.obj 文件格式与.mtl 文件格式

最近在学习 obj 文件格式，上网查了些资料，很难找到比较全面的文章，尤其是对.mtl 文件的说明甚少。今天把最近搜索的资料整合了一下。

这里的 obj 文件格式指的是 Wavefront 公司为它的一套基于工作站的 3D 建模和动画软件 "Advanced Visualizer"开发的一种文件格式。
OBJ 文件是一种标准的 3D 模型文件格式，很适合用于 3D 软件模型之间的互导。OBJ 文件是一种文本文件格式，这就意味着你可以直接用写字板打开进行查看修改。 目前几乎所有知名的 3D 软件都支持 OBJ 文件的读写，不过很多软件需要通过插件才能 做到这一点。另外，作为一种优秀的文件格式，很多游戏引擎也都支持 OBJ 文件。
OBJ3.0 格式支持多边形(Polygon)，直线 (Lines)，表面(Surfaces)，和自由形态曲线(Free-form Curves)。直线和多角形通过它们的点来描述，曲线和表面则根据于它们的控制点和依附于曲线类型的额外信息来定义。这些信息支持规则和不规则的曲线， 包括那些基于贝塞尔 (Bezier)曲线，B 样条(B-spline)，基数(Cardinal/Catmull-Rom 样条)，和泰勒方程(Taylor equations)的曲线。

OBJ 文件特点

-1- OBJ 是一种 3D 模型文件，因此不包含动画、材质特性、贴图路径、动力学、粒子等信息。

-2- OBJ 文件主要支持多边形(Polygons)模型。
虽然 OBJ 文件也支持曲线(Curves)、表面(Surfaces)、点组材质(Point Group Materials)，但 Maya 导出的 OBJ 文件并不包括这些信息。

-3- OBJ 文件支持三个点以上的面，这一点很有用。
很多其它的模型文件格式只支持三个点的面，所以我们导入 Maya 的模型经常被三角化了，这对于我们对模型的再加工甚为不利。

-4- OBJ 文件支持法线和贴图坐标。

OBJ 文件基本结构

OBJ 文件不需要任何种文件头(File Header)，尽管经常使用几行文件信息的注释作为文件的开头。OBJ 文件由一行行文本组成，注释行以一个“井”号(#)为开头，空格和空行可以随意加到文件中以增加文件的可读性。有字的行都由一两个标记字母也就是关键字 (Keyword)开头，关键字可以说明这一行是什么样的数据。多行可以逻辑地连接在一起表示一行，方法是在每一行最后添加一个连接符(\)。注意连接符 (\)后面不能出现空格或 tab 格，否则将导致文件出错。

下列关键字可以在 OBJ 文件使用【关键字根据数据类型排列，每个关键字有一段简短描述】

顶点数据(Vertex data)：

v 几何体顶点 (Geometric vertices)

vt 贴图坐标点 (Texture vertices)

vn 顶点法线 (Vertex normals)
vp 参数空格顶点 (Parameter space vertices)

自由形态曲线(Free-form curve)/表面属性(surface attributes):
deg 度 (Degree)
bmat 基础矩阵 (Basis matrix)
step 步尺寸 (Step size)
cstype 曲线或表面类型 (Curve or surface type)

元素(Elements):
p 点 (Point)
l 线 (Line)
f 面 (Face)
curv 曲线 (Curve)
curv2 2D 曲线 (2D curve)
surf 表面 (Surface)

自由形态曲线(Free-form curve)/表面主体陈述(surface body statements):
parm 参数值 (Parameter values)
trim 外部修剪循环 (Outer trimming loop)
hole 内部整修循环 (Inner trimming loop)
scrv 特殊曲线 (Special curve)
sp 特殊的点 (Special point)
end 结束陈述 (End statement)

自由形态表面之间的连接(Connectivity between free-form surfaces):
con 连接 (Connect)

成组(Grouping):
g 组名称 (Group name)
s 光滑组 (Smoothing group)
mg 合并组 (Merging group)
o 对象名称 (Object name)

显示(Display)/渲染属性(render attributes):
bevel 导角插值 (Bevel interpolation)
c_interp 颜色插值 (Color interpolation)
d_interp 溶解插值 (Dissolve interpolation)
lod 细节层次 (Level of detail)
usemtl 材质名称 (Material name)

mtllib 材质库（Material library）
shadow_obj 投射阴影（Shadow casting）
trace_obj 光线跟踪（Ray tracing）
ctech 曲线近似技术（Curve approximation technique）
stech 表面近似技术（Surface approximation technique）

Maya 导出的 OBJ 文件:

在 Maya 中创建一个多边形立方体,选中这个立方体 ,导出格式为 OBJ ,文件名为"cube.obj".（如果没有此格式 ,请在 Plug-in Manager 中载入"objExport.mll"）用写字板打开"cube.obj"。可以看到如下代码：

```
# The units used in this file are centimeters.
g default
v -0.500000 -0.500000 0.500000
v 0.500000 -0.500000 0.500000
v -0.500000 0.500000 0.500000
v 0.500000 0.500000 0.500000
v -0.500000 0.500000 -0.500000
v 0.500000 0.500000 -0.500000
v -0.500000 -0.500000 -0.500000
v 0.500000 -0.500000 -0.500000
vt 0.000000 0.000000
vt 1.000000 0.000000
vt 0.000000 1.000000
vt 1.000000 1.000000
vt 0.000000 2.000000
vt 1.000000 2.000000
vt 0.000000 3.000000
vt 1.000000 3.000000
vt 0.000000 4.000000
vt 1.000000 4.000000
vt 2.000000 0.000000
vt 2.000000 1.000000
vt -1.000000 0.000000
vt -1.000000 1.000000
vn 0.000000 0.000000 1.000000
vn 0.000000 0.000000 1.000000
vn 0.000000 0.000000 1.000000
vn 0.000000 0.000000 1.000000
```

vn 0.000000 1.000000 0.000000
vn 0.000000 1.000000 0.000000
vn 0.000000 1.000000 0.000000
vn 0.000000 1.000000 0.000000
vn 0.000000 0.000000 -1.000000
vn 0.000000 0.000000 -1.000000
vn 0.000000 0.000000 -1.000000
vn 0.000000 0.000000 -1.000000
vn 0.000000 -1.000000 0.000000
vn 0.000000 -1.000000 0.000000
vn 0.000000 -1.000000 0.000000
vn 0.000000 -1.000000 0.000000
vn 1.000000 0.000000 0.000000
vn 1.000000 0.000000 0.000000
vn 1.000000 0.000000 0.000000
vn 1.000000 0.000000 0.000000
vn -1.000000 0.000000 0.000000
vn -1.000000 0.000000 0.000000
vn -1.000000 0.000000 0.000000
vn -1.000000 0.000000 0.000000
s off
g pCube1
usemtl initialShadingGroup
f 1/1/1 2/2/2 4/4/3 3/3/4
f 3/3/5 4/4/6 6/6/7 5/5/8
f 5/5/9 6/6/10 8/8/11 7/7/12
f 7/7/13 8/8/14 2/10/15 1/9/16
f 2/2/17 8/11/18 6/12/19 4/4/20
f 7/13/21 1/1/22 3/3/23 5/14/24

这个文件看起来稍复杂一些，用到了许多关键词，你可以对照前面的列表查看一下每个关键词的意思。

解释一下：

"vt 1.000000 0.000000"这句"vt"代表点的贴图坐标。

"vn 0.000000 0.000000 -1.000000"这句"vn"代表点的法线。

"s off"表示关闭光滑组。

"usemtl initialShadingGroup"表示使用的材质。

"f 7/13/21"这时在面的数据中多了贴图坐标 uv 点和法线的索引号，索引号分别用左斜线(/)隔开。

格式："f 顶点索引/uv 点索引/法线索引"。

"g pCube1"表示组，这里的成组与 Maya 中的成组不一样，这里的成组是指把"g pCube1"

后出现的面都结合到一起，组成一个整的多边形几何体。

把"cube.obj"文件修改一下就知道成组的意思了。把"s off"这句后面的代码替换成以下代码：

usemtl initialShadingGroup
g pCube_Face1
f 1/1/1 2/2/2 4/4/3 3/3/4
g pCube_Face2
f 3/3/5 4/4/6 6/6/7 5/5/8
g pCube_Face3
f 5/5/9 6/6/10 8/8/11 7/7/12
g pCube_Face4
f 7/7/13 8/8/14 2/10/15 1/9/16
g pCube_Face5
f 2/2/17 8/11/18 6/12/19 4/4/20
g pCube_Face6
f 7/13/21 1/1/22 3/3/23 5/14/24

导入 Maya 后可以看到，立方体的每个面是分离的，每个面的名称分别是"pCube_Face(1~6)"，可见组的名称其实就是单独几何体的名称。

****

用中文**命名几何体(组)**并不总是顺利，把"g 立方体面 1"这行改为"g 选择"再试试看，这回导入时模型根本无法出现，只会出现如下的错误信息：

// Error: line 1: Your OBJ file contains a line which is too long to be parsed. Please edit your obj file. //

// Error: line 1: Error reading file. //

由此可见，物体命名的不规范也是导致 OBJ 文件出错的原因之一。关于 Maya 的物体命名，英文名是很保险的，标点符号中只有下划线(_)可用，数字不能用放到名称的开头，尽量不要用中、日、韩等双字节文字。

OBJ 文件不支持有孔的多边形面。举个例子：

选择 Maya 的创建多边形工具(Polygons -> Create Polyon Tool)，在视图中画一个四边形，不要按回车，按 Ctrl 在四边形中间点一下，可以继续在四边形中挖一个洞。把这个有孔的多边形存成 OBJ 格式，在导入 Maya 时，会发现多边形少了一块。如果你把这也看成错误，现在至少你已经知道错误的原因了，就是 OBJ 文件不支持有孔的多边形面。

****

OBJ 文件不包含面的颜色定义信息，不过可以引用**材质库**，材质库信息储存在一个后缀是".mtl"的独立文件中。关键字"mtllib"即材质库的意思。

材质库中包含材质的漫射(diffuse)，环境(ambient)，光泽(specular)的 RGB(红绿蓝)的定义值，以及反射(specularity)，折射(refraction)，透明度(transparency)等其它特征。

"usemtl"指定了材质之后，以后的面都是使用这一材质，直到遇到下一个"usemtl"来指定新的材质。

下面的例子说明了指定材质的方法。

Cube with Materials：

```
# This cube has a different material
# applied to each of its faces.
mtllib master.mtl
v 0.000000 2.000000 2.000000
v 0.000000 0.000000 2.000000
v 2.000000 0.000000 2.000000
v 2.000000 2.000000 2.000000
v 0.000000 2.000000 0.000000
v 0.000000 0.000000 0.000000
v 2.000000 0.000000 0.000000
v 2.000000 2.000000 0.000000
# 8 vertices
g front
usemtl red
f 1 2 3 4
g back
usemtl blue
f 8 7 6 5
g right
usemtl green
f 4 3 7 8
g top
usemtl gold
f 5 1 4 8
g left
usemtl orange
f 5 6 2 1
g bottom
usemtl purple
f 2 6 7 3
# 6 elements
```

****

贝塞尔片面(Bezier Patch)：

Maya 不能导出 OBJ 格式的贝塞尔片面，却能够导入它。导入的贝塞尔片面自动转换为 Nurbs 表面。

```
# 3.0 Bezier patch
v -5.000000 -5.000000 0.000000
v -5.000000 -1.666667 0.000000
v -5.000000 1.666667 0.000000
v -5.000000 5.000000 0.000000
v -1.666667 -5.000000 0.000000
v -1.666667 -1.666667 0.000000
v -1.666667 1.666667 0.000000
v -1.666667 5.000000 0.000000
v 1.666667 -5.000000 0.000000
v 1.666667 -1.666667 0.000000
v 1.666667 1.666667 0.000000
v 1.666667 5.000000 0.000000
v 5.000000 -5.000000 0.000000
v 5.000000 -1.666667 0.000000
v 5.000000 1.666667 0.000000
v 5.000000 5.000000 0.000000
# 16 vertices
cstype bezier
deg 3 3
# Example of line continuation
surf 0.000000 1.000000 0.000000 1.000000 13 14 \
15 16 9 10 11 12 5 6 7 8 1 2 3 4
parm u 0.000000 1.000000
parm v 0.000000 1.000000
end
# 1 element
```

****

基数曲线(Cardinal Curve)：

Maya 好像不支持 OBJ 格式的曲线，导入时不会出现错误信息，却也不会出现曲线。

```
# 3.0 Cardinal curve
v 0.940000 1.340000 0.000000
v -0.670000 0.820000 0.000000
v -0.770000 -0.940000 0.000000
v 1.030000 -1.350000 0.000000
v 3.070000 -1.310000 0.000000
# 6 vertices
cstype cardinal
deg 3
```

curv 0.000000 3.000000 1 2 3 4 5 6
parm u 0.000000 1.000000 2.000000 3.000000 end
1 element

****

贴图映射(Texture-Mapped)：

A 2 x 2 square mapped with a 1 x 1 square
texture stretched to fit the square exactly.
mtllib master.mtl
v 0.000000 2.000000 0.000000
v 0.000000 0.000000 0.000000
v 2.000000 0.000000 0.000000
v 2.000000 2.000000 0.000000
vt 0.000000 1.000000 0.000000
vt 0.000000 0.000000 0.000000
vt 1.000000 0.000000 0.000000
vt 1.000000 1.000000 0.000000
4 vertices
usemtl wood
The first number is the point,
then the slash,
and the second is the texture point
f 1/1 2/2 3/3 4/4
1 element

****

说下**顶点索引**吧。

我 们知道，对于每一个三角形，都需要用 3 个顶点来表示。例如在上面的立方体模型中，一共有 6×2×3=36 个顶点。仔细想想就会知道， 在这 36 个顶点中，又相当数量的顶点是重合的。如果把这些重合的顶点都一一表示出来，就太浪费存储空间了。于是，我们提出了顶点索引的想法，解决空间占用 问题。顶点索引的思想是建立两个数组，一个数组用于存储模型中所有的顶点坐标值，另一个数组则存储每一个表面所对应的三个顶点在第一个数组中的索引。建立 这样的顶点索引显然更加节约存储空间。

假设 Indices:array of Integer 是顶点索引数组，Vertices:array of TVertex 是顶点数组，使用下面的代码段就可以把整个顶点索引对应的所有三角形绘制出来：

```
procedure DrawIndex(Indices:array of Integer;Vertices:array of TVertex);
var i :Integer;
begin
  glBegin(GL_TRIANGLES);
  for i := 0 to (High(Vertices)+1) div 3 -1 do
```

```
    begin
      glVertex3fv(@Vertices[Indices[i*3]]);
      glVertex3fv(@Vertices[Indices[i*3+1]]);
      glVertex3fv(@Vertices[Indices[i*3+2]]);
    end;
    glEnd;
end;
```

以此类推，我们可以为模型中所有的法线、纹理坐标都建立起相应的索引，以节省更多的空间。而事实上，OBJ 文件就是这么做的。

在 一个 OBJ 文件中，首先有一些以 v、vt 或 vn 前缀开头的行指定了所有的顶点、纹理坐标、法线的坐标。然后再由 一些以 f 开头的行指定每一个三角形所对应的顶点、纹理坐标和法线的索引。在顶点、纹理坐标和法线的索引之间，使用符号“/”隔开的。一个 f 行可以以下面几 种格式出现：

f 1 2 3

这样的行表示以第 1、2、3 号顶点组成一个三角形。

f 1/3 2/5 3/4

这样的行表示以第 1、2、3 号顶点组成一个三角形，其中第一个顶点的纹理坐标的索引值为 3，第二个顶点的纹理坐标的索引值为 5，第三个顶点的纹理坐标的索引值为 4。

f 1/3/4 2/5/6 3/4/2

这样的行表示以第 1、2、3 号顶点组成一个三角形，其中第一个顶点的纹理坐标的索引值为 3，其法线的索引值是 4；第二个顶点的纹理坐标的索引值为 5，其法线的索引值是 6；第三个顶点的纹理坐标的索引值为 6，其法线的索引值是 2。

f 1//4 2//6 3//2

这样的行表示以第 1、2、3 号顶点组成一个三角形，且忽略纹理坐标。其中第一个顶点的法线的索引值是 4；第二个顶点的法线的索引值是 6；第三个顶点的法线的索引值是 2。

值得注意的是文件中的索引值是以 1 作为起点的，这一点与 Delphi 中以 0 作为起点有很大的不同。在渲染的时候应注意将从文件中读取的坐标值减去 1。

****

最后说说**.mtl 文件**吧

三维模型处理会要读取.mtl 文件来获得材质信息。

.mtl 文件（Material Library File）是材质库文件，描述的是物体的材质信息，ASCII 存储，任何文本编辑器可以将其打开和编辑。一个.mtl 文件可以包含一个或多个材质定义，对 于每个材质都有其颜色，纹理和反射贴图的描述，应用于物体的表面和顶点。

以下是一个材质库文件的基本结构：

newmtl mymtl_1

材质颜色光照定义
纹理贴图定义
反射贴图定义
newmtl mymtl_2
材质颜色光照定义
纹理贴图定义
反射贴图定义
newmtl mymtl_3
材质颜色光照定义
纹理贴图定义
反射贴图定义
……
注释：每个材质库可含多个材质定义，每个材质都有一个材质名。用 newmtl mtlName 来定义一个材质。对于每个材质，可定义它的颜色光照纹理反射等描述特征。
主要的定义格式如下文所示：
材质颜色光照
1。环境反射有以下三种描述格式，三者是互斥的，不能同时使用。

Ka r g b　\\用 RGB 颜色值来表示，g 和 b 两参数是可选的，如果只指定了 r 的值，则 g 和 b 的值都等于 r 的值。三个参数一般取值范围为 0.0~1.0，在此范围外的值则相应的增加或减少反射率;
Ka spectral file.rfl factor　\\用一个 rfl 文件来表示。factor 是一个可选参数，表示.rfl 文件中值的乘数，默认为 1.0;
Ka xyz x y z　\\用 CIEXYZ 值来表示，x，y，z 是 CIEXYZ 颜色空间的各分量值。y 和 z 两参数是可选的，如果只指定了 x 的值，则 y 和 z 的值都等于 r 的值。三个参数一般取值范围为 0~1。

2。漫反射描述的三种格式：
Kd r g b
Kd spectral file.rfl factor
Kd xyz x y z

3。镜反射描述的三种格式:
Ks r g b
Ks spectral file.rfl factor
Ks xyz x y z

4。滤光透射率描述的三种格式：
Tf r g b
Tf spectral file.rfl factor
Tf xyz x y z

5。光照模型描述格式：

illum illum_#
指定材质的光照模型。illum 后面可接 0~10 范围内的数字参数。各个参数代表的光照模型如下所示：
****

光照模型	属性
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

****

6。渐隐指数描述

d factor
参数 factor 表示物体融入背景的数量，取值范围为 0.0~1.0，取值为 1.0 表示完全不透明，取值为 0.0 时表示完全透明。当新创建一个物体时，该值默认为 1.0，即无渐隐效果。
与真正的透明物体材质不一样，这个渐隐效果是不依赖于物体的厚度或是否具有光谱特性。该渐隐效果对所有光照模型都有效。

d -halo factor
指定一种受观察者影响的渐隐效果。例如，对于一个定义为 d -halo 0.0 的球体，在它的中心是完全消隐的，而在表面边界处将逐渐变得不透明。
其中 factor 表示应用在材质上的渐隐率的最小值。而材质上具体的渐隐率将在这个最小值到 1.0 之间取值。其计算公式为：

dissolve = 1.0 - (N*v)(1.0-factor)

7。反射指数描述

Ns exponent
指定材质的反射指数，定义了反射高光度。
exponent 是反射指数值，该值越高则高光越密集，一般取值范围在 0~1000。

8。清晰度描述

Sharpness value
指定本地反射贴图的清晰度。如果材质中没有本地反射贴图定义，则将此值应用到预览中的全局反射贴图上。
value 可在 0~1000 中取值，默认 60。值越高则越清晰。

9。折射值描述

Ni ptical density
指定材质表面的光密度，即折射值。
ptical density 是光密度值，可在 0.001 到 10 之间进行取值。若取值为 1.0，光在通过物体的时候不发生弯曲。玻璃的折射率为 1.5。取值小于 1.0 的时候可能会产生奇怪的结果，不推荐。

****

纹理映射
纹理映射可以对映射的相应材质参数进行修改，这个修改只是对原有存在的参数进行叠加修改，而不是替换原有参数，从而纹理映射在物体表面的表现上有很好的灵活性。
纹理映射只可以改变以下材质参数：
- Ka (color)
- Kd (color)
- Ks (color)
- Ns (scalar)
- d (scalar)
除了以上参数，表面法线也可以更改。
纹理文件类型可以是以下几种：
1.纹理映射文件
.mpc：颜色纹理文件 color texture files ——可改变 Ka，Kd，Ks 的值
.mps：标量纹理文件 scalar texture files——可改变 Ns，d，decal 的值
.mpb：凹凸纹理文件 bump texture files——可改变面法线
2.程序纹理文件：

程序纹理文件是用数学公式来计算纹理的样本值。有以下几种格式：
.cxc
.cxs
.cxb

以下是 mtl 文件中对于纹理映射的描述格式：
1.map_Ka -options args filename
为环境反射指定颜色纹理文件(.mpc)或程序纹理文件(.cxc) ,或是一个位图文件。在渲染的时候，Ka 的值将再乘上 map_Ka 的值。
而 map_Ka 的可选项参数有以下几个：
-blendu on | off
-blendv on | off
-cc on | off
-clamp on | off
-mm base gain
-o u v w
-s u v w
-t u v w
-texres value

2.map_Kd -options args filename
为漫反射指定颜色纹理文件(.mpc)或程序纹理文件(.cxc)，或是一个位图文件。作用原理与可选参数与 map_Ka 同。

3.map_Ks -options args filename
为镜反射指定颜色纹理文件(.mpc)或程序纹理文件(.cxc)，或是一个位图文件。作用原理与可选参数与 map_Ka 同。

4.map_Ns -options args filename
为镜面反射指定标量纹理文件（.mps 或.cxs）。可选参数如下所示：
-blendu on | off
-blendv on | off
-clamp on | off
-imfchan r | g | b | m | l | z
-mm base gain
-o u v w
-s u v w
-t u v w
-texres value

5.map_d -options args filename
为消隐指数指定标量纹理文件（.mps 或.cxs）。作用原理和可选参数与 map_Ns 同。

6.map_aat on
打开纹理反走样功能。

7.decal -options args filename
指定一个标量纹理文件或程序纹理文件用于选择性地将材质的颜色替换为纹理的颜色。可选参数同 map_Ns。
在渲染期间， Ka, Kd, and Ks 和 map_Ka, map_Kd, map_Ks 的值通过下面这个公式来进行使用：
result_color=tex_color(tv)*decal(tv)+mtl_color*(1.0-decal(tv))
其中 tv 表示纹理顶点，result_color 是 Ka,Kd 和 Ks 的综合作用值。

8.disp -options args filename
指定一个标量纹理文件或程序纹理文件实现物体变形或产生表面粗糙。可选参数同 map_Ns。

9.bump -options args filename
为材质指定凹凸纹理文件（.mpb 或.cxb）,或是一个位图文件。
可选参数可为：
-bm mult
-clamp on | off
-blendu on | off
-blendv on | off
-imfchan r | g | b | m | l | z
-mm base gain
-o u v w
-s u v w
-t u v w
-texres value

****

反射贴图
在.mtl 文件中的定义格式为：
1.refl -type sphere -options -args filename
指定一个球体区域将指定的纹理反射映射至物体。filename 为一个颜色纹理文件，或可以映射的位图。

2.refl -type cube_side -options -args filenames

指定一个立方体区域将指定的纹理反射映射至物体。可以通过以下方式来指定纹理位置：

refl -type cube_top
refl -type cube_bottom
refl -type cube_front
refl -type cube_back
refl -type cube_left
refl -type cube_right

“refl”可以单独使用 ,或配合以下参数使用。使用时将参数置于“refl”和“filename”之间。

-blendu on | off
-blendv on | off
-cc on | off
-clamp on | off
-mm base gain
-o u v w
-s u v w
-t u v w
-texres value