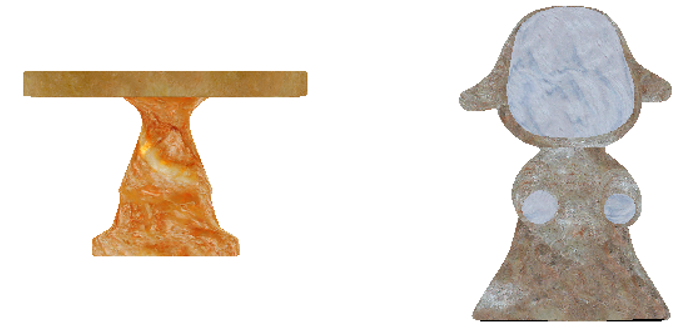
实验四 带纹理的OBJ文件读取和显示

1. 实验内容
2. 在程序中读取带纹理的obj文件，载入相应的纹理图片文件，将带纹理的模型显示在程序窗口中。参考实现效果如下，物体之间的位置自行设置，没有要求。



1. 具体内容
2. 读取带纹理的obj文件

实现TriMesh类中的readObj函数，使得vertex\_textures中存储UV坐标数据，vertex\_positions存储顶点坐标数据，vertex\_normals存储顶点法线数据，vertex\_colors存储顶点颜色数据；faces存储三角面片的顶点索引数据，normal\_index存储三角面片的顶点的法向量数据的索引下标，texture\_index存储三角面片的顶点的纹理坐标数据的索引下标。因obj文件无法存储颜色数据，可以使用法线作为颜色数值；参考实验4.1中程序生成物体的内容，确定不同数据的存储位置和方法。

1. 完善数据的读取

参考实验4.1，结合上一个步骤中对obj数据的读取和存储方式，将本实验留空代码的TriMesh.cpp中的storeFacesPoints函数补全。另外从模型的读取到数据存储完成可能需要一些时间，可以在编译的时候选择Release模式，这样生成的程序执行速度会快一些。如果有优化的想法也可以尝试优化一下代码，提高程序速度。

1. 模型和纹理显示

修改main.cpp中的init函数，使得程序可以正常运行，显示贴有纹理的玩偶模型和桌子模型。

1. Obj格式说明

实验中还提供了两个cube.obj文件实例，下面以cube.obj为例进行文件格式的说明：

当我们使用建模软件创建模型，并导出为obj格式后，obj文件一般会配套生成一个材质文件（.mtl后缀），如果有纹理的话还有纹理图片。

先介绍一下.mtl材质文件，这个文件里面会记录该模型材质相关的参数，“#”开头的为注释内容，“newmtl”开头的关键字后面会跟着一个名称，作为材质的名字，比如这里就有一个材质名字叫“Material”，后面跟着的内容都是这个材质的信息。

“Ns”开头的是材质的高光系数，“Ka”是环境光系数，“Ks”是镜面光系数，“Kd”是漫反射系数，“map\_Kd”后面跟着的是纹理图片的路径。除此之外可能还会有其他关键字，都是描述材质的参数，但是我们学习的光照模型用不到那些参数，所以可以不理。

# Blender MTL File: 'None'

# Material Count: 1

newmtl Material

Ns 323.999994

Ka 1.000000 1.000000 1.000000

Kd 0.800000 0.800000 0.800000

Ks 0.500000 0.500000 0.500000

Ke 0.000000 0.000000 0.000000

map\_Kd wall.jpg

mtl文件示例

obj文件的每一行都会以一个关键词或者字符开头，“#”开头的为注释内容，“mtllib”开头的关键字后面会跟着要使用的材质文件名字，“usemtl”开头的关键字后面会跟着材质文件中要使用的材质名字，然后下面就跟着顶点的各种数据，每类顶点数据的开头字符都不同，下面举例解释一下：

“v -0.500000 -0.500000 0.500000”中，“v”代表点的几何坐标。

“vt 1.000000 0.000000”的“vt”代表点的贴图坐标。

“vn 0.000000 0.000000 1.000000”中，“vn”代表点的法线。

“f 2/1/1 3/2/1 1/3/1”中，“f”开头表示面的数据，记录的是顶点索引，每个面由多个顶点组成，这里每个顶点的3个数值分别表示顶点索引下标、纹理的UV索引下标、法向量索引下标。索引号分别用左斜线(/)隔开，每个顶点的数据用空格分开，即格式：

f 顶点索引/uv点索引/法线索引

obj格式记录的模型中，面片f**一般是三角面片，也可能是四边形面片**，cube.obj和cube2.obj两个模型中一个是由三角面片构成，一个是四边形面片构成，具体差别可以打开这两个文件进行对比。除此之外还有其他关键字，不过我们这里不需要考虑。

本次实验提供的是三角面片组成的网格模型，而且本次实验不需要mtl文件。

参考资料：https://blog.csdn.net/linziping/article/details/100739991

# Blender v2.81 (sub 16) OBJ File: ''

# www.blender.org

mtllib cube.mtl

usemtl Material

v -1.000000 -1.000000 1.000000

v -1.000000 1.000000 1.000000

v -1.000000 -1.000000 -1.000000

v -1.000000 1.000000 -1.000000

v 1.000000 -1.000000 1.000000

v 1.000000 1.000000 1.000000

v 1.000000 -1.000000 -1.000000

v 1.000000 1.000000 -1.000000

vt 0.625000 0.000000

vt 0.375000 0.250000

vt 0.375000 0.000000

vt 0.625000 0.250000

vt 0.375000 0.500000

vt 0.625000 0.500000

vt 0.375000 0.750000

vt 0.625000 0.750000

vt 0.375000 1.000000

vt 0.125000 0.750000

vt 0.125000 0.500000

vt 0.875000 0.500000

vt 0.625000 1.000000

vt 0.875000 0.750000

vn -1.0000 0.0000 0.0000

vn 0.0000 0.0000 -1.0000

vn 1.0000 0.0000 0.0000

vn 0.0000 0.0000 1.0000

vn 0.0000 -1.0000 0.0000

vn 0.0000 1.0000 0.0000

f 2/1/1 3/2/1 1/3/1

f 4/4/2 7/5/2 3/2/2

f 8/6/3 5/7/3 7/5/3

f 6/8/4 1/9/4 5/7/4

f 7/5/5 1/10/5 3/11/5

f 4/12/6 6/8/6 8/6/6

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

f 4/4/2 8/6/2 7/5/2

f 8/6/3 6/8/3 5/7/3

f 6/8/4 2/13/4 1/9/4

f 7/5/5 5/7/5 1/10/5

f 4/12/6 2/14/6 6/8/6

cube文件示例

1. 提交内容 **【请各位务必严格按照以下要求执行（图片仅供参考）】**
2. **程序代码**：本次实验提供参考代码，**程序运行窗口标题设为：“学号\_姓名\_实验四”**（如果出现乱码，可能是因为代码文件的字符格式问题，实验提供的文件字符编码为utf-8，改成GBK格式就不会出现乱码了）。最终提交代码中与实验内容相关部分必须写注释。



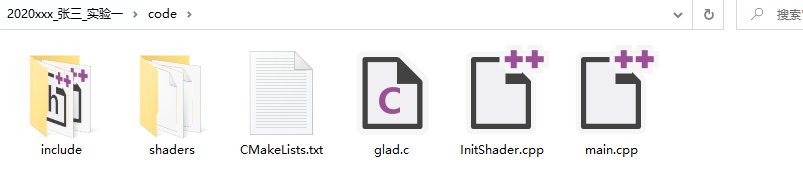
1. **实验报告**：内容完整，实验目的、实验步骤、实验结果、实验心得都要完成。 排版要整齐，字体要规范。每一实验内容有相应的文字描述和关键步骤的截图。
2. **上传格式**：按上述要求完成实验，一并提交电子版实验报告和源代码压缩包，文档和压缩包名称为“学号\_姓名\_实验四”。
   1. 提交文件包括：实验报告和源代码压缩包，命名格式均为“学号\_姓名\_实验四”。



* 1. 源代码压缩包内要求有两个文件夹，一个为代码文件夹，命名为“code”，一个为可执行文件夹，命名为“exe”。



* 1. 代码文件夹中只能包含代码和代码需要用到的资源文件（比如纹理图片、模型），其他由编辑器或者编译器创建项目时候生成的文件全部都不要加上，不清楚的同学可以询问助教。



* 1. 可执行文件夹中，只包含可执行文件以及执行所需的动态库文件和资源文件等，要求可以直接点击该程序就可正常执行（详情可以查看实验1.1的ppt或者word最后的内容，每个配置方法最后有一页内容讲解）。



1. **截止时间：2024年12月10日 23:59**