**Object文件解析：**

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 betweenfree-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)

**OBJLoader解析：**

1.整个函数是一个js立即执行函数，形如(function(){})()的就是立即执行函数，也叫立即调用函数。

作用就是：

Js中没有私有作用域的概念，如果在多人开发的项目上，你在全局或局部作用域中声明了一些变量，可能会被其他人不小心用同名的变量给覆盖掉，根据javascript函数作用域链的特性，可以使用这种技术可以模仿一个私有作用域，用匿名函数作为一个“容器”，“容器”内部可以访问外部的变量，而外部环境不能访问“容器”内部的变量，所以( function(){…} )()内部定义的变量不会和外部的变量发生冲突，俗称“匿名包裹器”或“命名空间”。

2.结构

函数：

Function //最外部的立即调用函数

function ParseState（）//object文件中各种属性（点，面，材质等）的提取与分析

startObject

startMaterial //新的usemtl声明overwrite一个继承的material

clone //复制

currentMaterial //现在的material

\_finalize //Ignore objects tail materials if no face declarations followed them before a new o/g started.

finalize //完成

parseVertexIndex //点索引

parseNormalIndex //面法向量索引

parseUVIndex //纹理贴图坐标索引

addVertex //读取点

addVertexLine //读取点line

addNormal //读取面法向量

addColor //读取颜色

addUV //读取UV（纹理坐标）

addUVLine //读取UVline

addFace //读取多个属性形成face

addLineGeometry //vertex和uv

function OBJLoader（）//构造器

load //通过object的url载入object,并且在载入过程后执行一些其他自定义函数

setPath //设定路径

setMaterials //设定materials

parse //对读入的整个文件的字符串进行处理

**MTL文件解析：**

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。

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。

**MTLLoader解析：**

感觉这个MTLLoader和OBJLoader很像，里面的很多函数都很类似。

MTLLoader.prototype //

Load //通过url载入mtl文件

SetPath //赋值path

SetTexturePath //赋值texturepath

setBaseUrl //赋值baseurl

SetCrossOrigin //赋值crossorigin

Parse //对整个mtl字符串进行处理

MaterialCreator

setManager //设定manager

SetMaterials //设定信息

Convert //把material信息标准化处理

Preload

getIndex //取得索引

Create //若material为undefined，调用creatematerial

CreateMaterial\_ //创建material

resolveURL //对传入的url格式进行规范化

setMapForType //保存第一个遇到的texture

getTextureParams //得到texture的参数

loadtexture //通过url载入texture