

# IV. IMPLEMENTACIÓN

## IV.I. Herramientas de Diseño

En el desarrollo de un sistema de *Realidad Virtual*, particularmente en la creación de un *Paseo Virtual*, se utilizan herramientas de software que facilitan su desarrollo.

Los sistemas utilizados se aplicarán en las etapas del proceso de desarrollo del sistema:

- Se crearán los modelos que conforman el ambiente virtual del Palacio de Bellas Artes.
- Para la edición de imagen se tomarán directamente de las fotografías del edificio, de las cuales se obtendrán las texturas que envolverán a los objetos desarrollados en la etapa de modelado.

El objetivo es realizar cada uno de los elementos con un alto nivel de detallan y con ello se acerquen lo más posible a los elementos reales, produciendo en el usuario la sensación de realidad, introduciéndolo en el ambiente del paseo virtual.

### IV.I.I. Herramientas de Modelado

Las herramientas de modelado sirven para la creación de los elementos que conforman el sistema de realidad virtual, estos sistemas permitirán que se diseñe, tal cual se ha imaginado, cada uno de estos elementos, desde una simple mesa, hasta un edificio completo o los pájaros que están volando alrededor de un árbol, al que también le hemos “dado vida” con dichas herramientas.

*3DS Max*® de *Autodesk*® es un software de diseño muy poderoso, que integra modelado tridimensional, animación, *rendering* y composición, el cual les permite a los diseñadores y artistas poder realizar sus proyectos con una gran rapidez y al mismo tiempo contengan una gran calidad.<sup>163</sup>

*3DS Max 2009*® es el software utilizado en el desarrollo de todos los modelos del Paseo Virtual por el Palacio de Bellas Artes. Entre los motivos por los cuales se decidió el utilizar de este software son los siguientes:

- La disponibilidad de plugins estables para el exportado de archivos *NIF*.
- El conocimiento previo en el uso del software.
- La flexibilidad y sencillez en su manejo, en el modelado de objetos, aplicación de materiales, mapeo de texturas y asignación de mallas de colisión con facilidad, cubriendo los aspectos necesarios para la elaboración del paseo virtual.

---

<sup>163</sup> <http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=13567410>

#### IV.I.II. Herramientas para el Tratamiento de Planos y Edición de Texturas

Este tipo de software se encarga de aumentar el realismo del mundo virtual, permiten la creación, edición y tratamiento tanto de imágenes como fotografías, con ellos es posible dar una textura a los objetos creados, que se vean rugosos o lisos, para que así le parezcan más reales al usuario. Además de presentar una gran ventaja, ya que al utilizarlas para darle más realismo a los modelos es posible eliminar la cantidad de detalle a los modelos, sin perder calidad en ellos. Entre los softwares principales encontramos:

- *Photoshop®* de *Adobe®*.
- *CorelDRAW®* de *Corel Corporation®*.
- *Paint®* de *Microsoft®*.
- *GIMP®* (*GNU Image Manipulation Program®*) de *GNU General Public License®*.

*Photoshop®* en su suite CS4 de *Adobe®* es el software utilizado para el diseño, edición y tratamiento de texturas y planos. *Adobe Photoshop®* es un software dedicado a la creación de imágenes impactantes. Puede realizar selecciones complejas fácilmente, remover cualquier elemento de la imagen y rellenar espacios, además de efectuar pintura realista, crea imágenes sorprendentes en HD, permite la remoción de ruido, agregar el efecto de granulado y todas las demás herramientas de fotografía artística.

## IV.II. Principios Básicos de la Computación Gráfica

### IV.II.I. Computación Gráfica

La computación gráfica es una rama de las ciencias de la computación, se enfoca en la visualización por computadora, utilizándose equipo de cómputo para la generación de imágenes.

Este campo de la informática se puede dividir en las siguientes áreas:

- Dibujo por computadora 3D en tiempo real (*render*).
- Animación por computadora.
- Captura de video.
- Creación de video mediante render.
- Edición de efectos especiales (utilizados para cine y televisión).
- Edición de imágenes
- Modelado (Para aplicaciones médicas y de ingeniería, entre otras).

La computación gráfica es utilizada actualmente en muchas áreas de la industria, negocios, educación y entretenimiento. La lista de sus aplicaciones es enorme y está creciendo a la par de las capacidades gráficas de las computadoras permitiendo ambientes y productos más amigables.

Entre las áreas de aplicación encontramos (Figura IV.II):

- Interfaces de usuario.
- Ploteo interactivo en negocios, ciencia y tecnología.
- Cartografía.
- Medicina.
- Diseño asistido por computadora.
- Sistemas multimedia.
- Simulación y animación para visualización científica y entretenimiento.<sup>164</sup>

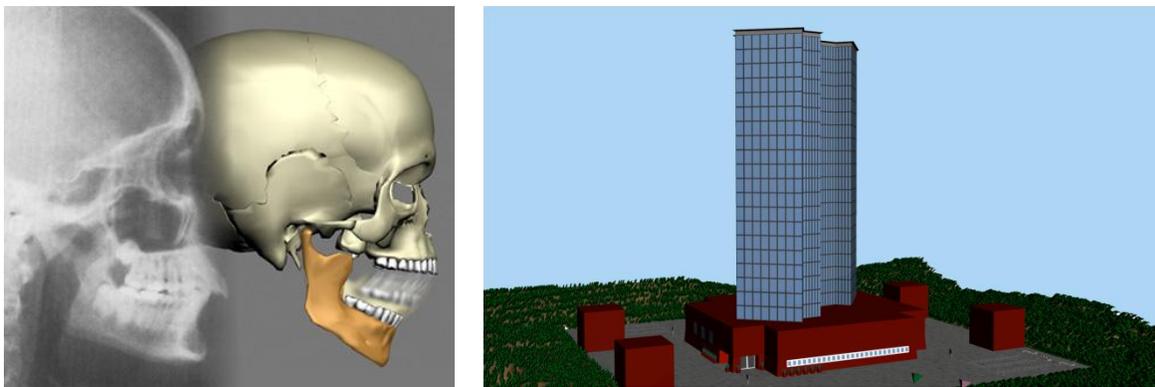


Figura IV.I Aplicaciones computación gráfica (Izquierda medicina, derecha arquitectura).

<sup>164</sup> Foley James, *Ibidem*.

### IV.II.II. Pipeline Gráfico

Son las etapas mediante las cuales una escena es almacenada, leída y dibujada en pantalla. Las etapas del *pipeline* gráfico son las siguientes (Figura IV.II):

- Modelado. En esta etapa es creado el modelo 3D que será dibujado en escena. El programa dibuja las imágenes especificando su posición en el espacio tridimensional. Consiste en la descripción de vértices, líneas y polígonos (usualmente triángulos y cuadrados), dando las coordenadas  $x$ ,  $y$ ,  $z$  de los vértices.
- Selección del punto de vista. Es usualmente utilizada para el control del punto de vista del modelo 3D. En esta etapa, se coloca una *cámara* o *posición del punto de vista* y su *dirección*. Además se determinan el *rango* y *campo de visión*. Las herramientas matemáticas que se utilizan son las *proyecciones ortográficas* y las *transformaciones en perspectiva*. Se aplica la *matriz de proyección*.
- División en perspectiva. Las etapas anteriores utilizan un método de representación de los puntos en tres dimensiones o coordenadas homogéneas, las cuales utilizan vectores con cuatro componentes para representar puntos en tres dimensiones. Esta etapa se encarga de convertir las coordenadas homogéneas en las usuales coordenadas  $x$ ,  $y$ ,  $z$ . Las coordenadas “ $x$ ” e “ $y$ ” determinan la posición del vértice en la imagen final del gráfico. La coordenada “ $z$ ” se encarga de medir la distancia al objeto. La división en perspectiva meramente consiste en la división mediante un valor “ $w$ ”.
- Dibujado en pantalla. En esta etapa la escena es dibujada en pantalla o el dispositivo de visualización que se esté utilizando. Una ventana en una computadora consiste de un arreglo rectangular de píxeles. A cada píxel se le puede colocar un color y un brillo independiente. En las aplicaciones 3D las partes que no son visibles no son dibujadas, ya que esto provocaría perder tiempo de procesamiento innecesario ya que no sería mostrado en la pantalla. En esta etapa a los píxeles se les provee de color y de la información de la profundidad y los métodos de interpolación que son utilizados para rellenar el interior de los polígonos. Es la única etapa que depende de las características físicas del dispositivo de salida.<sup>165</sup>

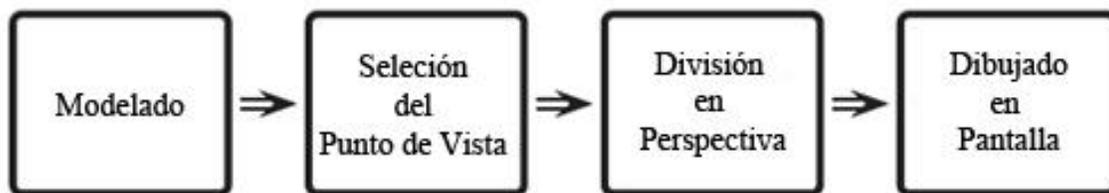


Figura IV.II Pipeline gráfico.

<sup>165</sup> Buss Samuel R., 3-D Computer Graphics: A Mathematical Introduction with OpenGL, 2003.

### IV.II.III. Sistemas de Coordenadas de Referencia

Tanto los programas de modelado como los sistemas de RV cuentan con un espacio, el cual en el sentido geométrico posee dimensiones a las cuales es posible hacerles referencia, mediante un sistema de coordenadas cartesianas ortogonales X, Y, Z (Figura IV.III).

Al situar una entidad geométrica o modelo en dichos espacios, se emplea el sistema de coordenadas universal y éste sirve como referencia para todos los demás sistemas de referencia que se utilicen. A esos otros sistemas de referencia se les denomina sistemas de coordenadas locales y pueden estar ubicados y orientados en cualquier dirección del sistema de coordenadas universal.<sup>166</sup>

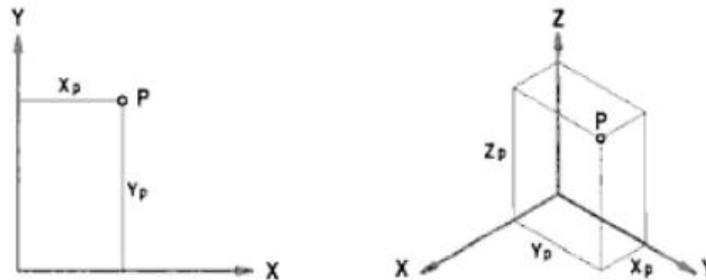


Figura IV.III Sistema de coordenadas cartesianas ortogonales.

A este tipo de sistema de coordenadas también se le denomina de coordenadas absolutas, ya que los valores que son especificados son las posiciones reales dentro del sistema de coordenadas que se esté utilizando.

Para poder dibujar una imagen en pantalla, lo primero que se necesita es seleccionar un sistema de coordenadas cartesianas adecuado, denominado sistema de coordenadas de mundo, que puede ser bidimensional o tridimensional. Posteriormente se describen los objetos de la imagen proporcionando sus especificaciones geométricas en términos de la posición dentro de las coordenadas de mundo. El elemento más pequeño que se puede dibujar dentro de un espacio 3D es un punto.<sup>167</sup>

<sup>166</sup> Alvarez Rafael, Roces Jorge, Introducción al Diseño Paramétrico: Con Autodesk Mechanical Desktop, 2005.

<sup>167</sup> Hearn Donald, Baker Pauline, Gráficos por Computadora con OpenGL, 2006.

#### IV.II.IV. Transformaciones Geométricas

Las principales funciones de las transformaciones geométricas son las que se utilizan para la traslación, la rotación y el cambio de escala.

- Traslación. Es el traslado de un objeto especificando un vector de traslación tridimensional, que determina cuánto va a ser movido en cada una de las tres dimensiones de coordenadas.

Un objeto se traslada en tres dimensiones, transformando cada una de las posiciones de coordenadas de definición para el objeto, y reconstruyendo después el objeto en la nueva localización. Para un objeto representado como un conjunto de superficies poligonales, trasladamos cada vértice de cada superficie y volvemos a mostrar las caras del polígono en las posiciones trasladadas (Figura IV.III).<sup>168</sup>

Esta operación se representa matemáticamente teniendo una posición  $\mathbf{P} = (x, y, z)$  en un espacio tridimensional, que se trasladará a la posición  $\mathbf{P}' = (x', y', z')$  añadiendo las distancias de traslación  $t_x, t_y, t_z$  a las coordenadas cartesianas de  $\mathbf{P}$ :

$$x' = x + t_x$$

$$y' = y + t_y$$

$$z' = z + t_z$$

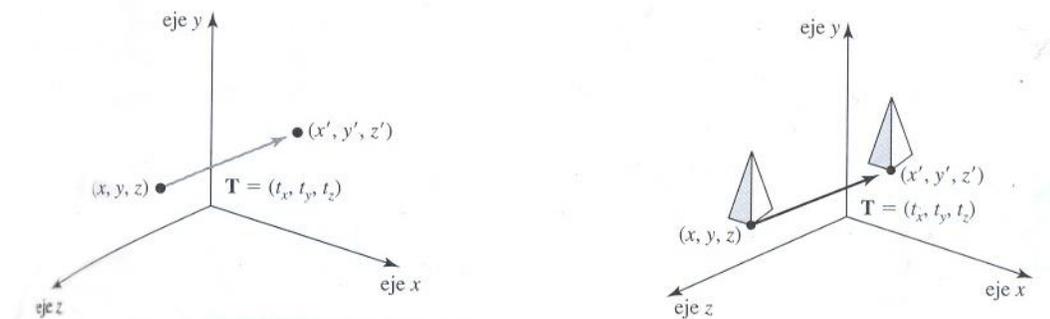


Figura IV.IV Cambio de posición de un objeto tridimensional usando un vector de traslación  $T$ .

Las posiciones de las coordenadas,  $\mathbf{P}$  y  $\mathbf{P}'$ , se representan en coordenadas homogéneas con matrices columna de cuatro elementos y el operador de traslación  $\mathbf{T}$  es una matriz de 4 por 4:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{T} * \mathbf{P}$$

<sup>168</sup> Hearn Donald, Ibídem.

- Rotación. Es posible rotar un objeto sobre cualquier eje en el espacio, pero la forma más fácil de llevar a cabo una rotación de ejes, es aquella que es paralela a los ejes de coordenadas cartesianas. También es posible usar combinaciones de rotaciones de ejes de coordenadas (utilizando las traslaciones apropiadas) para especificar una rotación sobre cualquier otra línea en el espacio. Por convenio, los ángulos de rotación positivos producen rotaciones en el sentido contrario de las agujas del reloj sobre un eje de coordenadas, utilizan un pivote que es cada uno de los ejes coordenados o paralelos a ellos para realizar rotaciones sobre los planos XY, XZ y YZ (Figura IV.III).

Las ecuaciones de **rotación para el eje z** son respectivamente:

$$\begin{aligned}x' &= x \cos\theta - y \sin\theta \\y' &= x \sin\theta + y \cos\theta \\z' &= z\end{aligned}$$

El parámetro  $\theta$  especifica el ángulo de rotación sobre el eje Z, y los valores de la coordenada Z no se pueden cambiar con esta transformación. En la forma de coordenadas homogéneas, las ecuaciones para la rotación tridimensional del eje Z son:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

De una forma más compacta se expresa:

$$\mathbf{P}' = \mathbf{R}_z(\theta) * \mathbf{P}^{169}$$

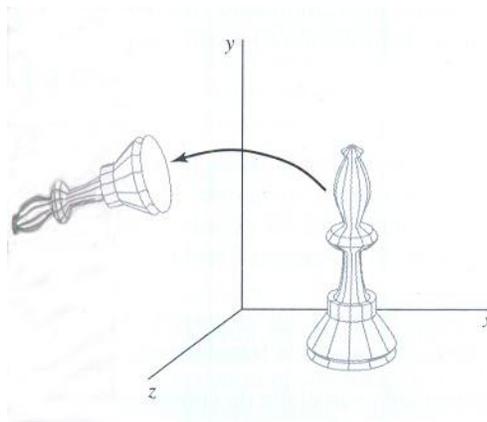


Figura IV.V Rotación de un objeto sobre el eje z.

<sup>169</sup> Hearn Donald, Ibídem.

- Escalamiento. Es la transformación que es utilizada para alterar el tamaño de un objeto. Se lleva a cabo multiplicando las posiciones de los objetos  $(x, y, z)$  por los factores de escala  $s_x$ ,  $s_y$  y  $s_z$  se les asignan valores positivos para producir las coordenadas transformadas  $(x', y', z')$  son(Figura IV.III):

$$x' = x * s_x, \quad y' = y * s_y, \quad z' = z * s_z$$

La expresión de la matriz para la transformación del cambio de escala tridimensional es:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

También puede representarse de la siguiente forma:

$$P' = S * P$$

Cambiar la escala de un objeto utilizando la matriz de escalamiento antes mencionada, cambia la posición del objeto respecto del origen de coordenadas. Un valor del parámetro superior a 1 mueve un punto alejándolo del origen en la correspondiente dirección de coordenadas y a la inversa, un parámetro menor a 1 acerca el parámetro al origen de coordenadas. Además de que si los parámetros de escalamiento no son los mismos, el objeto es deformado en la dirección correspondiente. La forma original de un objeto se conserva realizando un cambio de escala uniforme:  $s_x = s_y = s_z$ .<sup>170</sup>

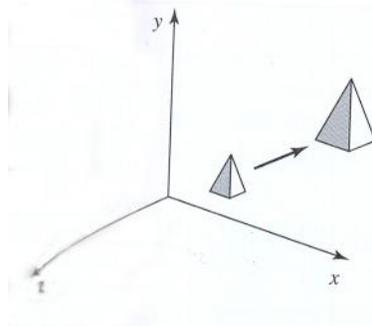


Figura IV.VI Escalamiento de un objeto al doble de su tamaño original.

<sup>170</sup> Hearn Donald, Ibídem.

### IV.III. Modelado

El modelado en computación gráfica es la creación y manipulación de la representación de un sistema. Los modelos gráficos se suelen denominar modelos geométricos, ya que las partes que componen el sistema se representan mediante entidades geométricas como son segmentos de línea, polígonos, poliedros, cilindros o esferas.<sup>171</sup>

Al modelar, la decisión que se debe tomar en cuenta, es cómo se va a construir el modelo a desarrollar y qué tipo de objeto se va a modelar. Para ello existen tres tipos de geometrías para la creación de modelos:

- Polígonos.
- Patch.
- NURBS.

Se puede utilizar cualquiera de las geometrías anteriores para crear el modelo deseado, pero esta decisión es de gran importancia, pues de ella dependerá que tan fácil o difícil será el proceso de desarrollo y cuanto tiempo será invertido en él. Cada una de ellas posee en sí misma ventajas y desventajas, las cuales serán recalçadas según sea el tipo de proyecto en el cual se trabajará, no importando el software a utilizar.

Una vez tomada la decisión de cómo construir el modelo, no es necesario quedarse solo con un tipo de modelado, si la forma del modelo lo requiere se puede utilizar un tipo de geometría como punto de partida y posteriormente pasar a otro tipo, para obtener un desarrollo “más preciso”.

Si al modelar se construyen formas orgánicas, es preferente utilizar *NURBS* o *superficies Patch*, ya que con ellas se tendrán superficies suaves con menos puntos de control, facilitando así su edición.

En cambio si se construyen superficies no orgánicas, como mesas, murallas o vehículos, es mucho más fácil desarrollarlos usando polígonos, ya que estos facilitan la creación de modelos que tienen muchas partes rectas y esquinas.<sup>172</sup>

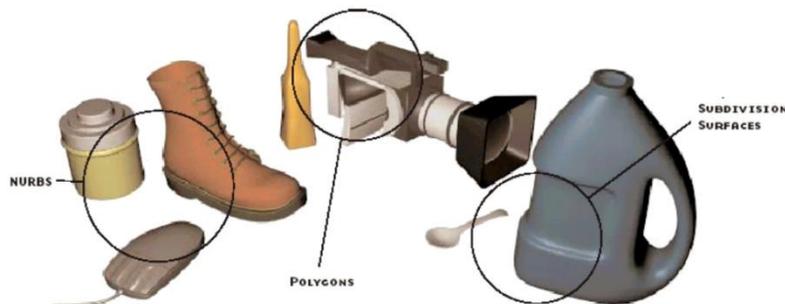


Figura IV.VII Objetos modelados por polígonos, parches y NURBS.<sup>173</sup>

<sup>171</sup> Hearn Donald, Ibídem.

<sup>172</sup> [http://student.vfs.com/~felipev/tutoriales/Manuales\\_PDF/Leccion\\_03\\_Nurbs.pdf](http://student.vfs.com/~felipev/tutoriales/Manuales_PDF/Leccion_03_Nurbs.pdf).

### IV.III.I. Modelado Poligonal

Un polígono es una forma de  $n$  lados, definida por un grupo ordenado de vértices, al conectar un vértice con otro forman bordes o edges. Un objeto poligonal es una colección de polígonos (llamadas caras poligonales o simplemente caras).

Los objetos poligonales se pueden representar por formas simples, como las primitivas poligonales, que a partir de sus formas y haciendo combinaciones entre varias de ellas, se pueden construir modelos complejos. Un objeto poligonal puede ser cerrado, abierto o conformado por capas, las cuales son piezas geométricas separadas.

También es posible construir modelos utilizando NURBS y convertirlas a geometría poligonal.

Los objetos poligonales se componen de:

- Vértices. Representan un punto en el espacio tridimensional por los ejes X, Y, Z; las conexiones entre ellos son las que determinan el resultado final del modelo poligonal.
- Bordes. Representan un lado de la cara de un polígono, se conforman por dos vértices conectados mediante una línea recta. Son muy útiles al trabajar con superficies poligonales desconectadas. Solo es necesario unir los bordes de las superficies para conectarlas.
- Caras. Se definen como la región que es conectada por la unión de bordes. Un objeto poligonal es la suma de muchas caras conectadas. Cuando el objeto es cerrado, las caras muestran su forma sólida. Son flexibles en su edición y texturizado.
- Sólidos u Objetos. Se componen por la unión de caras, que al cerrarse forman un volumen cerrado. Cada uno de los bordes en un sólido es compartido por dos caras. Un sólido se define por dentro y por fuera gracias a la dirección de sus normales.

Pueden ser creados utilizando primitivas (esferas, cilindros, conos, cubos o toroides), convirtiendo superficies abiertas a superficies cerradas, o desde objetos no sólidos utilizando operaciones poligonales.<sup>174</sup>

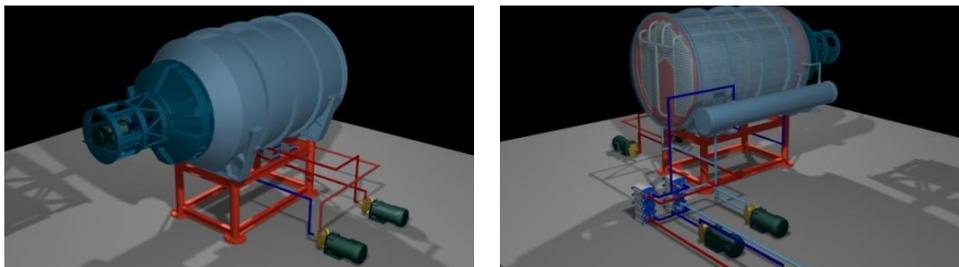


Figura IV.VIII Planta desaladora de compresión de vapor realizada mediante modelado poligonal.

<sup>173</sup> [http://student.vfs.com/~felipev/tutoriales/Manuales\\_PDF/Leccion\\_03\\_Nurbs.pdf](http://student.vfs.com/~felipev/tutoriales/Manuales_PDF/Leccion_03_Nurbs.pdf).

<sup>174</sup> Brooks Steven, *Ibidem*.

### IV.III.II. Modelado por Patch

El modelado por *Patch* (parche) basa su tecnología en el manejo y la manipulación de las curvas *bezier* y *splines*. El parche crea un “esqueleto” en forma de secciones para posteriormente recubrir todo con una superficie; esta superficie está constituida por subdivisiones de superficie de 3 ó 4 puntos llamadas parches.<sup>175</sup>

El modelado con curvas y con curvas de superficies *Patch* es diferente al modelado con líneas y polígonos. Para aproximar una curva, se necesita relativamente un gran número de segmentos de línea, mientras que la misma curva puede ser aproximada utilizando solo 3 segmentos de línea, siendo cada uno de los cuales una spline.

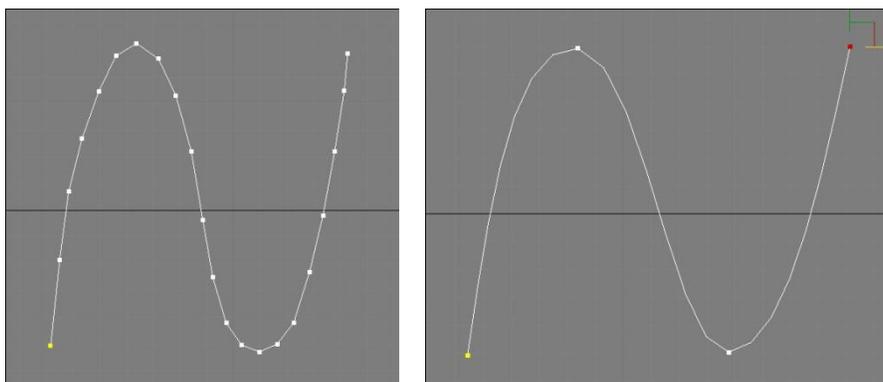


Figura IV.IX Representación de curvas usando segmentos de línea (izquierda) y spline (derecha).

El concepto anterior puede ser fácilmente aplicado a las superficies. Puesto que muchos polígonos son necesarios para poder aproximar una superficie curva como una esfera, mientras que solo 8 parches son necesarios para representar la misma esfera.

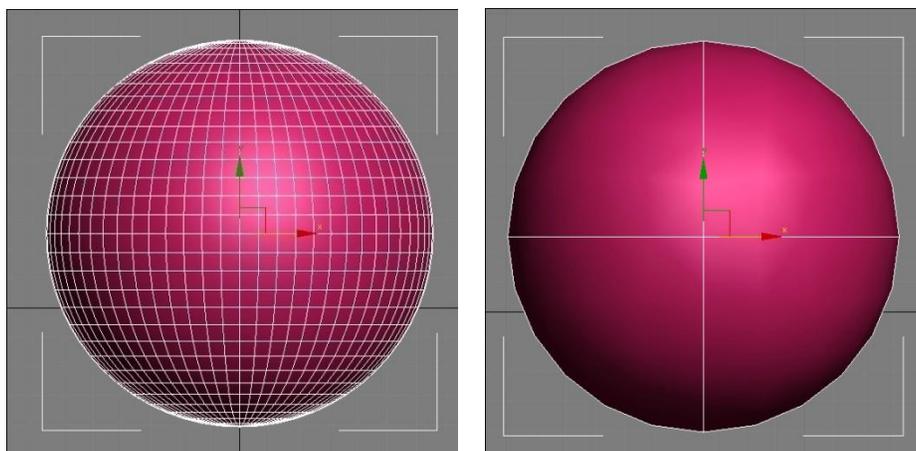


Figura IV.X Representación de una esfera usando polígonos (izquierda) y parches (derecha).

Los parches tienen una gran ventaja en el modelado orgánico. Pues, un único parche puede definir una superficie curva para la cual en el modelado poligonal sería necesario

<sup>175</sup> [http://www.loop.la/2003/docs/lo\\_ultimo\\_en\\_animacion\\_de\\_personajes.pdf](http://www.loop.la/2003/docs/lo_ultimo_en_animacion_de_personajes.pdf)

requerir de una densa colección de polígonos para simular el efecto de una curva suave.<sup>176</sup>

La aproximación es mucho mejor entre más polígonos se utilicen. Esto es verdad en el modelado poligonal, pero, no es necesariamente verdadero para el modelado por parches. Pues, un modelo poligonal de una esfera y un modelo poligonal de un rostro humano, necesitará aproximadamente el mismo número de polígonos; mientras que al dibujar una esfera utilizando parches, solo necesita 8, en cambio, al modelar un rostro humano mediante parches, éste necesitará muchos más.<sup>177</sup>

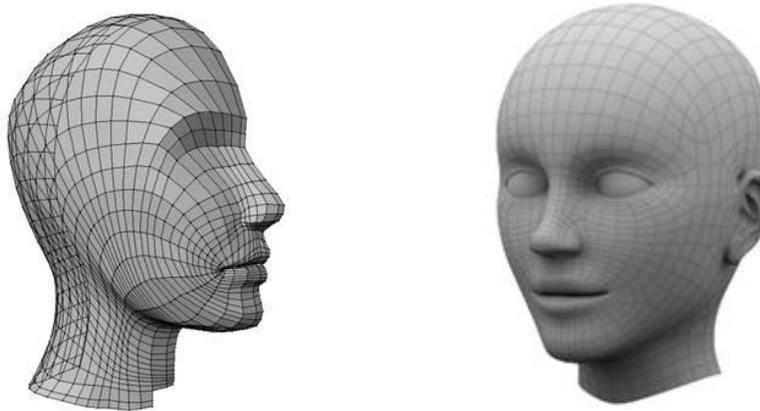


Figura IV.XI Modelo de rostro humano utilizando parches (izquierda) y polígonos (derecha).<sup>178179</sup>

Un modelo basado en parches requerirá una gran cantidad de ellos en áreas de mayor detalle, por el contrario, solo se necesitarán pocos parches en áreas de bajo detalle, no importando como sea la curva de la superficie (al contrario de los modelos basados en polígonos, los cuales necesitan un gran número de polígonos para poder aproximarse a cualquier superficie curva, independientemente del nivel de detalle).

Las operaciones que se pueden aplicar a las superficies *Patch* incluyen normalmente revolución, extrusión, barrido, entre otras.

La generación de los detalles en este tipo de modelado se realiza modificando los nodos y las tangentes de control de los vértices *bézier*, que es posible manipular en cualquiera de los ejes tridimensionales, obteniendo excelentes resultados solamente manipulando los nodos antes mencionados.

El modelado con superficies es más complejo que el modelado poligonal, pero con su ayuda es posible obtener modelos orgánicos 3D de un alto grado de dificultad, de gran calidad y con un bajo conteo de polígonos gracias a que estas superficies son generadas por las ecuaciones matemáticas correspondientes de las curvas.

<sup>176</sup> <http://www.suite101.com/content/what-is-3d-spline-modeling-and-patch-modeling-a108228>

<sup>177</sup> <http://www.jpach.com/tutorials/intro/>

<sup>178</sup> [http://www.3dlinks.com/oldsite/tutorials/GENERAL/Face\\_Modeling\\_1.cfm](http://www.3dlinks.com/oldsite/tutorials/GENERAL/Face_Modeling_1.cfm)

<sup>179</sup> <http://www.jpach.com/tutorials/intro/>

El modelado de superficies suele aplicarse a la industria automotriz, aeroespacial, procesos de manufactura, entre otras.

#### IV.III.III. Modelado por NURBS

Las *NURBS* (*Non-Rational Uniform B-Spline* o Curva Base Racional No Uniforme) son una variante o extensión de las curvas *B-Spline*, que se crean mediante vértices de control. Permite iniciar el modelado utilizando curvas que posteriormente serán utilizadas para la generación de superficies. Este proceso de desarrollo ofrece resultados muy precisos que pueden ser fácilmente controlados. Todas las superficies *NURBS* son superficies *Patch* de cuatro lados.

Una curva *NURBS* se define por sus vértices de control, los cuales controlan la forma de la curva. La curva resultante se aproxima por dichos vértices. Las curvas *NURBS* también cuentan con otros vértices como los puntos de edición (*edit points*) y *hulls* que nos ayudan a comprender y modelar la forma de la curva.

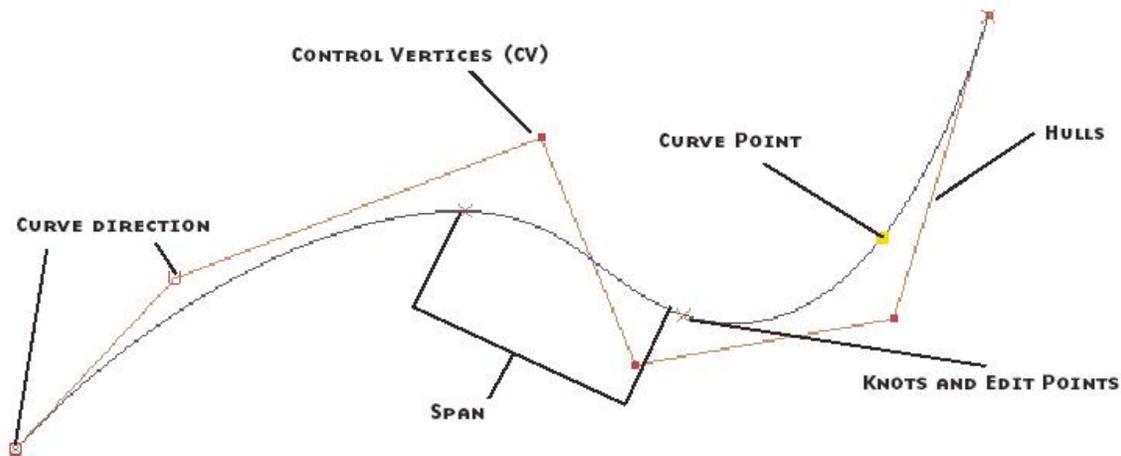


Figura IV.XII Elementos Curva NURB.<sup>180</sup>

Los elementos de una curva NURB son los siguientes:

- *Curve Direction*: El inicio de la curva está definido por un cuadrado y una pequeña “u” que es la que define la dirección de la curva.
- *Control Vertices (CV)*: Son los puntos que definen la forma de la curva.
- *Span*: Es una sección de la curva. Cada *span* es una pequeña curva que está relacionada con la siguiente *span*.
- *Curve Point*: Representa la medida de “U” a todo lo largo de la curva. El valor de “U” es dependiente del parámetro de la curva.
- *Hulls*: Línea recta que conecta los CVs con la curva. Al seleccionar un *hull* estamos seleccionando los CVs asociados.
- *Edit Point*: Puntos que definen el inicio y el término de los *spans*.

<sup>180</sup> [http://student.vfs.com/~felipev/tutoriales/Manuales\\_PDF/Leccion\\_03\\_Nurbs.pdf](http://student.vfs.com/~felipev/tutoriales/Manuales_PDF/Leccion_03_Nurbs.pdf).

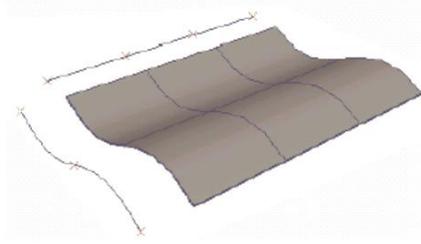


Figura IV.XIII Superficie construida mediante NURBS.

Una de las grandes ventajas de modelar utilizando superficies, es que éstas se definen matemáticamente, lo que significa que siempre se ven perfectas. La expresión general para evaluar una superficie *NURBS* es:

$$\vec{s}(u, v) = \frac{\sum_i \sum_j \omega_{ij} \vec{P}_{ij} N_i^m(u) N_j^n(v)}{\sum_i \sum_j \omega_{ij} N_i^m(u) N_j^n(v)}$$

Figura IV.XIV Ecuación general de evaluación de una superficie *NURBS*.

Haciendo una comparación con una superficie poligonal, esta vendría a ser como comparar un dibujo de vectorial, compuesto de trazos *bézier*, siempre perfectos con una imagen de mapa de bits, compuesta por pixeles, al ampliarla muestra los pixeles que la conforman.

El modelado *NURBS* resulta ideal para el diseño industrial. Esto trazando las curvas que conforman los bordes de un objeto partiendo de planos y fotografías, resultar indiscutiblemente práctico el utilizar este tipo de sistemas de subdivisión.

#### IV.III.IV. Modelado de Objetos

Para el modelado de objetos existe una gran variedad de técnicas y formas de hacerlo, ya que éstas dependen de varios factores como el modelo que se va a realizar, el ambiente para el cual será necesario el objeto, la cantidad de detalle requerida, el uso que se hará del modelo, se utilizará en una animación o en procesamiento en tiempo real, entre otras. Algunas consideraciones que deben tomarse en cuenta en el desarrollo de modelos, son las siguientes:

- Para el desarrollo del objeto es muy recomendable hacer un esquema o boceto del mismo, con ello se tendrá una base a la cual seguir, además de tener siempre en mente la meta a alcanzar, la cantidad de detalle y elementos que conformaran al objeto. No es imposible hacerlo sin una guía, pero un simple dibujo es una gran ayuda a la hora de crear el objeto.
- El modelo debe de ser estético, debe de verse bien, dado que en caso contrario el modelo no tendrá el efecto para el cual fue creado. Se tomará en cuenta características como: edad, material, forma, entre otras.

- El anime y el manga son buenas fuentes de inspiración en el desarrollo de modelos, particularmente de personajes.<sup>181</sup>
- El diseño debe de ser único, tener su propia identidad, deben tener un rasgo *único* que los caracterice, haberse planeado correctamente, tener un enfoque preciso de lo que se desea modelar.
- Pensando en consideraciones técnicas la más importante y que se debe de tener siempre en cuenta al desarrollar el objeto es la restricción *de la cuenta de polígonos*, pues impactará directamente en el diseño. Generalmente un personaje principal de un juego en tiempo real tiene en promedio 20,000 polígonos, 10,000 polígonos para la cabeza y 10,000 polígonos para el cuerpo, actualmente esto se maneja para PC's y el PlayStation 3®, presentando en este último a personajes como Snake del juego *Metal Gear Solid* con un conteo de polígonos de 60,000 a 80,000 y a Kratos del juego *God of War* con 20,000 polígonos.<sup>182</sup>

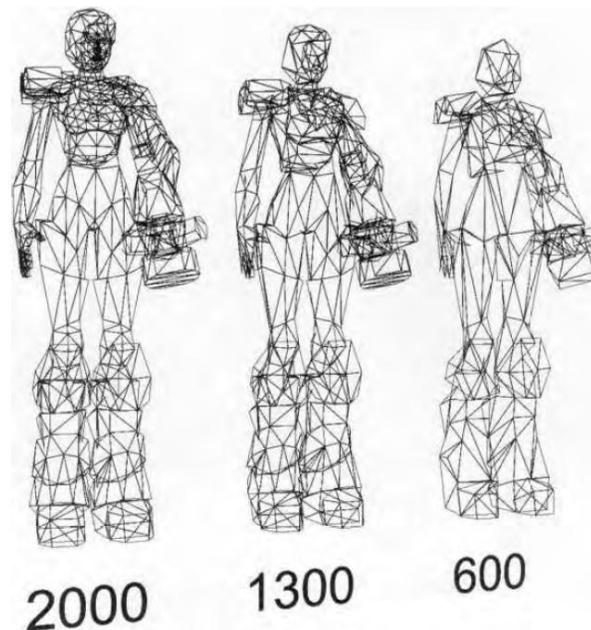


Figura IV.XV Nivel de Detalle de un Modelo.<sup>183</sup>

- Se debe de tener muy presente también nivel de detalle, el cual le dirá al motor del juego cuantos polígonos deberá de renderear dependiendo de la distancia que se esté manejando. Con lo anterior, dependiendo del modelo, este manejará un modelo con un alto, medio y bajo nivel de detalle.
- La forma en que se implementará el objeto en el motor y el sistema de huesos para la animación del objetos (si es que los requiere), son otras consideraciones técnicas a tomar en cuenta, pues, estas se basan principalmente en las animaciones que manejarán los objetos, dentro de un video juego.

<sup>181</sup> Steed Paul, *Animating Real – Time game Characters*, 2003.

<sup>182</sup> *Ibidem*.

<sup>183</sup> *Ibidem*.

- Al crear el modelo de un personaje es muy importante tener una referencia para su creación. Para ello se realizarán los esquemas necesarios para su desarrollo (la vista frontal, lateral y la superior). Por lo menos es necesario tener dos vistas cualesquiera para poder desarrollar un modelo muy preciso de cualquier objeto o personaje en 3D. La referencia debe de ser creada pensando y describiendo el carácter del personaje.



Figura IV.XVI Vistas frontal y lateral del boceto de un personaje y modelo resultante.<sup>184</sup>

- Lo más importante del aspecto de un modelo es su forma, ya que en ella recaen la masa y la identidad. Éstas son definidas por la propia distribución de vértices, bordes y caras. La eficiencia es de extrema importancia, cada vértice cuenta.
- Al construir un modelo, es recomendable que sea construido parte por parte; para poder tener un modelo de gran calidad, es necesario concentrarse en la parte que se está desarrollando en el momento, hacer que esta se vea bien y principalmente que le guste al modelador, antes de pasar a otra parte o área del modelo. No importa que las partes se guarden separadas, ya que al armar la versión final pueden ser combinadas. Lo importante es tener la satisfacción del modelo creado y que éste cumpla con las expectativas establecidas.
- “No es necesario reinventar la rueda”, si es posible reutilizar un objeto o una parte de un objeto creado con anterioridad, puede ser reutilizado, lo que significará menos horas de trabajo de modelado y más horas creativas que serán invertidas en el modelo.
- Es recomendable trabajar en una sola ventana, y no esforzarse por ver detalles al tener las cuatro vistas (perspectiva, frontal, lateral y superior) abiertas, con una tecla es posible cambiar entre las diferentes vistas.

<sup>184</sup> Steed Paul, *Ibidem*.

- Trabajar con simetrías es muy útil, si los modelos permiten esa facilidad, es recomendable enfocarse en la mitad a desarrollar, aplicar la *simetría* y soldar el modelo, evitando problemas y ahorrando trabajo de desarrollo.
- El programa que se utilice para modelar debe de ser lo más cercano al modelador como sea posible, debe sentirse como propio, entre más comfortable sea para el modelador y haga que éste realice su trabajo un mayor gusto, fluidez y exactitud, será mucho mejor. Desde el simple hecho de manejar los comandos abreviados, hasta un ambiente de trabajo más agradable fomentará un mayor desempeño.
- No temer a experimentar y la utilizar métodos alternativos de modelado, de esa forma se adquiere experiencia y técnicas propias de modelado. Esas técnicas se aplicarán a uno u otro modelo teniendo en mente que lo más importante es desarrollar un excelente trabajo.<sup>185</sup>

#### IV.IV. Texturizado.

Es un método utilizado para variar las propiedades de una superficie punto por punto con la finalidad de darle a la superficie una apariencia detallada que no está presente en la geometría de la superficie.<sup>186</sup>

Para lograrlo se crea una imagen bidimensional, que posteriormente es aplicada utilizando un método llamado *mapeo de texturas*, en el cual literalmente la textura se envuelve sobre el objeto tridimensional.

Si por ejemplo se toma una fotografía de una losa de mármol, posteriormente la fotografía es aplicada utilizando mapeo de texturas a un modelo, con la cual se le pedirá al procesador que ajuste las imágenes planas a las superficies curvas y planas, de tal forma que se obtenga como resultado un objeto que dé la impresión que se encuentra hecho de mármol.

Otro método utilizado para la simulación de superficies es la aplicación de texturas procedurales, éste método consiste en realizar un pequeño programa que dibujará y calculará matemáticamente la textura, que será aplicada al objeto tridimensional, en lugar de utilizar imágenes de mapas de bits. Por lo general estas imágenes ya han sido programadas y se encuentran en los programas de modelado más comunes que existen en el mercado, solo es necesario aplicarla al modelo deseado.<sup>187</sup>

---

<sup>185</sup> Steed Paul, *Ibidem*.

<sup>186</sup> Ebert David, Musgrave Kenton, Peachey Darwin, Perlin Ken, Worley Steven, *Texturing & Modeling A Procedural Approach*, 2003.

<sup>187</sup> *Ibidem*.



Figura IV.XVII Modelo de una tetera a la cual se le ha aplicado una textura de mármol.<sup>188</sup>

#### IV.IV.I. Materiales

Los materiales se encargan de especificar las propiedades ópticas de los modelos a los que son aplicados como son: la esencia del color y la opacidad, que cantidad de luz difusa que va a dispersar, que tan reflectivo será, entre otras. Al generar un *render*, la imagen del objeto es calculada tomando en cuenta sus propiedades ópticas, al mismo tiempo que calcula su color y transparencia.

Es un error muy común el confundir los materiales con texturas, por ese motivo se hace resaltar la diferencia que existe entre ellos. Como se mencionó anteriormente las texturas se encargan de simular en un modelo cualquier tipo de superficie, mientras que los materiales están asociados con los objetos en escena, son paquetes de información que dicen como reaccionaran los objetos a la luz.<sup>189</sup>

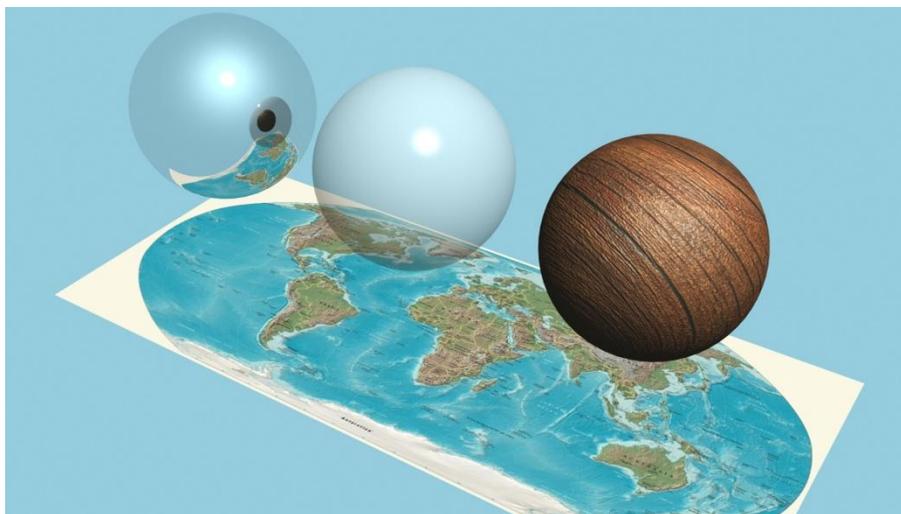


Figura IV.XVIII Materiales de metal reflejante, cristal y luz ambarina para madera.

<sup>188</sup> Raghavachary Saty, *Rendering for Beginners: Image synthesis using RenderMan*, 2005.

<sup>189</sup> *Ibíd.*

#### IV.IV.II. Texturas

Son aquellas que te entregan una experiencia visual por sí mismas. Una textura puede describir una gran variedad de características de las superficies, como pueden ser: terrenos, plantas, minerales, pelajes y pieles. La meta del cómputo gráfico desde su creación ha sido el poder el alcanzar el realismo visual de nuestro mundo; actualmente las texturas son los elementos más comúnmente empleados para la obtención de imágenes sintetizadas.<sup>190</sup>

Las texturas proveen a la superficie una interesante y detallada apariencia, en lugar de una simple y aburrida superficie de un único color uniforme. Con ellas se obtiene un realismo visual alto a un bajo costo en esfuerzo humano y en tiempo de cómputo.<sup>191</sup>

Las texturas pueden ser obtenidas de una gran variedad de fuentes, desde un dibujo a mano alzada, imágenes o incluso fotografías escaneadas. Pueden ser uniformes como un muro de ladrillos o un mosaico, o irregulares como el granulado de la madera o el mármol.



Figura IV.XIX Texturas de madera, mármol, mezclilla, estuco y pared de ladrillo como fondo.

Es difícil trabajar con las imágenes para generar texturas, pues de acuerdo con los ejemplos anteriores, los dibujos a mano son estéticamente agradables pero es muy difícil lograr fotorealismo a partir de ellos. Las fotografías, dan un excelente aspecto de realidad, pero generalmente tienen un tamaño inadecuado o no permiten la repetición si son usadas directamente en el mapeado.

<sup>190</sup> <http://graphics.stanford.edu/projects/texture>

<sup>191</sup> Ebert David, Ibídem.

Las texturas pueden ser divididas en dos grupos:

- Texturas No Procedurales. Son las antes mencionadas, imágenes que son mapeadas, en las que cada punto en una superficie 3D de un parche paramétrico corresponde a un punto particular 2D  $(u,v)$  en el espacio paramétrico. La correspondencia 2D a 3D implica que una textura de imagen bidimensional puede ser fácilmente mapeada en una superficie 3D, como lo hizo Catmull en 1974. Los parámetros  $(u,v)$  de cualquier punto en un parche pueden ser usados para computar la correspondiente localización del pixel en la textura de imagen.<sup>192</sup>
- Texturas Procedurales. Es una textura que es creada sintéticamente, es generada usando un programa o modelo en lugar de ser una imagen digitalizada o pintada. Por lo general se utiliza para la representación de elementos naturales como madera, mármol, granito, piedra, entre otros.



Figura IV.XX Texturas estuco, a la izquierda mediante una textura procedural incrementando el *bump* al máximo, izquierda textura no procedural usando un *bump* mínimo entregando mayor realismo.

---

<sup>192</sup> Ebert David, *Ibíd.*

### IV.IV.III. Mapeo de Texturas

Es una técnica para la síntesis de imágenes en la cual una textura de imagen es mapeada en la superficie tridimensional de una escena, como lo haría un papel tapiz al pegarse a una pared.<sup>193</sup>



Figura IV.XXI Imagen del mundo mapeada sobre un plano.

La ventaja del mapeo de texturas se presenta cuando la escena se encuentra cargada de una gran cantidad de detalle produciendo un incremento considerable en el tiempo de dibujado de la escena, en cambio al utilizar el método de mapeado de texturas, este por si mismo produce un modesto incremento en el detalle y si es aplicado profesionalmente produce resultados formidables usando texturas de alta definición sobre modelos de un bajo conteo de polígonos.

El mapeo de texturas no afecta a la eliminación de caras ocultas, pero sí agrega un tiempo adicional al cálculo del proceso de *shading*. Esta técnica se aplica fácilmente a superficies curvas.

El mapeo de texturas puede ser usado para definir muchos parámetros de la superficie además del color. Como son:

- Perturbaciones de la superficie de los vectores normales (*Bump Mapping*).
- Modulación de la opacidad de una superficie translúcida (*Transparency Mapping*).
- Variación del brillo de la superficie (*Specularity Mapping*).
- Modelar la distribución de la luz incidente de cualquier dirección (*Illumination Mapping*).<sup>194</sup>

<sup>193</sup> Heckbert Paul S., *Fundamentals Of Texture Mapping and Image Warping*, 1989.

<sup>194</sup> Heckbert Paul S., *Ibidem*.



Figura IV.XXII Imagen del mundo aplicando un mapa de normales a la izquierda y un mapa de opacidad a la derecha.

De todas las variedades de mapeo antes mencionadas, el mapeo geométrico es fundamental. El mapeo bidimensional es usado para definir la parametrización de una superficie y describir la transformación entre el sistema de coordenadas de textura y el sistema de coordenadas de la pantalla.

Al renderear una superficie texturizada se necesita realizar una conversión de las coordenadas de la imagen de textura para poder así dibujar la imagen en pantalla.

#### IV.IV.IV. Imágenes Deformadas

Es la acción de distorsionar una imagen fuente en una imagen destino de acuerdo a un mapeo entre un espacio fuente  $(u,v)$  y un espacio destino  $(x,y)$ . El mapeo es especificado usualmente por las funciones  $x(u,v)$  y  $y(u,v)$ . Se utiliza el término *deformación* o *warp* en lugar de sus sinónimos: *distorsión* y *transformación* por ser consistente y específico: Deformación específicamente sugiere el mapeo del dominio de una imagen, mientras transformación puede significar un mapeo en el rango de la imagen también.<sup>195</sup>

Entre otras aplicaciones de la deformación de imágenes se encuentran, en procesamiento de imágenes, para la corrección de distorsiones, para propósitos artísticos y efectos especiales, se aplica en programas interactivos de dibujo.

<sup>195</sup> Heckbert Paul S., Ibídem.

## IV.V. Iluminación.

En el mundo físico la luz interactúa con una gran variedad de materiales de una forma fascinante, enriqueciendo la diversidad visual. Desde la transparencia de una hoja vista a contra luz, entregando un brillo de un adorno de oro, hasta el simple reflejo especular de la luz sobre el cromo de un automóvil, nuestro ambiente está lleno de estos fenómenos ópticos.

Todos los días cada uno de nosotros utilizamos materiales naturales y artificiales, sin poner alguna atención en como la luz interactúa con ellos. De esa misma forma encontramos una gran diversidad de tipos de fuentes de luz, que van desde la luz solar, que es la que ilumina todo nuestro ambiente, iluminaciones interiores, como las que se encuentran dentro de nuestras casas, los faroles de los autos, hasta las linternas que funcionan a base de LED's.<sup>196</sup>

Al trabajar con fotorealismo se requieren dos elementos principalmente: representaciones precisas de las propiedades de la superficie y una buena descripción física de los elementos de iluminación en la escena. Estos efectos de iluminación incluyen la reflexión de la luz, la transparencia, las texturas de las superficies y las sombras.



Figura IV.XXIII Render de una escena aplicando diversos tipos de en la iluminación interior.

El modelar, usando efectos de iluminación, es un proceso complejo en el que interviene la física y la psicología. Los efectos de iluminación se describen mediante modelos que toman en cuenta la interacción de la energía electromagnética con las superficies de los objetos de la escena. Una vez que los rayos alcanzan nuestros ojos, se ponen en marcha procesos de percepción, que hace que veamos los que realmente vemos.

Los modelos físicos de iluminación toman en cuenta varios factores como son: las propiedades de los materiales, las posiciones de los objetos en relación con las fuentes de iluminación y con otros objetos y las características de las fuentes luminosas. Los objetos se componen de materiales opacos, o pueden tener un cierto grado de transparencia. Las superficies pueden ser brillantes o mates, además de exhibir ciertos patrones de textura superficial. Pueden utilizarse fuentes luminosas de diversas formas, colores y tamaños, además de las posiciones diversas que van a ocupar para la

<sup>196</sup> Raghavachary Saty, Ibidem.

iluminación de la escena. Todo lo anterior se utiliza para calcular la intensidad de la luz proyectada desde una posición concreta de la superficie en una dirección de visualización especificada.<sup>197</sup>

Para poder realizar todos esos cálculos en el cómputo gráfico se utilizan aproximaciones de las leyes físicas que describen los efectos de iluminación de una superficie. Para reducir los cálculos se utilizan modelos simplificados de cálculos de fotometría, esto debido a que el tiempo de rendereado con la interacción de las luces es muy alto y es necesaria una simplificación.

#### IV.V.I. Modelos de Iluminación

Son también llamados modelos de sombreado y son utilizados para calcular el color en cada posición iluminada en la superficie de un objeto. Los modelos más precisos de iluminación se encargan de calcular los resultados de las interacciones entre la energía radiante incidente y la composición del material del modelo.

El modelo básico de iluminación se divide en:

- Luz Ambiental. Utiliza una luz de fondo general para iluminar la escena con un cierto nivel de brillo, produciendo una luz ambiental uniforme que es igual para todos los objetos y que aproxima las reflexiones difusas globales a diversas superficies iluminadas. Ese nivel de luz ambiental de la escena dependerá de un parámetro de intensidad  $I_a$ . Esto quiere decir que la escena se iluminará por una luz de fondo y que sus reflexiones, son independientes de la dirección de visualización y de la orientación espacial de las superficies. Sin embargo la cantidad de luz ambiental incidente que se reflejará, dependerá de las propiedades ópticas de sus materiales, que determinarán que cantidad de luz se va a reflejar y qué cantidad de luz será absorbida.<sup>198</sup>
- Luz o Reflexión Difusa. La luz incidente se dispersa con igual intensidad en todas las direcciones, independientemente de la posición de visualización. A estas se les denomina reflectores difusos ideales. Si a todas las superficies las tratamos de esta forma, es necesario definir un parámetro  $k_d$  para poder determinar la fracción de la luz incidente que será necesario dispersar en forma de reflexiones difusas. Por lo tanto la reflexión difusa se da mediante una constante cuyo valor es igual a la intensidad de la luz incidente multiplicada por un coeficiente de reflexión difusa. Si una superficie es altamente reflejante, se asignará un valor a la constante cercano a 1.0, produciendo una superficie muy brillante, en la que la intensidad de la luz reflejada será casi igual a la luz incidente. Mientras que si el valor del coeficiente tiende a 0.0, se tendrá una superficie que absorberá prácticamente toda la luz incidente.<sup>199</sup>

---

<sup>197</sup> Hearn Donald, *Ibidem*.

<sup>198</sup> *Ibidem*.

<sup>199</sup> Hearn Donald, *Ibidem*.

- Luz o Reflexión Especular. El reflejo o reflexión especular es brillo que puede ser observado en las superficies brillantes, es el resultado de una reflexión total o casi total, de la luz incidente en una región concentrada alrededor del ángulo de reflexión especular. El ángulo de reflexión especular es igual al ángulo de la luz incidente, midiendo ambos ángulos en lados opuestos del vector unitario  $\mathbf{N}$  normal a la superficie.

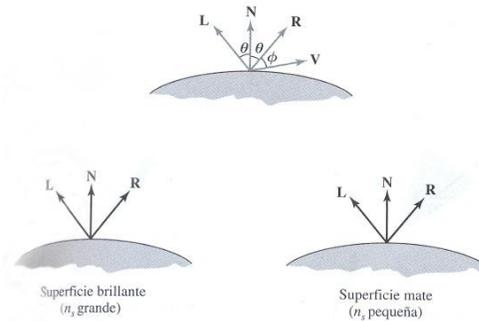


Figura IV.XXIV El ángulo de reflexión especular es igual al ángulo de incidencia  $\theta$  (Arriba), modelado de reflexiones especulares a partir del parámetro  $n_s$ .<sup>200</sup>

$\mathbf{R}$  representa el vector unitario en la dirección de la reflexión especular ideal,  $\mathbf{L}$  es el vector unitario dirigido hacia la fuente luminosa puntual y  $\mathbf{V}$  es el vector unitario que apunta hacia el observador desde la posición seleccionada de la superficie. El ángulo  $\Phi$  es el ángulo de visualización relativo a la dirección de reflexión especular  $\mathbf{R}$ . Para un reflector ideal (un espejo perfecto), la luz incidente se refleja solo en la dirección de reflexión especular y solo sería posible ver la luz reflejada cuando los vectores  $\mathbf{V}$  y  $\mathbf{R}$  coincidieran ( $\Phi = 0$ ).

Todos los objetos que no son reflectores ideales exhiben reflexiones especulares en un rango finito de posiciones de visualización en torno al vector  $\mathbf{R}$ . Las superficies brillantes tienen un rango de reflexión especular más estrecho, mientras que las superficies mates tienen un rango de reflexión más amplio.<sup>201</sup>

#### IV.V.II. Tipos de Luces

En 3D MAX existen dos tipos las luces estándar y las luces fotométricas. A continuación se mencionaran cada una de las luces que las componen y sus características:

Luces Estándar (*Standard Lights*). Con este tipo de luces se puede generar cualquier tipo de iluminación, desde la luz de una lámpara hasta la luz del sol. A cada luz es posible asignársele un nombre, para su posterior identificación, al igual que editar los parámetros que se presentan en la casilla del panel parámetros generales.

<sup>200</sup> Ibídem.

<sup>201</sup> Ibídem.

Entre los parámetros que pueden ser modificados encontramos:

- Tipo de Luz (*Light Type*). Permite realizar el cambio de luz seleccionando el tipo de luz deseado.
- Encendido de la Luz (*ON*). Enciende y apaga la luz del foco.
- Encendido de las Sombras (*OM*). De forma predeterminada las sombras no están activas en MAX, para evitar el tiempo de las muestras del render. Este modificador sirve para activar y desactivar las sombras.
- Extrude. Con este parámetro los objetos seleccionados son excluidos de la recepción de la luz seleccionada y no proyectan sombras.
- Tipos de Sombras. En MAX existen varios tipos de sombras. Como son:
  - Sombras de Trazado de Rayos (*Ray Traced Shadow*). Este tipo de sombras son generadas dibujando el recorrido de los rayos trazados desde la fuente de luz, son ideales para objetos transparentes.
  - Sombras de Trazado de Rayos Avanzadas (*Advanced Ray Traced Shadow*). Son similares a las sombras *Ray Traced*, diferenciándose en que ellas permiten una mayor manipulación de parámetros de la sombra.
  - Sombras por área (*Area Shadow*). Permite el dibujado de sombras en un área determinada.
  - Mapas de Sombra Mental Ray (*Mental Ray Shadow Map*). Se generan utilizando el renderizador *Mental Ray*, genera sombras mediante el algoritmo *Mental Ray*, utiliza un motor que calcula desde sombras simples hasta segmentadas, compilación fotónica emitida por luces conocido como *Photon Map* y posee su propio sistema de aceleración.
  - Mapa de Sombras (*Shadow Map*). Este tipo de sombras un mapa de bits, se genera una visualización en la fase previa a la renderización de la escena, no admite transparencias; las sombras son creadas probando si los pixeles son visibles desde las fuente de luz, comparando su buffer de profundidad con la vista de la fuente de luz, almacenándose en forma de texturas.
- Uso de Parametros Globales (*Use Global Parameters*). Al activar esta casilla todas las luces de la escena tienen el mismo tipo de sombreado.
- Intensidad de la Luz (*Multiplier*). Es la cantidad de energía radiante que proyecta la luz.
- Color. Determina el color con que la luz es proyectada.
- Tipo de Disminución (*Decay*). Determina como la luz se va a perder de acuerdo con la distancia en que esta se proyecta.
- Atenuación Cercana y Atenuación Lejana (*Near Attenuation & Far Attenuation*). Determina la distancia en que la luz se empieza difuminar, especificando un punto de inicio y un punto final, así como si será visible en los visores.

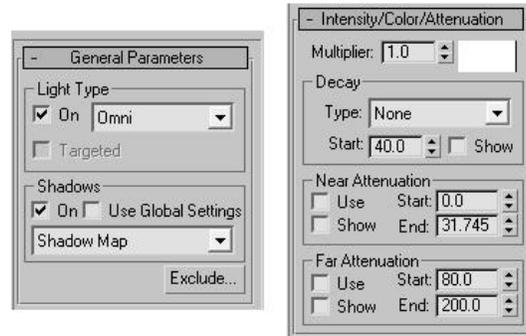


Figura IV.XXV Parámetros generales e intensidad de la luz en 3DS MAX®.

Entre las luces estándar existen ocho tipos de luces, entre las que encontramos:

- Luz Omnidireccional (*Omni*). Este tipo de luz dispersa sus rayos en todas direcciones, es ideal para utilizarlas como relleno en la iluminación de la escena.
- Foco Libre (*Free Spot*). Dispersa sus rayos en una sola dirección, es la típica luz *Spot* la cual conforme se aleja del punto de inicio el cono va aumentando su tamaño.
- Foco con Objetivo (*Target Spot*). Genera sus rayos en una sola dirección, contando con el beneficio del objetivo, con el cual es posible dirigir el rayo de luz, sin modificar la posición de la fuente.
- Direccional Libre (*Free Direct*). Genera rayos de luz en un solo sentido, haciendo que todos los rayos sean paralelos, simulando un cilindro o tambor de rayos; este tipo de luz trabaja de forma similar a como lo hacen los rayos solares, por lo que es idónea para su simulación.
- Direccional con Objetivo (*Target Direct*). Funciona de la misma forma que la luz Direccional Libre, contando con la ventaja de tener un objetivo.
- Luz Cenital (*Skylight*). Produce una cúpula imaginaria en la escena, haciendo que sus rayos reboten a lo largo de las superficies de la escena; es ideal para la simulación de exteriores. Este tipo de luz funciona a través de trazado de rayos.
- Luz Omnidireccional Mental Ray (*Mr Area Omni*). Funciona de forma idéntica a la Luz Omnidireccional, con la diferencia a utilizar el motor *Mental Ray* al dibujar la imagen.
- Foco de Área Mental Ray (*Mr Area Spot*). Trabaja de igual manera que el Foco con Objetivo, aprovechando el motor de render del *Mental Ray*.

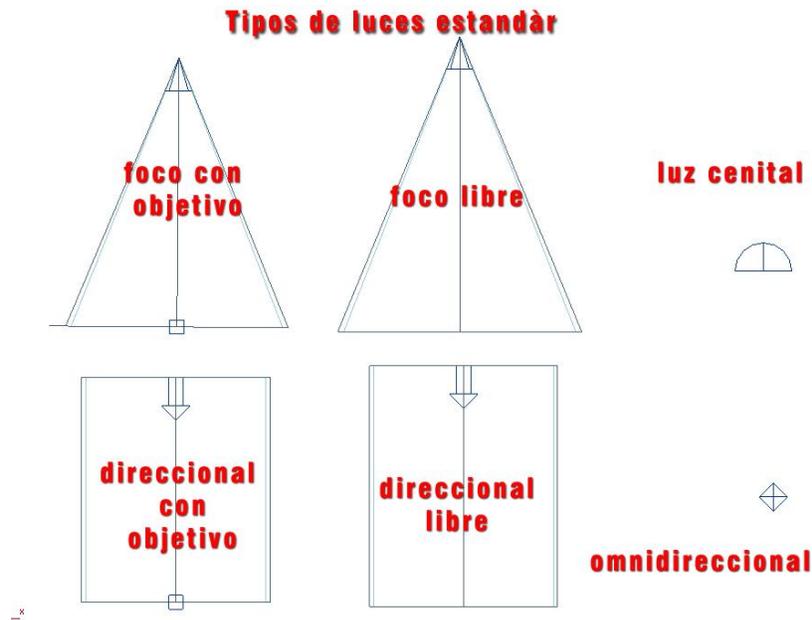


Figura IV.XXVI tipos de luces estándar en 3DS MAX®.

El otro tipo de luces que maneja MAX son las Luces Fotométricas, las cuales son más precisas en sus cálculos que las luces estándar, permitiendo que los trabajos que realicen con ellas sean más exactos y obteniendo una mejor simulación del mundo real. Debido a que se utilizan los datos proporcionados por empresas encargadas de la fabricación de luces. Este tipo de luces por defecto utilizan atenuación inversa al cuadrado. Al igual que las anteriores es posible cambiar sus parámetros, como son el encendido, si proyectarán sombras o no lo harán, entre otros.

Otros parámetros importantes en las Luces Fotométricas son los siguientes:

- Distribución (*Distribution*). Se encarga de determinar la forma en que los rayos de luz son emitidos.
- Isotrópica (*Isotropic*). Distribuye la luz de igual forma en todas las direcciones.
- Reflector (*Spotlight*). Distribuye la luz hacia una dirección de forma similar a un spot o cono.
- Red (*Web*). Esta distribución es tridimensional; los fabricantes son los encargados de distribuir este tipo de luces en archivos: IES, LTLI y CIBSE.
- Difusa (*Diffuse*). Distribuye la luz desde una superficie.
- Luz. Despliega una lista de lámparas como: *white, daylight fluorescent, halogen*, entre otras.
- Kelvin. Este determina el color de la luz por medio del ajuste de los contadores de temperatura del color.
- Filtro de Color (*Filter Color*). Produce un filtro frente a la luz, con la finalidad de determinar un color.
- LM. Mide la cantidad de la luz en flujo luminoso, medida en lumen.
- CD. Se encarga de medir la intensidad luminosa, medida en candelas.

- LX. Mide la cantidad de luz que llega a una superficie a una determinada distancia, medida en lux.
- Longitud (*Lenght*). Determina el largo del área donde se proyectará la luz.
- Ancho (*Witdh*). Determina el ancho del área donde se proyectará la luz.

Los diferentes tipos de *Luces Fotométricas* son las siguientes:

- Luz Puntual con Objetivo (*Target Point Light*). Este tipo de luz emite rayos desde una forma geométrica definida hasta un objetivo.
- Luz Puntual Libre (*Free Point Light*). Es igual a la Luz Puntual con Objetivo, solo que esta carece de objetivo.
- Luz Lineal con Objetivo (*Target Lineal Light*). Este tipo de luz emite sus rayos desde la línea que conforma la luz hasta llegar al objetivo.
- Luz Lineal Libre (*Free Linear Light*). Esta luz emite rayos desde la línea en donde inicia la luz, esta luz carece de objetivo.
- Luz de Área con Objetivo (*Target Area Light*). Esta luz genera un área rectangular, que es donde se generan los rayos, hasta el objetivo de la misma.
- Luz de Área Libre (*Free Area Light*). Esta luz es similar a la anterior, crea un área rectangular desde donde se generan los rayos de luz, pero carece de objetivo.
- Luz Solar IES (*IES [Illuminating Engineering Society] Sun Light*). Esta luz es ideal para la simulación de la luz solar, al añadir este tipo de luz es necesario activar el control de exposición logarítmico y seleccionar la opción de luz diurna y exterior, así como introducir en la escala física de 90,000, los parámetros se localizan en efectos de entorno control de exposición.
- Luz de Cielo IES (*IES Skylight*). Es ideal en la simulación de la luz cenital (la luz dispersa en la atmosfera).

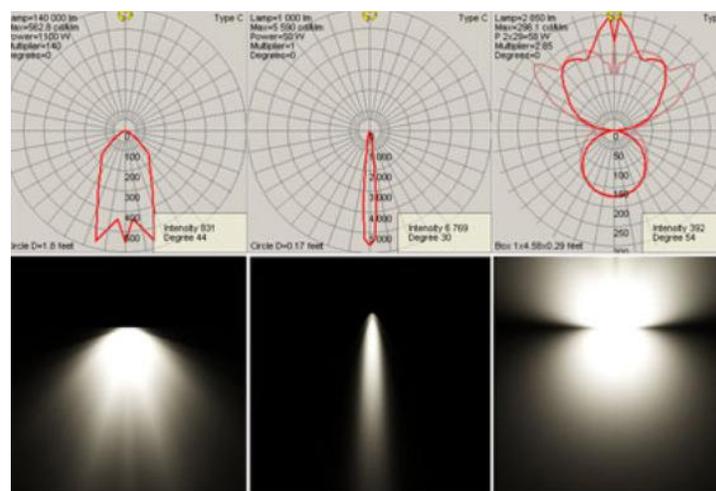


Figura IV.XXVII Diagramas de la representación de diferentes luces IES con sus renders.

### IV.V.III. Sombras

Las sombras son secciones de una superficie que no son visibles desde la fuente luminosa que está situada encima de la superficie.<sup>202</sup>

Para poder determinar esas áreas que no se encuentran iluminadas por la luz, se utilizan métodos de la detección de la visibilidad. Con la posición de visualización situada en la ubicación de una fuente luminosa es posible determinar que secciones de una superficie en una escena no son visibles. Estas áreas se denominan áreas de sombra. Una vez determinadas las áreas de sombra todas las fuentes luminosas, las sombras pueden tratarse como patrones superficiales y almacenarse en matrices de memoria de patrones.

### IV.V.IV. Modelos de Sombreado

De acuerdo a la sección anterior es posible sombrear cualquier superficie calculando las normales de cualquier punto visible de una superficie y aplicarle la iluminación deseada a cada punto. Desafortunadamente esto es muy costoso en tiempos de procesamiento y es el principal motivo por el cual existen modelos de sombreado.

Sombreado Constante (*Constant Shading*). Es el modelo de sombreado más simple para un polígono. Es también conocido como sombreado de caras o sombreado plano. Esta aproximación aplica un modelo de iluminación para determinar un único valor de intensidad que será utilizado para sombrear por completo un polígono.<sup>203</sup>

En esencia se simplifica el valor de la ecuación de iluminación para cada polígono y manteniendo el valor en el polígono usando solo una normal por cara para la reconstrucción de la sombra, produciendo un efecto de *facetado*.

Este método es válido si son válidos los siguientes supuestos:

- La fuente de iluminación está en el infinito, luego  $N \cdot L$  es constante para el polígono.
- El observador está en el infinito, por tanto  $N \cdot V$  es constante para cada polígono.
- El objeto es un poliedro y no una aproximación de un objeto con una superficie curva.

Sombreado por Interpolación (*Interpolated Shading*). Es una alternativa al evaluar la ecuación de iluminación, en la cual la información del sombreado es interpolada linealmente a través de un triángulo con valores determinados por sus vértices. Gouraud generalizó esta técnica al aplicarla a polígonos. Este método es particularmente sencillo para un algoritmo de salida de interpretado o render que interpola el valor de la profundidad  $z$  en un lapso de valores calculados para los valores del intervalo  $z$ .<sup>204</sup>

---

<sup>202</sup> Hearn Donald, Ibídem.

<sup>203</sup> Foley James D., Ibídem.

<sup>204</sup> Ibídem.

Esta interpolación es físicamente correcta, asumiendo que un polígono es plano, pero este tipo de sombreado es solo una aproximación del modelo de iluminación en cada punto del polígono.

Sombreado de Gouraud (*Gouraud Shading*). Llamado también sombreado de interpolación de la intensidad o sombreado de interpolación de color, elimina las discontinuidades de intensidad.<sup>205</sup>



Figura IV.XXVIII Escena dibujada aplicando *Sombreado de Gouraud*.<sup>206</sup>

Extiende el concepto de sombreado por interpolación aplicándolo a polígonos individuales interpolando valores de iluminación a los vértices de los polígonos, que tomaran en cuenta la superficie que será aproximada. El proceso del sombreado de Gouraud requiere que las normales sean conocidas para cada vértice de la malla poligonal.

El siguiente paso en el sombreado de Gouraud es identificar las intensidades de los vértices utilizando las normales de los vértices con el modelo de iluminación deseado. Finalmente cada polígono es sombreado por una interpolación lineal de las intensidades de los vértices a lo largo de cada borde.

---

<sup>205</sup> Foley James D., *Ibídem*.

<sup>206</sup> *Ibídem*.

Sombreado de Phong (*Phong Shading*). Es conocido como el sombreado de interpolación de los vectores normales, interpola la superficie del vector normal, en lugar de intensidad. La interpolación ocurre a lo largo de un polígono en su línea de exploración, entre las normales iniciales y finales.<sup>207</sup>



Figura IV.XXIX Escena dibujada aplicando *Sombreado de Phong*.<sup>208</sup>

Las normales se interpolan ellas mismas a lo largo de los bordes de las normales de los vértices que son computados como en el sombreado de Gouraud.



Figura IV.XXX Escena dibujada aplicando *Sombreado de Phong y Sombreado de Gouraud*.<sup>209</sup>

#### IV.V.V. Trazado de Rayos

Es la generalización del procedimiento de básico de proyección de rayos, no se limita a localizar la superficie visible desde cada posición de pixel, sino que el rayo correspondiente al pixel continua rebotando a lo largo de la escena, con la finalidad de recopilar las diversas contribuciones de intensidad. Proporcionando una técnica simple y potente de representación para obtener efectos globales de reflexión y transmisión.

<sup>207</sup> Foley James D., *Ibídem*.

<sup>208</sup> *Ibídem*.

<sup>209</sup> *Ibídem*.

El algoritmo básico de trazado de rayos detecta las superficies visibles, identifica las áreas de sombra, permite representar efectos de transparencia, genera vistas de proyección en perspectiva y admite múltiples efectos de iluminación con múltiples fuentes luminosas.

Las imágenes de escenas generadas utilizando esta técnica, pueden llegar a ser enormemente realistas, especialmente cuando la escena está conformada por un número considerable de objetos brillantes, la desventaja es que el tiempo que llevará visualizar el render utilizando este algoritmo será considerable.<sup>210</sup>



Figura IV.XXXI Escena dibujada utilizando el método de Ray Trace.

---

<sup>210</sup> Hearn Donald, *Ibíd.*

## IV.VI. Cámaras.

Son aquellas que nos permiten observar una escena desde cualquier posición. Con ellas se pueden generar imágenes fijas (*renders*), recorridos a lo largo de una escena (en una animación) y en tiempo real son aquellas que nos permiten observar, pasear y conocer el mundo que con el que estamos interactuando como usuarios.

La cámara es una de las herramientas más poderosas para el creador 3D, ya que al utilizarlas con destreza afecta tanto a los modelos como a las animaciones; el ángulo de la cámara, la distancia focal, el campo de visión, así como su propio movimiento, son aspectos que influyen con importancia en cualquier animación.

### IV.VI.I. Proyecciones

Los paquetes gráficos que se encuentran en el mercado son capaces de soportar proyecciones paralelas como proyecciones en perspectiva.

En una proyección paralela, las coordenadas se transfieren al plano de visualización a partir de una serie de líneas paralelas. Este tipo de proyección preserva las proporciones relativas de los objetos, es el método que se utiliza por lo general en diseño por computadora para poder generar imágenes a escala de los objetos tridimensionales.<sup>211</sup>

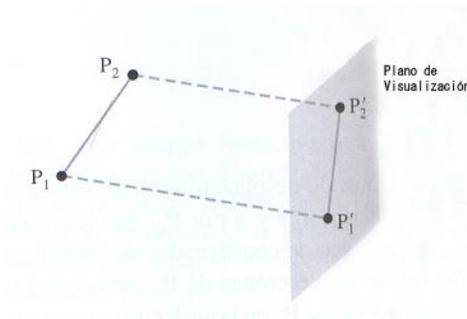


Figura IV.XXXII Proyección paralela de un segmento de recta sobre el plano de visualización.<sup>212</sup>

Todas las líneas paralelas de una escena se muestran así cuando se les contempla mediante una proyección paralela.

Existen dos métodos para poder obtener una vista en una proyección paralela de un objeto:

- Proyectar el objeto utilizando líneas que sean perpendiculares al plano de visualización.
- Utilizar un ángulo oblicuo con respecto al plano de visualización.

<sup>211</sup> Hearn Donald, *Ibídem*.

<sup>212</sup> *Ibídem*.

Para una *proyección en perspectiva*, las posiciones de los objetos se transforman a las coordenadas de proyección según una serie de líneas que convergen en un punto situado detrás del plano de visualización.<sup>213</sup>

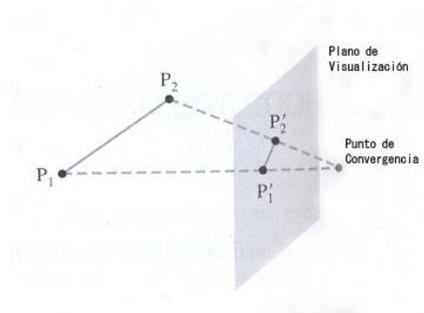


Figura IV.XXXIII Proyección en perspectiva de un segmento de recta sobre el plano de visualización.<sup>214</sup>

A diferencia de una proyección paralela, la proyección en perspectiva no conserva las proporciones relativas de los objetos. Mientras que esto es contrarrestado con el realismo de este tipo de vistas a comparación de la proyección paralela, debido a que los objetos tienen un tamaño más pequeño en la imagen proyectada.

La proyección ortogonal u ortográfica es aquella que se presenta cuando se realiza una transformación de un objeto a un plano de visualización usando líneas paralelas al vector normal al plano de visualización. Esto produce una transformación de proyección paralela en la que las líneas de proyección son perpendiculares al plano de visualización.<sup>215</sup>

Las proyecciones ortogonales se suelen utilizar para generar las vistas frontal, lateral y superior de un objeto (sin olvidar las vistas posterior, lateral izquierda o derecha según el caso y la vista inferior).

Las proyecciones ortogonales frontal, lateral y posterior de un objeto se denominan *elevaciones*, mientras que la proyección ortogonal superior se denomina *vista plana*. Este tipo de proyecciones se emplean principalmente en ingeniería y en arquitectura por su utilidad al representar de manera precisa longitudes y ángulos, ya que es posible medirlos en los mismos gráficos.

<sup>213</sup> Ibídem.

<sup>214</sup> Ibídem.

<sup>215</sup> Hearn Donald, Ibídem.

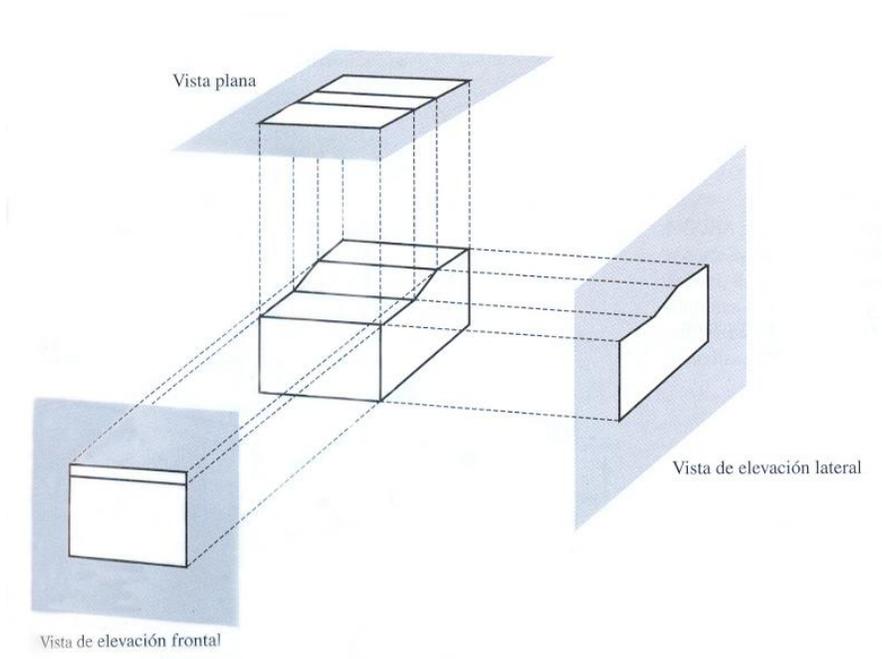


Figura IV.XXXIV Proyección ortogonales de un objeto, lateral, superior y frontal.<sup>216</sup>

#### IV.VI.II. Tipos de Cámaras

En 3DS MAX<sup>®</sup> existen dos tipos de cámaras:

- Cámara con Objetivo (*Target Camera*). Se conforma de dos elementos, la cámara y su objetivo. Este tipo de cámara es la más utilizada debido a la facilidad de su manejo y que puede animarse con sencillez, con el simple hecho de modificar la posición del objetivo.
- Cámara Libre (*Free Camera*). Se compone de un solo elemento que es la cámara. Puesto que este tipo de cámara carece de objetivo su visión se enfoca hacia donde este apuntando la lente de la cámara. Este tipo de cámara es ideal en la realización de animaciones de recorridos a través de escenarios.



Figura IV.XXXV Cámara con objetivo (izquierda), cámara libre (derecha).

<sup>216</sup> Ibídem.

### IV.VI.III. Características de las Cámaras

La interfaz de la cámara está compuesta por los siguientes elementos:

- Lentes (*Lens*). Determinan la distancia focal de la cámara, el objetivo puede ampliarse o reducirse, produciendo que el campo visual registrado por la cámara se estreche o se incremente.
- Campo Visual (*FOV*). Determina el tamaño del área que será visualizada por la cámara. El campo visual puede ajustarse horizontal, vertical y diagonalmente.
- Proyección Ortográfica. (*Orthographic Projection*). Al activar esta opción la generación de la vista de la cámara es ortográfica, sus líneas son paralelas; mientras que cuando se encuentra desactivada la proyección es cónica, en la cual sus líneas de proyección se encuentran en un punto común.
- Objetivos Disponibles. Son los objetivos estándares que existen, estos se encuentran ya predefinidos, solo es necesario seleccionarlos para poder aprovecharlos. Los objetivos más pequeños aumentan el campo de visión, pero también aumentan la distorsión, en cambio los objetivos más grandes disminuyen el campo de visión pero corrigen la perspectiva.
- Tipo de la Cámara (*Type*). Con esta opción se especifica el tipo de la cámara.
- Mostrar el Cono (*Show Cone*). Muestra el cono de la cámara aunque ésta no se encuentre seleccionada.
- Mostrar el Horizonte (*Show Horizon*). Con esta opción se muestra el horizonte; que es una línea que se encuentra a la altura de la cámara y es paralela al sistema de coordenadas universales.
- Rango de Proximidad (*Near Range*). Determina donde comenzarán a observarse los efectos atmosféricos definidos en la escena.
- Rango de Lejanía (*Far Range*). Es la opción que determina hasta donde terminarán los efectos atmosféricos definidos dentro de la escena.
- Recorte Manual (*Clip Manually*). Esta opción se encarga de definir los límites de corte de la cámara.
- Recorte Próximo y Lejano (*Clip Near, Far*). Determina la zona de la escena que será visible por la cámara aplicando los planos de corte próximo y lejano.



Figura IV.XXXVI Parámetros generales de las cámaras en 3DS MAX®

## IV.VII. Colisiones.

Una colisión es una configuración en la que dos objetos ocupan parte de una misma porción de espacio al mismo tiempo. Una colisión se produce como resultado del movimiento de los objetos; cuando esto ocurre con dos objetos estáticos es denominado interferencia entre los dos objetos.

La Detección de Colisiones se encarga de obtener, en algunos casos, una medida de la cercanía a la que se encuentran los modelos, mientras que en otros casos, tan solo si simplemente dos o más objetos se encuentran en contacto, no importando su complejidad.<sup>217</sup>

Los algoritmos de detección de colisiones tradicionales han requerido de una gran cantidad de pruebas de interacción geométrica, verificando si cada uno de los polígonos que conforman la superficie del modelo se intersecan con algún polígono de otro objeto en la escena, determinando de esta forma si dos objetos están colisionando.<sup>218</sup>

La tendencia en esta área es proponer algoritmos que reduzcan el número de llamadas para comprobar la intersección entre dos primitivas geométricas, planeando un tipo de volúmenes envolventes, organizados en una estructura jerárquica, evitando con ello la constante verificación de los pares de primitivas geométricas.

Se ha demostrado que estos algoritmos son muy efectivos y en la mayoría de las aplicaciones solamente se verifica unos cuantos pares de primitivas. Entonces el objetivo es verificar, si se presenta una colisión y si fuese así aplicar una prueba de verificación mucho más precisa.

Las primitivas más utilizadas para la comprobación de colisiones son: las esferas, las cajas, los rayos o segmentos de línea y los polígonos en general. Siendo las esferas y las cajas las limitantes volumétricas más populares y sencillas, mientras que las líneas y polígonos se aplican a modelos complejos. A continuación se describirán cada una de ellas más a fondo:

Las esferas son probablemente el tipo de primitiva más simple dentro del modelado geométrico. Debido a que las podemos representar utilizando solamente cuatro escalares, tres para su centro y el cuarto para la longitud del radio, por lo que es muy sencillo su almacenamiento. Además de son invariantes a las rotaciones haciéndolas el candidato ideal para la limitación de colisiones. El inconveniente de este tipo de volumen envolvente radica en que no se ajusta completamente a todos objetos, produciendo la dificultad de encontrar una esfera que se ajuste a un determinado objeto.

---

<sup>217</sup> Jiménez Juan José, Detección de colisiones Mediante Recubrimientos Simpliciales, 2006.

<sup>218</sup> OpenGL ARB, Shreiner Dave. *OpenGL Reference Manual*, 1999.

Debido a su simplicidad, la detección de colisiones entre esferas no es difícil. Entonces, dos esferas A y B se intersectan si la distancia entre sus centros  $C_A$  y  $C_B$  es casi la suma de sus radios  $\rho_A$  y  $\rho_B$ :

$$A \cap B \neq \emptyset \equiv ||CA - CB|| \leq \rho_A + \rho_B$$

Debido a que es necesario evadir el cálculo de las raíces, para evitar la pérdida de tiempo de cómputo, la ecuación se reescribe de la siguiente forma:

$$A \cap B \neq \emptyset \equiv ||CA - CB||^2 \leq (\rho_A + \rho_B)^2$$

Con ello el cálculo se realiza utilizando solo operaciones sencillas. Entonces la distancia entre el par de esferas es la distancia entre sus centros menos la suma de sus radios.

En las cajas alineadas a los ejes (comúnmente conocidas como *AABB Axis Aligned Bounding Box*) sus caras se encuentran orientadas de manera que sus normales paralelas a los ejes coordenados. Las ventajas que presenta este tipo de volumen es que es la limitante volumétrica más ampliamente utilizada debido a que su computo es muy sencillo, presentan un fácil almacenamiento y su testeo de intersección es muy rápido. Y al igual que la esfera de colisión, no produce un ajuste apropiado para muchos objetos, además de que cuando ésta rota el volumen envolvente suele ser más costoso que otros cuando rotan.

Para la comparación de cajas alineadas a los ejes utilizando la representación mínima – máxima, la prueba se realiza simplemente mediante la comparación extrema:

$$[p_1, q_1] \cap [p_2, q_2] \neq \emptyset \equiv p_1 \leq q_2 \text{ y } p_2 \leq q_1$$

Probando la intersección de un par de cajas en representación centro - extendida no es muy difícil:

$$[c_1 - h_1, c_1 + h_1] \cap [c_2 - h_2, c_2 + h_2] \neq \emptyset \equiv |c_1 - c_2| \leq h_2 \leq h_1$$

La prueba centro–extendida utiliza unas pocas más operaciones pero realiza tres comparaciones escalares y los saltos en el código son menores que en la comparación mínimo–máximo.

El trazado de rayos es otra técnica que se aplica en la detección de colisiones. Para explicarlo primeramente se definirá al rayo, como aquel que se representa mediante un vector que denota el comienzo y un vector (usualmente normalizado) que muestra la dirección del rayo. Esencialmente un rayo comienza en su punto de inicio y sigue la dirección del vector. Por lo tanto la ecuación del rayo se representa de la siguiente manera:

$$\text{Punto del Rayo} = \text{Inicio del Rayo} + t * \text{Dirección del Rayo}$$

Donde:

$t$  es un valor flotante en el intervalo de  $[0, \text{infinito})$ .

Con el valor de 0 se obtiene el punto de inicio, al sustituir el resto de los valores se consiguen los puntos correspondientes a lo largo del rayo.

Punto del Rayo, Inicio del Rayo, Dirección del Rayo son vectores  $(x,y,z)$ . Con esto es posible calcular intersecciones con planos, cilindros, entre otros utilizando el trazado de rayos.

Entonces, como se ha visto, el utilizar volúmenes envolventes puede disminuir el tiempo de cálculo de las colisiones entre objetos, sin embargo, el número de parejas de objetos sobre los que se realiza la prueba no cambia, por lo que el tiempo de cómputo es el mismo solamente que se reduce por una constante. Uniendo los volúmenes envolventes a una jerarquía es posible reducir la complejidad logarítmica, construyendo con ello una *jerarquía de volúmenes envolventes*.<sup>219</sup>

Lo anterior quiere decir que para un número de objetos  $n$  es posible construir una jerarquía de volúmenes envolventes en forma de árbol de la siguiente forma: primeramente se obtienen los volúmenes envolventes de los objetos individuales, estos son colocados como nodos (hoja del árbol que representa dicha jerarquía). Los nodos son colocados un nuevo volumen envolvente de manera recursiva hasta obtener un volumen envolvente que agrupe a todos los objetos (este será representado por la raíz del árbol). Con esta jerarquía se comprobaría la colisión entre los hijos de un nodo solo si se produce una colisión entre los nodos padre.

En una jerarquía de volúmenes envolventes no es necesario que un volumen padre agrupe a los volúmenes hijos, sino que solamente este debe abarcar a los dos objetos que representan a los volúmenes envolventes. Además de que dos volúmenes envolventes del mismo nivel pueden tener partes comunes entre ellos.

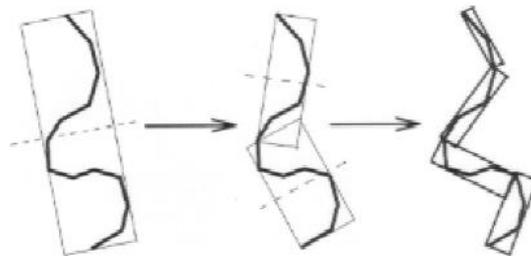


Figura IV.XXXVII Jerarquía de Volúmenes Envolventes.<sup>220</sup>

<sup>219</sup> Jiménez Juan José, *Ibidem*.

<sup>220</sup> *Ibidem*.

Una jerarquía de volúmenes envolventes puede ser utilizada en la organización de un los objetos de una escena o bien pueden aplicarse jerarquías de volúmenes envolventes a cada uno de los objetos de la escena.

### IV.VIII. Animación.

En el capítulo anterior se le dedicó un extenso apartado a la teoría sobre animación, es por ello que en este apartado se hablará sobre la animación en 3DS MAX® y en general para los sistemas de software 3D que se encuentran en el mercado.

Primeramente se verá desde el enfoque de animación tridimensional:

La animación agrega el parámetro tiempo a la representación de objetos, que es el conjunto de puntos que comprenden a un objeto en función del tiempo. La animación es creada al cambiar los atributos de una escena a través del tiempo. Existen dos tipos de movimientos que corresponden a diferentes atributos en una escena:

- El camino más común para animar un modelo es alterando las transformaciones en los nodos de transformación de la escena. A este tipo de animación se le puede llamar como cambio de posición. El cambio de posición es el resultado de las transformaciones de traslación, rotación y escalamiento no uniforme de un modelo.
- Las animaciones también pueden ser creadas cambiando la geometría y cambiando la forma de las primitivas. Este tipo de movimiento es llamado deformación. Este se aplica más comúnmente a objetos conformados por mallas de polígonos. Aquí la posición de los vértices de los polígonos dependen del tiempo. Las deformaciones pueden aplicarse a simulaciones de fluidos, ropas o piel.<sup>221</sup>

3DS MAX® de Autodesk® tiene una gran flexibilidad permitiendo animar casi cualquier cosa, desde una animación sencilla en la que a un modelo se le aplican las transformaciones básicas, como la traslación y la rotación.

El principio en que se basa 3DS MAX® al igual que la mayoría de los programas de animación actuales es utilizando las imágenes en secuencia mejor conocidos como fotogramas, las cuales trazan la ilusión del movimiento del objeto conforme estas van pasando una tras otra. Para que un objeto presente movimiento en una animación es necesario que este cuente con un inicio y un final, a estos se les denominarán puntos clave o *keyframes*. Los *keyframes* son puntos en los que se realiza un cambio del estado en que se encuentra el objeto, como un cambio de posición, de dirección de tamaño o un cambio en los parámetros propios del objeto en relación a su estado anterior.

En MAX solo es necesario definir los fotogramas y la cantidad necesaria de ellos para realizar la animación, ya que la secuencia de imágenes comprendida entre ambos *keyframes* se realiza automáticamente.

---

<sup>221</sup> Van den Bergen Gino, Collision Detection in Interactive 3D Environments, 2004.

Existen dos formas para generar *keyframes*:

- *Autokey*. Es la forma más sencilla de generar un *keyframe*, ya que al aplicar esta opción los *keyframes* se generan automáticamente en el momento en que se presenta un cambio de estado en el objeto.
- Generación del *key*. En este tipo de generación es necesario ir generando llave por llave cada vez que se define la acción a realizar.

3DS MAX muestra por default solamente 100 fotogramas en un inicio, que equivalen a 3.33 [s] de animación, esa cantidad de fotogramas puede ser cambiada de una manera sencilla presionando sobre el icono de configuración de tiempo, en el cual se determinará el formato de salida de la animación (Film, PAL, NTSC), el tipo de presentación del regulador de tiempo permitiendo la medición en fotogramas, minutos, segundos, impulsos (tomando en cuenta que un fotograma equivale a 160 impulsos), entre otros.

En estas opciones también se puede seleccionar que la animación solo sea reproducida en la vista que se encuentre activa, que se produzca un *loop* continuo en las animaciones, es decir que se repitan infinitamente, se puede determinar la velocidad a la que será reproducida la animación, tomando en cuenta en que 1x es la velocidad a la que normalmente reproduce MAX, 30 fotogramas por segundo, permitiendo que esta se reproduzca a 1/4x, 1/2x, 1x, 2x, 4x. La dirección de reproducción de la animación, no afecta el render de salida.

Ya en la animación se cuentan con las opciones de tiempo de inicio, el cual determina en qué momento y el fotograma en que será iniciada la animación, estos valores también pueden ser negativos. El tiempo de término, es aquel que determinará el fotograma final en el regulador de tiempo. La cuenta de fotogramas muestra la cantidad de fotogramas con los cuales contará la animación. La longitud se encarga de determinar la cantidad de tiempo existente entre el fotograma inicial y el final. Y finalmente el fotograma actual es el encargado de mostrar el fotograma en que se encuentra en ese momento exacto la animación.

Con esto se terminan de revisar los conceptos básicos de animación en 3DS MAX además de los elementos básicos que hay que tomar en cuenta al momento de realizar una animación utilizando este software.

## IV.IX. Representación Tridimensional.

En esta sección se procederá a mostrar cómo fue que se realizaron cada uno de los pasos hasta llegar a la creación del paseo virtual, comenzando por cómo se realizó el modelado de cada una de las piezas, grabados herrería y el mismo edificio del Palacio de Bellas Artes y sus exteriores, continuando con el texturizado de sus elementos, posteriormente se explicará paso a paso la inclusión de los modelos tridimensionales al motor del juego y finalmente se mostrarán los resultados obtenidos corriendo en el motor gráfico los objetos tridimensionales en tiempo real.

### IV.IX.I. Modelado de Objetos

Para la realización de cada uno de los elementos que componen al palacio de Bellas Artes se utilizó el sistema de modelado tridimensional *3DS MAX 2009*® de *Autodesk*® como ya se había comentado en páginas anteriores. Este sistema ofrece una gran flexibilidad en el modelado de los objetos tridimensionales, permitiendo la edición de vértices, bordes, caras y objetos completos; además de una facilidad de movimiento y visión de cada uno de los ángulos de la pieza que se esté creando con su amigable interfaz de usuario y manejo de vistas, además de su reciente incorporación del modelo de vistas en la parte superior derecha gracias a la compra del software de *MAYA*®. *3DS MAX* cuenta con muchas más ventajas que se explicarán a lo largo de esta parte del capítulo mientras se muestran como se desarrollaron los objetos en este amigable sistema.

Se comenzará explicando cómo se modelaron algunos de los elementos que se desarrollaron en la elaboración de esta tesis.

Como ayuda en el modelado de la herrería se utilizaron fotografías que se tomaron directamente del edificio. Se sacaron más de 2,500 fotografías del Palacio de Bellas Artes y de sus alrededores, tomando la arquitectura del edificio, los elementos por los que está compuesto y sus grabados como son: los mascarones y florones, los jardines y las fuentes, para la obtención de materiales de varios bloques de mármol que componen las paredes y pisos del edificio, entre otros.

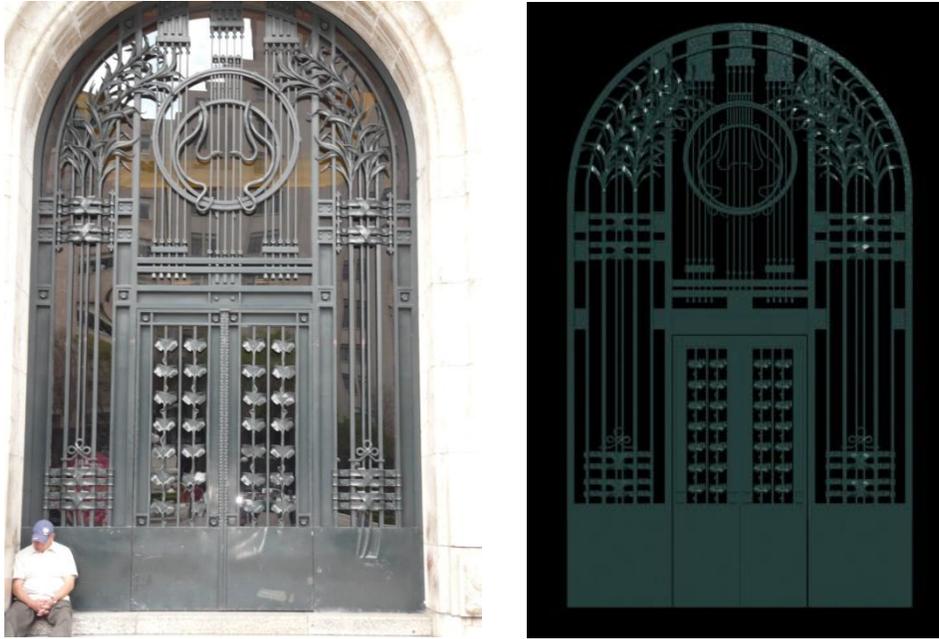


Figura IV.XXXVIII Fotografía entrada Palacio de Bellas Artes (izquierda), puerta modelada (derecha).

En el modelado de la herrería de las puertas y ventanas como se mencionó anteriormente solo se utilizaron fotografías tomadas de frente de las mismas. Debido a que las puertas y las ventanas básicamente se componen por barrotes y placas de acero, para el modelado de estas se utilizaron cajas alineándolas de acuerdo a su forma mostrada en las fotografías.

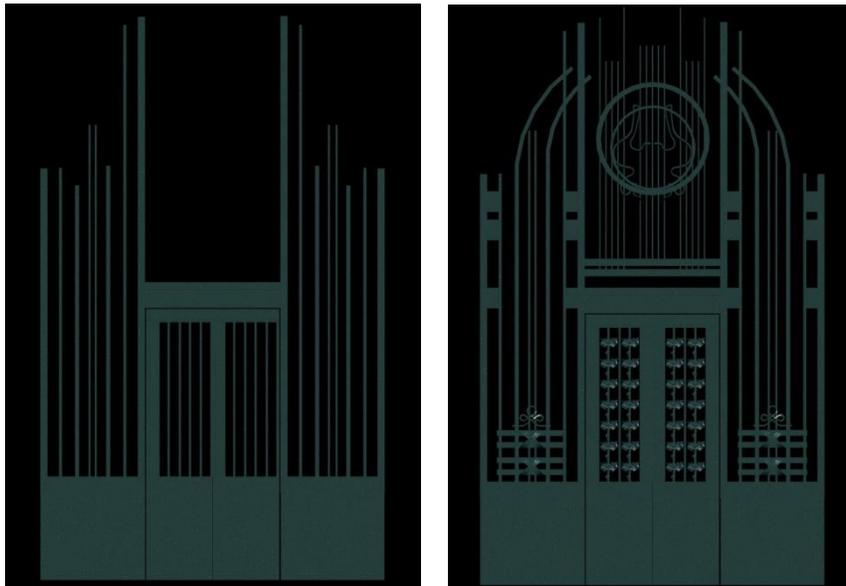


Figura IV.XXXIX Progreso del desarrollo del modelo de la puerta del Palacio de Bellas Artes.

Como se puede ver en los *renders* anteriores la puerta lateral fue construida colocando cajas alargadas en sus posiciones acorde con las fotografías tomadas. Se utilizaron cajas alargadas para emular los barrotes y cajas de forma cuadrada y rectangular para emular las diversas placas de acero de las puertas y ventanas.

De acuerdo con esta técnica solo es necesario tener una referencia de la cual partir para realizar el modelo que emule al objeto del mundo real.



Figura IV.XL Emulación de objetos del mundo real al crearlos digitalmente en 3D.

Como se comentó prácticamente toda la herrería del palacio de Bellas Artes fue desarrollada a partir de esta forma de construcción.

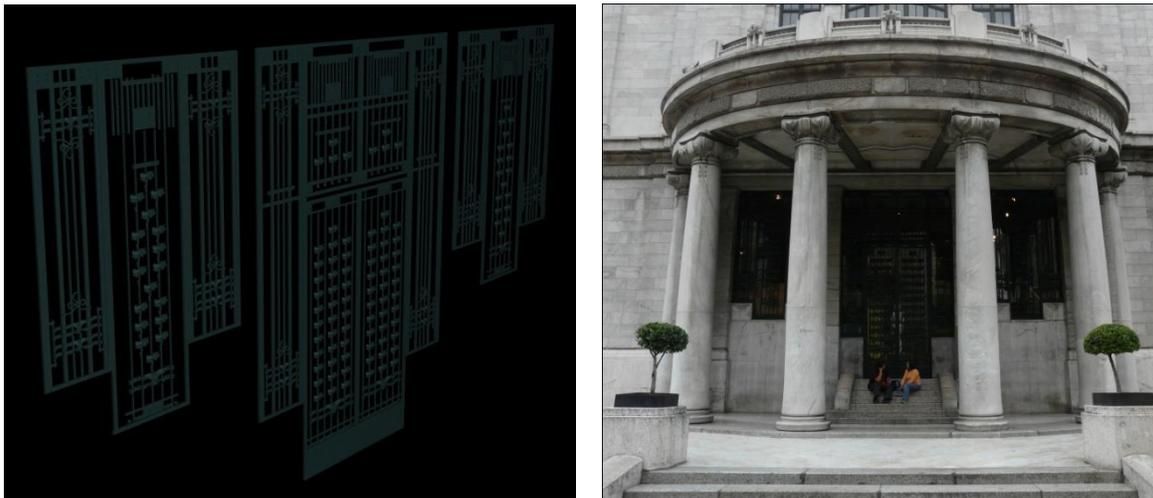


Figura IV.XLI Muestra de la herrería de las ventanas laterales del Palacio de Bellas Artes.

Las puertas y ventanas contienen también diversos adornos en su construcción. Para su realización se utilizó una técnica de modelado en la cual se coloca la imagen con la forma que se va a modelar. Se utilizaron las fotografías que se tomaron del Palacio de Bellas Artes, para colocarlas en un plano dentro de *3DS MAX*®, se tomaron en cuenta sus medidas, el largo y el ancho, las cuales correspondían a las dimensiones del mismo plano donde se colocaría la fotografía, para que así no se presentaran distorsiones en la imagen.



Figura IV.XLII Mapeo de la fotografía de un adorno sobre un plano.

Ya habiendo asignado el material al plano, se procede a dibujar literalmente sobre el plano, la forma de la figura que se desea obtener. En el ejemplo se dibujaron la rama y las dos hojas con que cuenta el adorno de la puerta. Para ello es necesario ir al menú de formas y tomar la herramienta línea, con la cual se procederá a trazar todo el contorno de la forma a dibujar, que en este caso serían el tallo y las hojas.

Una vez dibujada la forma de las hojas cuidadosamente, se agregaron vértices donde fue necesario, corrigiendo las posiciones de los vértices ya colocados en el trazo original y los posteriores hasta dejar el dibujo de la forma en el estado deseado, se procedió a convertir el trazo en un plano (aplicando la herramienta polígono editable), para convertir a polígono editable.

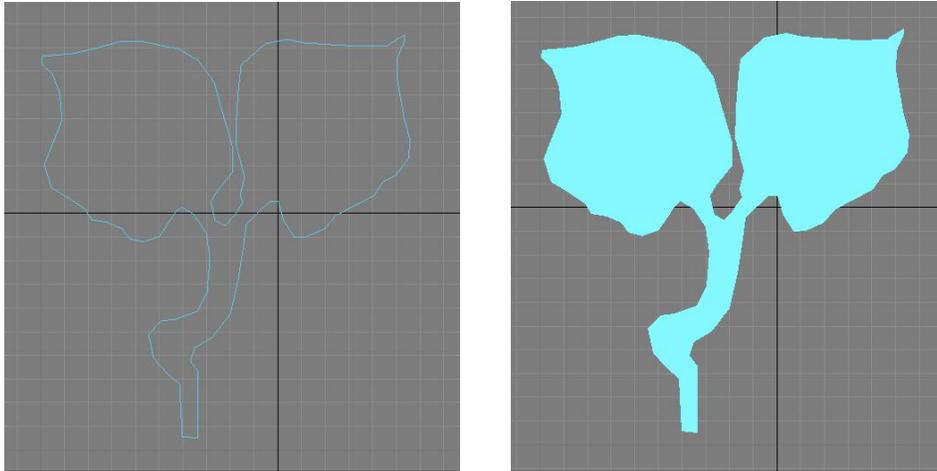


Figura IV.XLIII Dibujo del borde de la hoja del adorno (izquierda), polígono editable del trazo del adorno (derecha).

El trabajo que se realiza posteriormente depende del modelo en el cual se esté trabajando, por ejemplo se puede, extruir las formas para darle volumen al modelo, como se realizó en los moños que se encuentran en la puerta; para la forma en particular en la cual se trabajó, una vez transformada a polígono, se tomaron nuevamente los vértices y se le fue dando la forma tridimensional de las hojas y el tallo, de tal forma que la apariencia sea realmente que la hoja tiene surcos y dobleces, además de tener ciertas inclinaciones en algunos puntos. Al generar el render resultante se obtuvieron resultados excelentes. Se agregó el extruido de la forma, para obtener un poco más de grosor en las hojas.

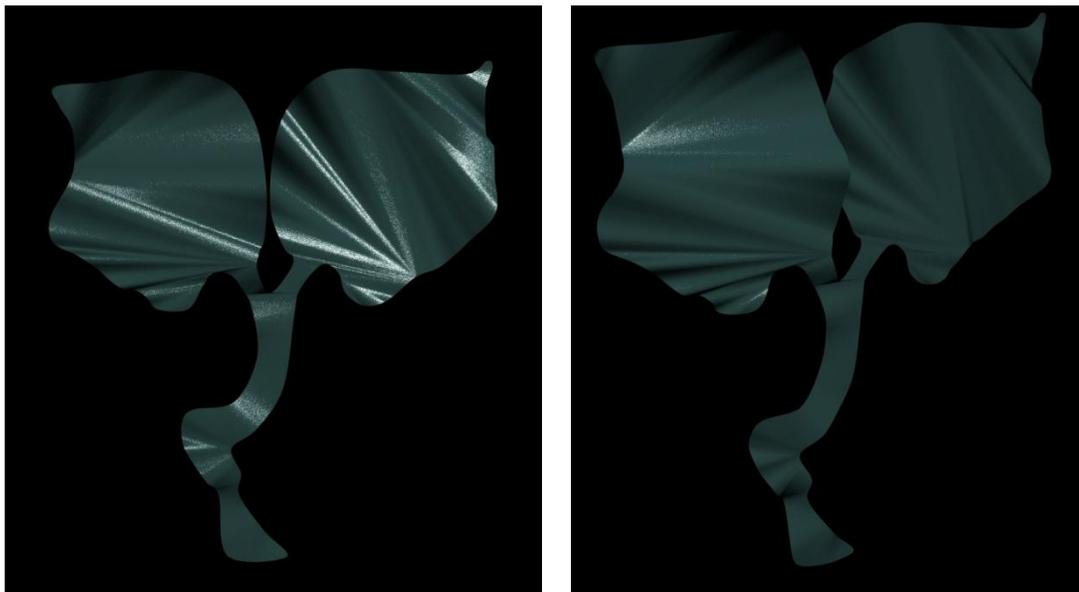


Figura IV.XLIV Muestras del resultado final del modelo del adorno aplicando la técnica del trazado.

Muchos de los adornos que se realizaron en el modelado del palacio de Bellas Artes, fueron realizados de la forma que acabo de explicar. Simplemente en el desarrollo del modelado de la puerta, tanto los moños, las cruces y las serpientes fueron desarrollados utilizando esta técnica.



Figura IV.XLV Serpientes y adornos modelados utilizando la técnica del trazado.

En la herrería se desarrollaron una gran cantidad de formas y adornos, se modelaron utilizando la técnica antes descrita. La siguiente muestra es la puerta que contiene más adornos en todo el exterior del Palacio de Bellas Artes.



Figura IV.XLVI Puerta del costado del Palacio de Bellas Artes desarrollada mediante trazos.

En la figura anterior es posible observar las dos técnicas explicadas antes, el posicionamiento para colocar los barrotes y trazar la forma de la puerta, mientras que para los espirales, las hojas, las ramas que sostienen las hojas, el arpa central de las serpientes, entre otras, fueron trazados dibujando las líneas, convirtiendo a polígonos editables y extruyendo sus formas.

Para otras formas, como son las esculturas, se empleó el modelado poligonal y el modelado orgánico.



Figura IV.XLVII La técnica del trazo puede ser una alternativa al modelado orgánico.

Para el desarrollo de estos modelos inicialmente se trazó el borde del rostro de la serpiente, posteriormente se convierte en polígono editable y es extruida la forma, para tener un cuerpo inicial del modelo, para introducir un realce la forma de los colmillos y la lengua de la serpiente, al igual que con el resto de los realces que se observan en el rostro, nuevamente se dibuja la forma a realzar, se transforma y se realiza una extrusión o se aplica la herramienta *Bevel* para obtener un escalamiento de la forma y una extrusión. Al utilizar esta técnica se puede decir que estamos armando un rompecabezas al colocar cada forma en una posición adecuada y con ello conseguir el resultado deseado.

Otra técnica que también se utilizó en el modelado de objetos para esta tesis fue la de construcción hacia arriba, nombrada de esta manera, ya que en esta técnica se comienza utilizando una forma simple como un rectángulo o un círculo, o en su caso la forma necesaria para el modelo. Esta forma de construcción es muy útil en la elaboración de modelos con formas verticales o semi regulares. De esta forma se construyeron los pilares, las bases de las estatuas, las jardineras y las astas del Palacio de Bellas Artes, entre otras.

Se ejemplificará esta técnica de modelado, explicando el desarrollo de las astas de plaza del Palacio de Bellas Artes recurriendo a las fotografías que fueron tomadas.



Figura IV.XLVIII Fotografías del asta del Palacio de Bellas Artes, tomadas como muestra para realizar el desarrollado por modelado poligonal, proyectando los bordes de la figura hacia arriba.

Para iniciar es necesario tomar una forma básica como lo es un plano, lo dibujamos presionando la *tecla Ctrl*, para poder obtener un cuadrado perfecto, posteriormente se convierte el plano en un polígono editable, pasamos a la selección por vértices y en ella seleccionamos la herramienta *Chamfer* que será aplicada a los cuatro vértices del cuadrado para conseguir un achatado en cada una de las esquinas.

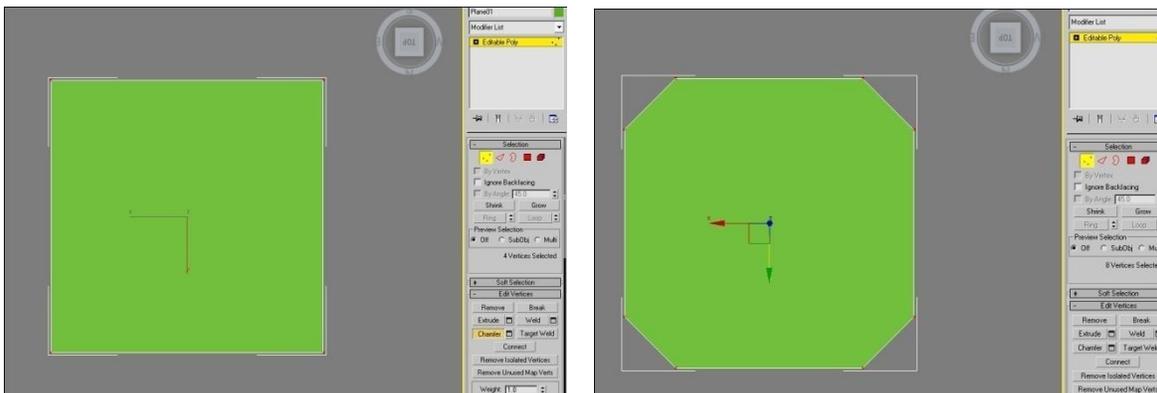


Figura IV.XLIX Plano convertidos a polígono editable (izquierda), aplicación de la herramienta *Chamfer* al plano (derecha)

Una vez que se obtuvo la forma octagonal para simular la base del asta, es necesario seleccionar por el borde de la forma y prolongarla en dirección del eje Z, es decir hacia arriba con ayuda de la transformación de traslación o movimiento y presionando la *tecla shift*. Con lo anterior empezamos a construir la forma que requerimos para la base del asta. Para continuar con la construcción y poder generar inclinaciones en la forma, es necesario aplicar la secuencia de modelado anterior para prolongar el borde de la forma nuevamente hacia arriba y para finalizar se aplica la transformación de escalamiento uniforme, hasta obtener el resultado deseado.

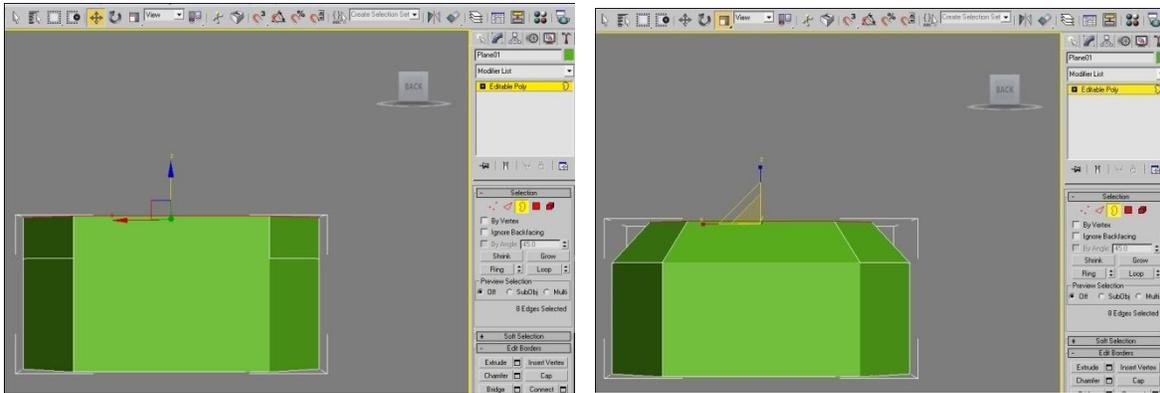


Figura IV.L Proyección hacia arriba (izquierda) y escalamiento uniforme (derecha) del contorno de la forma octogonal del plano.

Se continúa de esta forma aplicando las transformaciones básicas hasta obtener el modelo de la base deseado. Se puede decir que esta es la forma más básica y a la vez la más sencilla de modelado poligonal, ya que a partir de esta forma de construcción es posible realizar una gran cantidad de modelos en poco tiempo obteniendo muy buenos resultados.

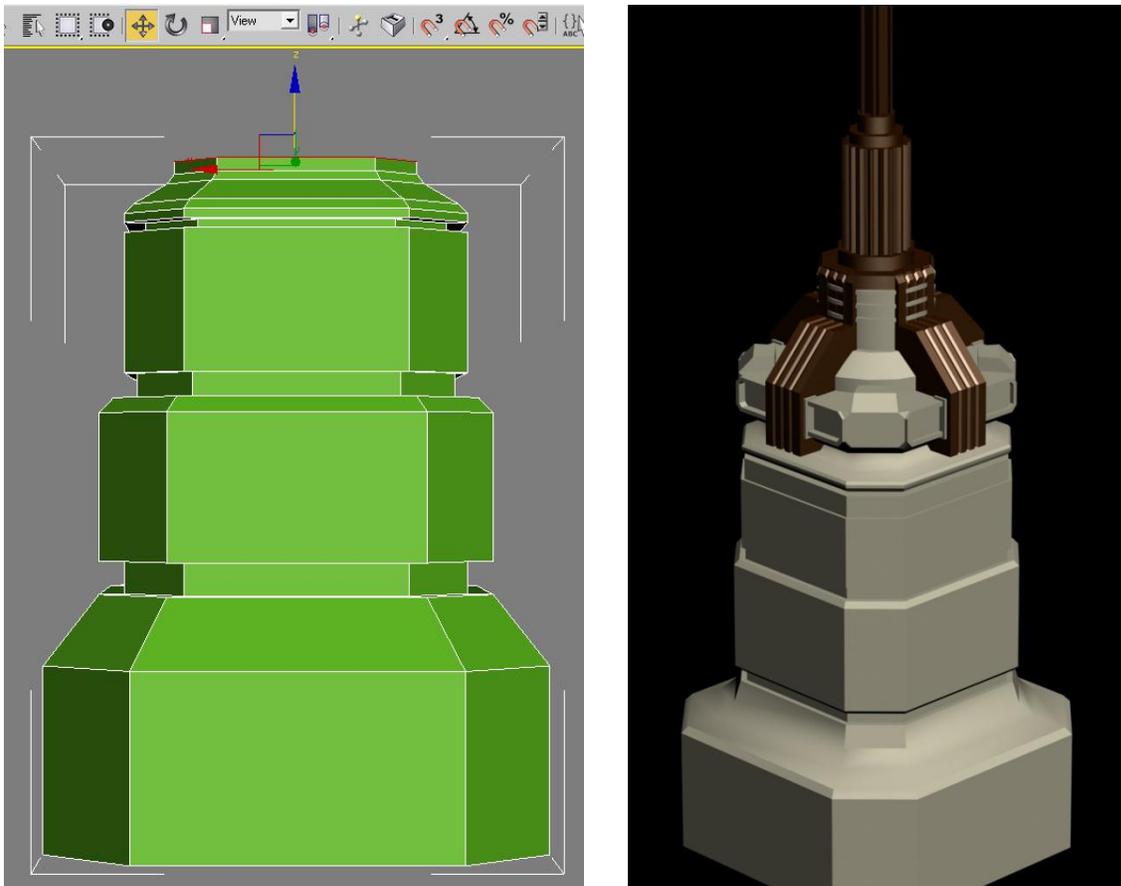


Figura IV.LI Muestra de la base del asta después de aplicar proyecciones verticales y escalas uniformes.

Para poder desarrollar la estructura del Palacio de Bellas Artes fue necesario recurrir métodos más precisos y así lograr los resultados esperados, además del uso de fotografías como referencia, también fue necesario utilizar los planos de la construcción del edificio, tomados del libro “El Palacio de Bellas Artes”<sup>222</sup>.

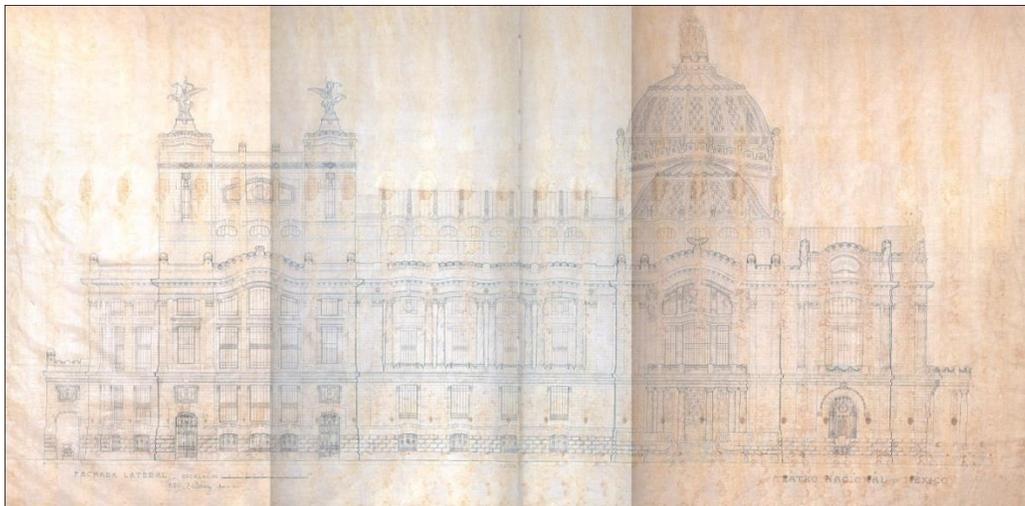


Figura IV.LII. Muestra del plano de la vista lateral del Palacio de Bellas Artes.<sup>223</sup>

Al recurrir a los planos se consiguió tener una base de la cual partir en la construcción el edificio, teniendo una exactitud mayor, debido a que con ellos solo era necesario empezar a construir con ayuda de un plano y prolongar sus bordes o tomar la cara o caras del mismo y extruirlas y con ello obtener las salientes del edificio o los bordes de las ventanas, entre otras formas.

Inicialmente es necesario colocar la imagen del esquema del Palacio de Bellas Artes como un material, posteriormente asignarla al plano con el tamaño de la imagen que va a ser asignada, posteriormente se elige el punto donde se va comenzar a construir, para esta explicación se elegirá la torre izquierda al sur del edificio. A continuación se coloca un plano que cubra la forma de la saliente del esquema del Palacio de Bellas Artes.

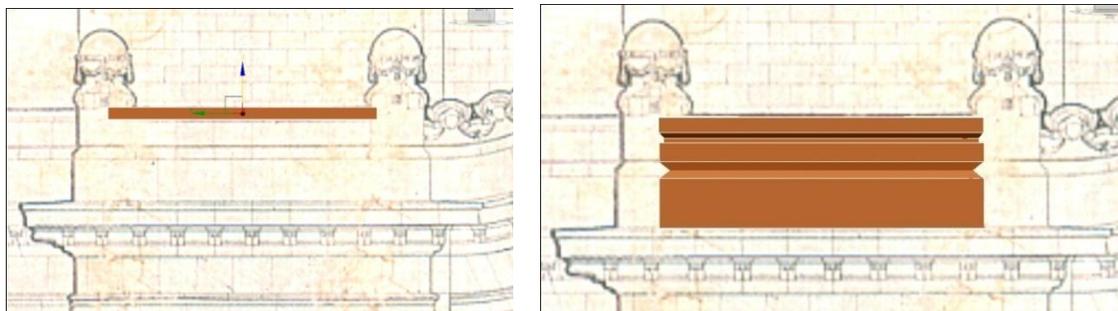


Figura IV.LIII Se mapea imagen del plano del Palacio de Bellas Artes sobre un plano, siguiendo la forma de este se coloca un plano el cual será proyectado y escalado hasta tomar la forma deseada.

<sup>222</sup> Mark Mogilner, El Palacio de Bellas Artes, 2004.

<sup>223</sup> Ibídem.

Debido a que era muy complicado realizar la construcción del edificio en una sola pieza o en grandes partes de construcción, se tomó la decisión de construirlo en pequeños bloques de construcción, de tal forma como si se estuviera construyendo realmente. De esta forma se colocaba un nuevo plano para realizar el siguiente bloque de construcción. Al proseguir con la construcción, en las imágenes anteriores se puede observar una nueva saliente, para ello es necesario colocar un nuevo plano y continuar a prolongar las salientes del plano y construir hacia abajo.

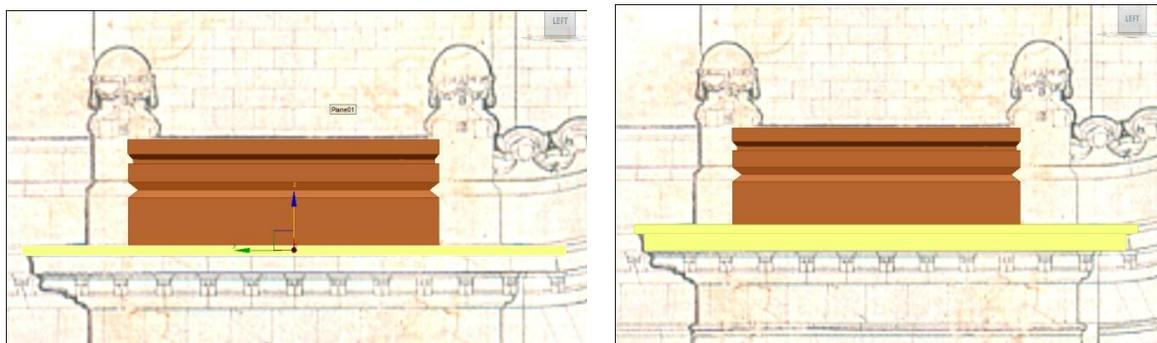


Figura IV.LIV Muestras de construcción por bloques para la realización del Palacio de Bellas Artes.

De esta forma se agruparon uno a uno los bloques de construcción, mientras se le daba forma a cada plano, o forma trazada, para alcanzar las formas que tiene el edificio, para los huecos de las ventanas o para los arcos del edificio.



Figura IV.LV Muestra de construcción por bloques colocados sobre el plano del Palacio de Bellas Artes.

Posteriormente a los planos modelados se integraron las ventanas, las puertas los pilares, tanto del primero como segundo pisos y los grabados que habían sido previamente modelados.



Figura IV.LVI Muestras del resultado final de la construcción por bloques, colocando ventanales y puertas a los costados del Palacio de Bellas Artes.

La última técnica de modelado que se utilizó en el desarrollo del Palacio de Bellas Artes, es de modelado poligonal, muy importante en el desarrollo de objetos de formas no regulares. Debido a la forma y la dificultad que presentan este tipo de formas, el modelado poligonal mostrado anteriormente no es fácil aplicarlo para este tipo de desarrollos, ya que está diseñado principalmente para formas rectas y con esquinas. Esta técnica fue aplicada en el desarrollo de los grabados y esculturas del edificio. Esta técnica consiste en que por lo menos necesitamos dos vistas del objeto a modelar, ya sea la frontal y la paralela, o la frontal y la superior, aunque lo ideal sería contar con las tres vistas del objeto.

Una vez que se tienen las vistas, lo que procede es colocar las imágenes sobre planos en *3DS MAX®* esto se logra, conociendo las dimensiones de las imágenes, se crean los planos de acuerdo a las mismas y se asignan los materiales a cada uno de los planos con las vistas del modelo.

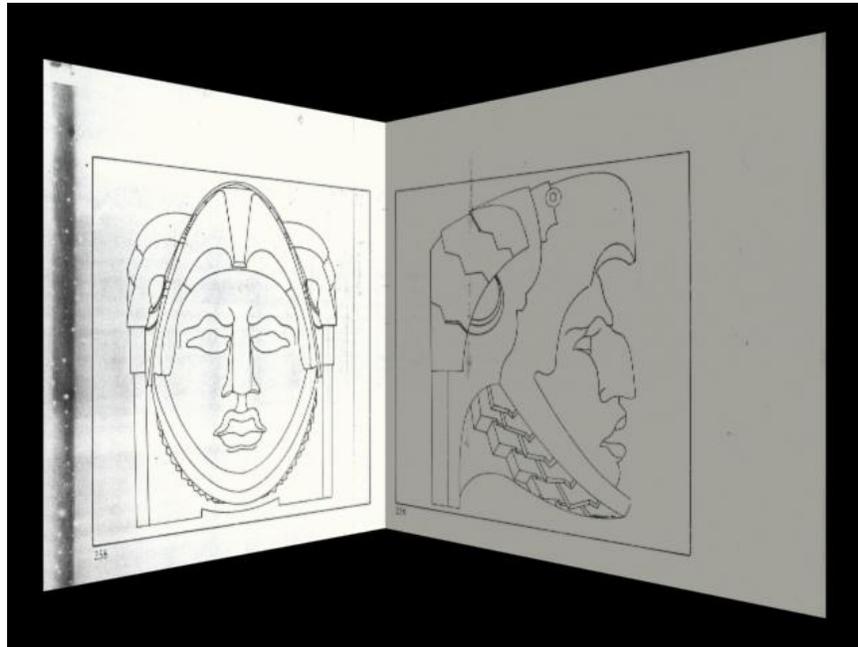


Figura IV.LVII Imágenes de muestra colocadas en los planos frontal y lateral para la realización del mascarón del caballero águila.

Ya que se tienen las vistas de los modelos, se crea un pequeño plano en alguna de las formas que tenga la imagen o fotografía de la vista a elegir del modelo, reiterando se puede utilizar cualquiera de las tres vistas. Este plano es transformado en un polígono editable, permitiendo que los vértices y bordes puedan ser editados como se desee. Una vez que el plano es un polígono editable, es necesario que podamos ver la forma de la vista que se encuentra detrás del plano para así poder dibujarla, para ello se presiona el comando *Alt+X* dándole al plano un efecto de transparencia.

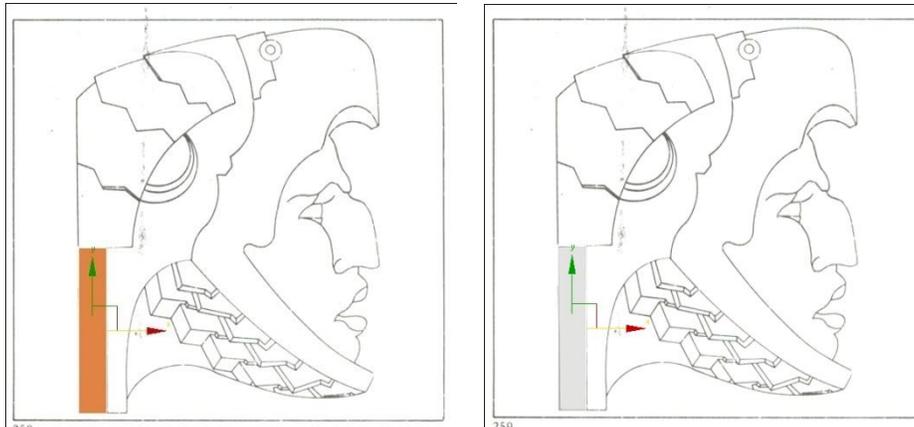


Figura IV.LVIII Posicionamiento y aplicación de transparencia de un plano para el dibujo del rostro del caballero águila.

Con el plano transparente lo que precede es dibujar literalmente la forma de la vista, en las imágenes de muestra anteriores puede observarse que se utilizaron las imágenes del rostro del Guerrero Águila que se encuentra sobre las puertas del Palacio de Bellas Artes. Entonces, se dibujó literalmente su rostro, esto puede resultar un tanto difícil en las primeras ocasiones en que se aplica esta técnica, ya que requiere que se tenga una idea de cuantos vértices serán necesarios para mostrar un efecto de realce en los bordes o si se presenta un área redonda en el modelo a realizar habrá que calcularse el número de caras necesarias para que el polígono muestre un efecto de redondeado.

El dibujado consiste en tomar los bordes necesarios para trazar la forma de la imagen que se encuentra detrás del plano.



Figura IV.LIX Dibujo del rostro del caballero águila realizado mediante la proyección de bordes.

Una vez que se ha trazado totalmente la forma de la imagen a modelar, es el momento en el que intervienen el resto de las vistas del objeto, debido a que en este paso se toman uno a uno los vértices del plano dibujado y son proyectados hasta la posición que lo indique la otra vista con la que se esté trabajando. Para ello es recomendable conocer los comandos de *3DS MAX*® para el cambio de vistas y así poder manejar y observar en pantalla completa el modelo que se está desarrollando. Entre los comandos para las vistas se tienen: superior: tecla t, frontal: f, izquierda: l, perspectiva: p, ortográfica: u, entre otros.

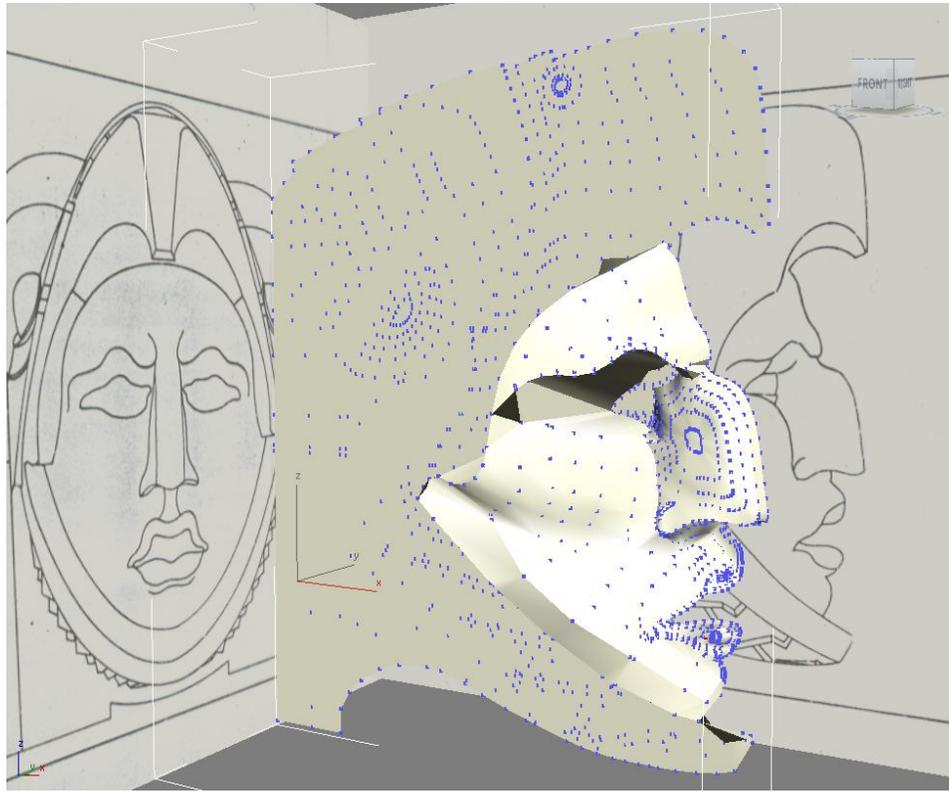


Figura IV.LX Proyección de vértices utilizando la vista lateral como apoyo en el desarrollo del modelo.

Esto se realiza hasta terminar con los vértices. También se debe de tomar en cuenta constantemente los resultados que se van obteniendo con el reposicionamiento de los vértices de tal forma que se obtenga el resultado deseado.

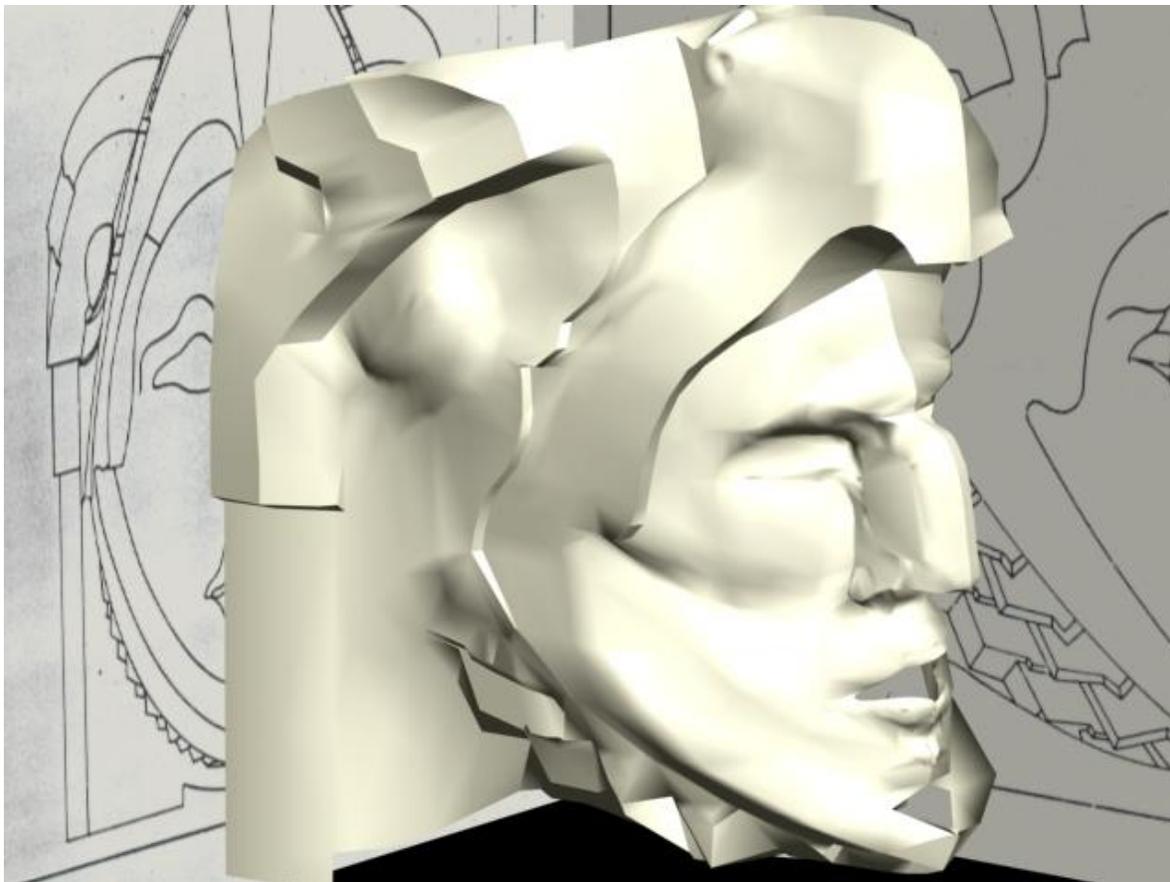


Figura IV.LXI Muestra final del mascarón del caballero águila.

Debido a que incluso cuando aparentemente se tiene el resultado que se quiere, al observar la malla poligonal esta presenta pequeñas deformaciones no logrando el efecto deseado y con ello que no se vea realmente como debe de verse el objeto.

Una herramienta de gran utilidad para reforzar esta técnica es la herramienta *Smooth* de la *lista de modificación*. La herramienta *smooth* produce un suavizado en la malla sin la necesidad de aplicar una gran cantidad de polígonos a la misma.

Al utilizar esta técnica se debe dedicar un cuidado especial a los pliegues y realces de la figura, ya que empleando de manera incorrecta esta herramienta se suavizará y se perderá la forma que se deseaba que tuviese la malla. Para evitar esto, se posicionan los vértices de manera que las formas que se desea que prevalezcan se tengan más marcadas.



Figura IV.LXII Aplicación de la herramienta *smooth* al mascarón del caballero águila.

Las técnicas antes mencionadas se emplearon en el desarrollo de las partes restantes, ya sea en combinaciones o individualmente.

Una vez que se han desarrollado los objetos para Palacio de Bellas Artes, se texturizaron y posteriormente se integraron al sistema de *Oblivion* mostrando los resultados finales del paseo virtual por el palacio de Bellas Artes. A continuación se explicará el procedimiento.

#### IV.IX.II. Texturizado y Manejo del Software NifSkope

Una vez terminado el modelo procedemos a texturizarlo, para ello se abre la ventana de materiales y se busca la textura que va a utilizarse. Se tomará uno de los pilares de la planta baja del Palacio de Bellas Artes para ejemplificar. En la ventana de materiales, se presiona el botón que se encuentra al lado derecho del color difuso, con ello se abrirá una ventana de diálogo en la cual se seleccionará la opción de imagen de mapa de bits, abriéndose un cuadro de diálogo para buscar la imagen que se utilizará como textura.

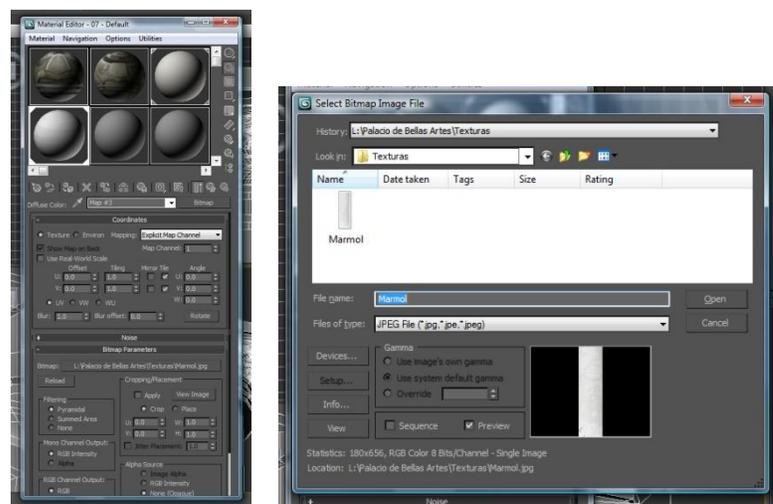


Figura IV.LXIII Ventana de materiales y cuadro de diálogo de selección de imagen.

Una vez seleccionada la imagen, en la ventana de materiales, se selecciona la esfera que contiene la imagen que se utilizará como textura, se presiona el botón *asignar material a la selección*, con ello la imagen será asignada como material al modelo del pilar. Logrando esto se procede a mapear la imagen como textura, para ello se presiona sobre el menú modificar → lista de modificación y se selecciona la opción de *mapeo UVW*, este modificador, permitirá que se aplique la textura en diferentes representaciones, dependiendo de la forma del modelo, plana, cilíndrica, esférica, envolviéndolo, por cara o transformando las coordenadas XYZ a UVW.

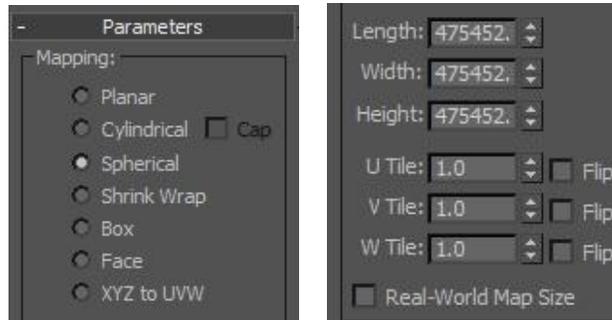


Figura IV.LXIV Parámetros de la herramienta *UVW Mapping*.

Aplicando un mapeado por cara de la textura se tiene el siguiente resultado:



Figura IV.LXV Mapeado por caras de la textura de mármol sobre el pilar del Palacio de Bellas Artes.

Una vez que el modelo ha sido texturizado el paso a proceder es exportarlo al formato *.NIF*, que es formato en que se manejan los objetos en el juego de *Oblivion*.

Para que las texturas puedan ser identificadas por el set de construcción y a la vez por el juego de *Oblivion* es necesario convertirlas al formato *.DDS* (*Digital Data Storage*), utilizado para el almacenamiento y respaldo de datos, que en este caso es para utilizarlo como envoltorio de las texturas de los modelos para el juego de *Oblivion*.

Para poder utilizar este formato en *Photoshop CS4*<sup>®</sup> fue necesario descargar el *plugin DDS* de *Photoshop*<sup>®</sup>, permitiendo almacenar archivos en dicho formato. Entonces para tener la textura que deseamos en formato *.DDS*. Para la instalación del *plugin*, es necesario guardarlo en una ruta específica dentro de la carpeta donde se encuentra instalado el software de *Photoshop*, para esta tesis la ruta fue: *C:/Program Files/Adobe/Photoshop/Plugins/File Formats*.

Una vez colocado guardado el *plugin*, se inicia el programa *Photoshop* y solo es necesario abrir la imagen deseada en *Photoshop*<sup>®</sup>, para este caso se utilizará una textura de mármol obtenida de las fotografías que fueron tomadas del Palacio de Bellas Artes, una vez abierta la imagen es necesario ir al menú archivo → guardar como, nombrar y seleccionar el archivo en formato *.DDS*.

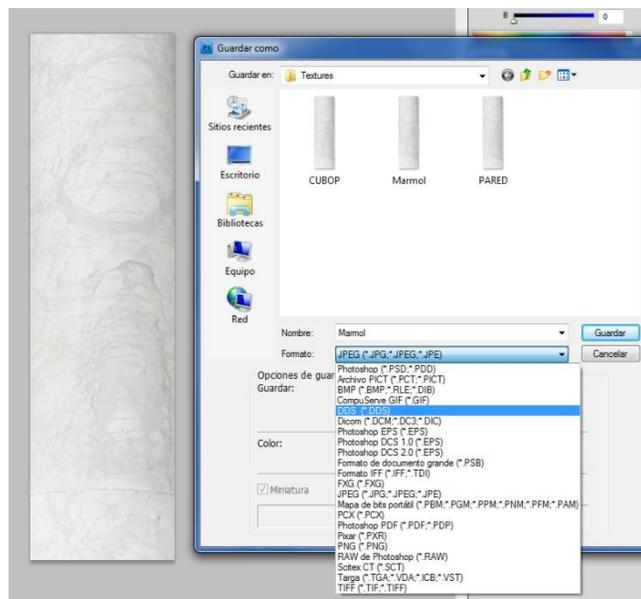


Figura IV.LXVI Almacenamiento de la textura en formato *DDS*.

Ya que se tiene el modelo con extensión *.NIF* y la textura que le será integrada en formato *.DDS* se utilizará el software *NifSkope* de código abierto que es un programa gráfico con el cual es posible abrir archivos *.NIF*, ver su contenido y editarlos. Con él se pueden realizar rápidamente cambios a las propiedades específicas de los archivos *.NIF* como cambiar sus texturas, agregar transparencias, entre otras; además de entregar una vista 3D del contenido de los archivos *.NIF*, permitiendo observar los objetos 3D, editarlos y obtener vistas previas de los mismos.<sup>224</sup>

<sup>224</sup> <http://niftools.sourceforge.net/wiki/NifSkope>

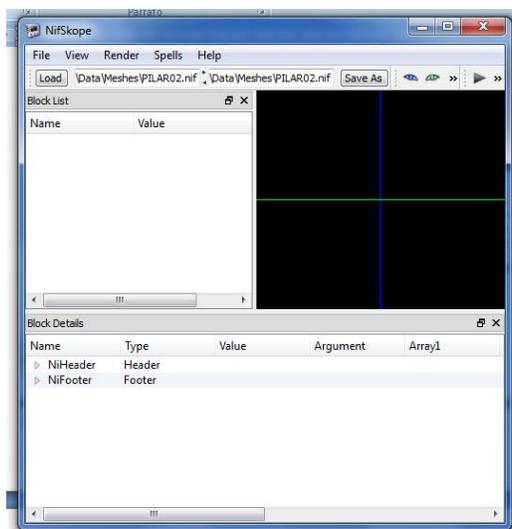


Figura IV.LXVII Ventana principal del software de *NifScope*.

El programa *NifScope* permite ver vistas previas de los modelos tridimensionales, ver sus vistas superior, frontal y lateral, además de sus propiedades como sus materiales y texturas. La importancia que tiene este programa para esta tesis radica en el cambio de la textura del modelo por la textura con formato *.DDS*, además de que en él pueden visualizarse los modelos con todas sus características como lo es su malla de colisión.

Entonces primeramente es necesario cargar el modelo con formato *NIF*, estos deben encontrarse en la ruta: *C:\Oblivion\Data\Meshes*, para ello es necesario ir al menú *archivo* → *cargar*, o directamente presionando el botón cargar. Una vez cargado el archivo a mano derecha se encuentra la vista previa del modelo cargado, mientras que a la izquierda tenemos las propiedades del objeto.

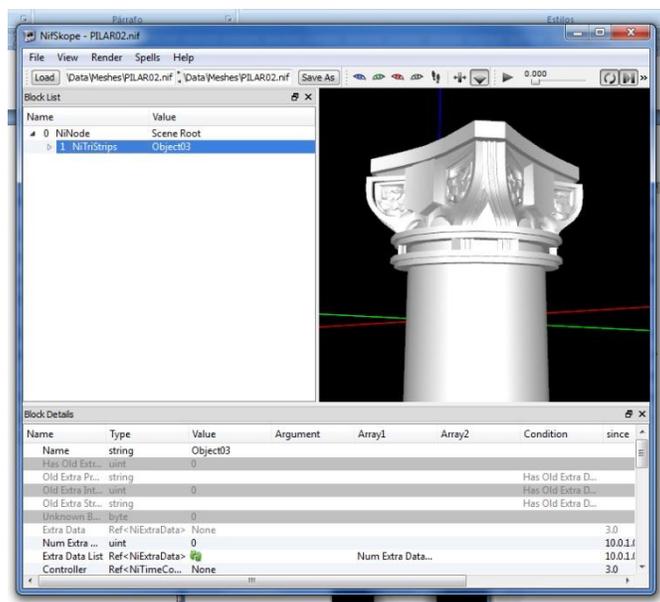


Figura IV.LXVIII Carga del modelo del pilar en formato NIF en *NifScope*.

En las propiedades del objeto se presiona sobre la flecha que se encuentra a la izquierda de *NiTriStrips* que son las partes por las que está conformado el objeto, cada una puede tener una textura distinta. Una vez presionada la flecha se desplegarán las diferentes propiedades del objeto, como son sus materiales y sus texturas. Para acceder a las texturas nuevamente es necesario presionar sobre la flecha que se encuentra a mano izquierda de *NiTexturingProperty*, desplegándose las propiedades de las texturas, como la selección de la fuente de la textura, para acceder a esta es necesario presionar sobre la flor lila. Abriéndose el cuadro de diálogo de selección de textura, ahí se buscará la textura con formato *DDS*, para que esta pueda ser vista dentro del set de construcción y del mismo juego de *Oblivion*.

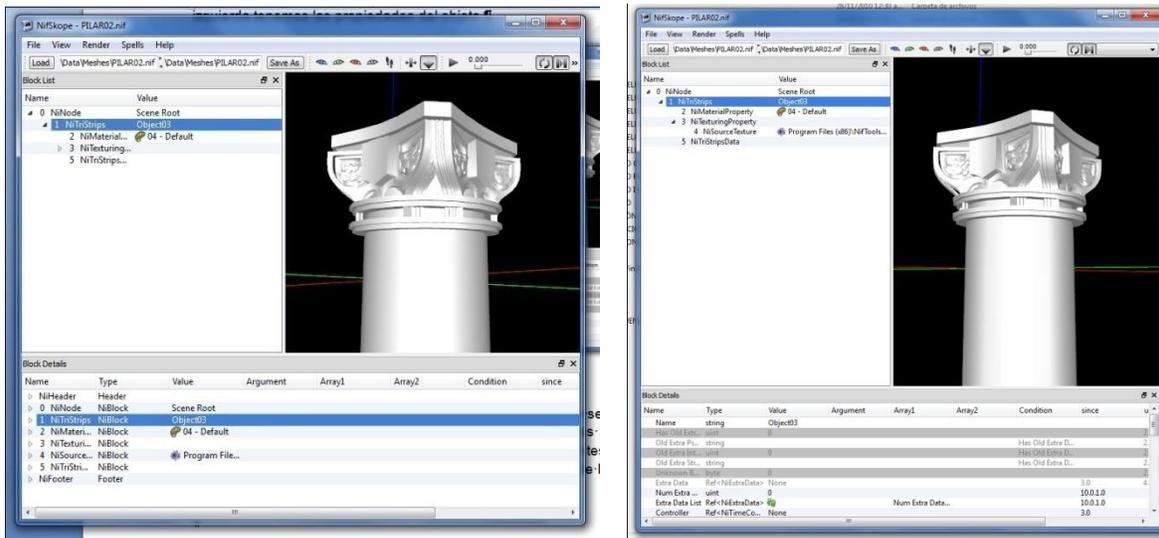


Figura IV.LXIX Despliegue de las propiedades del modelo (izquierda) y de las propiedades de textura (derecha).

Esa es la forma más sencilla de texturizado, si se requiere texturizar una zona en concreto del objeto, se presiona *click* izquierdo del ratón sobre el objeto en la ventana de render, dibujándose la malla del mismo sobre este, permitiéndonos trabajar sobre ella.

Incluso se pueden integrar texturas con mapas de relieve y de reflejos, solamente es necesario guardarlas con el nombre de la textura seguido de un guión bajo y de esta una *n*, presentando con esto el mapa de relieve o *normal mapping*, en el caso de la textura del pilar, nombrada *marmol.dds*, entonces si se integra una textura de mapa de normales sería necesario nombrarla *marmol\_n.dds*.

Con lo anterior se han completado los elementos necesarios para ser insertados dentro del set de construcción y así poder visualizarlos dentro de la ejecución del *Oblivion*.

### IV.IX.III. Manejo e Inserción de Colisiones al Set de Construcción

En esta sección se detallará el proceso de entregarle a objeto tridimensional la propiedad de ser un cuerpo rígido, esto con la finalidad de que este pueda presentar colisiones y a su vez el motor de física Havok® pueda interpretar la malla de colisión del modelo y así interactuar con él.

Entonces se describirá el proceso de cómo se insertaron las mallas de colisión a los modelos desarrollados para el paseo virtual de Palacio de Bellas Artes.

Una vez que se tiene el modelo texturizado, está casi listo para ser exportado, pero antes de ello es necesario introducirle la malla de colisión, para ello es necesario colocar una caja sobre el modelo que se desea que tenga la caja de colisión, cubriendo el área que va a tener la malla de colisión. Para ejemplificar se utilizará la base de los pegasos de la plaza del palacio de bellas artes, debido a que esta es un sólido completo, se cubrirá totalmente por la caja de colisión.

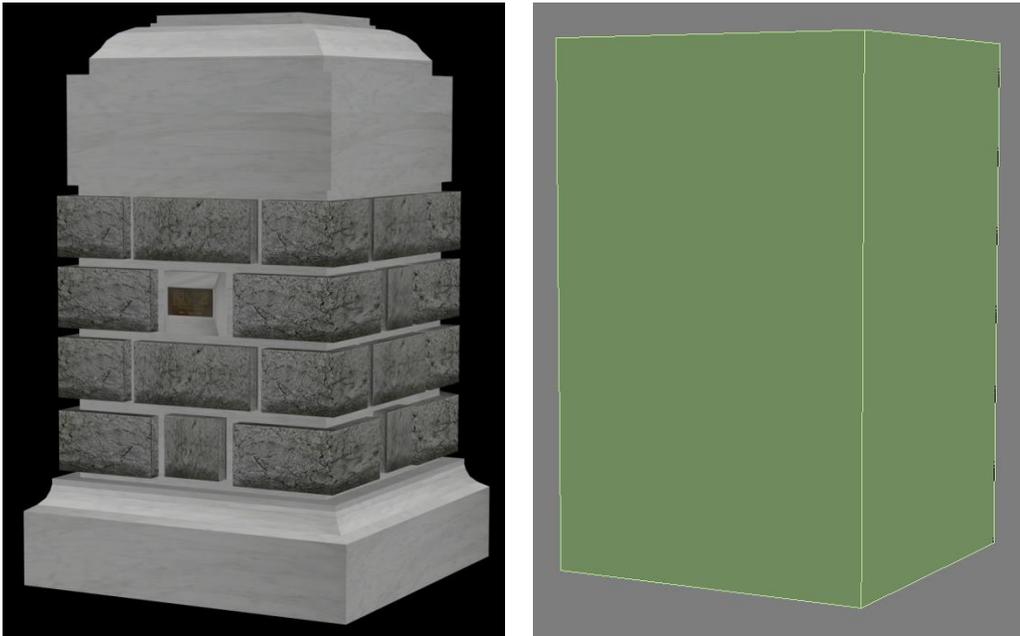


Figura IV.LXX Base de los pegasos texturizada izquierda, caja cubriendo en su totalidad el modelo derecha.

El siguiente paso es asignarle a la caja de colisión el modificador *bhkRigidBody*, esta matriz es la que se encargará de darle a la caja que cubre el modelo la cualidad de ser identificada como una colisión y que no pueda ser atravesado al estar en una posición en la que ya se encuentra el objeto.

Para este caso debido a que la base de los pegasos es de mármol en los parámetros del modificador *bhKRigidBody* se seleccionará en material *pedra*, en *Boundig Volume caja alineada a los ejes*.

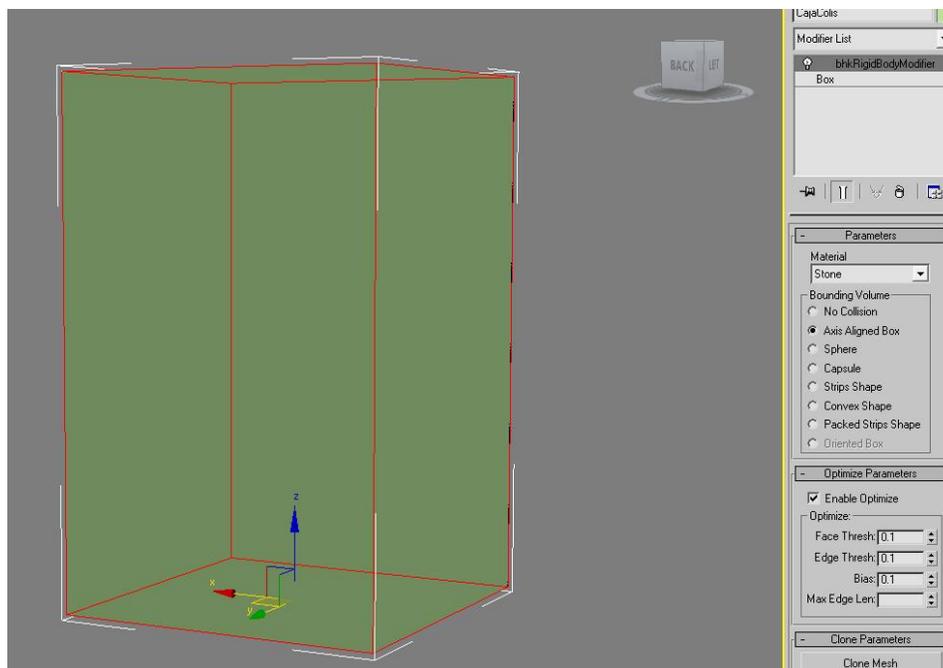


Figura IV.LXXI Caja de colisión teniendo aplicado el modificador *bvhRigidBody* y cambiando los parámetros *Material* y *Bounding Volume*.

Para terminar de asignarle la caja de colisión a los objetos es necesario seleccionar tanto el modelo como la caja de colisión, una vez seleccionados estos se agrupan para tomarlos como un solo objeto, para ello se toma el menú *Grupo* → *Grupo*, presentándose un cuadro de diálogo pidiendo la asignación de un nombre para el nuevo grupo.

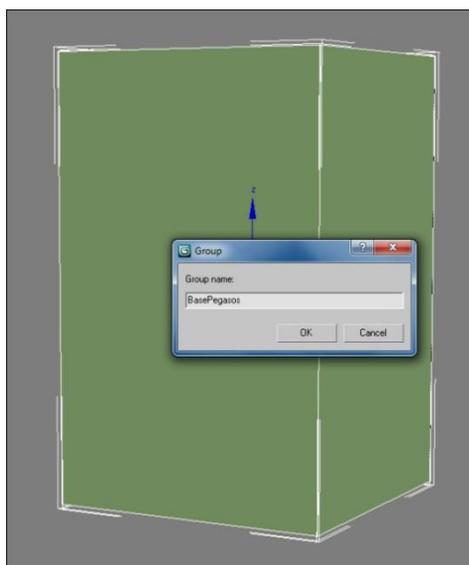


Figura IV.LXXII Se agrupan los objetos y se le asigna un nuevo nombre al nuevo grupo.

Teniendo los dos objetos agrupados lo único que falta es exportar el modelo mediante el menú *Archivo* → *Exportar* seleccionando como tipo de archivo a exportar *NIF*, lo cual se explicará a detalle en la siguiente sección.

#### IV.IX.IV. Integración de Modelos Utilizando el Set de Construcción

En el capítulo anterior se presentó una introducción sobre los mods de videojuegos, sobre el videojuego en el que se basa este trabajo “*The Elder Scrolls IV: Oblivion*” y sobre su set de construcción. En este apartado se explicará primeramente como realizar una correcta instalación del Set de Construcción, su utilización y la integración de los modelos creados para el Palacio de Bellas Artes.

Para ello se necesitan se utilizaron los siguientes archivos:

- OBMM Setup.
- TES Construction Set 1.0 y 1.2.404
- Patch Construction Set v1.2.404
- NifTools Max Plugins 3.4.2.4632

Los cuales pueden ser descargados de la página principal del Set de Construcción: [http://cs.elderscrolls.com/constwiki/index.php/Main\\_Page](http://cs.elderscrolls.com/constwiki/index.php/Main_Page). El primero *OBMM Setup* u *Oblivion Mod Manager* es el archivo de instalación con el cual se realizará la modificación del juego de Oblivion. *TES Construcción Set* y el archivo *Patch*, son las herramientas que permite editar y crear cualquier dato que será usado en *The Elder Scrolls IV: Oblivion*, permitiendo la lectura de archivos *.ESM* o *.ESP* que son los archivos almacenados en los archivos del juego, permitiendo la creación de áreas propias como: pueblos, calabozos e islas y poblarlas con personajes, edificios, creaturas, ítems, entre otros.<sup>225</sup> El archivo ejecutable *NifTools* permite al software *3DS MAX®* de *Autodesk®* exportar archivos en formato NIF, es un formato de archivos de modelos 3D, en este formato serán exportados los modelos del Palacio de Bellas Artes para ser integrados mediante el set de construcción.

Para su instalación, primeramente es necesario el *OBMM Setup*, para ello es necesario ejecutarlo e instalarlo en la carpeta donde se encuentran instalados los archivos del juego de *Oblivion*, esto a partir de Windows Vista, para evitar problemas con las direcciones, es necesario instalarlos directamente en la raíz del disco C. Por ello se instalarán los archivos en la dirección *C:\Oblivion*.



Figura IV.LXXIII Ventana de instalación de Oblivion Mood Manager.

<sup>225</sup> [http://cs.elderscrolls.com/constwiki/index.php/Main\\_Page](http://cs.elderscrolls.com/constwiki/index.php/Main_Page)

Al terminar la instalación será lanzado el *Oblivion Mod Manager*, este es el asistente que muestra los mods que están instalados en el sistema. Para poder realizar la modificación del juego solo es necesario activar la casilla con el nombre *Oblivion.esm* en la parte superior izquierda del *Mod Manager*, que es el archivo principal del juego de *Oblivion*.

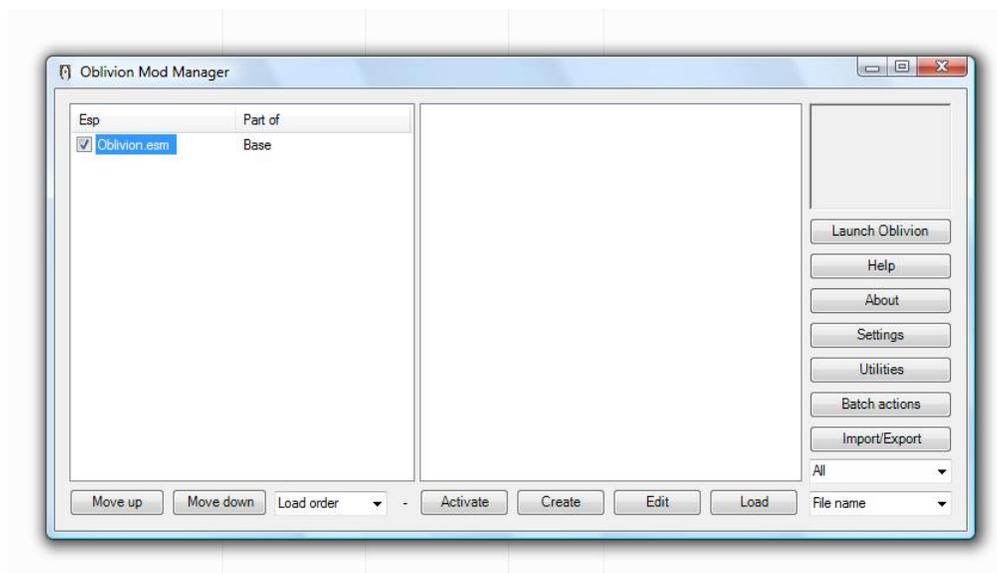


Figura IV.LXXIV Selección del archivo maestro de Oblivion en la ventana de *Oblivion Mod Manager*.

Para su configuración y poder evitar problemas con las texturas es necesario ir a la opción *Utilities* → *Archive Invalidation* y marcar las opciones que se muestran en la siguiente imagen.

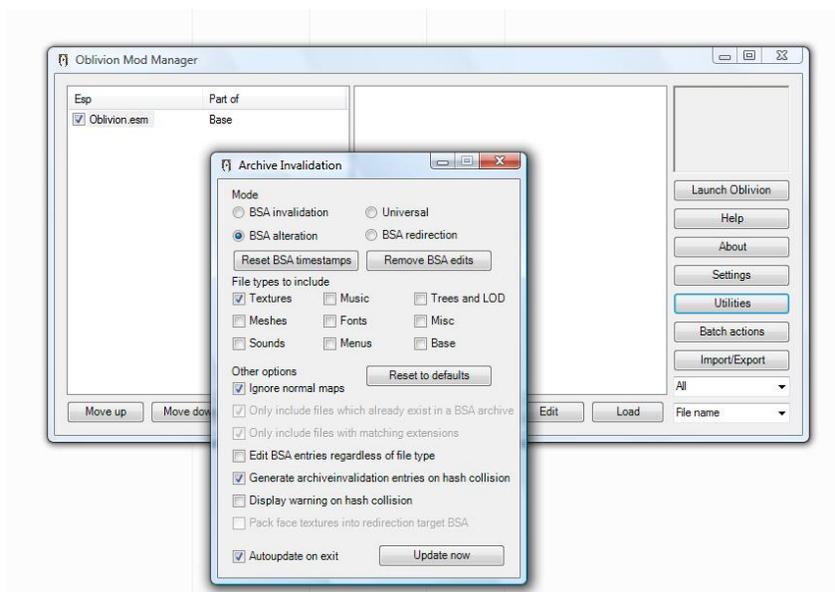


Figura IV.LXXV Configuración Oblivion Mod Manager.

Con lo anterior se sustituirán los archivos principales de texturas BSA, por archivos más recientes de forma fiable que se encuentran en la carpeta de texturas, y con ello se evitan problemas al cargar de texturas durante la ejecución del sistema.

Posteriormente se presiona *Update now* y en la pantalla anterior *Activate*. El siguiente paso es la instalación del set de construcción y su parche.

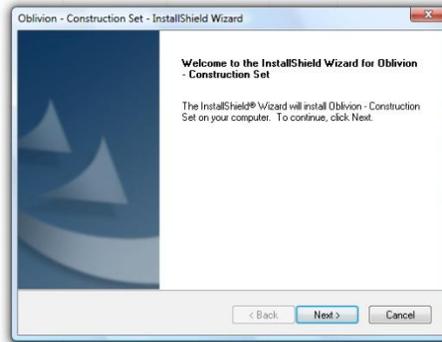


Figura IV.LXXVI Ventana de instalación del set de construcción.

Una vez instalados se tiene la siguiente pantalla:

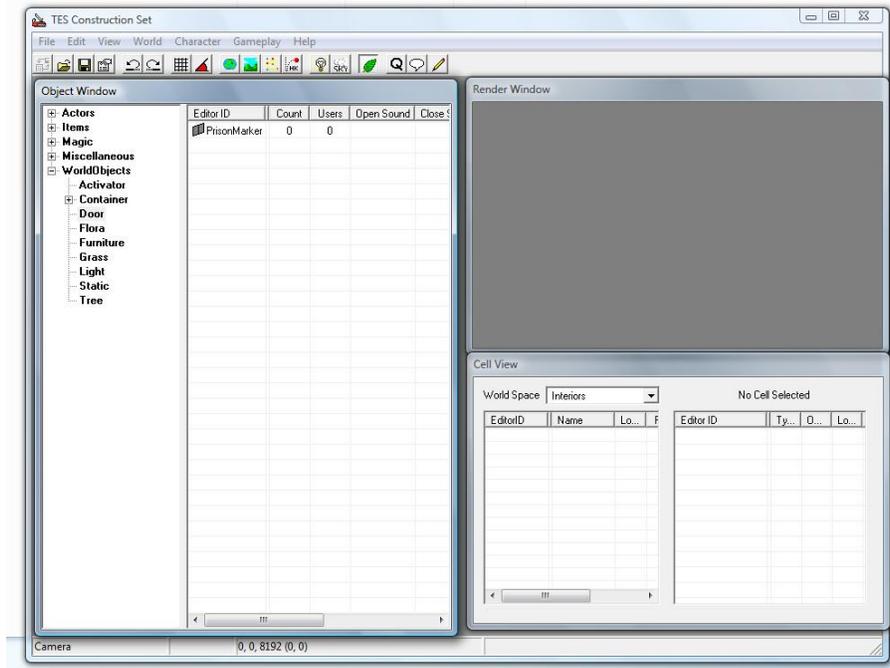


Figura IV.LXXVII Ventana principal del set de construcción.

Posteriormente es necesario entrar al menú *File* → *Data* y marcar el archivo maestro Oblivion.esm, para que este pueda ser editado por el set de construcción. Una vez marcado el archivo maestro, tenemos la ventana principal del set de construcción, esta cuenta con tres sub ventanas:

- La ventana objeto desde la cual los objetos son lanzados a la ventana de renderizado, contiene una base de datos con todos los objetos que están dentro del juego, se encuentran almacenados por categorías dependiendo de la función del objeto.
- La ventana de vista de celdas permite la selección de las celdas o mundos del juego. A la izquierda se encuentra la lista de selección de las celdas y a la derecha la lista con todos los elementos que están contenidos en las celdas. La lista desplegable permite moverte de los interiores a los diferentes mundos exteriores. Al presionar dos clicks sobre la celda seleccionada en la ventana de renderizado se dibujará la celda.

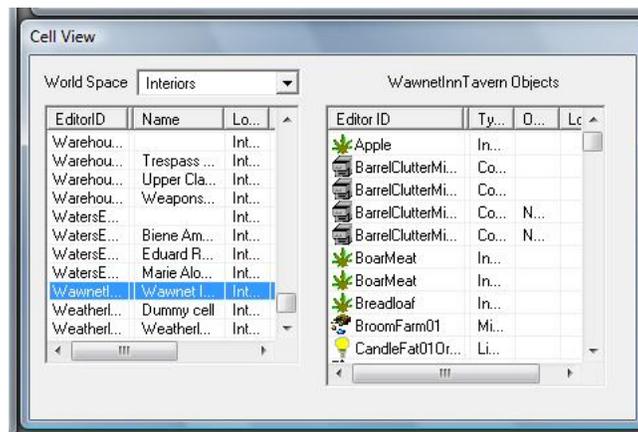


Figura IV.LXXVIII Ventana de vista de celdas

- La ventana de renderizado es aquella en la que es posible ver los resultados, en otras palabras es la vista del mundo que se está creando o modificando. Para ello es necesario tener un buen manejo de la cámara en esta ventana, debido a que locaciones de las celdas pueden ser muy grandes y contener elementos muy pequeños dentro de ellas. Entonces para poder desplazarse dentro de la ventana se utilizan las teclas de la cruz direccional del teclado o manteniendo presionado la barra espaciadora y mientras el ratón es desplazado para manejar los ejes X y Y, mientras que para el manejo de la profundidad del eje Z se utilizará el scroll del ratón.

Para deseleccionar los elementos es necesario dar un *click izquierdo* con el ratón sobre el área gris de la ventana o presionar el *botón d*. Para seleccionar los elementos es necesario dar un *click izquierdo* del ratón o dibujando un cuadro de selección sobre el elemento, de esta forma aparecerá una caja multicolor sobre el elemento seleccionado, esta caja define las dimensiones del elemento seleccionado a la vez que es usada como la caja de colisión por el motor de física. Al presionar *doble click* sobre ella aparecerá una caja de diálogo entregando información sobre ella.

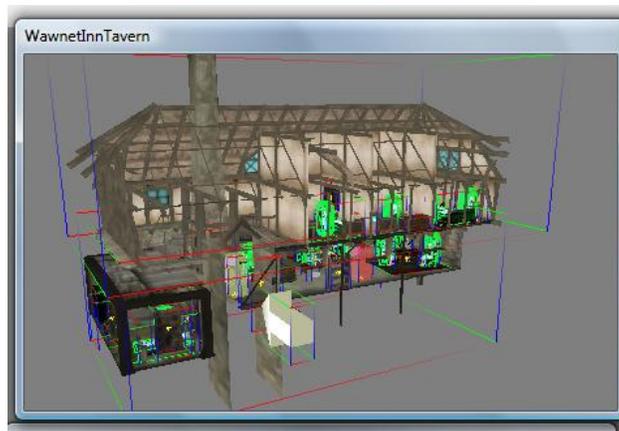


Figura IV.LXXIX Ventana de renderizado del set de construcción.

La rotación de la cámara se realiza presionando la *tecla shift* en conjunción con el movimiento del ratón; la cámara rota alrededor del objeto seleccionado. Si se presiona la *tecla c* la vista de la cámara es centrada a dicho objeto.

Para poder manipular los objetos primeramente es necesario seleccionar el objeto, mantener presionado el botón izquierdo del ratón y mover a este último hasta posicionar al objeto deseado en su nueva localización, con esto se tiene un movimiento en los ejes X y Y, para manipularlo en el eje Z, es necesario tomar el objeto y presionar la *tecla z*, presentándose con esto un movimiento en profundidad. Para rotar los objetos en el set de construcción es necesario seleccionar el objeto a rotar, posteriormente presionar el botón derecho del ratón, desplazar a este último, con ello se produce una rotación sobre el eje Z, para producir rotaciones en los ejes X y Y, es necesario mantener presionadas esas respectivas teclas y presionar el botón derecho del ratón, mientras este es desplazado.

Para realizar la introducción de modelos al set de construcción, en 3DS MAX se abre el archivo que será introducido al juego. Es necesario que todos los elementos del modelo sean uno solo, para ello se utiliza la herramienta Attach, con esta herramienta se integraran uno a uno los elementos que se encuentran separados en el archivo, formando así un solo elemento. Posteriormente, se exporta el modelo, para ello se presiona sobre el menú archivo → exportar, abriéndose un cuadro de diálogo donde se introducirá el nombre con el que se identificará el archivo exportado y abajo del recuadro del nombre del archivo seleccionar en la lista desplegable la opción “netinmerse gamebryo / KF, NIF”, con la cual se selecciona el formato NIF para el archivo a exportar y se presiona guardar.

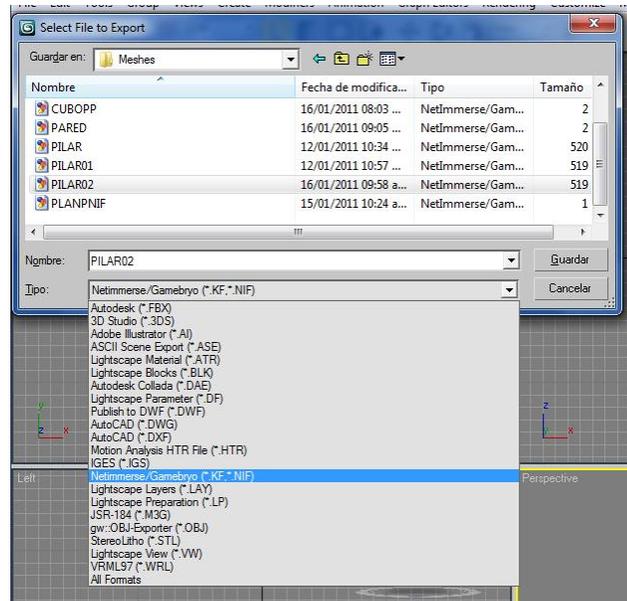


Figura IV.LXXX Exportado de modelos en formato NIF desde 3DS MAX®.

Una vez que el *archivo NIF* ha sido texturizado y ha pasado por el software *NIFScope*, en el set de construcción, en la ventana objeto se introduce un *click izquierdo* sobre los objetos de mundo desplegándose una lista con las categorías en las que se encuentran almacenados todos los objetos del juego de Oblivion, se presiona sobre la categoría que sea más acorde al objeto a introducirse y en la lista de la derecha aparecerán todos los objetos que corresponden a la categoría seleccionada, se selecciona cualquiera de ellos y presiona el botón derecho del ratón, de la lista que se despliega se selecciona la opción nuevo, apareciendo el siguiente cuadro de diálogo:

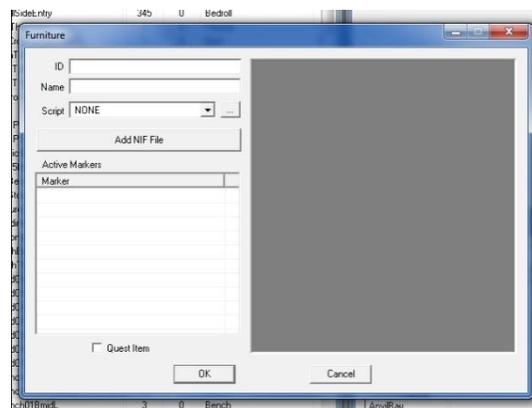


Figura IV.LXXXI Ventana de introducción de nuevos objetos al set de construcción.

En la ventana anterior, al presionar sobre el *botón agregar archivo NIF* se abre el cuadro de diálogo que se muestra a continuación en el cual se seleccionará el archivo NIF a cargar en el set de construcción.

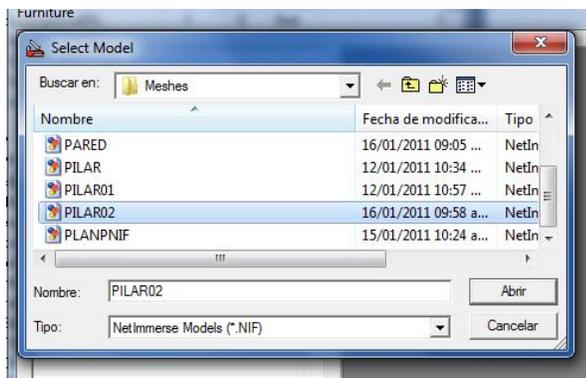


Figura IV.LXXXII Selección del objeto NIF que será introducido al set de construcción

Ya que se tienen los modelos NIF que serán cargados en el set de construcción, se procederá a introducirlos en el mundo, para ello, primeramente será necesario seleccionar el área exterior designada para tal acción. En la ventana de la vista de celdas se presionará en el menú desplegable y se seleccionará la opción *Tamriel*, ya que en esta área del mundo de *Oblivion* se encuentran grandes zonas prácticamente despejadas que serán utilizadas para aplicar nuestros propósitos.

Las áreas que prácticamente están desiertas aparecen con el nombre de *Wilderness*, una vez seleccionada una de ellas se presiona *doble click izquierdo* sobre ella y en la ventana de render aparecerá dibujada esta área.

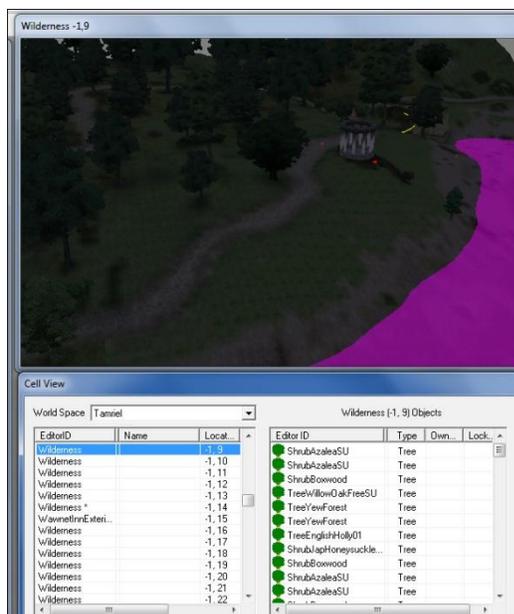


Figura IV.LXXXIII Selección del área donde serán colocados los modelos del Palacio de Bellas Artes.

En ella se procederá a introducir los modelos del Palacio de Bellas Artes, editar el terreno y presentarlo de tal forma que el paseo virtual de una impresión de real, para ello es necesario dirigirnos a la ventana de objetos e ir seleccionando uno a uno los modelos a insertar, ya que fue seleccionado el objeto se toma presionando *click izquierdo* sobre él y se arrastra hasta la ventana de render lo más cercano a la posición que se desee que este vaya a ocupar.

De esta forma se insertaran los modelos desarrollados anteriormente hasta conformar por completo el Palacio de Bellas Artes y sus exteriores.



Figura IV.LXXXIV Entrada principal introducida usando el procedimiento de inserción explicado anteriormente.

### IV.IX.V. Inserción de Diálogos a los Personajes

Para cumplir con el objetivo de que el paseo virtual sea una herramienta educativa y a su vez cultural, es necesario que el sistema tenga la capacidad de entregar información acerca de la historia del Palacio de Bellas Artes, es por ello que se tomó la decisión de insertar personajes dentro del ambiente virtual, los cuales al hablar entregarán la información en forma de texto, explicando algunos de los aspectos más representativos de la historia de este importante edificio.

Para ello es necesario ejecutar el software del set de construcción de *Oblivion*, una vez abierto se necesita definir una *quest* para poder agregar un dialogo, entonces se presionará el botón que tiene una Q dibujada sobre él.

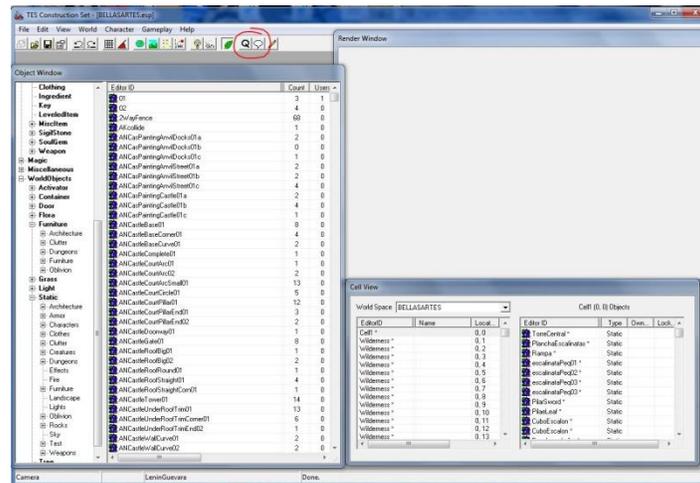


Figura IV.LXXXV Botón Quest del set de construcción de *Oblivion*.

Una vez presionado este botón, el set de construcción dibujará la ventana de las *quest* del juego.

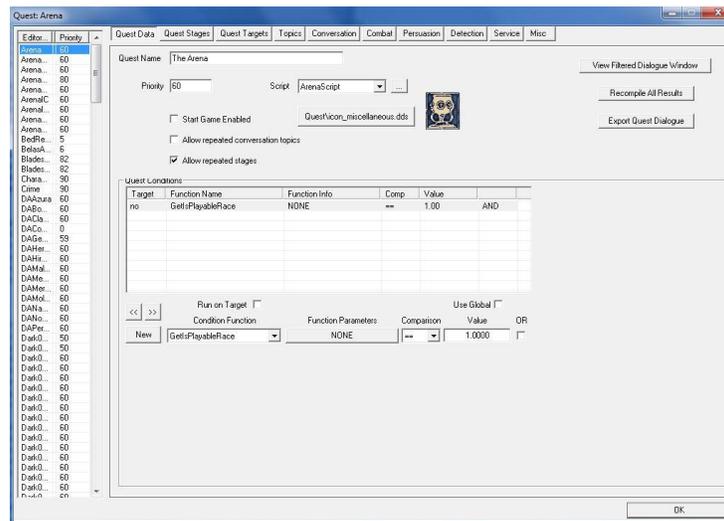


Figura IV.LXXXVI Ventana de las *Quest* del juego del set de construcción de *Oblivion*.

En la ventana de las *quest* a la izquierda se encuentra una tabla con los campos *EditorID* y *Priority*, para poder crear una *quest* es necesario colocar el cursor sobre cualquiera de los elementos que se encuentran en el campo *EditorID*, presionar *click* derecho con el ratón y seleccionar la opción *new*, dibujándose una pequeña ventanita en la cual se introducirá el nombre de la *quest* nueva, que para este caso fue *Belasart01*.

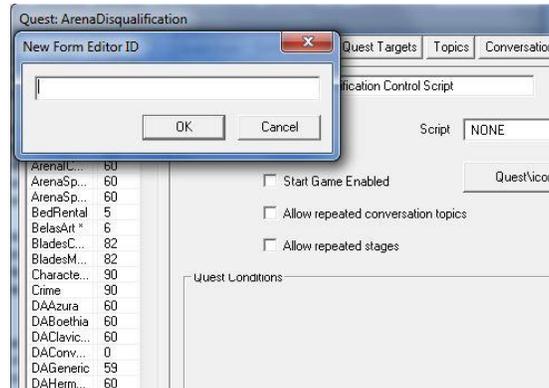


Figura IV.LXXXVII Ventana nueva *Quest*.

El siguiente paso es seleccionar la nueva *quest* que acaba de ser creada, al hacerlo se mostraran las opciones de ésta. Es necesario cambiar la opción *priority* por el valor de 6, al hacer esto los saludos que sean agregados al personaje tendrán una prioridad mayor a lo normal y serán los primeros en mostrarse. También se debe de habilitar la opción *start game enabled*, en la lista de condiciones es necesario agregar el valor *GetIsPlayableRace == 1*, la opción *Target = no* y finalmente la opción *info = none*. Estas opciones se muestran en la siguiente imagen:

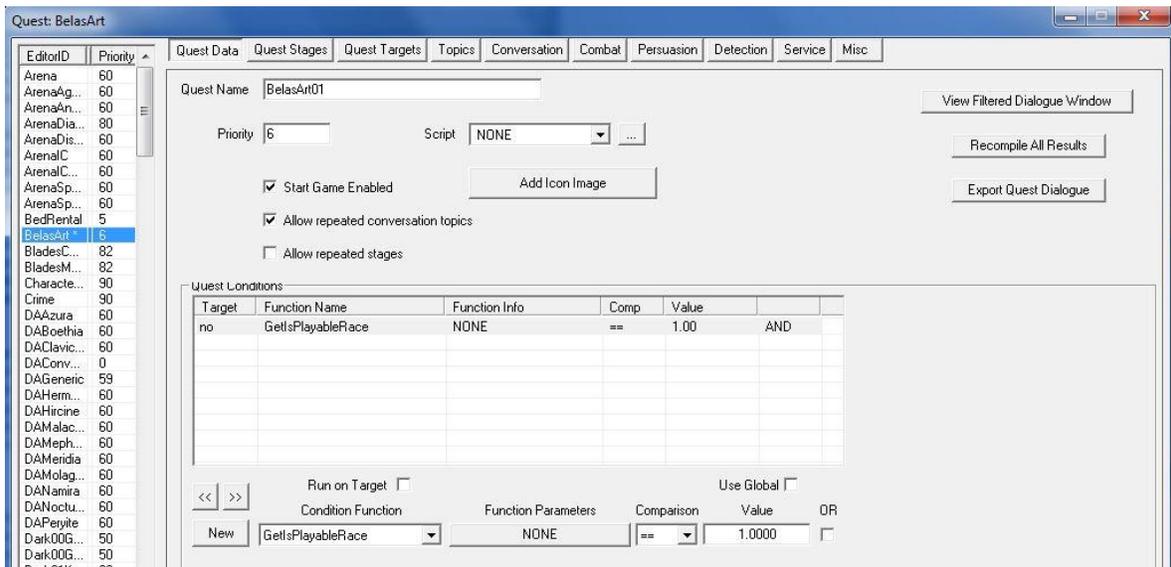


Figura IV.LXXXVIII Valores de la ventana *Quest: BelasArt01*.

El siguiente paso es presionar el botón *Topics*, presentándose una nueva ventana.

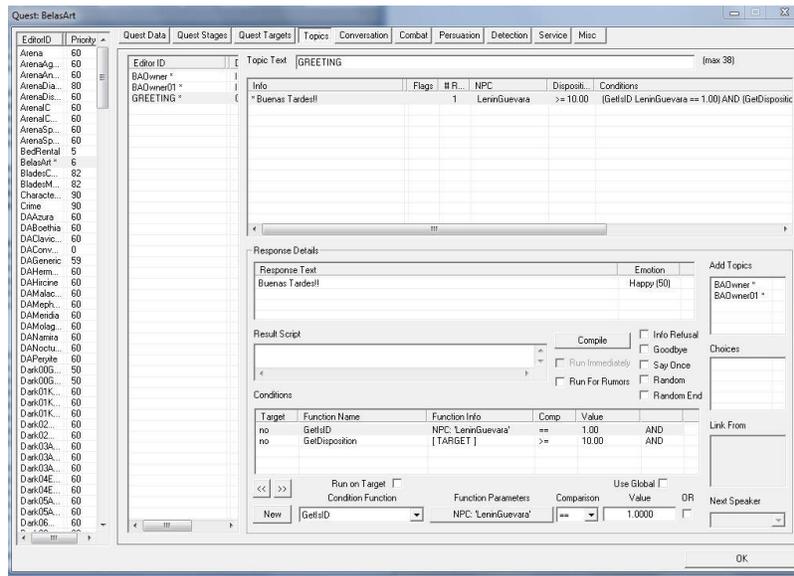


Figura IV.LXXXIX Valores de la ventana *Topics de BelasArt01*.

Entonces se procederá a crear un tema o *topic*, al cual se le insertará la información, que posteriormente será desplegada por el personaje al hablar con él. Para crear el *topic* se posiciona el cursor sobre la lista del *Editor ID*, se presiona el *click* derecho del ratón y se elige la opción *Add Topic*, con lo anterior se dibujará una nueva ventana con una lista de *topics*.

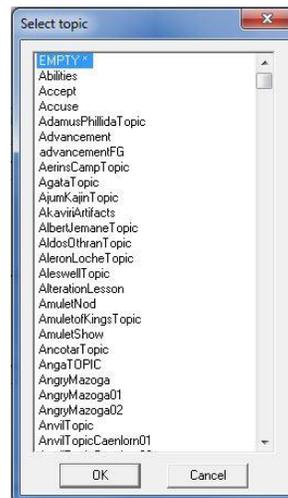


Figura IV.XC Ventana de selección de un nuevo *Topic*.

De la lista se seleccionará el *topic* existente de nombre *Greeting*, al seleccionarlo podrá ser visualizado este nuevo *topic* en el *Editor ID*, pero la información de éste aún se encuentra vacía.

Una vez que se tiene el tema creado se procede finalmente a la inserción de la información a éste. Para ello es necesario colocar el cursor sobre la lista en el campo *info*, presionar *click* derecho y seleccionar la opción *new*. Dibujándose una nueva ventana, en la cual se insertará la información que queremos que diga el personaje.

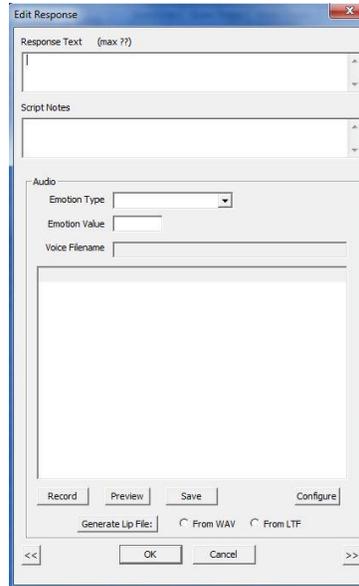


Figura IV.XCI Ventana de inserción de información del nuevo tema.

Al regresar a la ventana de topics al fondo se puede observar la lista de condiciones, en ella se presionará el botón *new*, insertando los siguientes valores: Target: no, Getdisposition Target  $\geq 10$  (significando que la información será hablada si la disposición del personaje es mayor o igual que 10) y el más importante en Function Info: se selecciona al personaje que se desea que despliegue la información al hablar.



Figura IV.XCII Personaje desplegando el saludo indicado.

Al hablar con el personaje es posible desplegar un menú con opciones de información que se desea que este diga. Para ello es necesario volver a la ventana de *quest / topics*, nuevamente se posiciona el curso sobre la lista del *Editor ID* se presiona *click* derecho y se selecciona la opción *Add Topic*, en la lista de selección de *topic* se presiona *click* derecho y se elige la opción *new*, con ello la lista permitirá que se inserte el nombre del nuevo *topic*, al cual se le dará el nombre de *Información01*.

Se procederá siguiendo los pasos anteriores de inserción de valores e información, con la excepción de que se marcará la casilla *Run For Rumors*. Con ello, al hablar con el personaje se desplegará una lista con la información que este tiene para conversar e informar al usuario que esté utilizando el paseo virtual.



Figura IV.XCIII Personaje desplegando el menú de información.



Figura IV.XCIV Personaje desplegando la información que está contenida en el menú de información.

#### IV.IX.VI. Resultados Finales

Ya que se conoce el procedimiento mediante el cual se realizó el trabajo del Paseo Virtual del palacio de Bellas Artes sobre la plataforma de *Oblivion*, a continuación se mostrarán los resultados obtenidos en este proyecto.

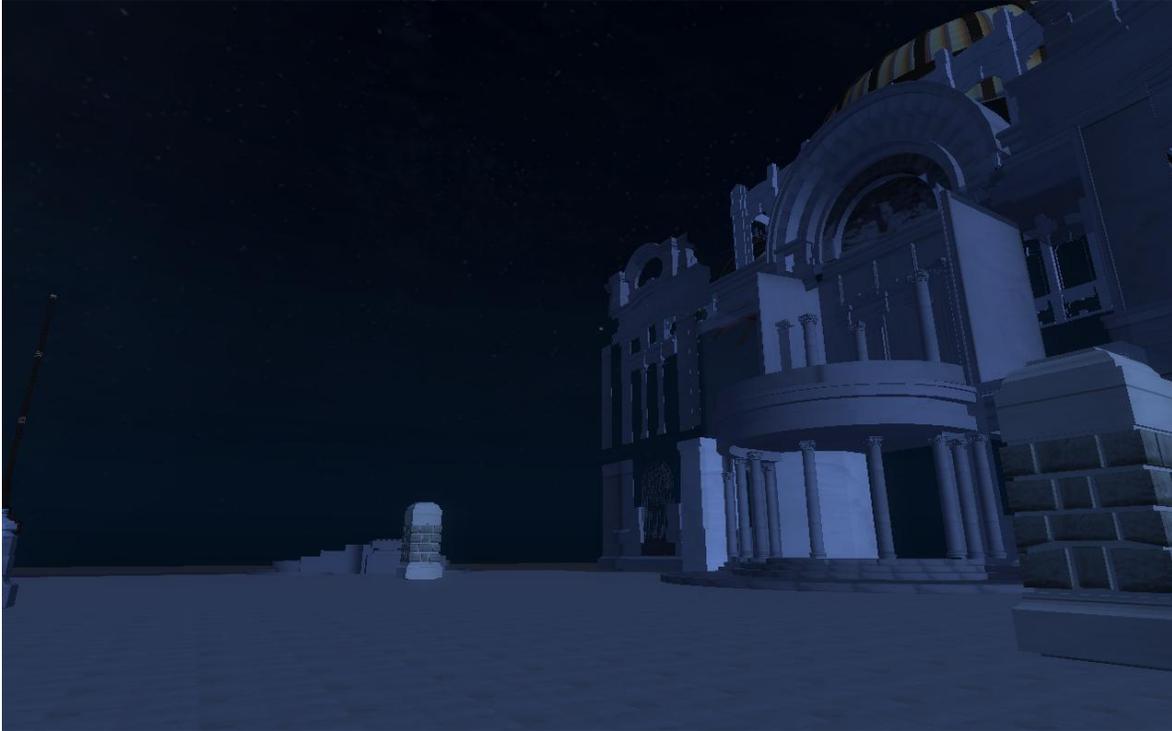


Figura IV.XCV Palacio de Bellas Artes Frente y plaza, modelos ejecutándose sobre el sistema de Oblivion.

El principal motivo por el cual se eligió el software de *Oblivion* son sus ambientes e iluminación, produciendo una inmersión de realidad inmensa. Este permite una gran flexibilidad en el manejo de climas (despejado, nublado, lluvioso, entre otros), en el cambio de iluminación a lo largo del día (luz solar, amanecer, atardecer y paisaje nocturno) e implementación de iluminación no natural (permite la inserción de luces spot, hogueras, antorchas, entre otros diversos tipos). Como puede ser apreciado en las siguientes pantallas:

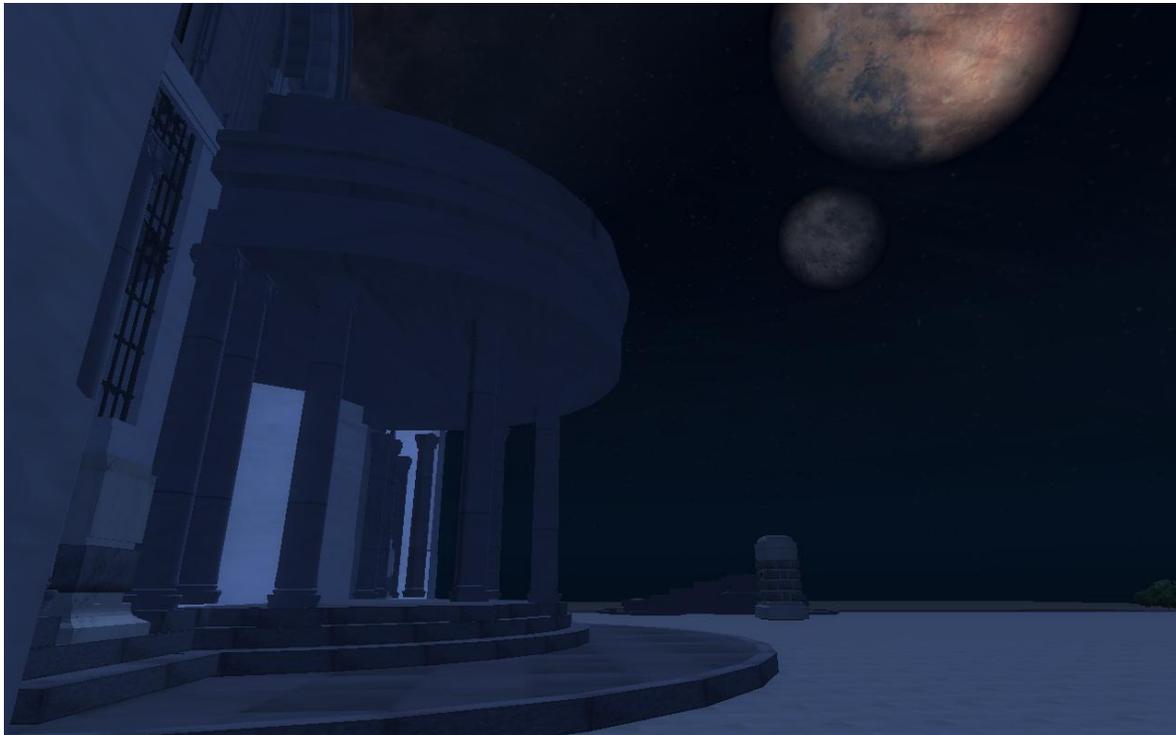


Figura IV.XCVI Escalinata Principal y plaza durante el paisaje nocturno.

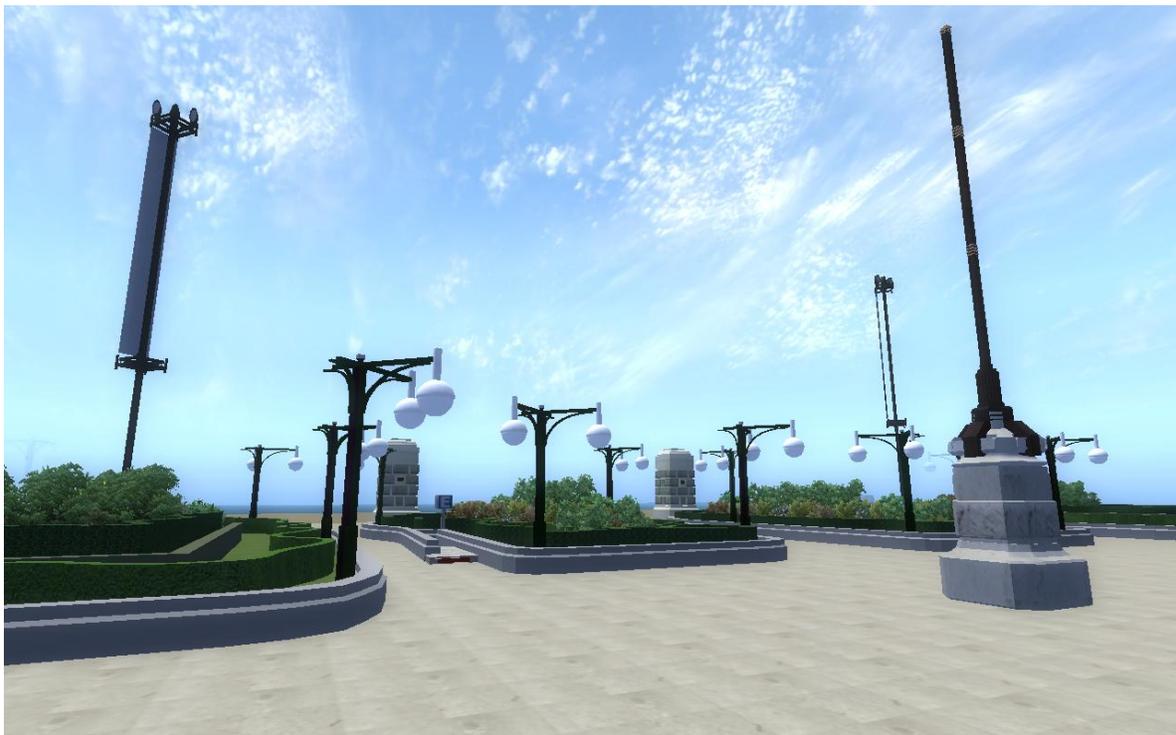


Figura IV.XCVII Plaza del Palacio de Bellas Artes en un ambiente despejado.

Otro de los motivos por los que se decidió el uso de este sistema es su manejo de los algoritmos de recorte, ya que mientras se recorre la amplia plaza el sistema nos hace

notar que se está cargando un área nueva, manejando un dibujado por aproximación, esto es conforme mi personaje o punto de vista se aproxima a los objetos estos se dibujan y cambian de tonalidad de ser transparentes a los colores sólidos de las texturas.



Figura IV.XCVIII Parte superior dibujada parcialmente, mientras que la parte izquierda dibujada en pantalla, pero aún transparente debido a la aproximación

El beneficio del recorte se presenta en la carga de objetos con un alto conteo de polígonos, este fue uno de los objetivos desde el inicio del desarrollo de este proyecto, el desarrollo de modelos de gran calidad y detalle, recurriendo al mapeado de texturas lo menos posible. Presentando un *lag* mínimo e incluso nulo, como en la plaza que tiene gran número de objetos. Para ejemplificar, las puertas laterales con las serpientes se desarrollaron con un poco más de 30,000 polígonos.

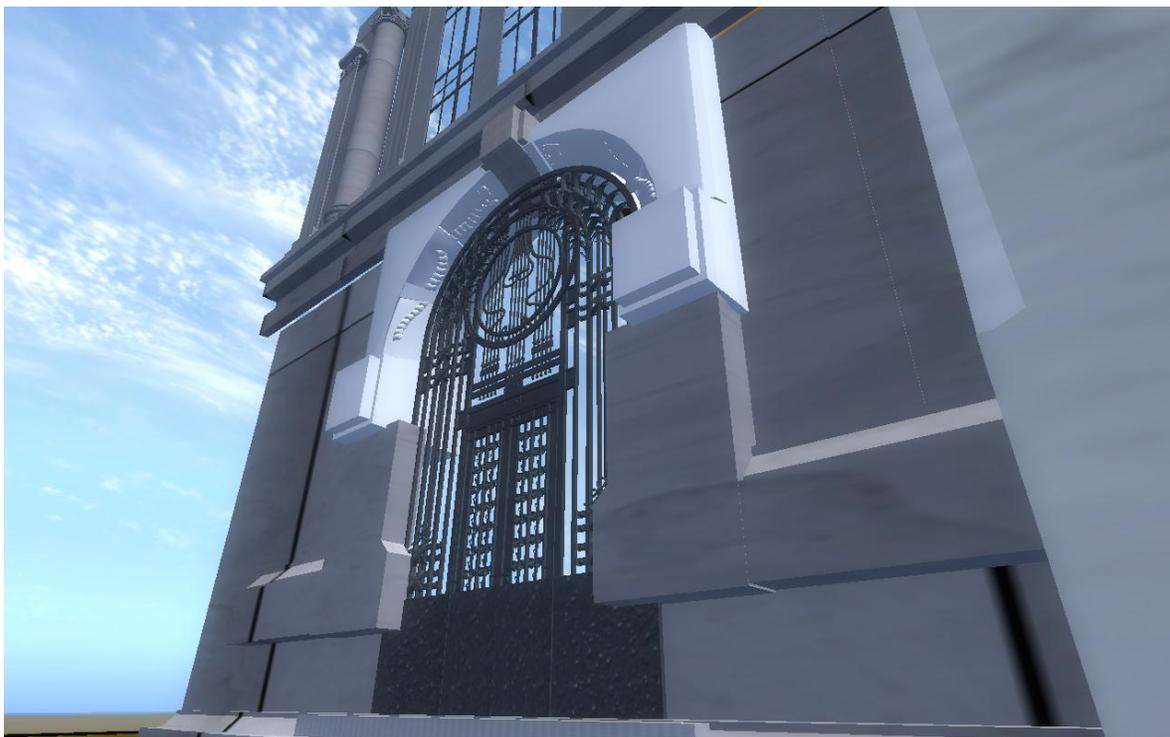


Figura IV.XCIX Puerta lateral detallada en diferentes ambientes y puntos de vista.

Otro punto a notar es la física del motor *Havok*®, presentado en todos los objetos de la plaza, en las escalinatas, la plancha, las jardineras, las fuentes, las bases de las estatuas y los pegasos, los semáforos, letreros y paredes. Permitiendo que el personaje suba

escaleras, que los objetos no sean atravesados cuando el personaje se acerca a su posición gracias a la malla de colisión, que sea posible una libre interacción con las jardineras, las fuentes, calles y en general a lo largo de todo el ambiente del Palacio de Bellas Artes.



Figura IV.C Personaje mostrándose parado sobre el borde de una de las jardineras.

Se mostrarán algunos renders más del proyecto. Mostrando el detalle y texturizado en los objetos del edificio y la plaza.











Figura IV.CI Detalle y texturizado de los objetos del Palacio de Bellas Artes y su plaza.

