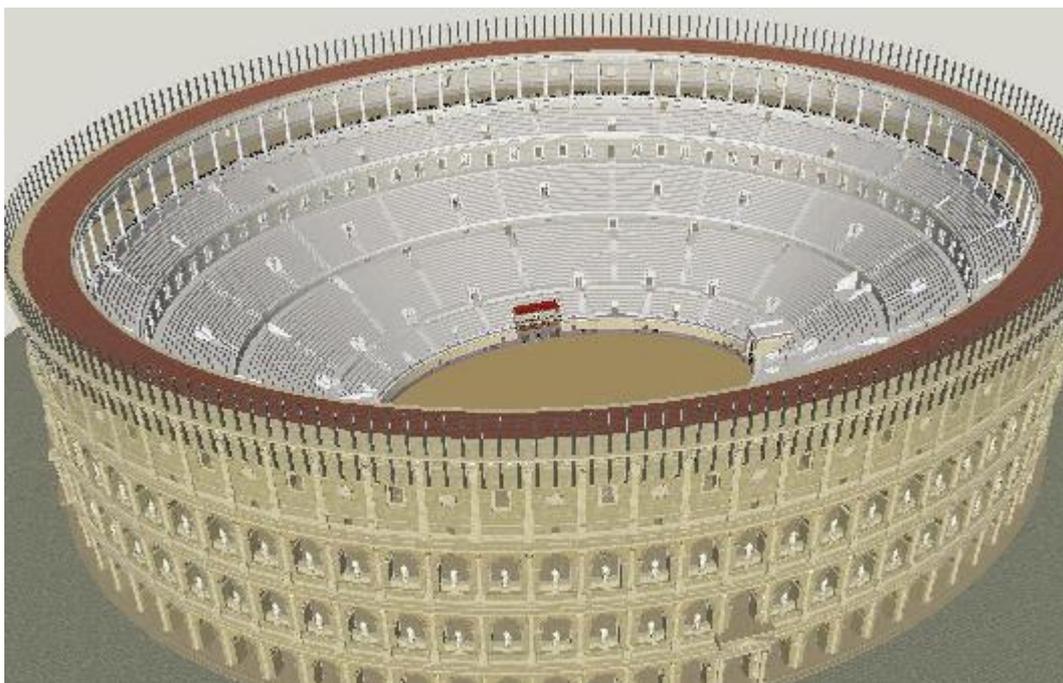


Figura 2.6.1.2. Modelo 3D del coliseo (reconstrucción)

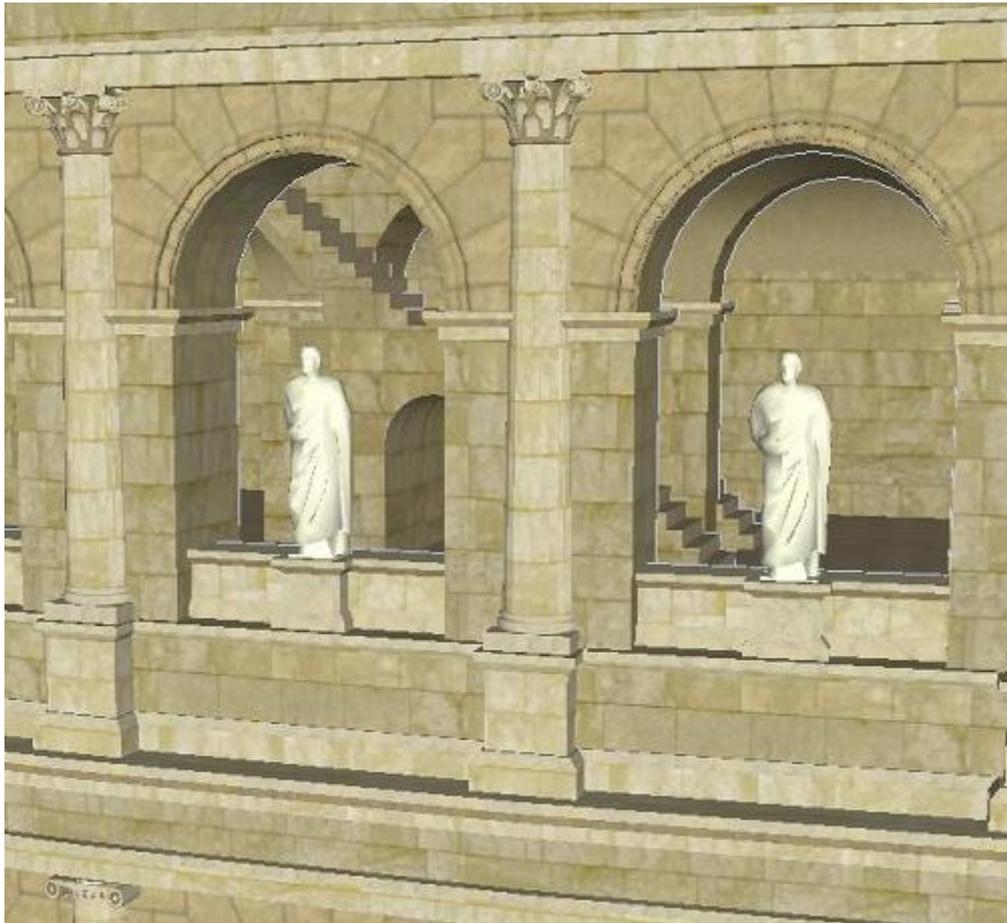


Figura 2.6.1.3. Vista frontal del Coliseo Romano

Para este modelo se utilizó la herramienta de diseño *Google SketchUp*. Como puede observarse en las figuras 2.6.1.2 y 2.6.1.3, la estructura tiene buenos detalles estructurales (ej. Estatuas, palco), pero no existen buenos detalles de sombras, iluminación, y ambiente en general.

En cuanto a la posibilidad de recorrer el modelo, éste debe hacerse mediante la herramienta *SketchUp*, por lo que se necesita que el usuario conozca la herramienta. Al recorrerla, se puede ingresar a la estructura y observar las estatuas, pasillos, escaleras, palco, la arena, etc.

2.4.2 - Foro de Augusto

El Foro de Augusto es uno de los foros imperiales de Roma, construido por el emperador Augusto. Incluye el templo de Mars Ultor («Marte el Vengador»). La construcción del foro comenzó alrededor del año 20 a. C., este se erigió tanto para albergar un templo en honor a Marte como para proporcionar espacio para procedimientos legales, puesto que el Foro Romano estaba ya saturado. Antes de la batalla, los generales militares salían del templo de Marte, después de celebrar una ceremonia.

En la figura 2.4.2.1 se puede apreciar el estado actual del foro de Augusto.



Figura 2.6.2.1. Foto del Foro de Augusto en la actualidad

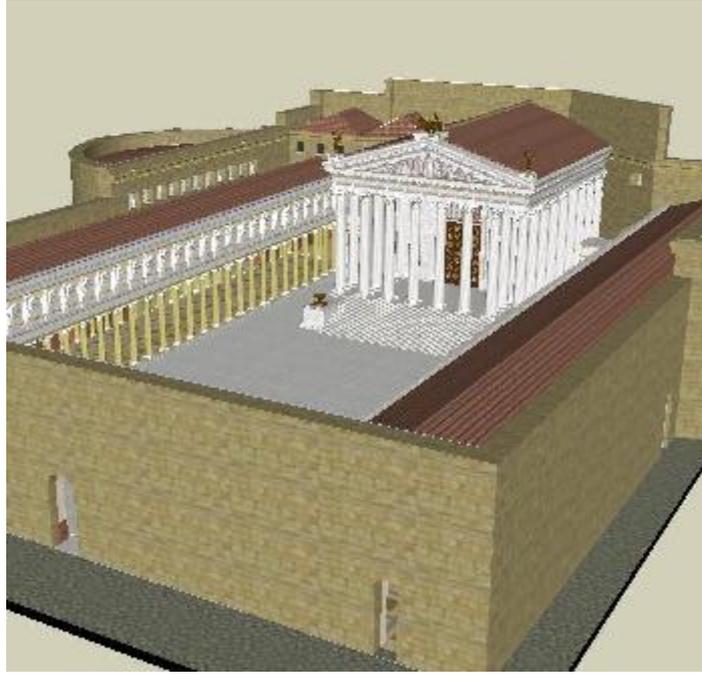


Figura 2.6.2.2. Modelo 3D del coliseo (reconstrucción)



Figura 2.6.2.3. Vista frontal superior de la entrada del Foro

Ésta es una buena reconstrucción del foro del emperador Augusto en Roma realizado con la herramienta *Google SketchUp*. Las figuras 2.6.2.2 y 2.6.2.3 muestran un buen nivel de detalle ya que es posible apreciar todo el interior, incluso se puede ver el templo de

Marte. Pero, al igual que la reconstrucción del coliseo romano, no tiene buenos detalles en el ambiente (iluminación, sombreado, texturas, etc.).

En cuanto a la posibilidad de recorrer el modelo, este debe hacerse mediante la herramienta *SketchUp*, por lo que se necesita que el usuario conozca la herramienta. Al recorrerla, se puede ingresar a la estructura y observar los detalles [22].

2.6.3 - The Palace

Este es un diseño realizado por una consultora de arquitectura para un cliente específico, no es una reconstrucción histórica, sólo un modelo 3D de un edificio que no existe físicamente. Se utilizó la herramienta Autodesk 3DS MAX para la construcción del modelo, y el software Vray1.5 para la renderización.



Figura 2.6.6.1. Renderización del modelo 3D realizado en 3DS Max.

Como puede observarse (figura 2.6.6.1), el modelo tiene un gran nivel de detalle, tanto en la construcción del mismo como en los detalles ambientales. Han logrado una excelente vista del modelo, gracias a la iluminación, sombreado, y a aquellos objetos que embellecen la escena (ej. autos, plantas, arboles, etc.)

El modelo no está disponible, por ende, no se puede acceder al mismo. Sólo se puede ver la foto provista por el autor [23].

2.6.4 - Tjuvholmen F1

Este edificio pertenece al proyecto Tjuvholmen de la ciudad Noruega de Oslo. Al igual que el modelo anterior, ésta no es una reconstrucción histórica, sólo un modelo 3D de un edificio que no existe físicamente. Se utilizó la herramienta Autodesk 3DS Max [24].



Figura 2.6.7.1. Renderización del modelo 3D realizado en 3DS Max.

El modelo tiene un gran nivel de detalle. La imagen lograda es de muy buena calidad, puede observarse una buena iluminación, sombreado y efectos de reflexión (figura 2.6.7.1).

El modelo no está disponible, por ende, no se puede acceder al mismo. Sólo se pueden ver las fotos provistas por el autor.

2.6.8 - Lone House

Proyecto de paisaje campestre con edificaciones realizado con la herramienta Blender [25].



Figura 2.6.8.1. Renderización del modelo 3D realizado en Blender

Como puede observarse (figura 2.6.8.1), el modelo tiene un excelente nivel de detalle en cuanto al ambiente: iluminación, sombreado, efecto de atardecer, y los elementos que rodean la escena.

El modelo no está disponible, por ende, no se puede acceder al mismo. Sólo se pueden ver las fotos provistas por el autor.

2.6.6 - *Castillo Medieval*

Proyecto de edificación medieval de un castillo hipotético [26].



Figura 2.6.9.1. Renderización del modelo 3D realizado en Blender

La figura 2.6.9.1 muestra, al igual que el modelo anterior, un excelente nivel de detalle, iluminación, sombreado, niebla, etc.

El modelo no está disponible, por ende, no se puede acceder al mismo. Sólo se pueden ver las fotos provistas por el autor.

2.6.7 - Casa Histórica de la Independencia Argentina

También conocida como Casa de Tucumán, es un edificio histórico localizado en la ciudad argentina de San Miguel de Tucumán, donde un cuerpo de delegados de la mayoría de las Provincias Unidas del Río de la Plata, conocido como el Congreso de Tucumán, proclamó la declaración de independencia de la Argentina, el 9 de julio de 1816. De estilo colonial típico, y caracterizada por tener molduras que representan a las columnas salomónicas a ambos costados de la puerta principal, pertenecía originalmente a Doña Francisca Bazán de Laguna, quien permitió usarla para el Congreso, luego la recuperó. En 1941-43 fue reconstruida basada en documentos antiguos y declarada Monumento Histórico Nacional. Hoy la casa es un museo, custodiado por el Batallón de Infantería Nro. 10 del ejército nacional, que cuenta con 9 salas que exhiben elementos de la época colonial, un laboratorio de restauración, y 3 patios con árboles de la región; el último además cuenta con un sector de placas de homenaje y al final de éste, un mástil con la bandera nacional frente a 2 altorrelieves de Lola Mora.

La figura 2.6.3.1 es una fotografía que muestra la fachada de la casa de Tucumán.



Figura 2.6.3.1. Foto de la casa de Tucumán en la actualidad



Figura 2.6.3.2. Modelo 3D de la casa de Tucumán

Este modelo fue realizado con la herramienta *Google SketchUp*. El modelo no está disponible para el público (la figura 2.6.3.2 es una muestra provista por el autor) por lo que no se puede observar los detalles del mismo, y por ende, tampoco se puede analizar el modelo [27].

2.6.8 - Iglesia De Los Capuchinos - Córdoba, Argentina

La Iglesia del Sagrado Corazón está ubicada en la ciudad de Córdoba, Argentina en la intersección de las calles Buenos Aires y Obispo Oro. Es conocida como la iglesia de los padres capuchinos (brazo de la orden Franciscana que se distinguían por el uso de “IlCapucchio”). Es un fiel exponente de la arquitectura neogótica, diseñada y realizada por el célebre artista e ingeniero italiano Augusto Ferrari, quien trae desde Roma al personal especializado para semejante estructura. Su construcción fue iniciada en 1926 y terminada en el año 1934, en la cual llama la atención la torre trunca, cuya explicación aduce a la materia que perece, mientras

la otra, erguida representa el alma que asciende. Cada uno de los elementos que integran la fachada principal tienen un sentido o significación para el artista destacando un sin número de columnas de distintos estilos a la altura de las naves y que representan todas las culturas antes del advenimiento de Jesucristo. Su rico interior impacta por el cielo representado en su bóveda, que genera una idea de absoluta inmensidad y se encuentran pinturas que nos hablan de la vida de San Francisco de Asís, patrono de la orden. Posee importantes imágenes y un bellissimo pesebre en cartapesta restaurado en el 2006 por la artista plástica Susana Araoz.

La figura 2.6.4.1 muestra la Iglesia de los Capuchinos en la actualidad.



Figura 2.6.4.1. Foto de la Iglesia de los Capuchinos

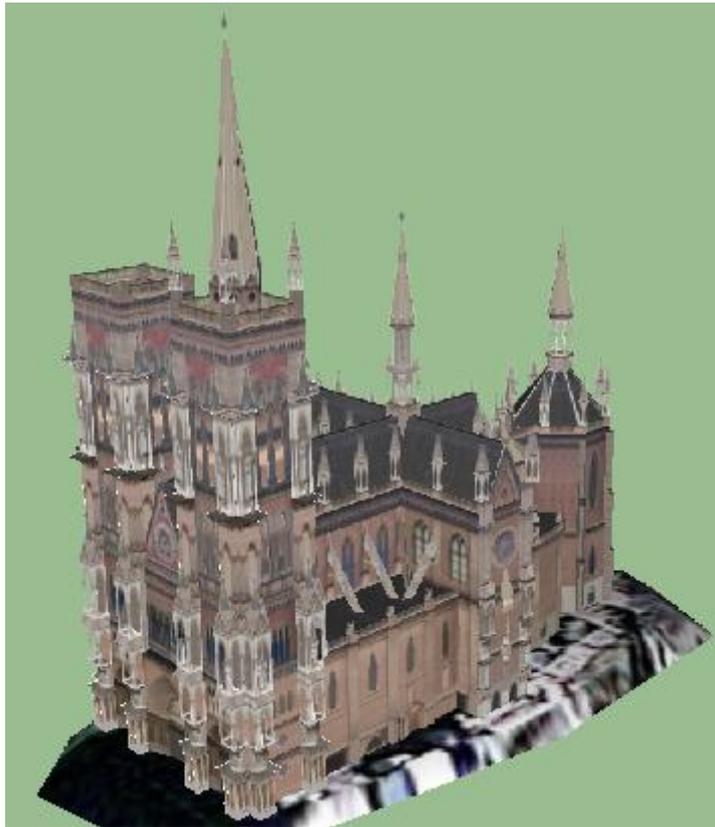


Figura 2.6.4.2. Modelo 3D de la Iglesia de los Capuchinos

Este modelo fue realizado con la herramienta Google SketchUp. Como se puede observar en la figura 2.6.4.2, el modelo es una buena representación estructural, pero sólo está modelada la parte externa de la iglesia, es decir, el “casarón”. No posee detalles ambientales tales como iluminación, sombreado, etc. El modelo puede recorrerse a través de la herramienta SketchUp.

2.6.9 - Congreso de la Nación Argentina

El Palacio del Congreso de la Nación argentina es el edificio donde desarrolla sus actividades el Congreso de la Nación Argentina. Se encuentra ubicado en la Ciudad de Buenos

Aires, en la manzana limitada por la Avenida Rivadavia, Combate de Los Pozos, Avenida Entre Ríos e Hipólito Yrigoyen, con una superficie de manzana de 12.079,60 m².

El estilo de la obra es un estilo grecorromano, dentro del academicismo italiano de Víctor Meano. El edificio se distingue por su cúpula, que alcanza los 80 metros de altura.

La figura 2.6.5.1 es una fotografía de la fachada del Congreso de la Nación Argentina.



Figura 2.6.5.1. Foto del Congreso de la Nación Argentina



Figura 2.6.5.2. Modelo 3D del Congreso de la Nación Argentina



Figura 2.6.5.3. Vista superior del modelo 3D del Congreso de la Nación Argentina

Al igual que el modelo anterior, éste está construido con la herramienta *SketchUp*; sólo está modelada la parte exterior del edificio y no posee detalles ambientales (tal como se puede observar en las figuras 2.6.5.2 y 2.6.5.3). El recorrido puede hacerse por medio de la herramienta. [29]

2.6.10 - Fortaleza Protectora Argentina – Maqueta

Otra forma de obtener una representación 3D de un sitio histórico, es la reconstrucción física a escala del sitio, es decir, maquetas. Si bien ésta no es una representación mediante CAD afecta directamente a nuestro proyecto. En la sala histórica del Batallón de V Cuerpo de Ejército de Bahía Blanca se encuentra una maqueta de la Fortaleza Protectora Argentina (figura 2.6.10.1, 2.6.10.2) realizada por un oficial del Ejército Argentino.

La maqueta es una muy buena representación de la Fortaleza Protectora Argentina. Está sobre una base de 1,80mts por 1,80mts recubierta por un vidrio protector, dentro de la misma pueden apreciarse los detalles representados, tales como cañones, soldados, animales, casas, polvorines, el puente levadizo, la vegetación, etc.

La maqueta se encuentra en una sala histórica, lo que quiere decir que no tiene acceso libre al público, a diferencia de un museo que si lo tiene, pero cabe destacar que la maqueta esta en buenas condiciones y bien preservada. La visita a la sala histórica, donde se encuentra la maqueta, se debe coordinar previamente y se hará bajo supervisión del personal de Ejército, con una visita guiada a cargo de personal idóneo en la historia de la Fortaleza Protectora Argentina.



Figura 2.6.10.1. Fotografía de la maqueta de la Fortaleza Protectora Argentina



Figura 2.6.10.2. Fotografía de la maqueta de la Fortaleza Protectora Argentina

2.7 – Herramientas de Modelado y

Renderizado

Una herramienta de modelado 3D permite al usuario crear modelos 3D, estos brindan facilidades para la edición interactiva de los mismos. Algunos proveen, además del modelado, la posibilidad de renderizar el modelo y crear animaciones.

Las herramientas de modelado proveen la facilidad de agregar y editar objetos predefinidos que constituyen a las formas básicas utilizadas en el proceso de modelado, tales como cubos, esferas, líneas, conos, toroides, etc. Además permiten crear materiales y texturas que se podrán aplicar a dichos objetos.

A continuación se detallan algunas de las herramientas de modelado 3D sugeridas por profesionales en el ámbito de la arquitectura, ingeniería civil y computación gráfica.

2.7.1 - *Google SketchUp*

SketchUp permite conceptualizar y modelar imágenes en 3D de edificios, autos, personas y cualquier objeto o artículo que imagine el diseñador o dibujante. Además el programa incluye una galería de objetos, texturas e imágenes listas para descargar. Es un programa desarrollado y publicado por Google. *SketchUp* fue diseñado con el objetivo de que pudiera usarse de una manera intuitiva y flexible.

Como ventajas, *SketchUp* es simple de usar y se puede integrar el modelo realizado a Google Maps y Google Earth. La principal desventaja es que es un programa privativo cuya

licencia tiene actualmente un costo de u\$s495; otra desventaja es que los diseños obtenidos no tienen una gran calidad visual (sombras, reflejos, texturas, etc.), como veremos a continuación. Otra desventaja notable en los ejemplos anteriores es que no puede lograrse modelos con gran realismo, tales como iluminación y sombreado, entre otros efectos (figura 2.6.1.1)

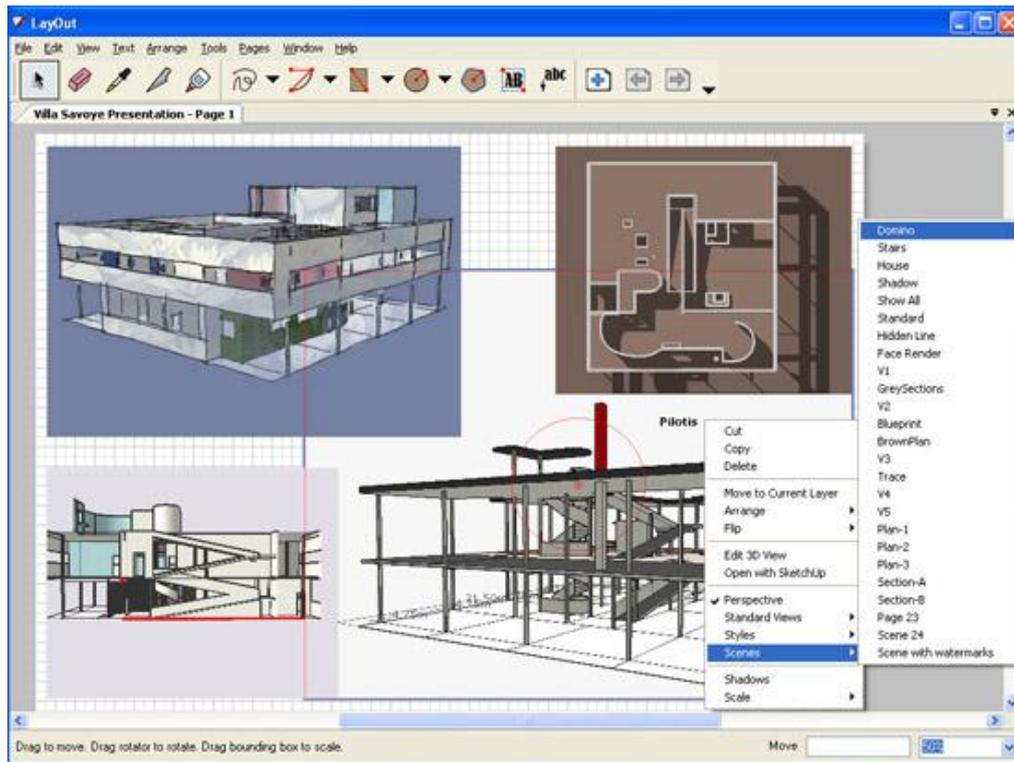


Figura 2.7.1.1. Aquí se puede observar la interfaz con el usuario del programa SketchUp

2.7.2 - Autodesk 3ds Max

3D Studio Max (como era denominado anteriormente) es un programa de creación de gráficos y animación 3D desarrollado por Autodesk, en concreto la división Autodesk Media & Entertainment (anteriormente Discreet). Creado inicialmente por el Grupo Yost para Autodesk, salió a la venta por primera vez en 1990 para DOS.

3ds Max (figura 2.7.2.1), con su arquitectura basada en *plugins*, es uno de los programas de animación 3D más utilizado, especialmente para la creación de video juegos, anuncios de televisión, en arquitectura o en películas. Hasta el momento este programa sólo funciona bajo el sistema operativo de Microsoft Windows.

La gran ventaja es que permite realizar diseños de gran calidad, y posee un repositorio de objetos muy importante. Como desventaja, es privativo y la licencia del 3ds Max 2012 cuesta US\$3.495 mas US\$1.745 (opcionales) de *upgrade* (es decir, que incluye objetos y características adicionales).

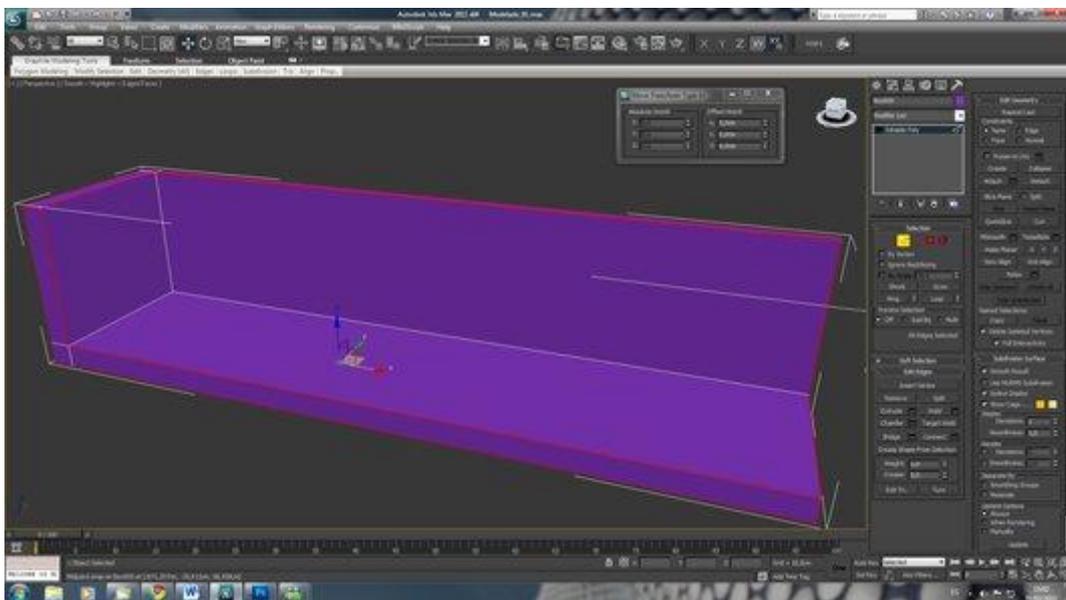


Figura 2.7.2.1. Aquí se puede observar la interfaz con el usuario del programa 3ds Max 2011.

2.7.3 - Blender

Blender es un programa multiplataforma, dedicado especialmente al modelado, animación y creación de gráficos tridimensionales.

El programa fue inicialmente distribuido de forma gratuita pero sin el código fuente, con un manual disponible para la venta, aunque posteriormente pasó a ser software libre.

Actualmente es compatible con todas las versiones de Windows, Mac OS X, Linux, Solaris, FreeBSD e IRIX.

Como desventaja, tiene una muy peculiar interfaz gráfica de usuario, criticada como poco intuitiva, pues no se basa en el sistema clásico de ventanas; pero tiene a su vez ventajas importantes sobre éstas, como la configuración personalizada de la distribución de los menús y vistas de cámara (figura 2.7.3.1).

Algunas ventajas:

- Multiplataforma, libre, gratuito y con un tamaño de origen realmente pequeño comparado con otros paquetes de 3D, dependiendo del sistema operativo en el que se ejecuta.
- Capacidad para una gran variedad de primitivas geométricas, incluyendo curvas, mallas poligonales, vacíos, NURBS, *metaballs*.
- Junto a las herramientas de animación se incluyen cinemática inversa, deformaciones por armadura o cuadrícula, vértices de carga y partículas estáticas y dinámicas.
- Edición de audio y sincronización de video.
- Características interactivas para juegos como detección de colisiones, recreaciones dinámicas y lógica.
- Posibilidades de renderizado interno versátil e integración externa con potentes trazadores de rayos o "raytracer" libres como *kerkythea*, *YafRay* o *Yafrid*.4
- Lenguaje *Python* para automatizar o controlar varias tareas.
- Blender acepta formatos gráficos como TGA, JPG, Iris, SGI, o TIFF. También puede leer ficheros Inventor.
- Motor de juegos 3D integrado, con un sistema de ladrillos lógicos. Para más control se usa programación en lenguaje Python.

- Simulaciones dinámicas de objetos blandos, partículas y fluidos.
- Modificadores apilables, para la aplicación de transformación no destructiva sobre mallas.
- Sistema de partículas estáticas para simular cabellos y pelajes, al que se han agregado nuevas propiedades entre las opciones de *shaders* para lograr texturas realistas.

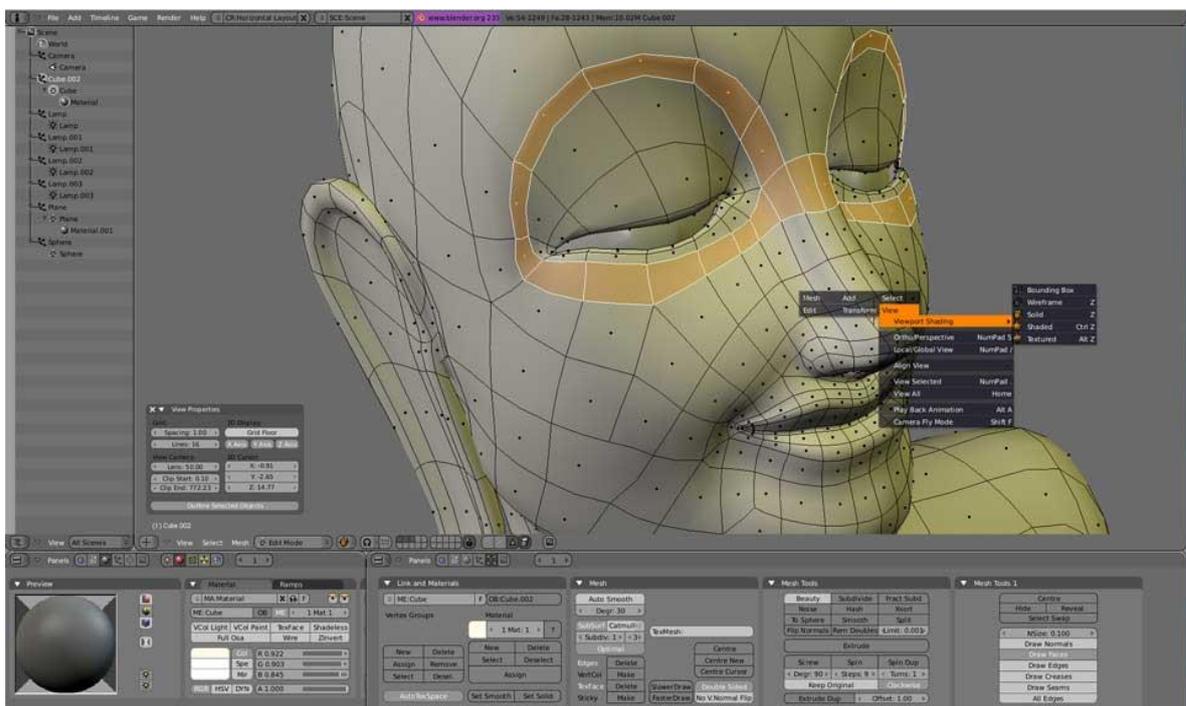


Figura 2.7.3.1. Aquí se puede observar la interfaz con el usuario del programa Blender.

2.7.4 - Unity 3

Unity, es una herramienta para el desarrollo de juegos 3D (motor de juegos) muy simple de usar y potente. Está disponible para la Plataforma Windows y Mac OS X, y permite crear juegos para Windows, Mac, Xbox 360, PlayStation 3, Wii, iPad, iPhone, y Android. Gracias

al Plug-In Web de Unity, también se pueden desarrollar juegos que se ejecutan en el navegador, para Windows y Mac.

El motor provee módulos de renderización, luces, modelado de terreno, física, sonidos, programación y redes.

Una particularidad de este motor es que los objetos son llamados “Assets”, que pueden ser importados, o puede usarse los ya implementados por el motor.

Es muy bueno para la interacción y renderización, pero no es una herramienta completa para modelado (figura 2.7.4.1)

Unity 3d tiene varias versiones disponibles al público. La versión gratuita es denominada Unity 3d, a diferencia de aquellas que son pagas, que se denominan Unity 3d Pro (posee muchas características que no están disponibles en la versión gratuita). El valor de Unity 3d Pro es u\$s 1500 sin ningún soporte para algunos sistemas (ej. iOS o Android.).

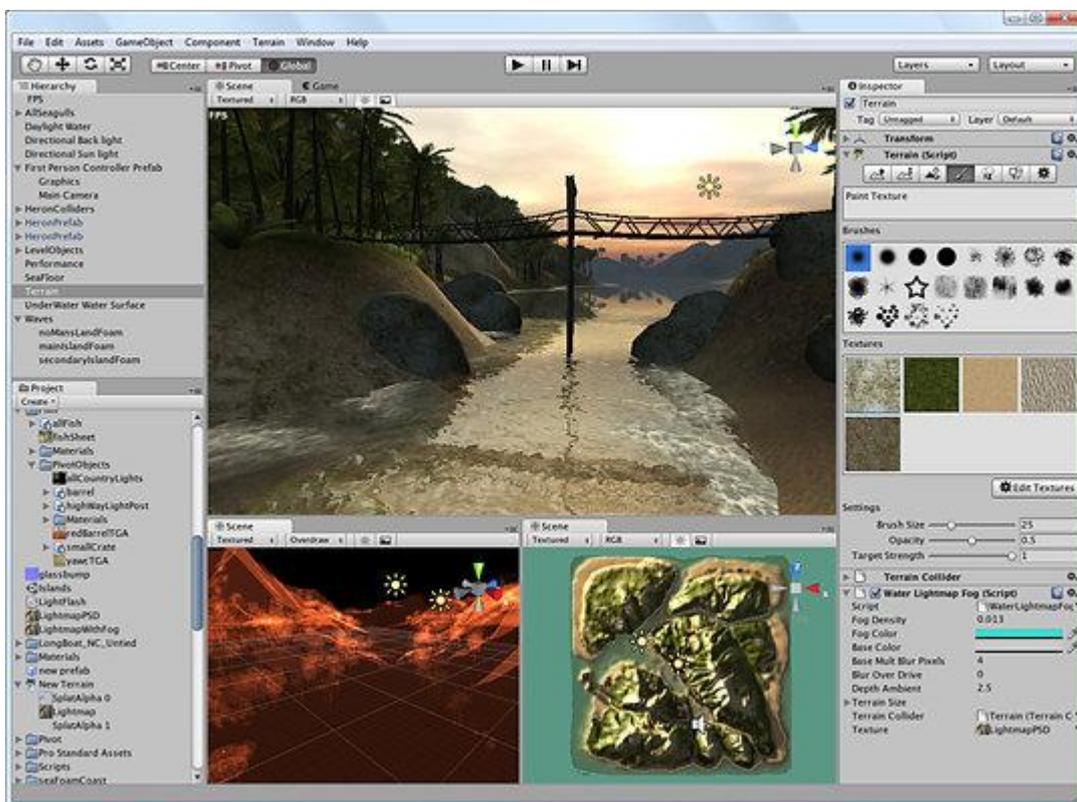


Figura 2.7.4.1. Aquí se puede observar la interfaz con el usuario del programa Unity.

2.7.5 - Autodesk AutoCAD 2012

Autodesk AutoCAD es un programa de diseño asistido por computadora para dibujo en dos y tres dimensiones. Actualmente es desarrollado y comercializado por la empresa Autodesk [30].

AutoCAD es un software reconocido a nivel internacional por sus amplias capacidades de edición, que hacen posible el dibujo digital de planos de edificios, piezas mecánicas (figura 2.7.5.1) o la recreación de imágenes en 3D.

AutoCAD es uno de los programas más usados, elegido por arquitectos, Ingenieros y diseñadores industriales.

AutoCAD gestiona una base de datos de entidades geométricas (puntos, líneas, arcos, etc.) con la que se puede operar a través de una pantalla gráfica en la que se muestran éstas, el llamado editor de dibujo. La interacción del usuario se realiza a través de comandos, de edición o dibujo, desde la línea de órdenes, a la que el programa está fundamentalmente orientado. Las versiones modernas del programa permiten la introducción de éstas mediante una interfaz gráfica de usuario o en RUSO GUI, que automatiza el proceso.

Esta herramienta procesa imágenes de tipo vectorial, aunque admite incorporar archivos de tipo fotográfico o mapa de bits, donde se dibujan figuras básicas o primitivas (líneas, arcos, rectángulos, textos, etc.), y mediante herramientas de edición se crean gráficos más complejos. El programa permite organizar los objetos por medio de capas o estratos, ordenando el dibujo en partes independientes con diferente color y grafismo. El dibujo de

objetos seriados se gestiona mediante el uso de bloques, posibilitando la definición y modificación única de múltiples objetos repetidos.

Parte del programa AutoCAD está orientado a la producción de planos, empleando para ello los recursos tradicionales de grafismo en el dibujo, como color, grosor de líneas y texturas tramadas. AutoCAD, a partir de la versión 11, utiliza el concepto de espacio modelo y espacio papel para separar las fases de diseño y dibujo en 2D y 3D, de las específicas para obtener planos trazados en papel a su correspondiente escala

Es en la versión 11, donde aparece el concepto de modelado sólido a partir de operaciones de extrusión, revolución y las booleanas de unión, intersección y sustracción. Este módulo de sólidos se comercializó como un módulo anexo que debía de adquirirse aparte. Este módulo sólido se mantuvo hasta la versión 12, luego de la cual, AutoDesk, adquirió una licencia a la empresa Spatial, para su sistema de sólidos ACIS.

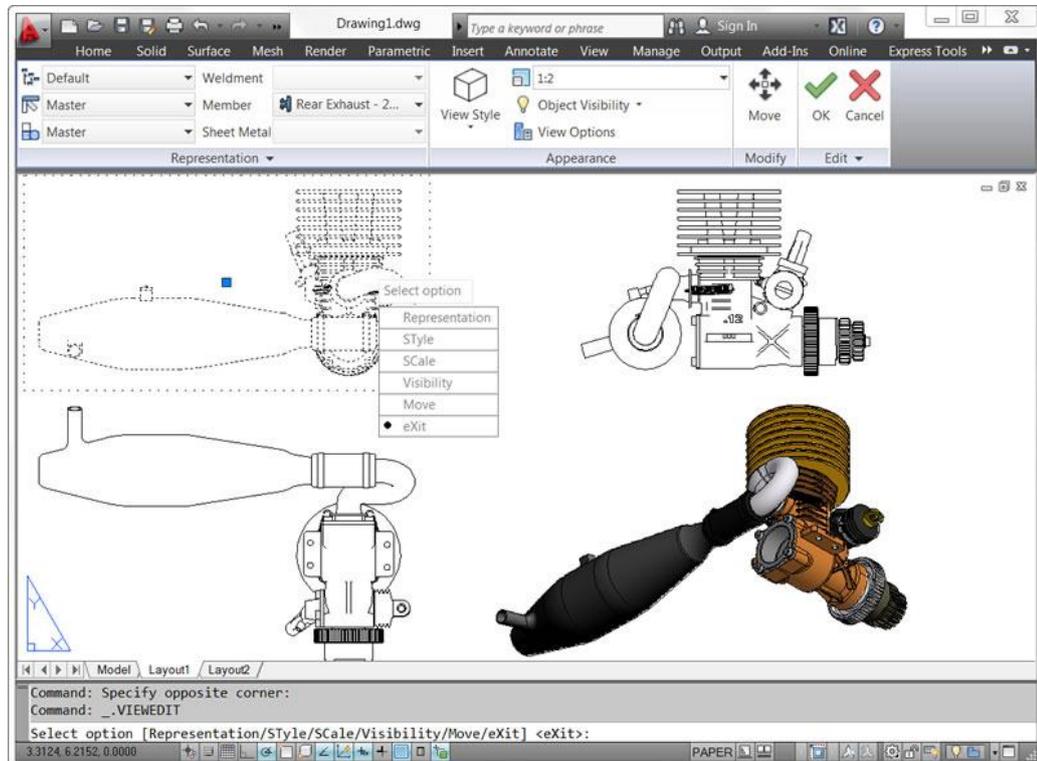


Figura 2.7.5.1. Aquí se puede observar la interfaz con el usuario del programa AutoCAD 2012.

2.7.6 - Autodesk Maya

Autodesk Maya (también conocido como Maya) es software dedicado al desarrollo de gráficos en 3d, efectos especiales y animación [31].

Maya se caracteriza por su potencia y las posibilidades de expansión y personalización de su interfaz y herramientas'. MEL (Maya Embedded Language) es el código que forma el núcleo de Maya, y gracias al cual se pueden crear scripts y personalizar el paquete. En la figura 2.7.6.1 se puede apreciar la interfaz de usuario de esta herramienta.

El programa posee diversas herramientas para modelado, animación, render, simulación de ropa y cabello, dinámicas (simulación de fluidos), etc.

Maya trabaja con cualquier tipo de superficie NURBS, Polygons y Subdivision Surfaces, incluye la posibilidad de convertir entre todos los tipos de geometría.

- Nurbs: Son figuras creadas a base de curvas y superficies cuyos componentes son básicamente los CV's (control vertex), las isoparms (isoparamétricas) y los hulls (loops enteros de isoparms).
- Polygons: Son los objetos más fáciles de modelar por su falta de complejidad y su mayor número de herramientas. Sus componentes básicas son las Faces (caras), Edges (aristas) y Vertex (vértices).
- Subdivisiones: Son un híbrido entre las Nurbs y los Polygons. Sin embargo no se pueden modelar usando ambos estilos a la vez, para ello hay que escoger en qué modo se desea modelar (Standard Mode o Polygon Mode). Poseen los mismos componentes que las Nurbs y los Polygons además de un modo de

refinamiento por niveles para obtener mayor subdivisión geométrica y conseguir así mayor detalle de modelado.



Figura 2.7.6.1. Aquí se puede observar la interfaz con el usuario del programa Autodesk Maya.

2.7.7 - Cinema 4D

Cinema 4D es un programa de creación de gráficos y animación 3D desarrollado por la empresa Maxon [32].

Permite modelado (primitivas, splines, polígonos), texturización y animación. Sus principales virtudes son una muy alta velocidad de renderización, una interfaz (figura 2.7.7.1) altamente personalizable y flexible, y una curva de aprendizaje (comparado con otros programas de prestaciones profesionales similares) muy vertical.

Una de las características más destacadas de Cinema 4D es la modularidad.

Partiendo de una versión básica de Cinema 4D pueden añadirse módulos especializados independientes en función de las necesidades del proyecto a realizar:

- *Advanced Render*: Iluminación global, cáusticas, oclusión de ambiente, profundidad de campo y otros efectos fotorrealistas, generador de cielos volumétricos. Este módulo, desde la versión 2.6, contiene también *PyroCluster*, una herramienta de generación de partículas complejas, llamadas volumétricas, como humo, polvo, llamas, vapor, etc.
- *Dynamics*: Gestión de la gravedad y los efectos físicos.
- *Mocca*: Animación esquelética de personajes. Permite aplicar controladores a personajes u objetos. Este módulo contiene también *Clothilde*, una herramienta de simulación de tela.
- *Hair*: Creación de cabello, pelo, y plumas con animación secundaria automática, así como simulación física.
- *ThinkingParticles*: Gestión avanzada de partículas (de manera que pueden reaccionar ante estímulos como colisión entre ellas mismas o volúmenes en la escena).
- *MoGraph*: Efectos de animación complejos, clonación de objetos modular.

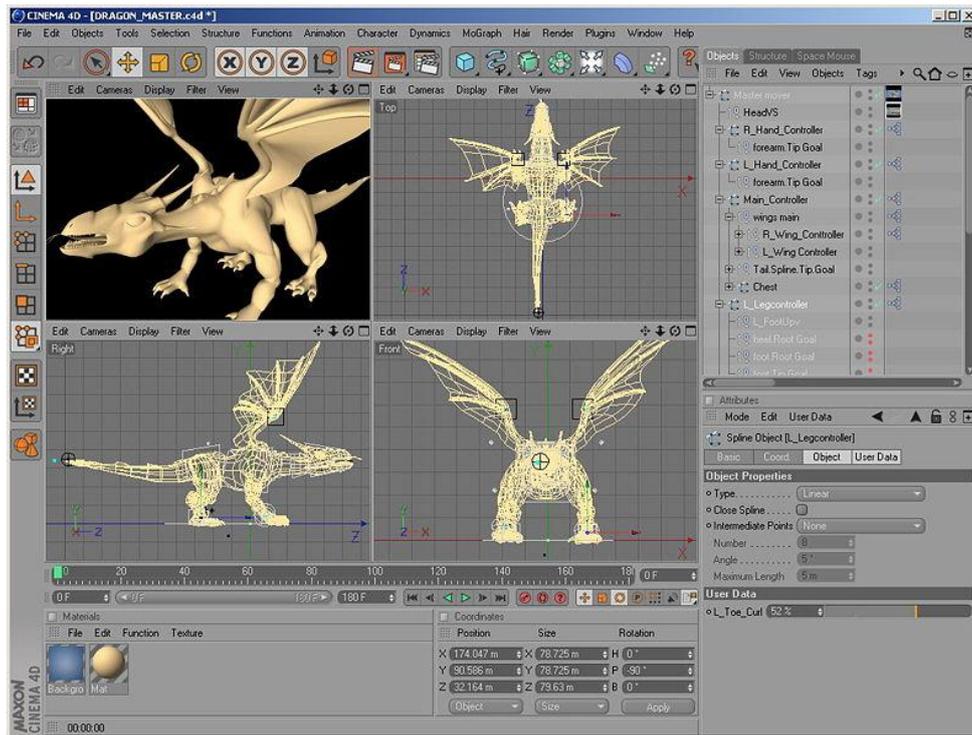


Figura 2.7.7.1. Aquí se puede observar la interfaz con el usuario del programa Cinema 4D.

2.7.8 - Rhinoceros 3D

Rhinoceros 3D es una herramienta de software para modelado en tres dimensiones basado en NURBS [33]. El programa es comúnmente usado para el diseño industrial, la arquitectura, el diseño naval, el diseño de joyas, el diseño automotriz, CAD/CAM, prototipación rápida, ingeniería inversa, así como en la industria del diseño gráfico y multimedia (Figura 2.7.8.1).

Rhinoceros 3D se especializa principalmente en el modelado libre mediante NURBS. Aunque hay disponibles varios agregados (*add-ons*) para el renderizado fotorrealístico raytracing (*Flamingo*), renderizado no fotorrealístico (*Penguin*) y la animación (*Bongo*). Como muchas aplicaciones de modelado 3D, Rhinoceros incorpora el lenguaje llamado *RhinoScript*, basado en Visual Basic.

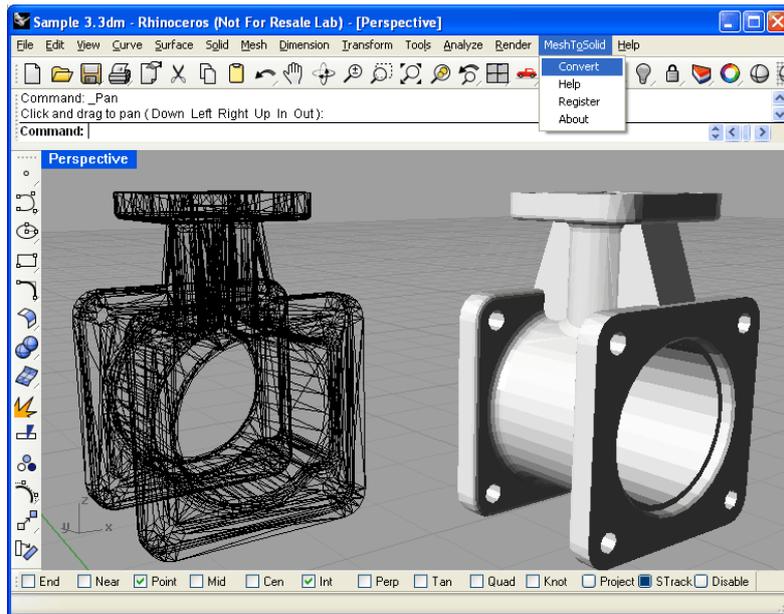


Figura 2.7.8.1. Aquí se puede observar la interfaz con el usuario del programa Rhinoceros 3D.

2.7.9 - TrueSpace

TrueSpace es un software de creación de gráficos y animación 3D desarrollado por *Caligari Corporation* [34].

Truespace es un programa de diseño 3D potente. El usuario se mueve por un espacio renderizado con una interfaz intuitiva y personalizable, con múltiples funciones (figura 2.7.9.1).

Permite crear y modelar cualquier tipo de objeto, colorearlo, añadirle atributos físicos, animarlo, etc. También ofrece una amplia gama de objetos ya creados y escenas que facilitan comenzar a usar el programa.

En cuanto a modelado, trueSpace permite modelado poligonal, NURBS, y subdivisiones.

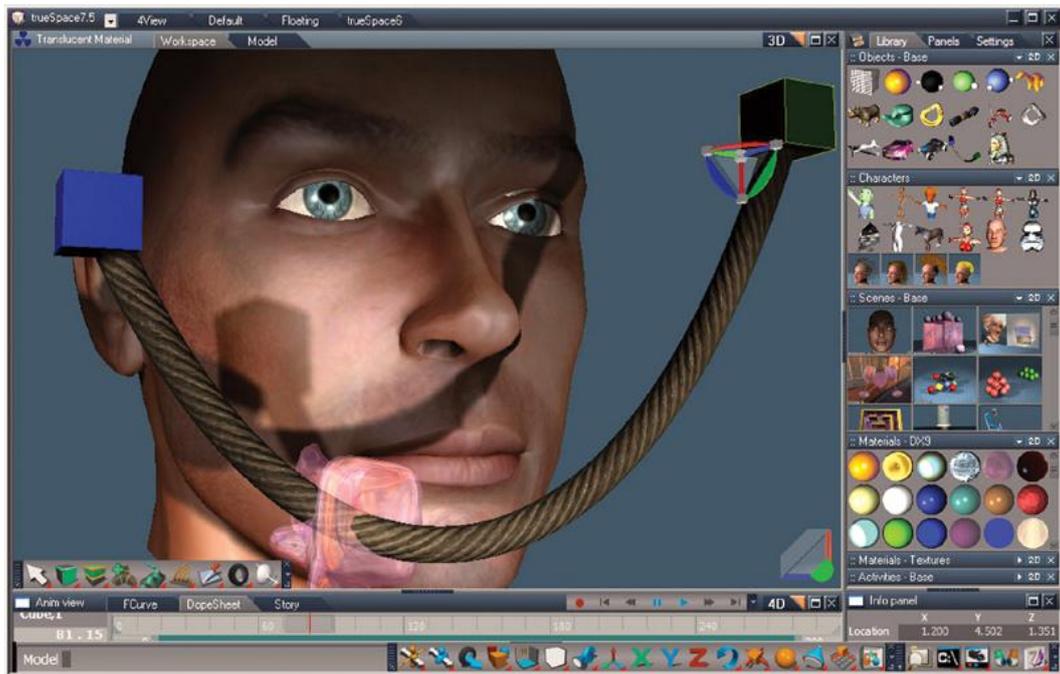


Figura 2.7.9.1. Aquí se puede observar la interfaz con el usuario del programa trueSpace.

Capítulo 3

Propuesta

Propuesta

Como se mencionó en los capítulos anteriores, no existe un modelo 3D digital de la Fortaleza Protectora Argentina. Existe un modelo 3D (maqueta) que se encuentra en una sala histórica del “V Cuerpo de Ejército” de Bahía Blanca, pero sólo se puede acceder mediante una visita guiada, con cita previa de esto se desprende que, no es simple poder contemplar cómo era este sitio histórico.

En este capítulo se describe la propuesta, que consiste en el modelado y recorrido digital 3D de la Fortaleza Protectora Argentina, que podrá estar disponible en exhibiciones y/o en la web.

3.1 - Descripción de la propuesta

En este trabajo se propone realizar un modelo 3D digital, con un recorrido interactivo de la Fortaleza Protectora Argentina, con fines turísticos y educativos.

El usuario verá el modelo con una visión de “primera persona”, esto quiere decir que observará el mundo desde la perspectiva del personaje protagonista (que será el usuario), esta vista da una sensación de mayor realismo y de presencia dentro del modelo. El usuario podrá recorrer el modelo en dos modalidades:

- Libre: En esta modalidad el usuario “caminará” por la fortaleza sin ningún tipo de restricción. Este modo de recorrido es similar al encontrado en los juegos de primera persona (FPS, *first-person-shooter*).
- Recorrido predefinido: Bajo esta modalidad el usuario no tendrá el control de la dirección del movimiento. Los movimientos serán predefinidos y la única interacción disponible al usuario será la de pausar el recorrido y reanudarlo. Esta modalidad se asemeja a la de un recorrido guiado, en donde los puntos por donde pasará el camino han sido prefijados.

3.2 – Modelado

El modelado de la Fortaleza Protectora Argentina se realizará mediante CAD (“Computer Aided Design”) basado en los planos históricos recopilados (figura 3.2.1) y en las imágenes obtenidas de la maqueta anteriormente mencionada. Se utilizará esta técnica debido a que la fortaleza real ya no existe; en su lugar se encuentran edificaciones correspondientes a las manzanas que están entre las calles Brown, Vieytes, Moreno, Estomba, Chiclana y O’Higgins (figura 3.2.2).

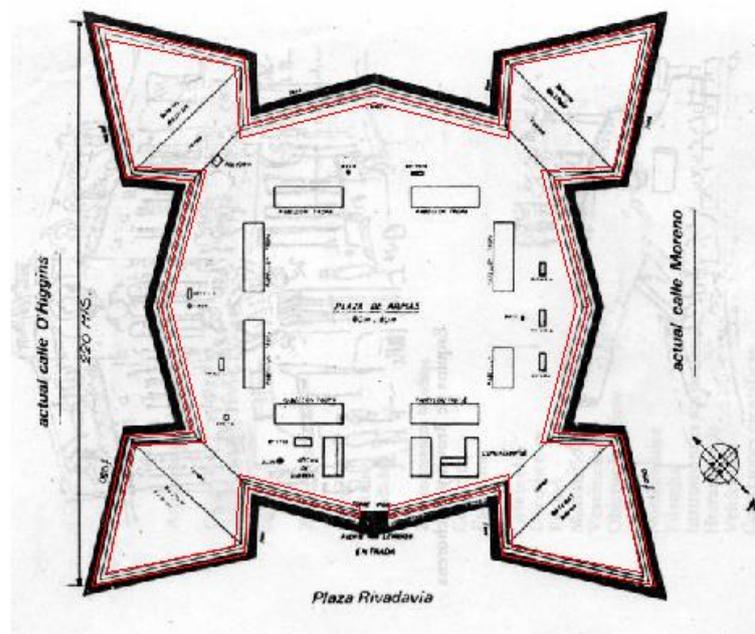


Figura 3.2.1. Estructura levantada en AutoCAD desde el plano

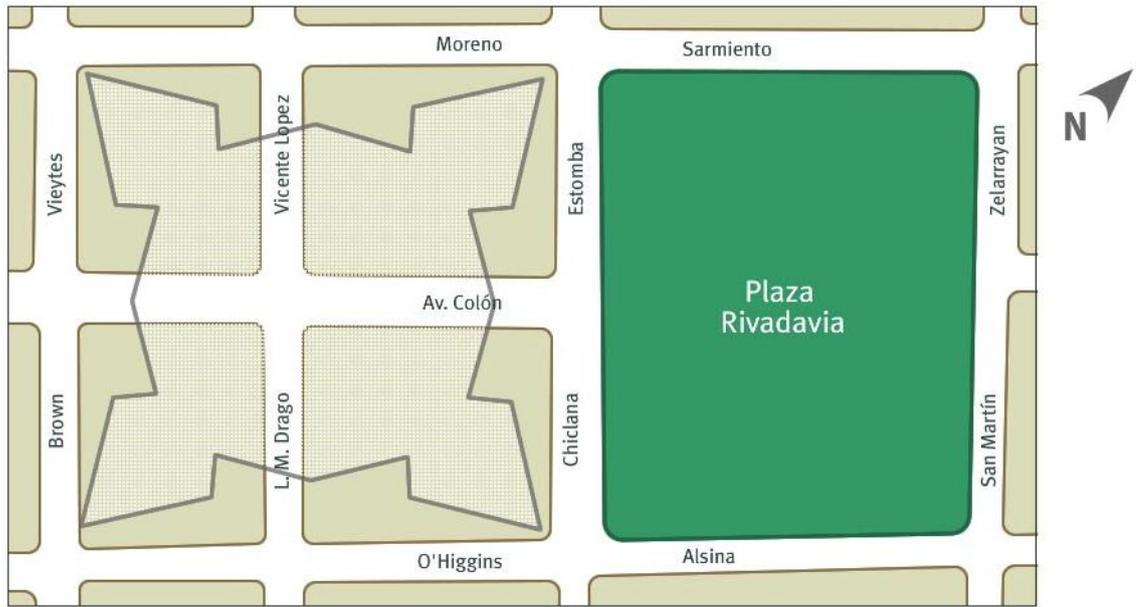


Figura 3.2.2: Ubicación de la Fortaleza Protectora Argentina.

3.3 – Rendering

El *render* que se utilizará será el que provee el motor de juegos Unity 3d, el cual implementa el método Ray Tracing. Este motor permite al desarrollador modificar algunas propiedades, como por ejemplo, el “*rendering path*” del pipeline gráfico (es decir, la secuencia de algoritmos que se utilizará en la etapa de *rendering*), o la calidad del *render* ([14,15]).

La calidad del *render* se corresponde con distintas categorías una de las cuales podrá ser elegida en el momento en que se ejecute la aplicación; las categorías son: *Fastest*, *Fast*, *Simple*, *Good*, *Beautiful* y *Fantastic*, (ordenadas por velocidad de mayor a menor, o por calidad de peor a mejor). El motor permite modificar las características (figura 3.3.1) de cada categoría, algunas de las más importantes son ([14,15]):

- *Pixel Light Count*: número máximo de luces por pixel a tener en cuenta.
- *Shadows*: qué tipo de sombras se permiten.
- *Anti Aliasing*: si se usa o no *antialiasing*.



Figura 3.3.1: Opciones de calidad.

3.4 – Interactividad

Se permitirá interactuar con el modelo por medio de realidad virtual no inmersiva, es decir, se permitirá recorrer con los controladores básicos de una computadora (Teclado, Mouse, Monitor) más Joystick.

3.5 – Licenciamiento

Se escogió la licencia de *Creative Commons*. Esta permite mantener los derechos de autor pero permite a otras personas copiar y distribuir nuestro modelo, siempre y cuando reconozcan la correspondiente autoría.

3.6 – Recopilación de datos

Los datos recolectados en un principio constaban de algunos planos rudimentarios, bosquejos y relatos que describen con poco nivel de detalle a la Fortaleza Protectora Argentina ([2], [3], [4], [5], [6]).

Se tenía poca información sobre la estructura de la fortaleza (los materiales utilizados, el aspecto de las paredes, el material, la textura, etc.), por lo que se suponía que las paredes eran de ladrillo o piedra con una inclinación de 90° con respecto al piso, y que en la parte superior de las paredes había un pequeño desnivel para que los soldados puedan caminar o poner cañones. La estructura que se imaginaba era similar a la fortaleza de Santa Teresa ubicada en Uruguay (Figura 4.1.2.4).



Figura 4.1.2.4. Fotografía de una de las paredes exteriores de la Fortaleza de Santa Teresa ubicada en el Departamento de Rocha, Uruguay

Pero una visita a la maqueta de la fortaleza que se encuentra en la sala histórica del IV Cuerpo de Ejército nos demostró que todo lo supuesto era falso (y que la bibliografía con la

Capítulo 4

Implementación

Implementación

Una vez recopilada parte de la información necesaria para construir las bases del modelo, se dispuso comenzar con la misma. En una primera instancia, se construyó el modelo 3D, para luego implementar las distintas maneras de recorrerla.

4.1 – Modelo de la Fortaleza Protectora Argentina

La construcción del modelo demandó la mayor parte del tiempo y esfuerzo. Se comenzó creando una estructura muy básica, pero exacta en cuanto a medidas, de lo que podríamos denominar el esqueleto de la fortaleza. Luego se moldeó la estructura principal para lograr un prototipo más detallada (es decir, una imagen con falta de realismo y una baja suavidad de los vértices). A partir de esto, con un arquetipo bastante preciso de lo que sería la fortaleza, se completa el modelo colocando mallas suaves, moldeadas a mano, sobre la estructura. Finalmente se aplican texturas y filtros según corresponda, terminando así con el modelo de la fortaleza.

4.1.1 – Estructura

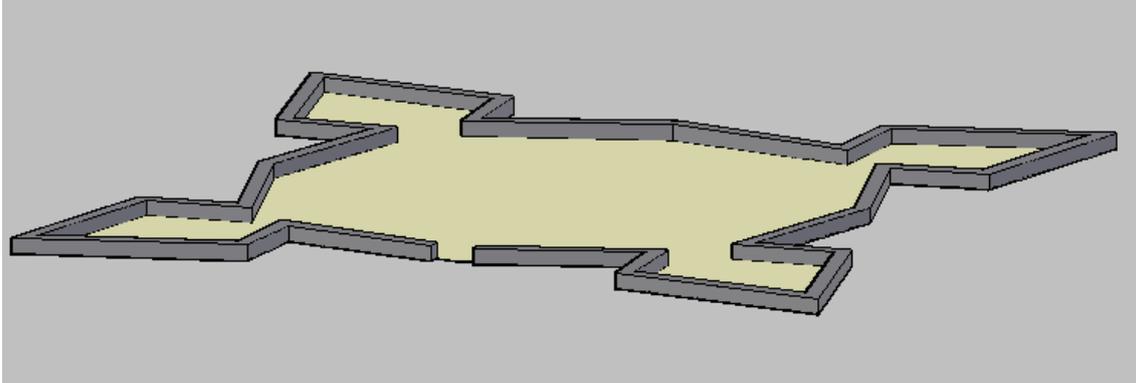


Figura 4.1.1.1. Estructura de la fortaleza (AutoCAD)

Se comenzó con la creación de una estructura simple de la fortaleza, sobre la cual iniciar el modelado. Para esto se utilizó *Autodesk AutoCAD* ya que es una herramienta que permite crear planos de construcciones con gran la exactitud y precisión, como la que se necesita en este caso. Luego se utilizara una herramienta específica de modelado ya que *Autodesk AutoCAD* no provee todas las facilidades de modelado interactivo, esto se debe a que esta herramienta es específica para la creación de planos de construcciones edilicias.

Para ello se configuró el AutoCAD en metros. Es decir, 1 unidad = 1 metro. Una vez seteadas las dimensiones se importó la imagen correspondiente al plano de planta (figura 3.2.1) y a partir de ésta se prosiguió a construir la base. Aquí se encontró la primera ambigüedad: los ángulos de los bastiones en el plano diferían en casi 7° con respecto a la información bibliográfica disponible. Teniendo en cuenta que ninguna de las dos fuentes es completamente exacta, se tomó la decisión de expandir el ángulo tanto como fuera necesario para mantener las medidas de las longitudes de los muros lo más cercanas a las que se manejaban. De esta manera, los ángulos resultantes sobrepasan en casi 4° a los ángulos del plano (ver figura 4.1.1.2).

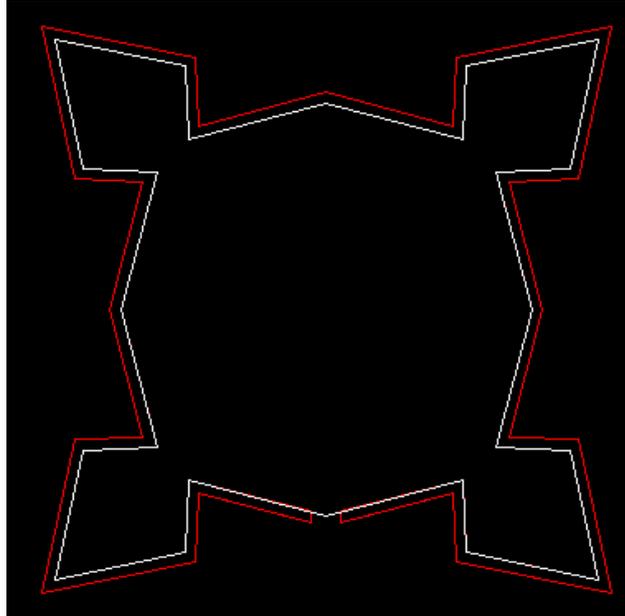


Figura 4.1.1.2. – Base de la fortaleza

Una vez logrado el modelo digital de la base inferior, se generó la fortaleza con las herramientas de simetría y escalado del AutoCAD. Posteriormente se observó que tener la estructura en una sola capa era incómodo para el moldeo. Por esta razón se decidió partir nuevamente de la base y levantar muro por muro, separando cada uno de ellos en una capa distinta. La decisión de utilizar una capa por muro (y no una capa para varios muros) radica en el realismo que se le quiere dar al moldeo: la fortaleza, como toda construcción, nunca va a ser completamente simétrica ni perfecta, por lo que cada muro, cada pared, cada fosa, debe ser moldeada por separado. Construir la fortaleza en capas nos permite cierta flexibilidad al momento del moldeo y texturizado que no se tiene con una sola capa.

4.1.2 – Moldeado

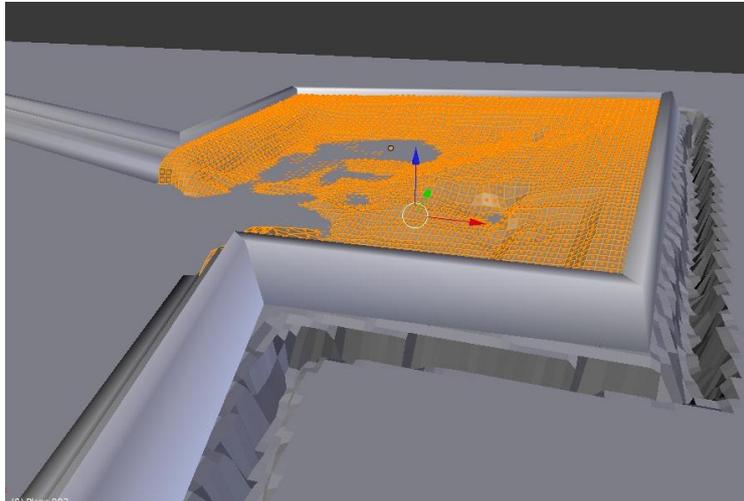


Figura 4.1.2.1 – Vista del moldeado

Blender es la herramienta de modelado que se utilizó para modificar la estructura creada de la fortaleza. Para poder hacerlo, primero se tuvo que importar el modelo creado anteriormente en AutoCAD 3D, pero como este es una herramienta de la empresa Autodesk, este permite exportar el modelo creado con el formato y extensión “.max” para la herramienta 3ds Max de la misma empresa y no en “.blender” o “.3ds” (versión anterior a “.max”). *Blender* permite importar únicamente “.max”, por lo que se debió realizar un paso obligado por dicha herramienta para poder hacer la transformación.

Así es que se obtuvo el modelo en el formato “.3ds” para poder importarlo a *Blender*, pero la importación de “.3ds” a *Blender* generó una estructura errónea formada por triángulos (Figura 4.1.2.2), algo no muy cómodo de manipular a la hora de modelar, por lo que se tuvo que buscar una alternativa a la importación en formato “.3ds”.

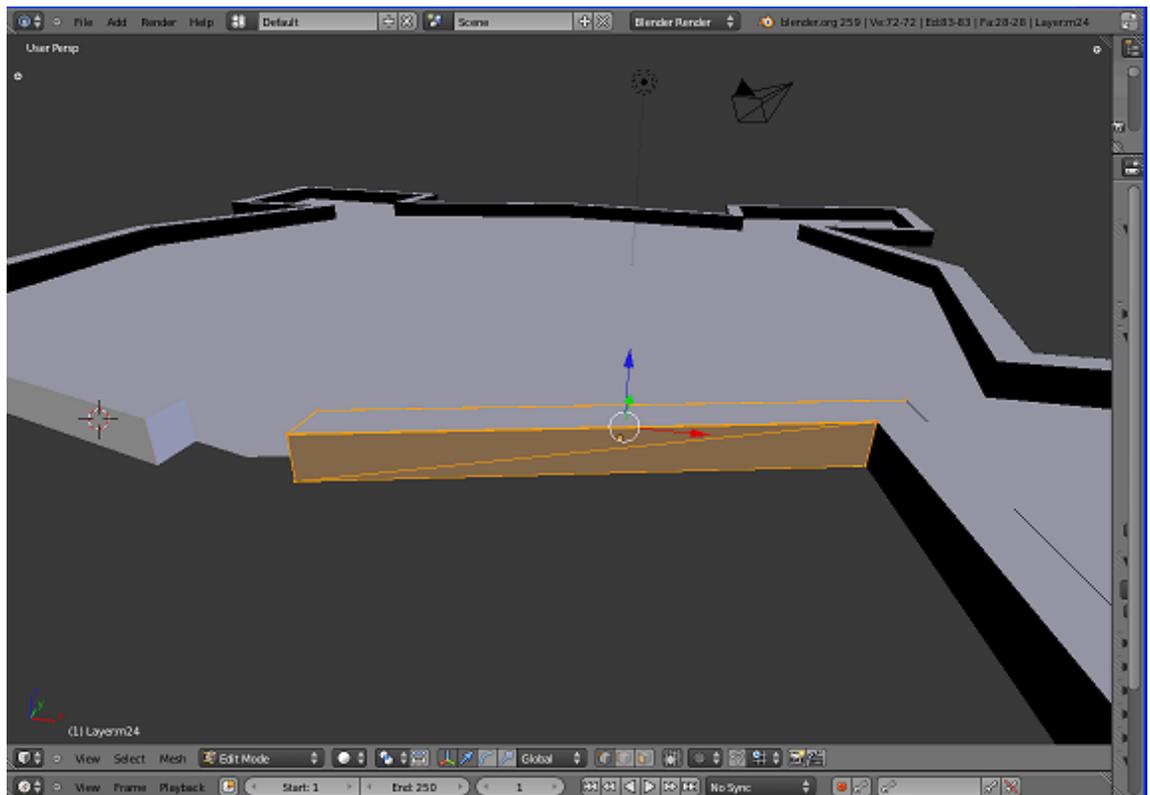


Figura 4.1.2.2. En esta captura de pantalla se puede apreciar los triángulos obtenidos a partir de la importación de .3ds

Entre las alternativas de exportación que permite 3DS Max se encuentra “.obj”, que es un formato que respeta el modelo original con sus vértices y caras, por lo que se utilizó este formato para poder importar el modelo a Blender con éxito (Figura 4.1.2.3).

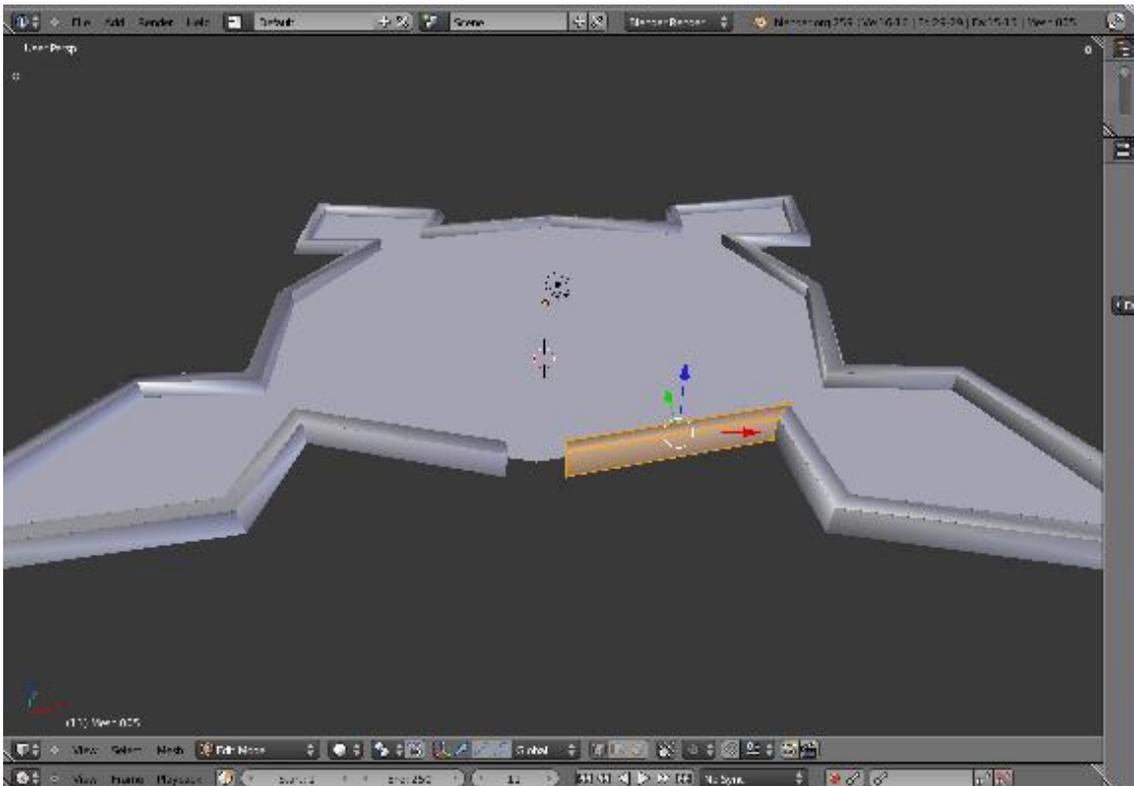


Figura 4.1.2.3. Aquí se puede observar la correcta importación del modelo .obj

Una vez importada la estructura de la fortaleza a Blender, se comenzaron a modelar los detalles de la misma. Para ello se tuvo que obtener un manejo importante de la herramienta, ya que Blender es un programa muy complejo y completo. También se debe conocer en profundidad los detalles de la fortaleza para poder recrearlos, algo que no se conocía en las primeras etapas. Debido a esto, se tuvieron que hacer varios cambios a la estructura original (agregar la inclinación externa a las paredes, crear los desniveles internos de las paredes y las puntas, y eliminar el desnivel de las partes superiores) y por último se agregó la fosa que rodea a la fortaleza.

En cuanto a las herramientas, la elegida tanto para modelar como para recorrer la estructura fue Blender, ya que este no sólo es una herramienta de modelado sino que también posee un motor de juegos denominado Blender Game que permite generar la interacción con el usuario. Por lo que se utilizó Blender para modelar y generar animaciones y Blender Game para recorrer la fortaleza libremente. Blender Game utiliza un renderizador distinto al que

utiliza Blender para el modelado (*Blender Render*), esto se debe a la eficiencia y renderización en tiempo real que necesita un juego. Blender Game tiene algunas particularidades, por ejemplo en el mapeo de texturas, que difiere al del renderizador común (*Blender Render*) [1]. El motor de juegos permitió realizar un recorrido básico sobre la estructura de la fortaleza, por medio de sensores, controladores y actuadores (provistos por el módulo *Logic Game*) definidos sobre la cámara. Entonces, hasta el momento se tenía una estructura básica de la fortaleza y la posibilidad de recorrerla a través del motor de juegos (Figura 4.1.2.6).

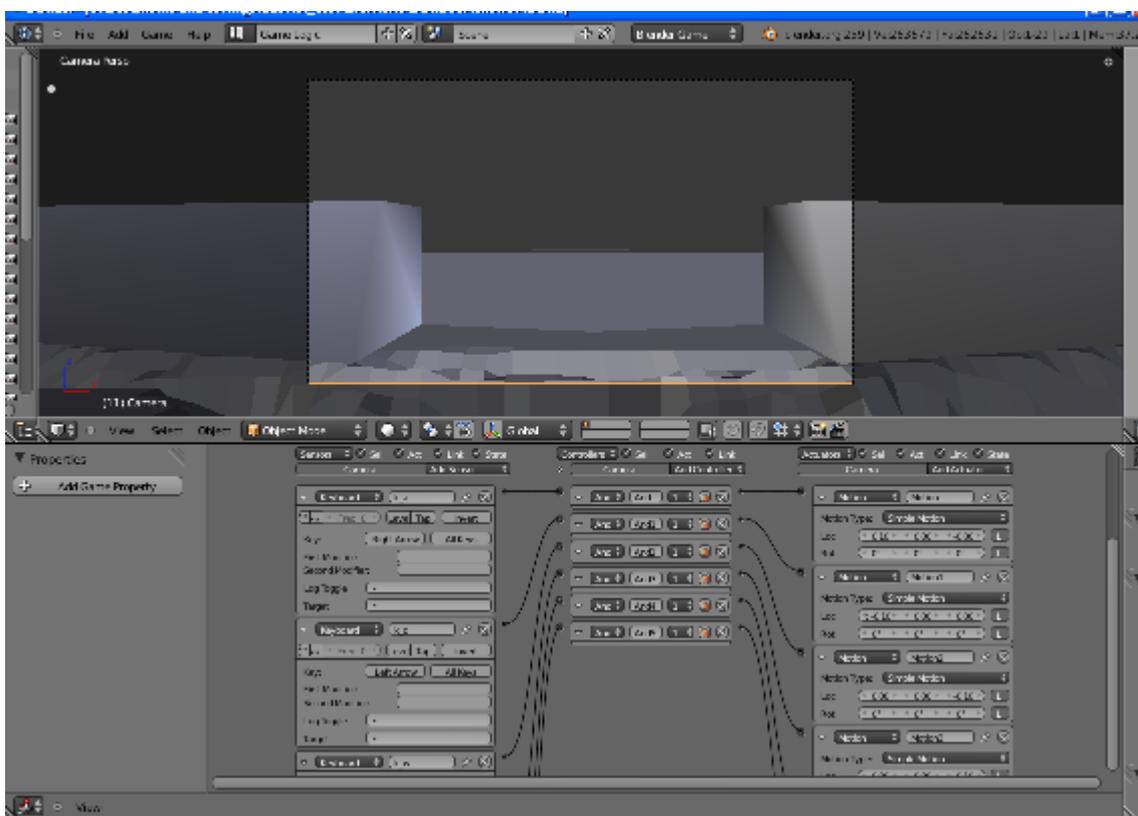


Figura 4.1.2.6. Captura de pantalla del Blender Game Engine y el módulo de Game Logic.

Una vez modelada la fortaleza geoméricamente, se deberían incorporar los atributos de la misma; es decir que debería simularse el material con el que estaba construida. Para ello se aplicaron las texturas correspondientes al modelo realizado hasta el momento. Éstas debían ser lo más realistas posible. En este punto, se decidió obtener las mismas en la web.

El primer objeto al que se le intentó dar estas características es al terreno, es decir, el suelo y la fosa. Como puede observarse en la fotografía de la maqueta el piso debería tener al menos dos texturas, una textura para las zonas donde no había vegetación, es decir los lugares donde había sólo tierra, y otra para las zonas donde si la había (yuyos, pasto, etc.). Las opciones que brinda *Blender* para lograr esto son dos:

- La primera es setear las coordenadas de textura para una cara y suavizar la diferencia de colores en los bordes de las mismas. Esta tarea manual debería hacerse en cada una de las caras en forma separada, lo cual implica una larga y ardua tarea ya que el terreno posee miles de caras. (Figura 4.1.2.5)
- La segunda opción es utilizar multitexturas, que consiste en aplicar más de una textura a un fragmento y luego por medio de una plantilla en gama de grises (*esténcil*), se le indica al renderizador que porcentaje del color de cada textura debe adquirir el fragmento. (Figura 4.1.2.6). Esta técnica es la más utilizada en el modelado de terrenos por su simpleza y facilidad de uso [2].

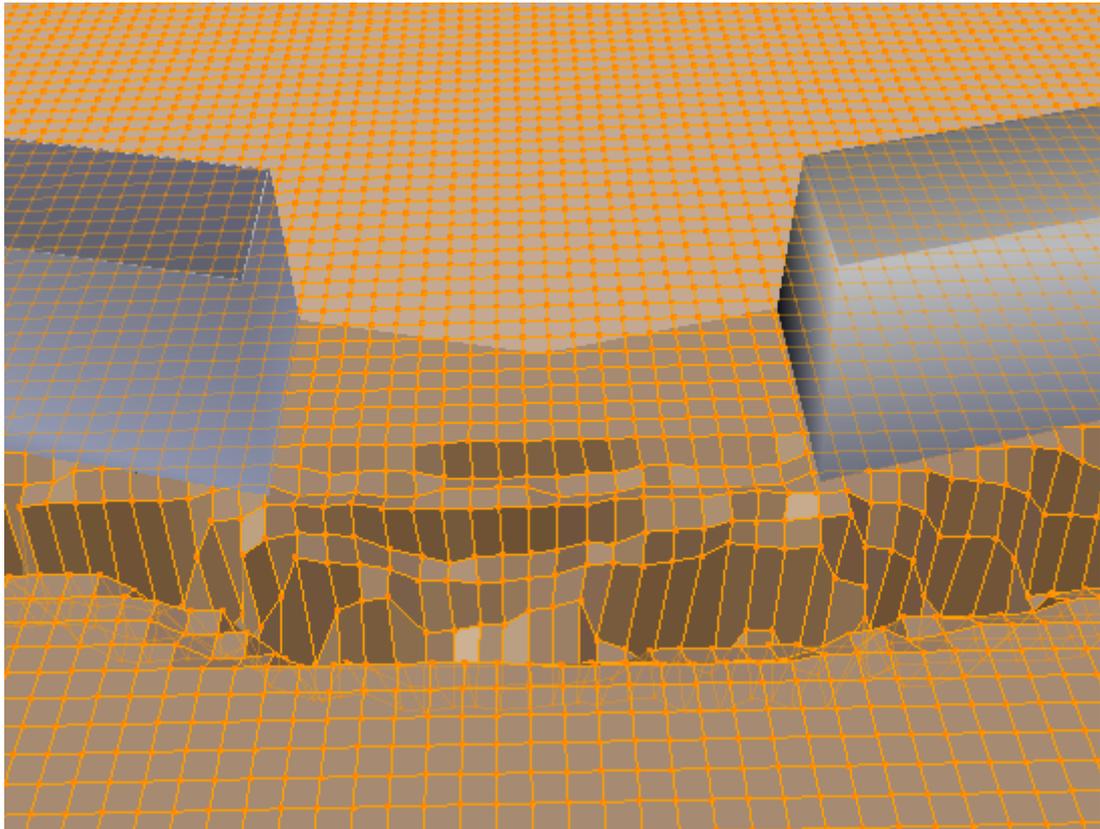


Figura 4.1.2.5. En esta captura de pantalla se puede apreciar la cantidad de caras de un sector del terreno.

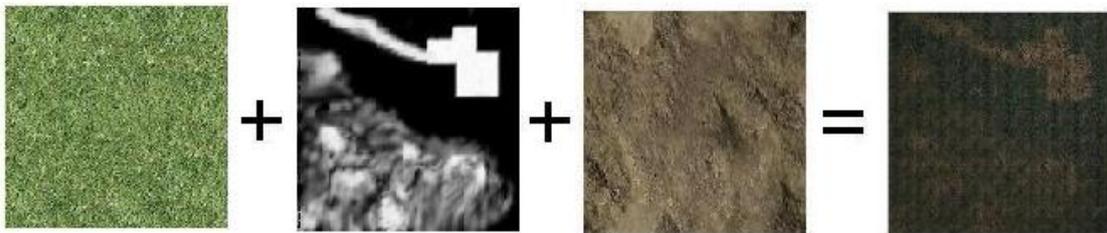


Figura 4.1.2.6. Ejemplo de aplicación de multitexturas con plantilla de grises.

Las opciones mencionadas las provee el renderizador de *Blender* (modelado) pero no el renderizador del motor de juegos *Blender Game*, por lo que se tuvo que buscar una alternativa a *Blender Game* para recorrer el modelo. Es así como se comenzó el análisis y estudio de la herramienta *Unity*.

4.1.3 – Suavizado y textura del terreno.

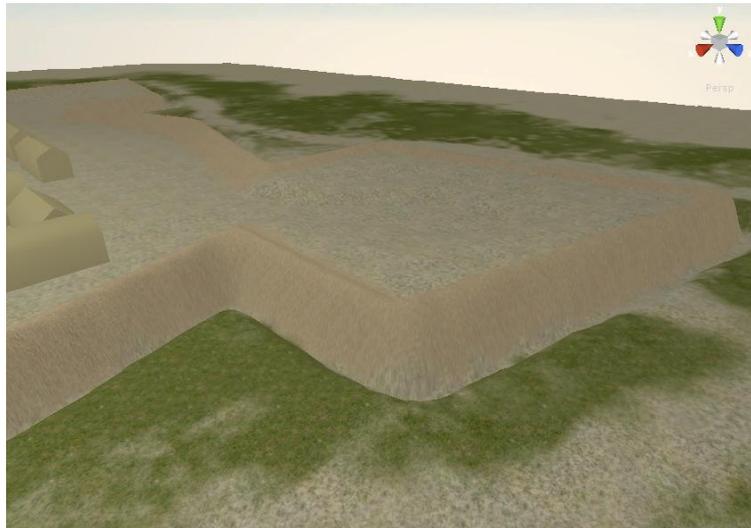


Figura 4.1.3.1 Vista del suavizado y texturado

Unity es un motor de juegos que permite aplicar múltiples texturas al terreno de forma fácil e interactiva (Figura 4.1.3.2). Tiene ya implementado un objeto (*Asset*, denominado así en la herramienta) de tipo terreno que permite realizar las modificaciones básicas y comunes que se necesitan en su modelado, por ejemplo la aplicación de múltiples texturas, o la deformación que sirve para crear montañas o depresiones (en nuestro caso, la fosa).

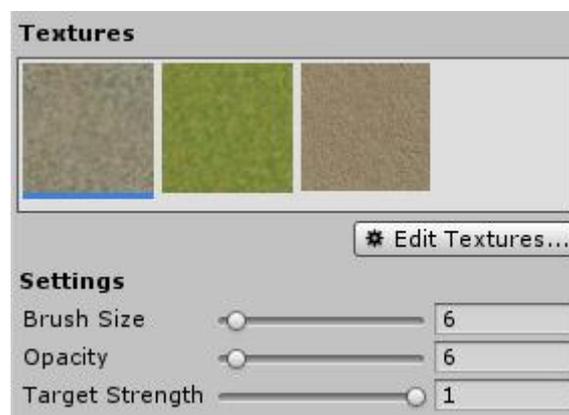


Figura 4.1.3.2. Texturas aplicadas al terreno, se selecciona una de ellas y se pinta el terreno con una brocha.

Debido a esta característica de modelado de terreno tan potente y fácil de usar, se optó por implementar el recorrido en *Unity* y continuar utilizando *Blender* en algunas cuestiones de modelado. Entonces por un lado tenemos desarrollado el modelo de la estructura de la fortaleza en *Blender*, y por otro lado, tenemos el modelado del terreno, la escena y la implementación del recorrido en *Unity*.

Luego de haber tomado las decisiones de diseño mencionadas, se debía importar el modelo de la fortaleza realizado en *Blender* y comenzar a modelar el terreno. Si bien en el manual de *Unity* consta que se pueden importar modelos de *Blender* de forma nativa (lo permite por defecto) a través de *FBX Exporter* [3], esto no fue posible. Por ello la conversión se realizó a través de archivos *.obj*. Una vez importado el modelo *.obj* al proyecto en *Unity*, cualquier cambio realizado al archivo *.obj* del proyecto se verá reflejado de forma automática en la escena de *Unity*.

Una vez finalizado el modelado del terreno y la aplicación de texturas al mismo, se procedió a texturizar el modelo de la fortaleza; podía realizarse de dos formas diferentes, un opción sería ponerle textura desde *Blender* y la otra sería amoldar el terreno, copiando la forma de la fortaleza, y así poder aplicar texturas desde *Unity*, es decir, se usaría el modelo de la fortaleza como una estructura guía cuya forma sería copiada y suplantada por el terreno creado en *Unity*. Se optó por la segunda alternativa debido a que no sólo es más simple la aplicación de texturas en *Unity* sino que además se logra una mejor vista ya que no hay diferencia entre el terreno y la fortaleza (en el primer caso, la fortaleza sería una entidad aparte del terreno) (Figura 4.1.3.3).



Figura 4.1.3.3. Terreno elevado que copia la forma de la fortaleza, con múltiples texturas.

Además del objeto terreno, Unity provee varios tipos de objeto y características que serán utilizados en la escena, tales como:

Controlador de primera persona: este objeto permite el recorrido de la escena en primera persona al ejecutar el juego. La interacción puede ser a través del mouse y del teclado.

Luces: se utilizarán dos tipos de luces, una luz ambiente y otra luz puntual simulando la luz solar.

- Mapeo ambiente cúbico: esto se logra por medio de materiales tipo Skybox que consiste en 6 texturas que se aplican a un cubo para poder emular, por ejemplo, un cielo levemente nublado

- Módulo de física: Unity permite agregar a cada objeto un componente físico, es decir, agregarle características físicas, entre ellas, la capacidad de indicar que un cuerpo es rígido o poder detallar cómo se comporta el material en caso de colisiones.

La única limitación presentada por Unity para este proyecto se encuentra al utilizar la versión gratuita de Unity 3d, ya que la misma no permite que se rendericen las sombras producidas entre los objetos, lo cual quita realismo.

4.2 - Implementación de las escenas.

Para poder lograr los objetivos fijados en el capítulo de propuesta, se crearon tres escenas, una para el recorrido predefinido, otra para el recorrido libre, y por último, una escena para el menú de inicio de la aplicación que es la que permite el ingreso a las otras dos escenas. Cada escena contiene el modelo de la fortaleza logrado con Unity.

4.2.1 - Implementación del recorrido predefinido

El motor de juegos Unity 3d posee también un editor de animaciones que ha sido de utilidad para implementar un recorrido predefinido sobre el modelo 3D creado.

A esta nueva escena se le agregó un objeto cámara, al que se le debió incorporar el componente "Animation" (Component->Miscellaneous->Animation) para poder crear una animación sobre la misma.

Luego se editó la animación de la cámara (figura 4.2.1.1). Básicamente, se agregan “*key-frames*” en la línea de tiempo para los parámetros de interés en este caso los parámetros que nos interesan para el movimiento de la cámara son: los tres parámetros de ubicación en el mundo, más los tres parámetros de rotación sobre los ejes. El editor interpola linealmente una línea entre los puntos y, en el caso de elegir el parámetro “*auto-smooth*” para el *key-frame*, la interpolación ya no será lineal generando movimientos más suaves.

El editor de animación provee varios métodos para de unir los *key-frames* tales como “*free*”, “*flat*” o “*broken*”, y permite cambiar las rectas tangente a derecha y a izquierda del *key-frame*.

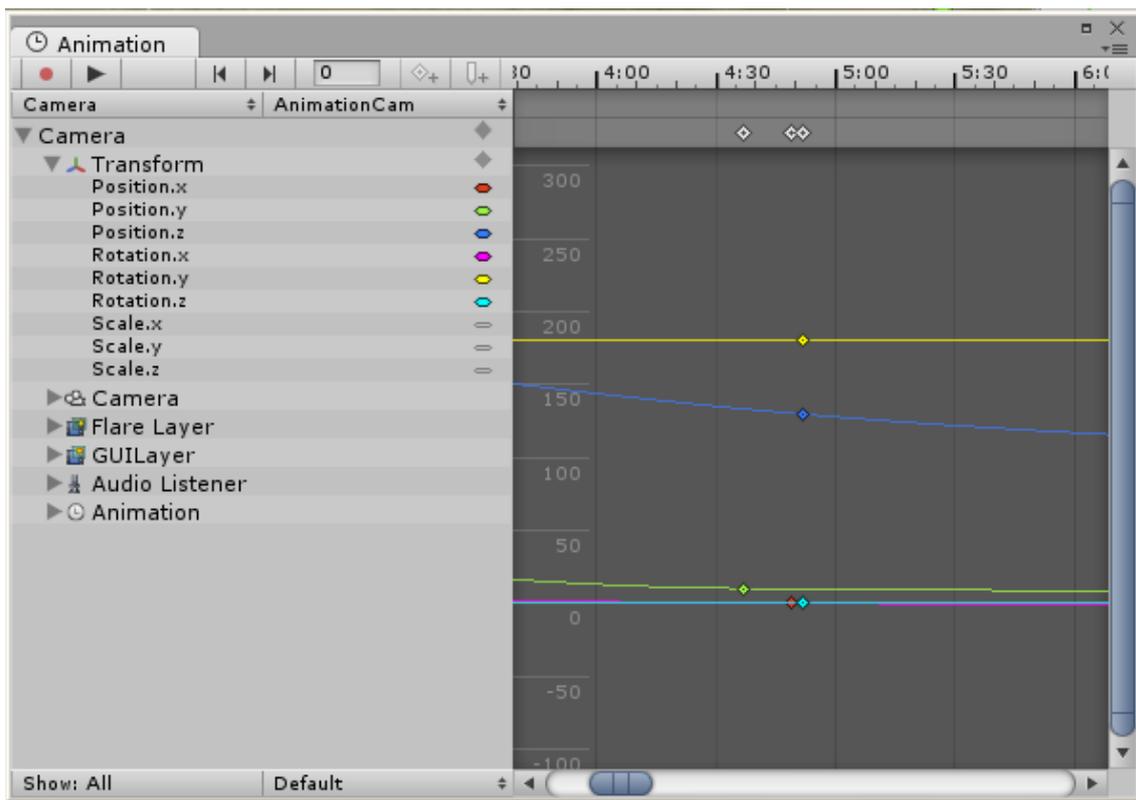


Figura 4.2.1.1 Panel de edición de animación.

4.2.2 - Implementación del recorrido libre

A la escena correspondiente al recorrido libre se le agregó un objeto “*First Person Controller*” (Figura 4.2.2.1) que permite recorrer el modelo creado en la modalidad de primera persona.

Este objeto predefinido provisto por Unity tiene un componente “*Character Controller*” que permite indicar los valores básicos del objeto, tales como la altura, el radio de impacto, etc.

Además, el objeto está provisto de tres *scripts* predefinidos que permiten la interacción con el usuario:

- *Mouse Look*: script que permite indicar la dirección de la vista a través del mouse.
- *Character Motor*: permite la interacción con el usuario a través del teclado para mover el objeto.
- *FPSInput Controller*: script que se encarga de leer la entrada y actuar en consecuencia.

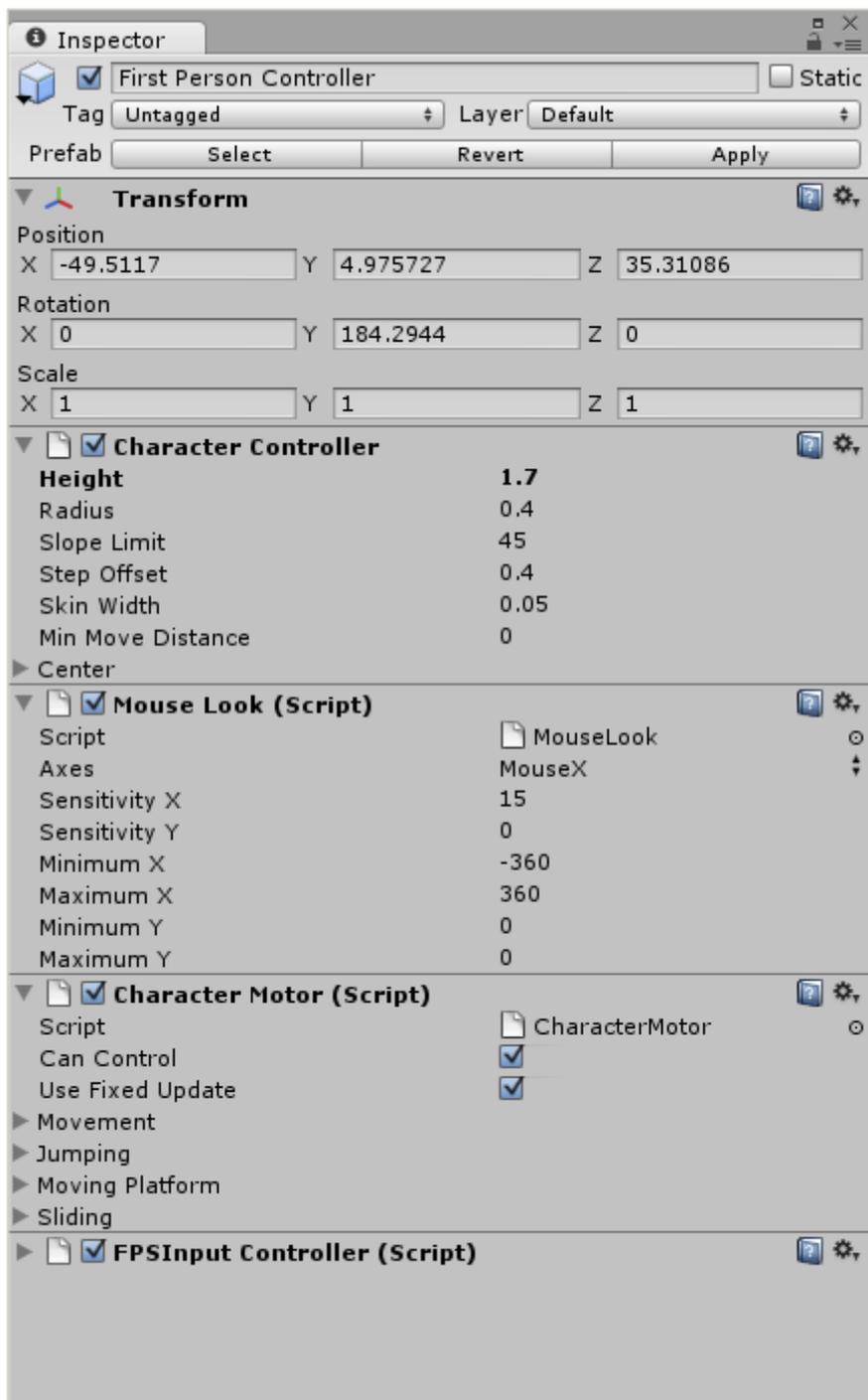


Figura 4.2.2.1 Panel de inspección sobre el objeto "First Person Controller".

4.2.3 Implementación del menú de inicio

Para esta escena se creó un objeto (Figura 2.2.3.2) modelado en Blender que simula los carteles de “referencia histórica” que pueden encontrarse en los puntos históricos de la ciudad de Bahía Blanca (Figura 2.2.3.1).

Este objeto se incluyó en la escena para incluir las opciones del menú de inicio. Básicamente, las opciones son objetos de texto “Text Mesh” para los cuales se tuvo que escribir un script (“TextControl”, Figura 4.2.3.3) que indique qué acciones se deber llevar a cabo cuando se haga click sobre alguno de ellos y para cuando el mouse esté por encima del texto para que, por medio de un cambio de color en el texto, el usuario note la opción sobre la que está posicionado.



Figura 4.2.3.1 Cartel de Referencia Histórica en Bahía Blanca.

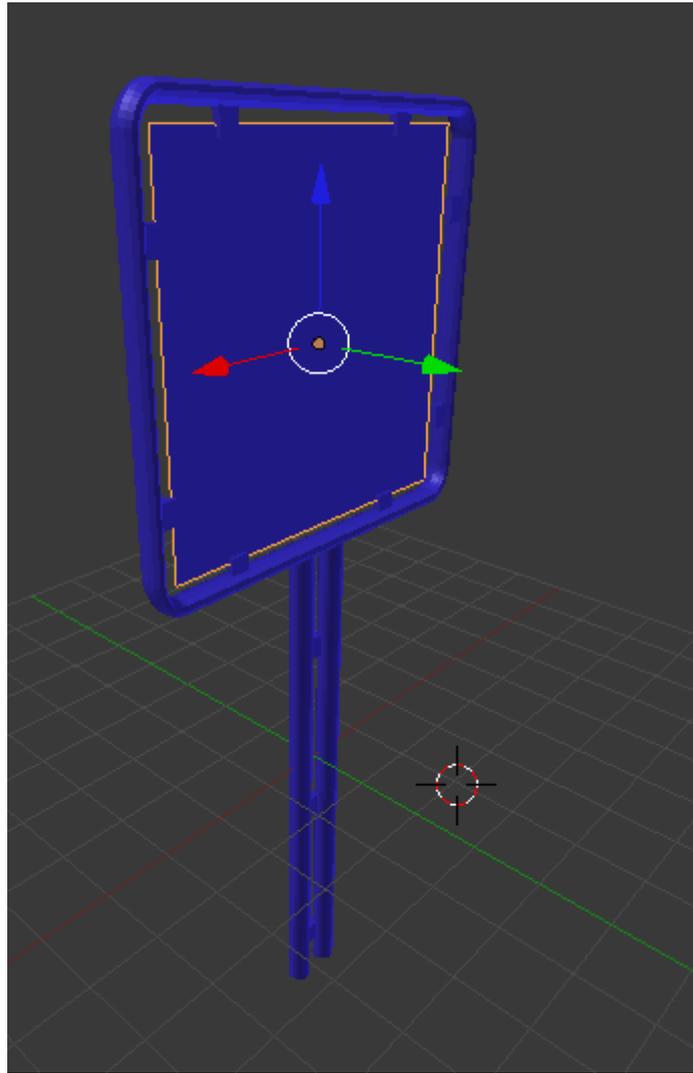


Figura 4.2.3.2 Objeto modelado en Blender simulando al cartel de referencia histórica.



Figura 4.2.3.3 Script para el control de los Text Mesh de opciones.

Conclusiones

Conclusiones

La tarea de reconstruir virtualmente sitios históricos, sobre todo aquéllos que ya no existen más o aquéllos a los que las personas no tienen acceso, es una actividad fundamental en la difusión de la cultura. En nuestro caso en particular, se reconstruyó un sitio histórico clave en la historia de la ciudad de Bahía Blanca, como lo es la Fortaleza Protectora Argentina. Esta fortaleza es considerada el comienzo de la ciudad, y sólo se contaba con una maqueta de la misma en el quinto cuerpo del ejército, y con escritos e ilustraciones de artistas. Contar con una única maqueta en toda la ciudad complica la disponibilidad y, en cierta forma, la utilidad de ésta. La existencia de una maqueta virtual permite mostrar con gran realismo y con una completa disponibilidad el ícono del nacimiento de Bahía Blanca.

Desde el comienzo del proyecto se estableció que no bastaba sólo con la construcción del modelo, sino también había que proveer un sistema de recorrido. Se investigaron distintos modelos de sitios históricos ya recreados y se concluyó que nuestra implementación debía ser llevada a cabo tanto mediante software de arquitectura como software de modelado. Para el recorrido se utilizó un motor de juegos debido a la facilidad que brindan para la interacción.

A pesar de los constantes cambios que sufrió nuestro modelo, debido a la falta de detalles en la información y de ambigüedades, algunas de las cuáles se resolvieron a medida que se desarrollaba el trabajo, se logró un modelo con un alto grado de detalle, el cual permite ser recorrido tanto libremente como siguiendo un camino guiado.

Trabajo futuro

Trabajo futuro

Para finalizar el informe se sugieren algunas mejoras y/o trabajos a futuro que podrían ser realizados a partir de este proyecto.

Sobre las mejoras al modelo de la fortaleza, se plantea el desarrollo de texturas propias para la fortaleza. Sin duda alguna, la calidad fotorealista de la fortaleza se acrecentaría enormemente con una textura exclusiva a ser aplicada en este caso. Otra mejora posible sería construir un puente más detallado con texturas nuevas.

Con respecto a los objetos esparcidos por todo el terreno (cañones, picos, etc.) son modelos prediseñados. La construcción de modelos propios con las especificaciones de la bibliografía correspondiente sería un trabajo interesante a realizar a futuro.

Respecto al recorrido se proponen 2 cosas. La primera es permitir el ingreso a las casas. Modelar las casas por adentro permitiría una inmersión casi total del usuario al mundo virtual de la fortaleza. La segunda propuesta es agregar recorridos predeterminados. Con el proyecto se entrega uno sólo, pero claro está que, al haber más recorridos, cada uno de estos podrá ser utilizado con distintos propósitos.

Agradecimientos

Queremos agradecer a la Sra. Sandra Di Luca, a la Soldado Voluntario Guillermina Schvert y al Teniente Coronel "VGM" Luis Angel Gonzalez por su ayuda y buena predisposición para/con nosotros al momento de realizar las observaciones y por el aporte de información durante nuestra visita al Batallón

Agradecemos también a Victoria Montagna y a Guillermo Gillich, estudiantes avanzados de Arquitectura e Ingeniería Civil respectivamente, por ayuda y colaboración al momento de relevamiento de software para diseño para la creación de la base de la fortaleza. A José Schneider por su ayuda en el licenciamiento. Queremos además agradecer a la Sra. Astrid Berg por la investigación y la recopilación de toda la información histórica de la fortaleza y a la Dra. Silvia Castro por la dirección de la tesis.

Por último, queríamos agradecer al Dr. Martin Larrea, por su ayuda, colaboración y dedicación durante estos meses, sin su acompañamiento hubiese sido imposible completar la implementación de la fortaleza con esta calidad en este tiempo.

Bibliografía

[1] Diario digital ElPais.com. "Zahi Hawass anuncia que cerrará la tumba de Tutankamón" -

http://www.elpais.com/articulo/cultura/Zahi/Hawass/anuncia/cerrara/tumba/Tutankamon/elpepucul/20110107elpepucul_1/Tes - 7 de enero de 2011.

[2] PRONSATO, Domingo. Estudio sobre los orígenes y consolidación de Bahía Blanca. 1956.

[3] PRONSATO, Domingo. Luces de mi tierra. 1954.

[4] PRONSATO, Domingo. Hacia otros horizontes. 1924.

[5] VIÑUALES, Graciela y ZINGONI, José. "Patrimonio urbano y arquitectónico de Bahía Blanca". 1990.

[6] Weinberg, F. y Otros. Manual de historia de Bahía Blanca - Ed. Universidad Nacional del Sur. Departamento de Ciencias Sociales – 1978.

[7] Sven Havemann Dieter Fellner. A versatile 3D Model Representation for Cultural Reconstruction – Institut für Computer Graphik, TU Braunschweig – 2002.

[8] S. El-Hakim, J. A. Beraldin, M. Picard and A. Vettore. Effective 3D Modeling of Heritage Sites - National Research Council of Canada - 2003.

[9] S. El-Hakim, J. Beraldin. Detailed 3D reconstruction of Monuments Using Multiple Techniques – National Research Council of Canada - 2002.

[10] M. Sormann, B. Reitinger, J. Bauer, A. Klaus, K. Karner. Fast and detailed 3D Reconstruction of Cultural Heritage - Institute for Computer Graphics and Vision, Graz University of Technology, Austria – 2004.

[11] M. Lo Brutto, M. G. Spera. Image-based and Range-based 3D Modelling of Archeological Cultural Heritage: THE TELAMON OF THE TEMPLE OF OLYMPIAN ZEUS IN AGRIGENTO (ITALY) - Department of Civil, Environmental and Aerospace Engineering University of Palermo, Italy – 2010.

[12] J. Roca, A. Marambio. Modelos Digitales de Nubes de Puntos de La Habana Vieja, Cuba - Universitat Politècnica de Catalunya, España – 2006.

[13] M. Cabral, M. Zuffo, S. Ghirotti, O. Belloc, L. Nomura, M. Nagamura, F. Andrade, R. Faria, L. Ferraz. An experience using X3D for Virtual Cultural Heritage - Laboratory of Integrated Systems, University of Sao Paulo, Brazil - 2007.

[14] Unity 3d Reference Manual. Documentación de componentes, Pipeline - <http://unity3d.com/support/documentation/Components/SL-RenderPipeline.html> - 25 de septiembre de 2010.

[15] Unity 3d Reference Manual. Documentación de componentes, Opciones de calidad - <http://unity3d.com/support/documentation/Components/class-QualitySettings.html> - 25 de septiembre de 2010.

[16] Blender. Game engine support limitations - <http://www.blender.org/development/release-logs/blender-248/realtime-gsl-materials/> -

[17] Wikibooks. Blender 3D: Noob to Pro/Landscape Modeling II: Texture Stenciling - http://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Landscape_Modeling_II:_Texture_Stenciling - 25 de febrero de 2011.

[18] Unity 3d Manual. Importing Objects From Blender - <http://unity3d.com/support/documentation/Manual/HOWTO-ImportObjectBlender.html> - 26 de enero de 2011.

[19] Jerry Isdale. What Is Virtual Reality? - <http://vr.isdale.com/WhatIsVR/noframes/WhatIsVR4.1-Types.html> - 1 de septiembre de 1998.

[20] Realidad Virtual. Realidad Virtual - <http://realidadvirtual6050sxxi.blogspot.com/> - 15 de diciembre de 2008.

[21] L.VII.C. Coliseo Romano - <http://sketchup.google.com/3dwarehouse/details?mid=84563dbb4d76098d251309806ab975db&ct=mdsa&prevstart=0> - 20 de noviembre de 2010.

[22] L.VII.C. Forum of Augustus - <http://sketchup.google.com/3dwarehouse/details?mid=7d40d6c8f173f516626df250d84698&ct=mdsa&prevstart=0> - 8 de mayo de 2009.

[23] Ismael Darang. The Palace - <http://forums.cgarchitect.com/67671-palace.html> - 2 de septiembre de 2011.

[24] Knut Ramstad. Tjuvholmen F1 - <http://www.vismasters.com/FORUMS/ShowPost.aspx?PostID=3428> - 11 de febrero de 2008.

[25] Blender Comunity. Lone House - <http://es.wikipedia.org/wiki/Blender> - 24 de noviembre de 2011.

[26] Softpedia. Imágenes de Blender - <http://www.softpedia.es/imagenes-Blender-16817.html>

[27] Néstor Suárez. Casa Histórica de la Independencia Argentina en 3D para Google Earth y Google Maps - <http://es.nesua.com/casaindependencia3d/> - 2011.

[28] Daniel Gonzalez. Iglesia De Los Capuchinos, Cordoba (Argentina) - <http://sketchup.google.com/3dwarehouse/details?mid=7c2704cffe178d3cfa73d437a6cdfa26&prevstart=0> - 30 de Mayo de 2010.

[29] Pablo Oneto. Congreso de la Nación Argentina - <http://sketchup.google.com/3dwarehouse/details?mid=86588e7374c662a6b436b8b346036ee6&prevstart=0> – 23 de Noviembre de 2008.

[30] Autodesk. Autocad official website- <http://usa.autodesk.com/autocad/> - 2011.

[31] Autodesk. Maya official website - <http://usa.autodesk.com/maya/> - 2011.

[32] Maxon. Cinema 4D official website - <http://www.maxon.net/> - 2011.

[33] Rhino 3D. Rhinoceros 3D official website - <http://www.rhino3d.com/> - 2010.

[34] Caligari. TrueSpace official site - <http://www.caligari.com/> - 2008.

Anexo I

Manual de instalación

Manual de instalación

Para instalar la aplicación no se requiere tener instalada ninguna librería ni software adicional. Los requerimientos mínimos para correr la instalación son los siguientes:

- Procesador: Intel® Pentium™ IV 2.3 GHz
- Placa de video con 64MB de VRAM y pixel shader.
- Memoria RAM: DDR2 667/800 Mhz, 1GB
- Espacio en disco: 550MB
- Sistemas operativos soportados: XP/Vista/7, 32b y64b

Para facilitar la instalación de la aplicación que permite el recorrido de la fortaleza se provee un instalador (“setup.exe”). Al ejecutarse, deben seguirse los pasos que se detallan a continuación:

1. La pantalla de bienvenida (Figura A_1) nos da una presentación de la instalación. Clickear “siguiente” para continuar.



Figura A_1 – Pantalla de presentación del instalador

2. La siguiente pantalla es la Licencia (Figura A_2), la cual es *Attribute-ShareAlike 3.0 Unported*. Esta licencia permite modificar, distribuir y reproducir, entre otras alternativas, siempre y cuando se haga referencia a los autores.

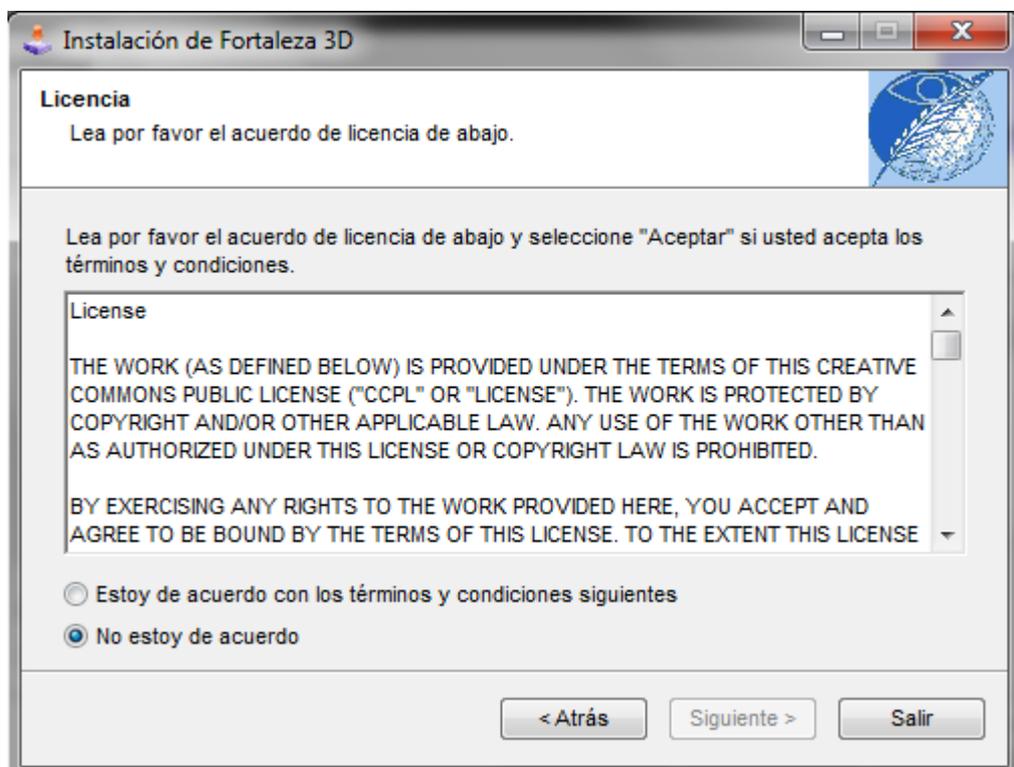


Figura A_2 – Licencia del programa

3. Ahora debemos elegir la carpeta de instalación del programa (Figura A_3). Ésta puede ser cualquiera del sistema.

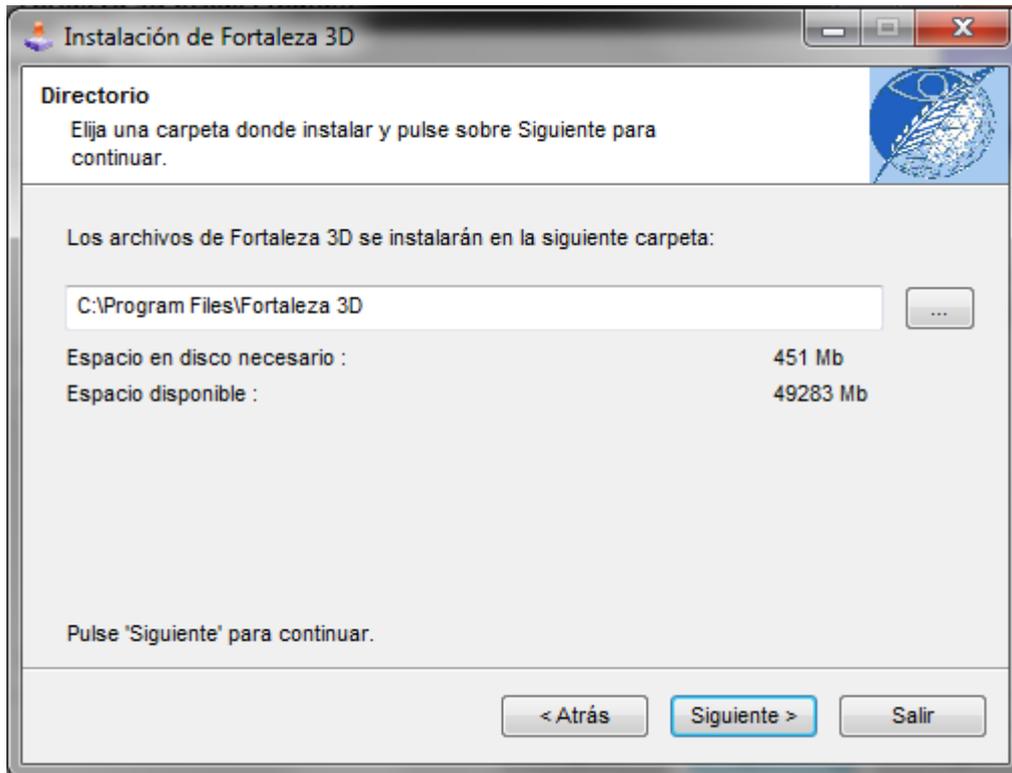


Figura A_3 – Directorio de instalación

4. Una vez realizado toda la configuración de la instalación, se predispone a instalar el programa. (Figura A_4)

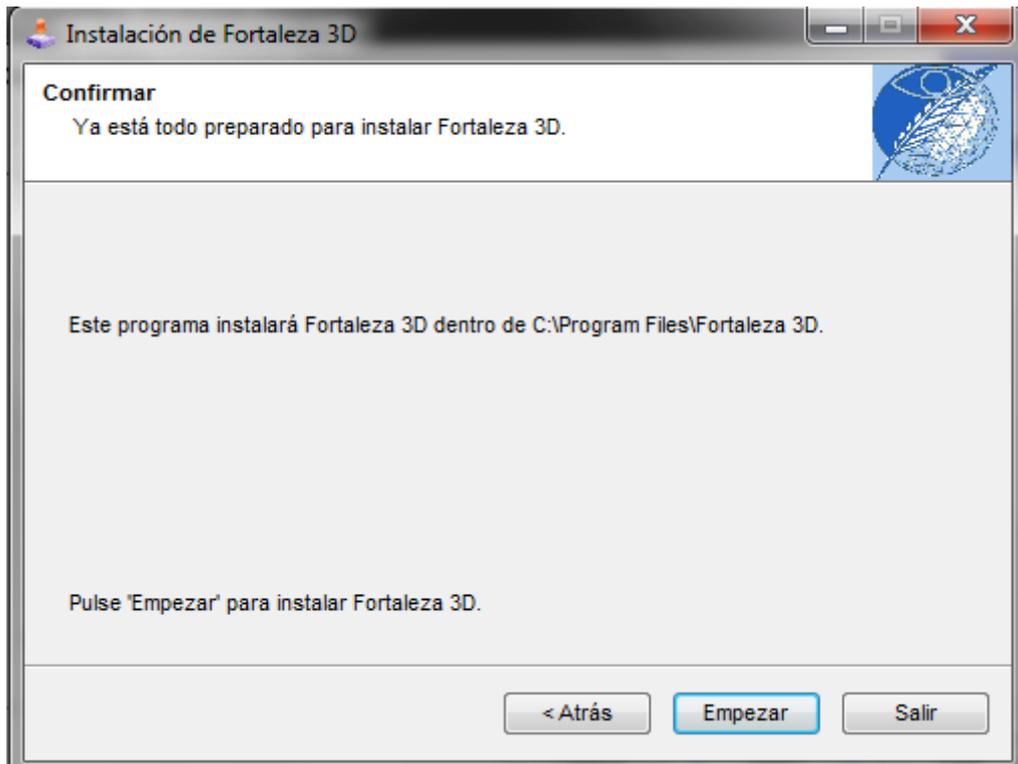


Figura A_4 – Comenzar la instalación

Anexo II

Manual de usuario

Manual de usuario

Una vez instalado el programa Fortaleza 3D ejecutarlo cliqueando sobre el ícono (ver Figura A_5) que se creó en el escritorio. En primera instancia se deben configurar los aspectos gráficos de la aplicación (ver Figura A_6). Esta opción está disponible cada vez que se ejecuta la aplicación. Entre los aspectos a configurar, encontramos las siguientes posibilidades, ordenadas ascendentemente teniendo en cuenta la calidad de los gráficos y descendentemente si tomamos como referencia la velocidad de la ejecución:

1. Fastest
2. Fast
3. Simple
4. Good
5. Beautiful
6. Fantastic



Figura A_5. Icono de la aplicación

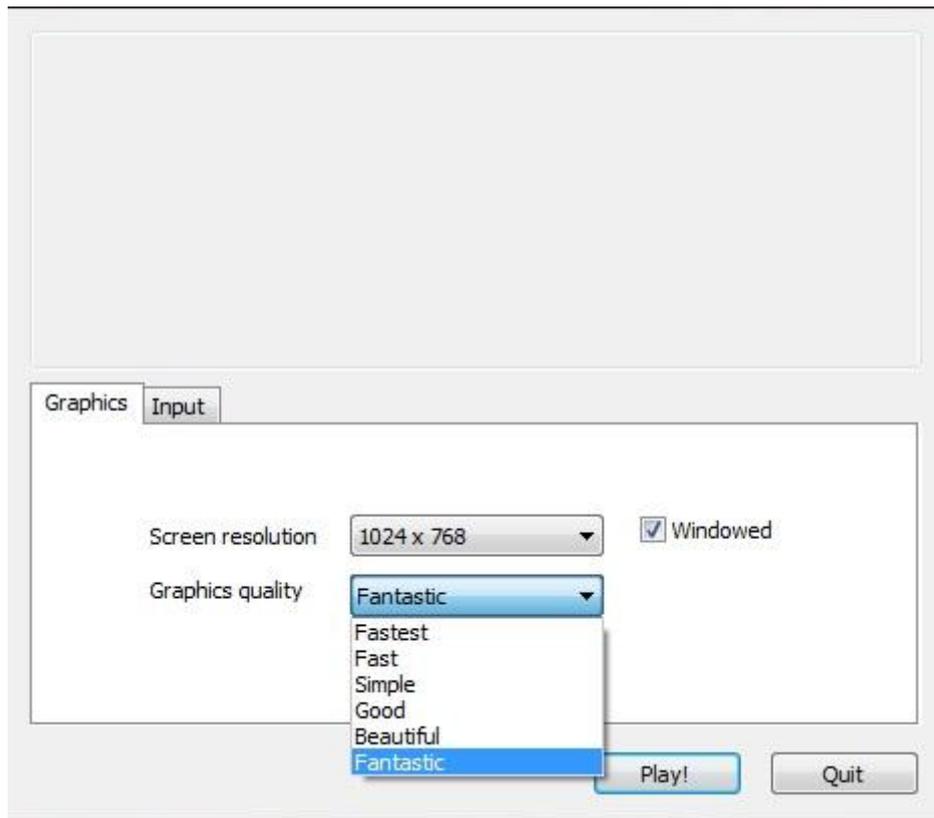


Figura A_6a. Configuración de los gráficos

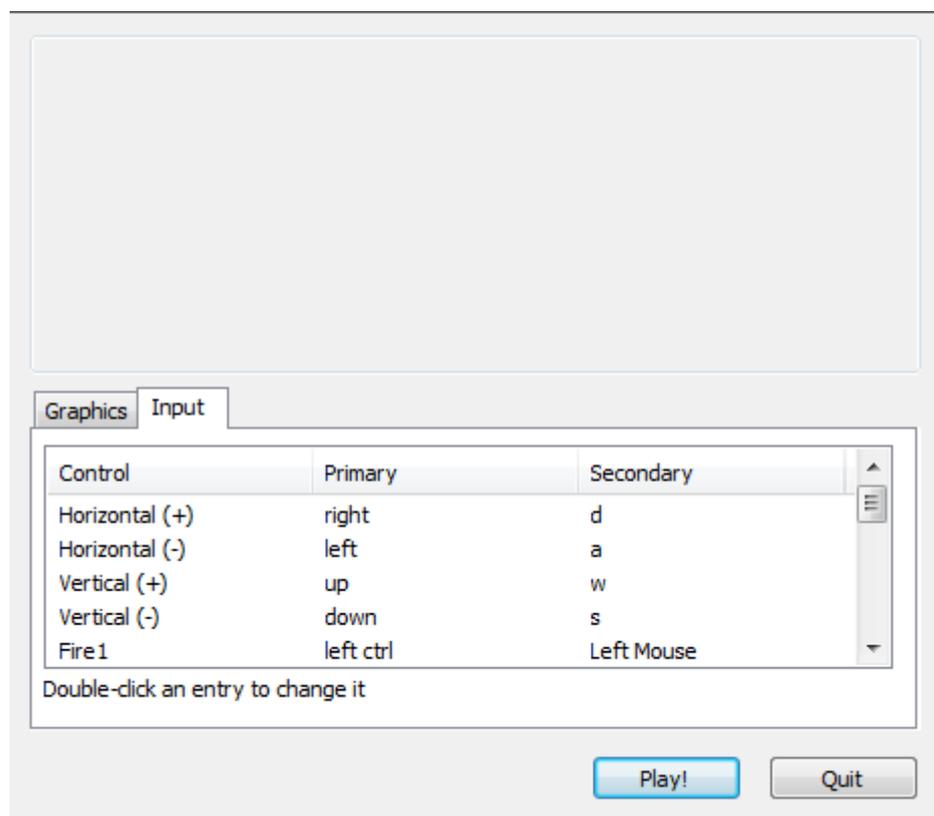


Figura A_6b. Configuración de los controles de entrada.

Si la opción “Windowed” está activada, la aplicación no se ejecutará en el modo de pantalla completa. Por defecto se interactuará con la aplicación mediante teclado y mouse; también se puede interactuar con Joystick. (Figura A_7); esto se configura en la solapa “input”.

Una vez configurado, se cargará la aplicación y aparece el menú (Figura A_8). Aquí podemos decidir si deseamos movernos libremente por la fortaleza o tomar el recorrido predefinido.



Figura A_8 – Menú de la aplicación.

Si se elige la opción “*Recorrido libre*”, las teclas para desplazarse por la fortaleza son las siguientes:

- “w” o flecha arriba: avanza,
- “s” o flecha abajo: retrocede,
- “a” o flecha izquierda: mueve a izquierda,
- “d” o flecha derecha: mueve a derecha,

Tanto la visión de la cámara como la dirección en la que se mueve el usuario por la fortaleza están dadas por el mouse.



Figura A_9 – Recorriendo la fortaleza

Si se opta por la opción del recorrido virtual, se avanza entre animación apretando la “barra espaciadora”, durante la animación simplemente mirar mientras éste se ejecuta.