**深 圳 大 学 实 验 报 告**

**课程名称： 计算机图形学**

**实验项目名称： 实验四 带纹理的OBJ文件读取和显示**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师： 徐鹏飞**

**报告人： 郑雨婷 学号： 2021150122 班级： 高性能**

**实验时间： 2023年 12月15日 -- 2023年12月17日**

**实验报告提交时间： 2023年12月 17日**

**教务部制**

|  |
| --- |
| 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。    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>      cube文件示例   1. 实验步骤   1.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。  读取obj文件前，首先需要清楚obj文件的格式，打开本次实验需要用到的table.obj文件。可以看到，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。根据不同的关键字需要进行不同的数据存储操作。因此，对于obj文件的每一行，首先读取关键字type，根据关键字来存储相应的数据。    图1 读取关键字  （1）当type为v时，其obj文件格式如下，后跟三个浮点数，代表顶点的几何坐标。通过输入流sin将本行之后的三个浮点数读入\_x,\_y,\_z中，再构造三维顶点坐标存入vertex\_positions容器中。    图2 type==v时几何坐标  （2）当type为vt时，其obj文件格式如下，后跟三个浮点数，代表顶点的纹理坐标。但是其实为纹理坐标是二维坐标，第三个浮点数我猜测只是为了格式统一，因此其值都为0。因为纹理坐标是二维坐标，只需要通过输入流sin将本行之后的两个浮点数读入\_x,\_y中，再构造二维顶点坐标存入vertex\_textures容器中。    图3 type==v时纹理坐标  （3）当type为vn时，其obj文件格式如下，后跟三个浮点数，代表顶点的法线。并且，我们根据法向量的数值来赋予颜色。通过输入流sin将本行之后的三个浮点数读入\_x,\_y,\_z中，再构造三维顶点坐标存入vertex\_normals和vertex\_colors容器中。    图4 type==vn时法向量  （4）当type为p时，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  通过输入流sin将每个顶点的顶点索引下标存入a、纹理的UV索引下标存入b、法向量索引下标存入c中，读入时需要用一个字符型变量slash将/接受。因为是三角面片，一个面片又三个顶点，需要存三组。然后顶点的下标压入faces中，把顶点对应的纹理的下标压入texture\_index向量中，把法向量对应的下标压入normal\_index中，因为法向量可以反应颜色，所以又把法向量下标压入color\_index中。    图5 type==p时面片的各个索引  需要注意，obj文件的下标是从1开始的，所以我们在压入数组之前需要把下标都减一，否则会出现一些毛刺，如图6所示，有些点的数读取失败。    图6 下标未减1时的错误  2.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。  在storeFacesPoints()函数中，已经实现物体的归一化和计算法向量了。看是否需要归一化，也就是do\_normalize\_size是否为1，如果需要的话，我们在vertex\_positions向量中分别寻找xyz值对应的最大和最小值，并最后把所有点的位置都做一下归一化。vertex\_normals向是的空话，意味着我们的法向量还没填充，那就调用computeVertexNormals()函数填充法向量。  我们需要完成的是根据每个三角面片的顶点下标存储要传入GPU的数据。我们根据在obj文件读取的下标，把点压入points中，把颜色压入colors中，把法向量压入normals，把纹理坐标压入textures，以便在主函数中把points、colors、normals和textures传到GPU中，绘制图像。    图7 storeFacesPoints()函数补充代码  3.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。  首先创建了一个指向 TriMesh 类型对象的指针 table。然后，通过调用 setNormalize 函数对模型进行了归一化操作，以确保模型尺寸在标准范围内。接下来，通过 readObj 函数从文件路径 ./assets/table.obj 中读取桌子模型的数据并存储到 table 对象中。随后，分别使用 setTranslation、setRotation 和 setScale 函数对模型进行了平移、旋转和缩放的处理，使其在场景中的位置、方向和大小得到了调整。最后，通过调用 painter->addMesh 函数将桌子模型添加到渲染器中进行渲染，指定纹理文件路径 "./assets/table.png" 以及顶点着色器和片段着色器的路径。这一系列操作使得桌子模型成功加载到三维场景中，并准备好进行渲染。    图8 init()中初始化table的代码部分  同理，绘制娃娃的代码如下：    图9 init()中初始化wawa的代码部分  从模型的读取到数据存储完成可能需要一些时间，因此在编译的时候选择Release模式，这样生成的程序执行速度会快一些。编译运行后的结果如图20，结果正确：    图10 最终运行结果   1. 实验结论：   在本次实验中，首先，通过实现TriMesh类中的readObj函数，成功实现了对带纹理的OBJ文件的读取。该函数的设计使得程序能够准确地提取UV坐标、顶点坐标、顶点法线、顶点颜色、三角面片的顶点索引、法向量数据的索引下标以及纹理坐标数据的索引下标，有效地存储这些数据，为后续的渲染和处理提供了基础。之后通过完善TriMesh.cpp中的storeFacesPoints函数，进一步优化了对模型数据的读取和存储。该函数的补全使得程序结构更加完整，确保了数据存储的准确性和可靠性。最后，在main.cpp的init函数中进行了必要的修改，确保程序能够正常运行并成功显示贴有纹理的玩偶模型和桌子模型。这一步骤是整个实验的最终目标，通过在渲染阶段验证纹理能够正确地贴在模型表面。  总之，通过这次实验，我对纹理贴图的应用和处理方式有了更为清晰的认识。 |

深圳大学学生实验报告用纸

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字： 2022年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。