

华东师范大学计算机科学与技术实验报告

实验课程：计算机图形学	年级：18级	实验成绩：
实验名称：导入三维模型	姓名：董辰尧	实验日期：2021/6/1
实验编号：13	学号：10185102144	实验时间：13:00~14:40
指导教师：李洋、王长波	组号：	

一、实验目的

实现三维模型的导入与渲染。

二、实验环境

- Windows 10
- D3DX
- OpenGL

三、实验内容

1. 读入.obj文件
2. 绘制mesh

四、实验过程与分析

- 本次实验没有给出代码，需要自己根据理解写，难度较大，首先需要读出来.obj文件的内容
几何实体必须以"V"、"VT"、"VN"或"F"开头，一个几何实体占据一行。所以在解析obj文件的时候，若一行不是以V\VT\VN\F这几个字母开头，就可以忽略掉这行。

下面是几个字母代表的几何实体类型：

V：代表顶点。格式为V X Y Z，V后面的X Y Z表示三个顶点坐标。浮点型

VT：表示纹理坐标。上面的立方体有24个纹理坐标，因为每个三角形面的三个顶点，都需要指定一个纹理坐标。格式为VT TU TV。浮点型

VN：法向量。同样，上面立方体也有24个法向量，因为每个三角形的三个顶点都要指定一个法向量。格式为VN NX NY NZ。浮点型

F：面。面后面跟着的整型值分别是属于这个面的顶点、纹理坐标、法向量的索引。面的格式为：

f Vertex1/Texture1/Normal1 Vertex2/Texture2/Normal2 Vertex3/Texture3/Normal3

比如这样一行索引f 7/9/21 1/10/22 5/11/23，包含三组索引，构成了一个面。

```
1  ObjLoader::ObjLoader(string filename)
2  {
3      string line;
4      fstream f;
5      f.open(filename, ios::in);
6      if (!f.is_open()){
```

```

7         cout << "Something Went Wrong When Opening Objfiles" <<
endl;
8     }
9     while (!f.eof()){
10         getline(f, line);
11         vector<string>parameters;
12         string tailMark = " ";
13         string ans = "";
14         line = line.append(tailMark);
15         for (int i = 0; i < line.length(); i++) {
16             char ch = line[i];
17             if (ch != ' ') {
18                 ans+=ch;
19             }
20             else {
21                 parameters.push_back(ans);
22                 ans = "";
23             }
24         }
25         //cout << parameters.size() << endl;
26         if (parameters.size() != 4) {
27             cout << "the size is not correct" << endl;
28         }
29         else {
30             if (parameters[0] == "v") {
31                 vector<GLfloat>Point;
32                 for (int i = 1; i < 4; i++) {
33                     GLfloat xyz = atof(parameters[i].c_str());
34                     Point.push_back(xyz);
35                 }
36                 vSets.push_back(Point);
37             }
38             else if (parameters[0] == "f") {
39                 vector<GLint>vIndexSets;
40                 for (int i = 1; i < 4; i++){
41                     string x = parameters[i];
42                     string ans = "";
43                     for (int j = 0; j < x.length(); j++) {
44                         char ch = x[j];
45                         if (ch != '/') {
46                             ans += ch;
47                         }
48                         else {
49                             break;
50                         }
51                     }
52                     GLint index = atof(ans.c_str());
53                     index = index--;
54                     vIndexSets.push_back(index);
55                 }
56                 fSets.push_back(vIndexSets);
57             }
58         }
59     }
60     f.close();
61 }

```

读取文件其实就是简单的调用函数，但是需要读文件的过程中就进行文件内容的处理。

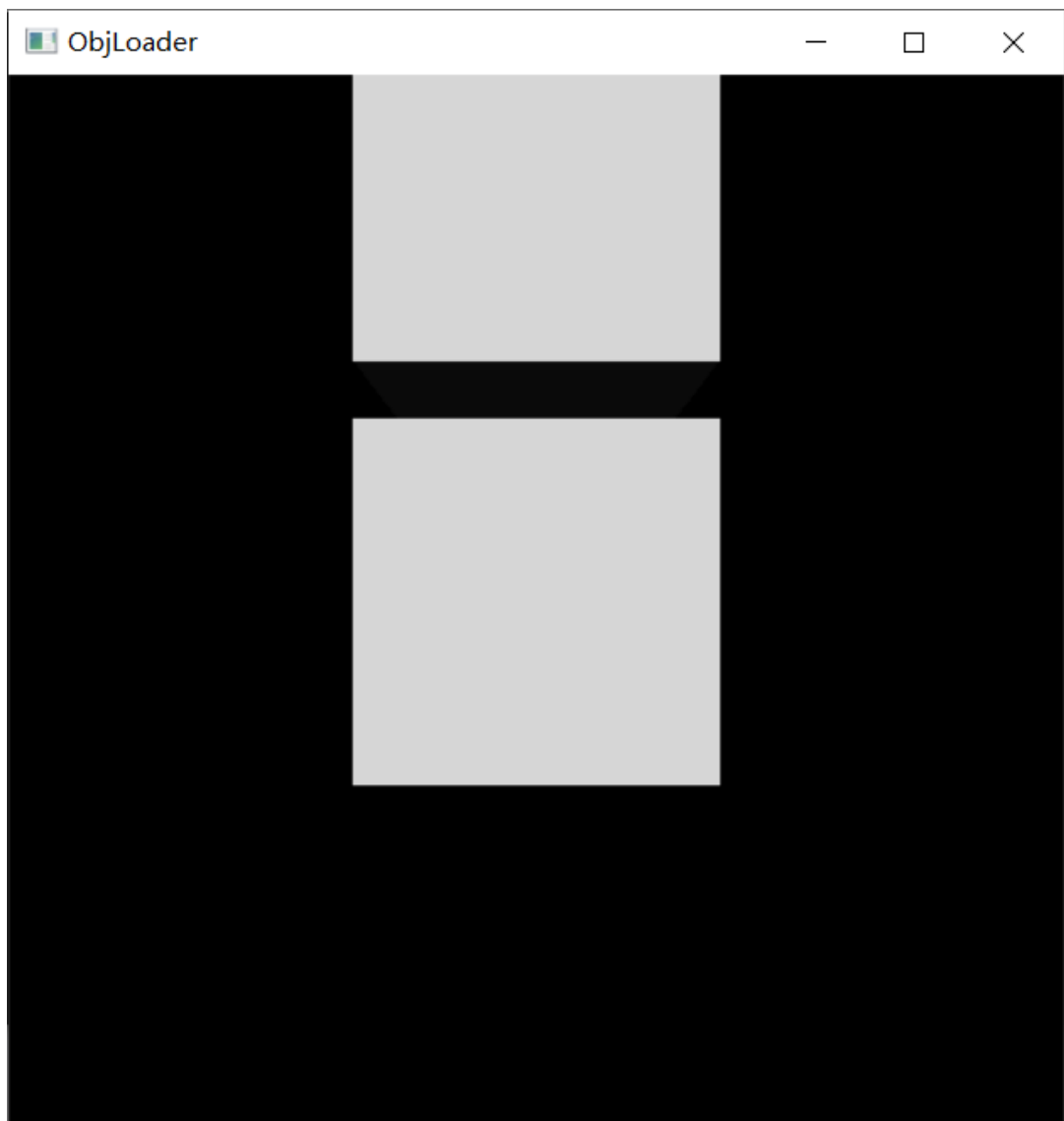
- 绘制mesh，主要代码如下

```
1 void ObjLoader::Draw(){
2
3     glBegin(GL_TRIANGLES);
4
5     for (int i = 0; i < fSets.size(); i++) {
6
7         GLfloat VN[3];
8
9         GLfloat SV1[3];
10        GLfloat SV2[3];
11        GLfloat SV3[3];
12
13
14        if ((fSets[i]).size() != 3) {
15            cout << "the fSetsets_size is not correct" << endl;
16        }
17
18        else {
19
20            GLint firstVertexIndex = (fSets[i])[0];
21            GLint secondVertexIndex = (fSets[i])[1];
22            GLint thirdVertexIndex = (fSets[i])[2];
23
24            SV1[0] = (vSets[firstVertexIndex])[0];
25            SV1[1] = (vSets[firstVertexIndex])[1];
26            SV1[2] = (vSets[firstVertexIndex])[2];
27
28            SV2[0] = (vSets[secondVertexIndex])[0];
29            SV2[1] = (vSets[secondVertexIndex])[1];
30            SV2[2] = (vSets[secondVertexIndex])[2];
31
32            SV3[0] = (vSets[thirdVertexIndex])[0];
33            SV3[1] = (vSets[thirdVertexIndex])[1];
34            SV3[2] = (vSets[thirdVertexIndex])[2];
35
36            GLfloat vec1[3], vec2[3], vec3[3];
37            //(x2-x1,y2-y1,z2-z1)
38            vec1[0] = SV1[0] - SV2[0];
39            vec1[1] = SV1[1] - SV2[1];
40            vec1[2] = SV1[2] - SV2[2];
41
42            //(x3-x2,y3-y2,z3-z2)
43            vec2[0] = SV1[0] - SV3[0];
44            vec2[1] = SV1[1] - SV3[1];
45            vec2[2] = SV1[2] - SV3[2];
46
47            //(x3-x1,y3-y1,z3-z1)
48            vec3[0] = vec1[1] * vec2[2] - vec1[2] * vec2[1];
49            vec3[1] = vec2[0] * vec1[2] - vec2[2] * vec1[0];
50            vec3[2] = vec2[1] * vec1[0] - vec2[0] * vec1[1];
51
52            GLfloat D = sqrt(pow(vec3[0], 2) + pow(vec3[1],
53                2) + pow(vec3[2], 2));
```

```
54         VN[0] = vec3[0] / D;  
55         VN[1] = vec3[1] / D;  
56         VN[2] = vec3[2] / D;  
57  
58         glNormal3f(VN[0], VN[1], VN[2]);  
59  
60         glVertex3f(sv1[0], sv1[1], sv1[2]);  
61         glVertex3f(sv2[0], sv2[1], sv2[2]);  
62         glVertex3f(sv3[0], sv3[1], sv3[2]);  
63  
64     }  
65 }  
66  
67 glEnd();  
68  
69 }
```

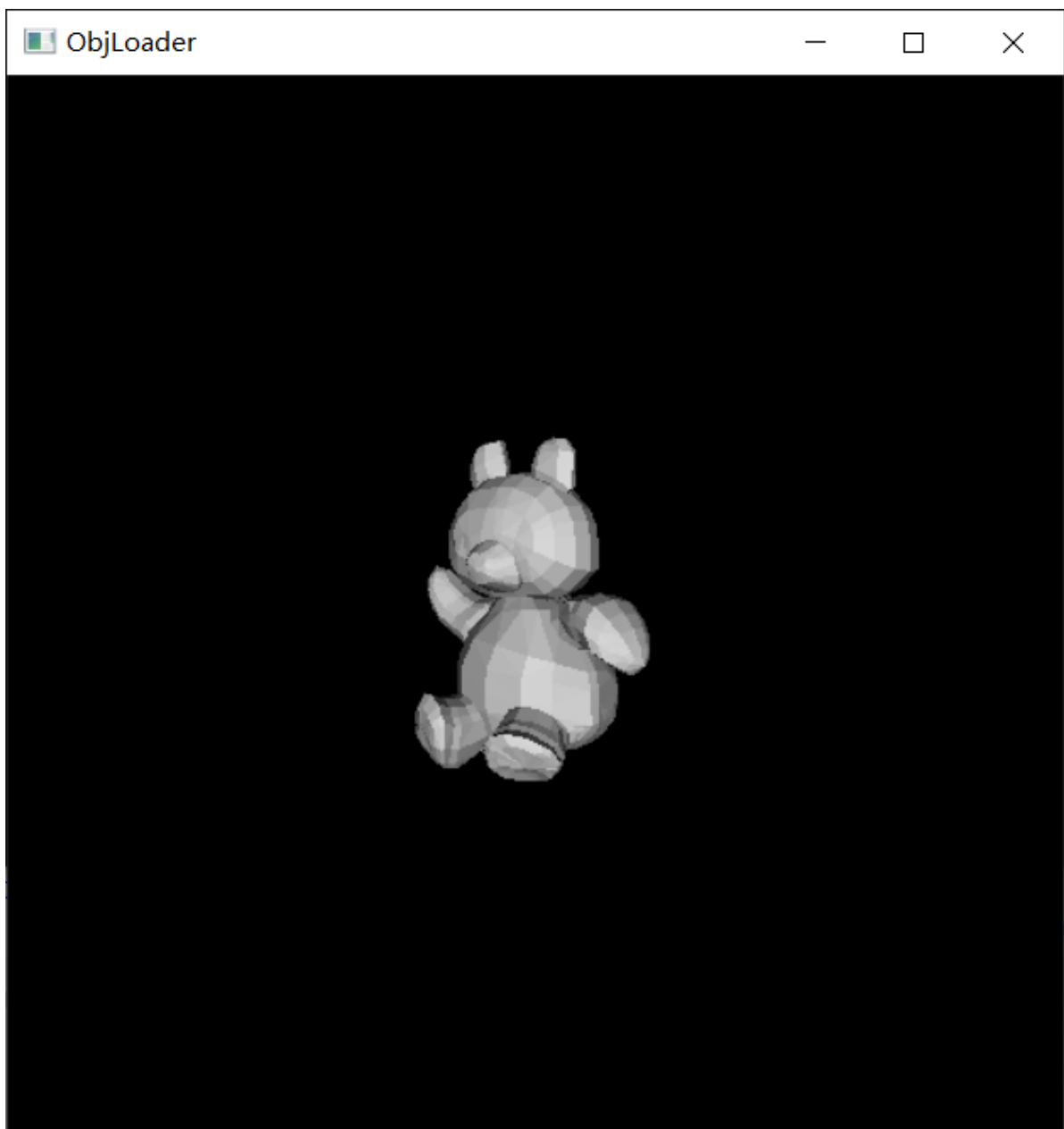
五、实验过程总结

- 结果截图：





![[image-20210601174922118](C:\Users\大草原\AppData\Roaming\Typora\typora-user-images\image-20210601174922118).]



这个是在网上找到的其他模型。