



## 6 Formatos de modelos 3D

En este proyecto se hizo uso del formato OBJ para exportar los modelos de huesos de 3ds Max a Mudbox y así remarcar detalles de importancia de cada hueso, también se usó el formato WRL para exportar las animaciones hechas en 3ds Max con el fin de integrarlas en Amira para su demostración.

### 6.1 El formato OBJ

El formato de archivo OBJ es un formato 3D creado por Wavefront Technologies para su producto Advanced Visualizer. Estos archivos pueden estar en formato ASCII (.obj) o formato binario (.mod).

Los archivos OBJ soportan polígonos que usan puntos, líneas y caras, también soporta objetos de forma libre que usan curvas y superficies.

Los archivos OBJ pueden tener los siguientes tipos de datos:

Comentarios que inician con el carácter #

Vértices que pueden ser de diferentes tipos:

- Vértices geométricos (v)
- Vértices de textura (vt)
- Vértices normales (vn)
- Vértices con parámetros de espacio (vp)
- Atributos de curvas y superficies (cstype)
- Grados (deg)
- Matriz base (bmat)
- Step zise (step)

Diversos elementos como:

- Punto (p)
- Línea (l)
- Cara (f)
- Curva (curv)
- Curva 2D (curv2)
- Superficie (surf)



Formas libres así como:

- Parámetros (parm)
- Bucle exterior de recorte (trim)
- Bucle interior de recorte (hole)
- Curva especial (scrv)
- Punto especial (sp)
- Fin (end)

Conectividad entre superficies en la forma libre (con)

Grupos como son:

- Nombre (g)
- Suavizado (s)
- Fusión (mg)
- Nombre de objeto (o)

Y finalmente atributos de visualización/render:

- Interpolación de orden (bevel)
- Interpolación de color (c\_interp)
- Interpolación de disolución (d\_interp)
- Nivel de detalle (lod)
- Nombre del material (usemtl)
- Librería de materiales (mtllib)
- Shadow casting (shadow\_obj)
- Ray tracing (trace\_obj)
- Técnica de aproximación de curva (ctech)
- Técnica de aproximación de superficie (stech)

Este es un ejemplo muy simple de un archivo OBJ que visualiza un cubo:

```
# Vértices: 8
# Puntos: 0
# Líneas: 0
# Caras: 6
# Materiales: 1

# Lista de vértices
v -0.5 -0.5 0.5
v -0.5 -0.5 -0.5
v -0.5 0.5 -0.5
v -0.5 0.5 0.5
v 0.5 -0.5 0.5
v 0.5 -0.5 -0.5
```



v 0.5 0.5 -0.5

v 0.5 0.5 0.5

# Punto/Línea/Lista de caras

usemtl Default

f 4 3 2 1

f 2 6 5 1

f 3 7 6 2

f 8 7 3 4

f 5 8 4 1

f 6 7 8 5

# Fin del archivo

## 6.2 El formato WRL

VRML (Virtual Reality Modeling Language) es un formato de archivo (con extensión .wrl) normalizado que tiene como objetivo la representación de escenas u objetos interactivos tridimensionales; diseñado particularmente para su empleo en la web.

El lenguaje VRML posibilita la descripción de una escena compuesta por objetos 3D a partir de prototipos basados en formas geométricas básicas o de estructuras en las que se especifican los vértices y las aristas de cada polígono tridimensional y el color de su superficie.

VRML permite también definir objetos 3D multimedia, a los cuales se puede asociar un enlace de manera que el usuario pueda acceder a una página web, imágenes, vídeos u otro fichero VRML de internet cada vez que haga clic en el componente gráfico en cuestión.

Los archivos WRL pueden tener los siguientes tipos de datos:

Comentarios que inician con el carácter #

Nodos que describen el contenido de la escena, así como luces, formas, sonidos, etc.

Nodos shape que construyen formas, las cuales son los bloques de construcción de un mundo, pueden ser geométricas:

- Box
- Cone
- Cylinder
- Sphere



- Text

y formas definidas en base a puntos como:

- Elevationgrid
- Extrusion
- IndexedFaceSet
- IndexedLineSet
- PointSet

Nodos de apariencia donde se define el material y textura de la forma

Nodos de agrupamiento como son:

- Transform que contiene el escalado, translación y rotación de formas
- Group que crea un grupo básico, donde cada nodo dentro del grupo es visualizado
- Switch que crea un grupo donde sólo uno de sus nodos es visualizado
- Billboard que crea un grupo con un sistema especial de coordenadas, por ejemplo rotar sobre el eje x
- Anchor crea un grupo activable el cual devuelve el contenido de una url cuando el usuario lo activa
- Inline crea un grupo especial a partir de otros archivos WRL seleccionados por una url
- Collision que detecta las colisiones del usuario con las formas pertenecientes al grupo
- LOD crea un grupo de formas que describen diferentes niveles de detalle o versiones de la misma forma

Este es un ejemplo de un cubo que se traslada sobre el eje X:

#Cubo

```
DEF Box01 Transform {
  translation 0 0 0
  children [
    DEF Box01-TIMER TimeSensor { loop TRUE cycleInterval 3.333 },
    DEF Box01-POS-INTERP PositionInterpolator {
      key [0, 0.03, 0.06, 0.09, 0.12, 0.15, 0.18, 0.21, 0.24, 0.27, 0.3,
        0.33, 0.36, 0.39, 0.42, 0.45, 0.48, 0.51, 0.54, 0.57, 0.6,
        0.63, 0.66, 0.69, 0.72, 0.75, 0.78, 0.81, 0.84, 0.87, 0.9,
        0.93, 0.96, 0.99, 1, ]
      keyValue [0 0 0, 0.04261 0 0, 0.167 0 0, 0.3678 0 0, 0.64 0 0,
        0.9783 0 0, 1.377 0 0, 1.832 0 0, 2.337 0 0, 2.888 0 0,
        3.478 0 0, 4.104 0 0, 4.758 0 0, 5.437 0 0, 6.136 0 0,
        6.848 0 0, 7.569 0 0, 8.293 0 0, 9.016 0 0, 9.731 0 0,
        10.43 0 0, 11.12 0 0, 11.78 0 0, 12.42 0 0, 13.02 0 0,
        13.59 0 0, 14.11 0 0, 14.58 0 0, 15 0 0, 15.36 0 0,
```



```
15.65 0 0, 15.88 0 0, 16.03 0 0, 16.1 0 0, 16.1 0 0,  
] },  
Transform {  
  translation 0 5 0  
  children [  
    Shape {  
      appearance Appearance {  
        material Material {  
          diffuseColor 0.5255 0.02353 0.02353  
        }  
      }  
      geometry Box { size 10 10 10 }  
    }  
  ] }  
]  
ROUTE Box01-TIMER.fraction_changed TO Box01-POS-INTERP.set_fraction  
ROUTE Box01-POS-INTERP.value_changed TO Box01.set_translation  
}  
  
#Fin del archivo
```