Obj

From Wikipedia, the free encyclopedia

OBJ geometry format

Filename extension	`.obj`
Internet media type	`application/x-tgif?`
Developed by	Wavefront Technologies
Type of format	3D model format

OBJ (or .OBJ) is a geometry definition file format first developed by Wavefront Technologies for its Advanced Visualizer animation package. The file format is open and has been adopted by other 3D graphics application vendors. For the most part it is a universally accepted format.

The OBJ file format is a simple data-format that represents 3D geometry alone — namely, the position of each vertex, the UV position of each texture coordinate vertex, normals, and the faces that make each polygon defined as a list of vertices, and texture vertices. Vertices are stored in a counter-clockwise order by default, making explicit declaration of normals unnecessary.

Contents

File format

Lines beginning with a hash character (#) are comments.

```
# this is a comment
```

An OBJ file contains several types of definitions:

```
# List of Vertices, with (x,y,z) coordinates.
v 0.123 0.234 0.345
v ...
...

# Texture coordinates, in (u,v[,w]) coordinates, w is optional.
vt 0.500 -1.352 [0.234]
vt ...
...

# Normals in (x,y,z) form; normals might not be unit.
vn 0.707 0.000 0.707
vn ...
...

# Face Definitions (see below)
f 1 2 3
f 3/1 4/2 5/3
f 6/4/1 3/5/3 7/6/5
f ...
...
```

Face definitions

Faces are defined using lists of vertex, texture and normal indices. Polygons such as quadrilaterals can be defined by using more than three vertex/texture/normal indices.

OBJ files also support free form curved surface objects such as NURB surfaces.

There are several ways to define a face, but each face line definition starts with "f" character.

Vertex

A valid vertex index starts from 1 and match first vertex element of vertex list previously defined. Each face can contain more than three elements.

```
f v1 v2 v3 v4 ...
```

Vertex/texture-coordinate

Each texture coordinate index must follow with no space the first slash. Texture coordinates index are optional. A valid texture coordinate index starts from 1 and match first texture coordinate element of texture coordinate list previously defined. Each face can contain more than three elements.

```
  f v1/vt1 v2/vt2 v3/vt3 ...
```

Vertex/texture-coordinate/normal

Each normal index must follow with no space the second slash. Normals index are optional. A valid normal index starts from 1 and match first normal element of normal list previously defined. Each face can contain more than three elements.

```
f v1/vt1/vn1 v2/vt2/vn2 v3/vt3/vn3 ...
```

Vertex/normal

As texture coordinates are optional, one can define geometry without them, but one must put the normal index after second slash.

```
f v1//vn1 v2//vn2 v3//vn3 ...
```

Referencing materials

Materials that describe the visual aspects of the polygons are stored in external .mtl files. The .mtl file may contain one or more named material definitions.

```
mtllib [external .mtl file name]
...
```

Named objects, polygon groups and material groups are specified via the following tags.

```
o [object name]
...
g [group name]
...
usemtl [material name]
...
```

The material name matches a named material definition in an external .mtl file. Each tag applies to all faces following, until another tag of the same type appears. The texture coordinate can be left out if the current material definition does not include a texture.

```
f v1//vn1 v2//vn2 v3//vn3
...
```

Smooth shading across polygons is enabled by smoothing groups

```
s 1
...
# Smooth shading can be disabled as well.
s off
...
```

More than one external MTL material file may be referenced from within the OBJ file.

Relative and absolute indices

OBJ files, due to their list structure, are able to reference vertices, normals, etc either by their absolute (1-indexed) list position, or relatively by using negative indices and counting backwards. However, not all software supports the latter approach, and conversely some software inherently writes only the latter form (due to the convenience of appending elements without needing to recalculate vertex offsets, etc), leading to occasional incompatibilities.

Software that supports OBJ

- 3DPaintBrush
- 3D Studio Max
- Art of Illusion
- Autodesk Mudbox
- Autodesk Softimage
- Ayam
- Blender
- CADdoctor
- Carrara
- CATIA
- Cheetah3D
- Cinema 4D
- CityEngine
- Curvy3D
- David Laserscanner
- DAZ Studio
- Display MINC tool (http://www.bic.mni.mcgill.ca/ServicesSoftware/HomePage) McConnell Brain Imaging Center, Montreal Neurological Institute, McGill University
- EnSight (CEI)
- Feature Manipulation Engine
- Fledermaus (IVS)
- FreeCAD
- Game Maker Through various user made DLL's and scripts
- GLC Player
- Google Sketchup With third-party plugins
- Hexagon
- Houdini
- Inkscape
- Irrlicht Engine
- Lightwave
- Mathematica
- Max/MSP
- Maya
- modo
- MeshLab
- Metasequoia
- Microstation
- MilkShape 3D
- Misfit Model 3d
- Open Cobalt
- OpenCTM
- Photoshop (CS4 and later)
- Poser
- Rhinoceros 3D
- SHOT
- Silo
- SketchUp
- SpeedTree
- Sweet Home 3D
- Szs Modifier
- TransMagic
- TrueSpace
- Unity (game engine)
- VisIt
- Vue
- VXL through the contrib/brl/bbas/imesh sub-library
- Wings 3D
- X-Plane Plane Maker
- ZBrush
- Zmodeler2
- Paint3D

Material template library

Synopsis

MTL material format

Filename extension	`.mtl`
Developed by	Wavefront Technologies
Type of format	3D texture format

In 3D Computer Graphics, one of the most common geometry interchange file formats is the .OBJ File Format. The .MTL File Format is a companion file format that describes surface shading (material) properties of objects within one or more .OBJ files. A .OBJ file references one or more .MTL files (called "material libraries"), and from there, references one or more material descriptions by name. (This author believes that expanding "MTL" to "Material Template Library" is a post-construction, as "mtl" is common shorthand for "material," and there is nothing template-ey about the use of materials in mtl files.)

Introduction

The **Material Template Library** format (MTL) is a standard defined by Wavefront Technologies for ASCII files that define the light reflecting properties of a surface for the purposes of computer rendering, and according to the Phong shading model. The standard has widespread support among different computer software packages, making it a useful format for interchange of materials.

MTL files are commonly accompanied by and referenced from **OBJ** files that define geometry upon which the materials of the MTL file are mapped.

The MTL format, although still widely used, is outdated and does not fully support later technologies such as specular maps and parallax maps. However due to the open and intuitive nature of the format, these can easily be added with a custom MTL file generator.

The MTL format defines a number of formats[1][2].

Basic materials

A single `.mtl` file define contain multiple materials. Materials are defined one after another in the file, each starting with the `newmtl` command:

```
# define a material named 'Colored'
newmtl Colored
```

The ambient color of the material is declared using `Ka`. Color definitions are in RGB where each channel's value is between 0 and 1.

```
Ka 1.000 1.000 1.000     # white
```

Similarly the diffuse color using `Kd`.

```
Kd 1.000 1.000 1.000     # white
```

And the specular color by `Ks`, weighted by the specular coefficient `Ns`.

```
Ks 0.000 0.000 0.000     # black (off)
Ns 10.000                # ranges between 0 and 1000
```

Materials can be transparent. This is referred to as being *dissolved*. Unlike real transparency, the result does not depend upon the thickness of the object.

```
d 0.9                    # some implementations use 'd'
Tr 0.9                   # others use 'Tr'
```

Multiple illumination models are available, per material. These are enumerated as follows:

```
0. Color on and Ambient off
1. Color on and Ambient on
2. Highlight on
3. Reflection on and Ray trace on
4. Transparency: Glass on, Reflection: Ray trace on
5. Reflection: Fresnel on and Ray trace on
6. Transparency: Refraction on, Reflection: Fresnel off and Ray trace on
7. Transparency: Refraction on, Reflection: Fresnel on and Ray trace on
8. Reflection on and Ray trace off
9. Transparency: Glass on, Reflection: Ray trace off
10. Casts shadows onto invisible surfaces
```

```
illum 2
```

Texture maps

Textured materials use the same properties as above, and additionally define texture maps.

```
newmtl Textured
Ka 1.000 1.000 1.000
Kd 1.000 1.000 1.000
Ks 0.000 0.000 0.000
d 1.0
illum 2
map_Ka lenna.tga           # the ambient texture map
map_Kd lenna.tga           # the diffuse texture map (most of the time, it will
                           # be the same as the ambient texture map)
map_Ks lenna.tga           # the specular texture map
map_d lenna_alpha.tga      # the alpha texture map
map_bump lenna_bump.tga    # the bump map
bump lenna_bump.tga        # some implementations use 'bump' instead of 'map_Bump'
```

See also

- 3DMLW is a markup language that shows OBJ files through common web browsers (Internet Explorer, Mozilla Firefox, Opera)

References

1. ^ "MTL Files - Material Definitions for OBJ Files" (http://people.sc.fsu.edu/~burkardt/data/mtl/mtl.html) . People.sc.fsu.edu. 2004-06-14. http://people.sc.fsu.edu/~burkardt/data/mtl/mtl.html. Retrieved 2010-11-26.
2. ^ Author. "Wavefront .mtl file format info - GRIPES and GRUMBLES - Wings - Wings3D - Official Development Forum - Message Board" (http://nendowingsmirai.yuku.com/forum/viewtopic/id/1723) . Nendowingsmirai.yuku.com. http://nendowingsmirai.yuku.com/forum/viewtopic/id/1723. Retrieved 2010-11-26.

External links

- Obj Specification (http://local.wasp.uwa.edu.au/~pbourke/dataformats/obj/)
- Mtl Specification (http://local.wasp.uwa.edu.au/~pbourke/dataformats/mtl/)
- Tools, libraries and example files (http://people.scs.fsu.edu/~burkardt/data/obj/obj.html)