

练习题报告

课程名称 计算机图形学

项目名称 OFF 格式的模型显示

学 院 计算机与软件学院

专 业 计算机科学与技术

指导教师 周虹

报 告 人 林浩晟 学号 2022280310

一、 练习目的

1. 了解和熟悉 OFF 模型文件格式。
2. 掌握读取 OFF 模型文件。
3. 了解基本 3D 图元的绘制。
4. 了解深度测试技术。
5. 了解面剔除技术。

二. 练习完成过程及主要代码说明

1. 补充代码，将顶点坐标和面片信息从文件中读取，储存到对应变量。

```
// 根据顶点数，循环读取每个顶点坐标，将其保存到vertices
for (int i = 0; i < nVertices; ++i) {
    float x, y, z;
    fin >> x >> y >> z;
    vertices.push_back(glm::vec3(x, y, z)); //读取每一个顶点坐标，保存到vertices
}

// 根据面片数，循环读取每个面片信息，并用构建的vec3i结构体保存到faces
for (int i = 0; i < nFaces; ++i) {
    int numVertices, a, b, c;
    fin >> numVertices;
    fin >> a >> b >> c;
    faces.push_back(vec3i(a, b, c));
}
```

2. 循环设置面片信息，并将面片顶点放入 points 中，最后设置每个顶点的颜色。

```
int i = 0;
for (const auto& face : faces) {
    // 每个三角形面片包含三个顶点索引
    glm::vec3 vertexA = vertices[face.a];
    glm::vec3 vertexB = vertices[face.b];
    glm::vec3 vertexC = vertices[face.c];

    // 添加三角形面片的三个顶点到points容器中
    points.push_back(vertexA);
    points.push_back(vertexB);
    points.push_back(vertexC);

    // 给每个顶点设置颜色，可以选择根据顶点索引使用vertex_colors
    colors.push_back(vertex_colors[i/2]);
    colors.push_back(vertex_colors[i/2]);
    colors.push_back(vertex_colors[i/2]);
    i++;
}
```

3. 这里补充深度缓存，同时进行清理颜色缓存和深度缓存。

```
void display(void)
{
    // @TODO: Task2:清理窗口，包括颜色缓存和深度缓存
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); //颜色缓存和深度缓存
    glDrawArrays(GL_TRIANGLES, 0, points.size());
}
```

4. 随后根据实验指引添加深度测试和关闭深度测试的代码，在启用深度测试后应清理颜色缓存和深度缓存，面剔除和线绘制同理，这里不做解释。最后打开注释即可。

```
// @TODO: Task2:启用深度测试
else if (key == GLFW_KEY_1 && action == GLFW_PRESS && mode == 0x0000) // 0x0000表示组合键为空
{
    cout << "depth test: enable" << endl;
    glEnable(GL_DEPTH_TEST); //深度测试
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); //清理颜色缓存和深度缓存
}

// @TODO: Task2:关闭深度测试
else if (key == GLFW_KEY_1 && action == GLFW_PRESS && mode == GLFW_MOD_SHIFT)
{
    cout << "depth test: disable" << endl;
    glDisable(GL_DEPTH_TEST);
}
```

```
// @TODO: Task3:启用反面剔除
else if (key == GLFW_KEY_2 && action == GLFW_PRESS && mode == 0x0000)
{
    cout << "cull back: enable" << endl;
    glEnable(GL_CULL_FACE); //面剔除
    glCullFace(GL_BACK); //反面
}

// @TODO: Task3:关闭反面剔除
else if (key == GLFW_KEY_2 && action == GLFW_PRESS && mode == GLFW_MOD_SHIFT)
{
    cout << "cull back: disable" << endl;
    glDisable(GL_CULL_FACE);
}

// @TODO: Task4:启用正面剔除
else if (key == GLFW_KEY_3 && action == GLFW_PRESS && mode == 0x0000)
{
    cout << "cull front: enable" << endl;
    glEnable(GL_CULL_FACE); //面剔除
    glCullFace(GL_FRONT); //正面
}

// @TODO: Task4:关闭正面剔除
else if (key == GLFW_KEY_3 && action == GLFW_PRESS && mode == GLFW_MOD_SHIFT)
{
    cout << "cull front: disable" << endl;
    glDisable(GL_CULL_FACE);
}
```

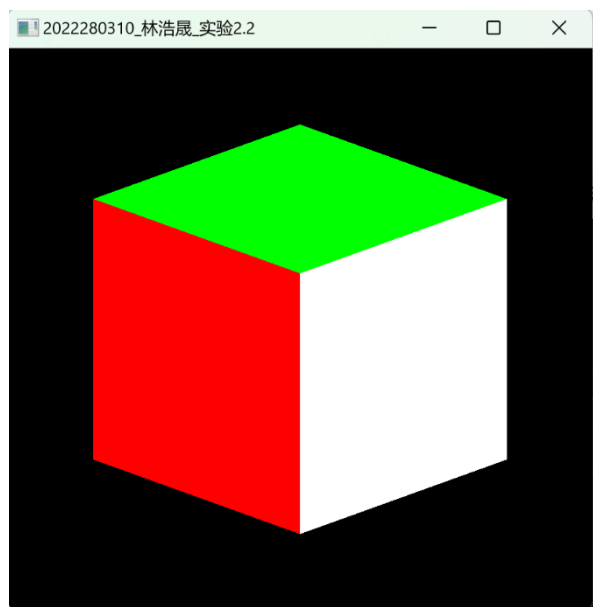
```

// @TODO: Task5:启用线绘制模式
else if (key == GLFW_KEY_4 && action == GLFW_PRESS && mode == 0x0000)
{
    cout << "line mode: enable" << endl;
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE); //线绘模式
}
// @TODO: Task5:关闭线绘制模式
else if (key == GLFW_KEY_4 && action == GLFW_PRESS && mode == GLFW_MOD_SHIFT)
{
    cout << "line mode: disable" << endl;
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
}

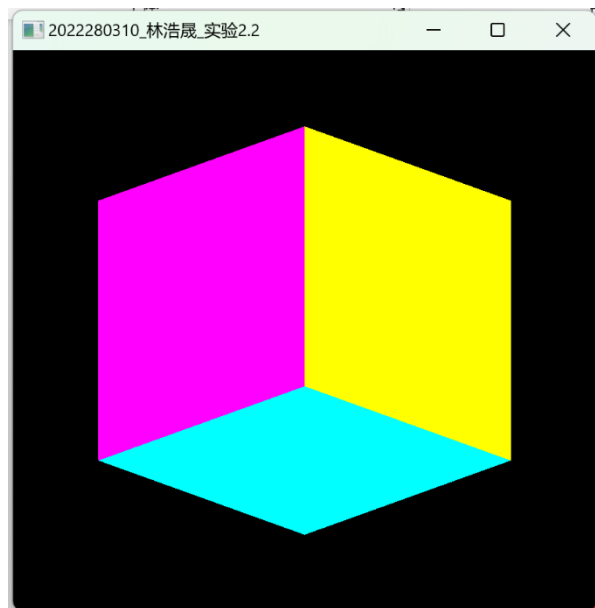
```

5.实验结果

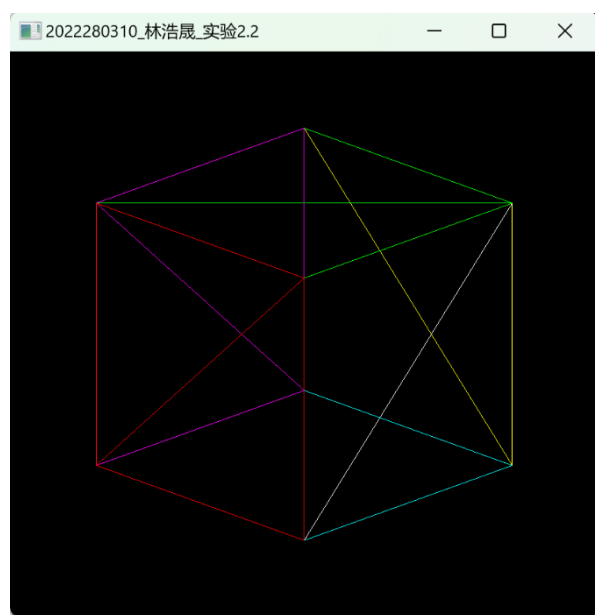
按下 1 开启深度测试



按下 3 开启正面剔除



按下 4 开启线模式



7.同时可以更改顶点颜色处代码，如下

```
colors.push_back(vertex_colors[face.a]);  
colors.push_back(vertex_colors[face.b]);  
colors.push_back(vertex_colors[face.c]);
```

8.可以得到如下结果

