

练习题报告

课程名称 计算机图形学

项目名称 三维模型的平移、缩放和旋转

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. 掌握三维模型顶点与三角面片之间关系。
2. 了解和掌握三维模型的基本变换操作。
3. 掌握在着色器中使用变换矩阵。

二. 练习完成过程及主要代码说明

1. 首先遍历 `cube_vertices`，将其中数据传给 `vertex_positions`，同理将 `basic_colors` 的数据传给 `vertex_colors`；再定义立方体的 6 个面。

```
for (auto& vertice : cube_vertices)
{
    vertex_positions.push_back(vertice);
}
for (auto& color : basic_colors)
{
    vertex_colors.push_back(color); //存放定义了的颜色
}
// faces再记录每个面片上顶点的下标
// 定义立方体的12个三角形（6个面，每个面2个三角形）
faces = {
    // 前面
    {0, 1, 2}, {1, 3, 2},
    // 后面
    {4, 6, 5}, {5, 6, 7},
    // 左面
    {0, 2, 4}, {2, 6, 4},
    // 右面
    {1, 5, 3}, {3, 5, 7},
    // 上面
    {2, 3, 6}, {3, 7, 6},
    // 下面
    {0, 4, 1}, {1, 4, 5}
};
```

2. 然后再遍历每一个面，将其中的位置信息和颜色信息传给 `points` 和 `colors`，先从 `faces` 中提取出索引，再利用索引获取位置信息；再循环 12 次（表示立方体一共由 12 个三角形构成），将每一个面的两个三角形赋值为同一颜色，保证一个面的颜色统一（0/2=0, 1/2=0, 以此类推）。

```

for (const vec3i& face : faces)//std::vector<vec3i> faces;
{
    glm::vec3 p1 = vertex_positions[face.x];
    glm::vec3 p2 = vertex_positions[face.y];
    glm::vec3 p3 = vertex_positions[face.z];
    /*
    glm::vec3 c1 = vertex_colors[face.x];
    glm::vec3 c2 = vertex_colors[face.y];
    glm::vec3 c3 = vertex_colors[face.z];
    colors.push_back(c1);
    colors.push_back(c2);
    colors.push_back(c3);
    */

    points.push_back(p1);//储存到points和colors
    points.push_back(p2);
    points.push_back(p3);
}
for (int i = 0;i < 12;i++)//一个三角形同一种颜色
{
    colors.push_back(vertex_colors[i/2]);
    colors.push_back(vertex_colors[i/2]);
    colors.push_back(vertex_colors[i/2]);
}

```

3. 将注释打开;

```

// #TODO: Task3 修改完TriMesh.cpp的代码后再打开下面注释, 否则程序会报错
glBufferSubData(GL_ARRAY_BUFFER, 0, mesh->getPoints().size() * sizeof(glm::vec3), &mesh->getPoints()[0]);
glBufferSubData(GL_ARRAY_BUFFER, mesh->getPoints().size() * sizeof(glm::vec3), mesh->getColors().size() * sizeof(glm::vec3), &mesh->getColors()[0]);

```

4. 首先使用 `scale` 函数获取缩放矩阵; 再用 `rotate` 函数分别获取各个方向旋转的矩阵, 其中 `glm::radians(rotateTheta.x)` 表示每次旋转角度, `vec3(1.0f, 0.0f, 0.0f)` 表示旋转轴为 `x` 轴, 其余同理, 最后再全乘起来, 用于三维空间中的旋转矩阵; 再用 `translate` 函数得到平移矩阵, 最后将三个矩阵乘起来得到最终的变换矩阵。

```

//缩放矩阵
glm::mat4 scaleMatrix = glm::scale(glm::mat4(1.0f), scaleTheta);

//旋转矩阵 (按 X、Y、Z 轴依次旋转)
glm::mat4 rotateX = glm::rotate(glm::mat4(1.0f), glm::radians(rotateTheta.x), glm::vec3(1.0f, 0.0f, 0.0f));
glm::mat4 rotateY = glm::rotate(glm::mat4(1.0f), glm::radians(rotateTheta.y), glm::vec3(0.0f, 1.0f, 0.0f));
glm::mat4 rotateZ = glm::rotate(glm::mat4(1.0f), glm::radians(rotateTheta.z), glm::vec3(0.0f, 0.0f, 1.0f));
glm::mat4 rotateMatrix = rotateZ * rotateY * rotateX ;//三维空间中旋转的矩阵

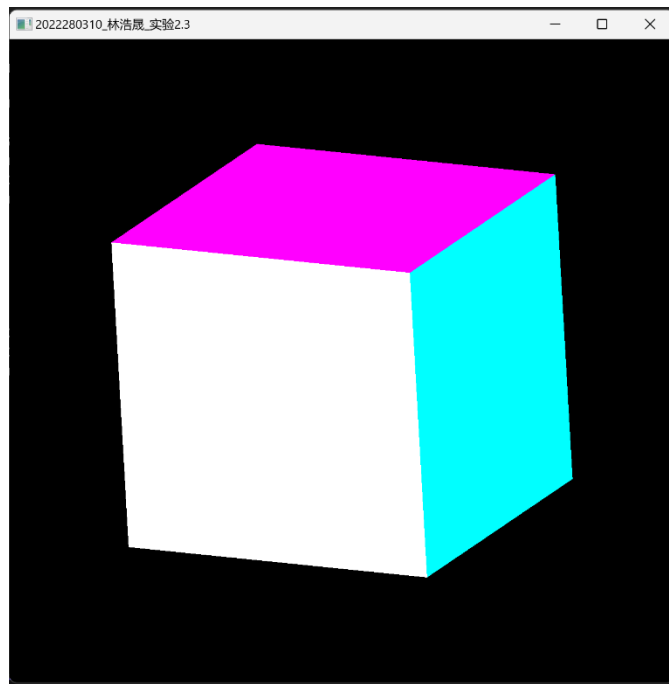
//平移矩阵
glm::mat4 translateMatrix = glm::translate(glm::mat4(1.0f), translateTheta);

//最终的变换矩阵
m = translateMatrix * rotateMatrix * scaleMatrix;

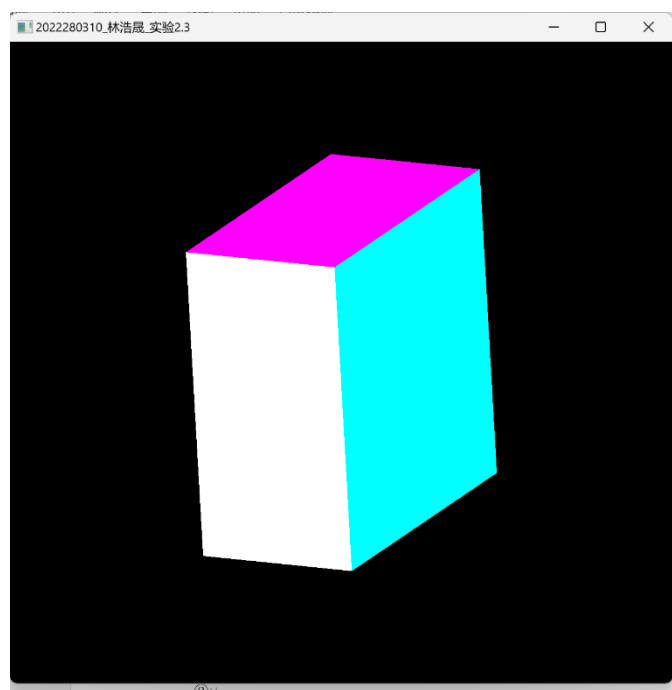
```

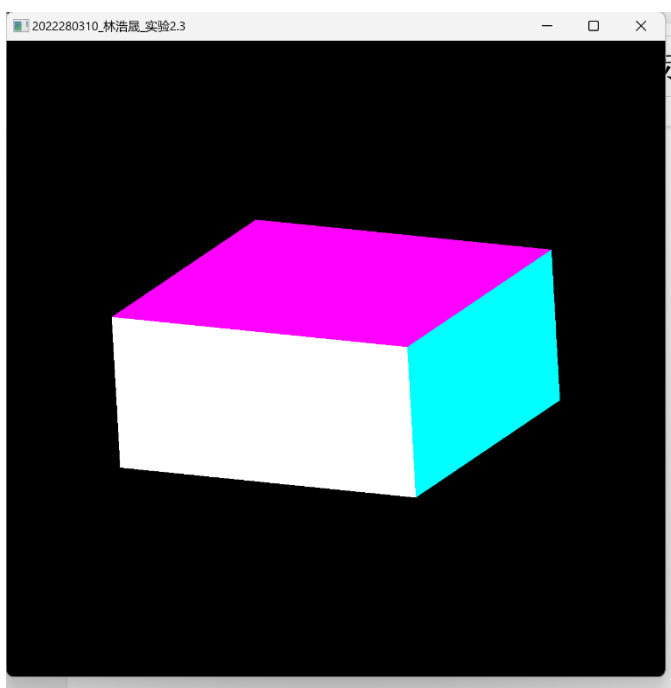
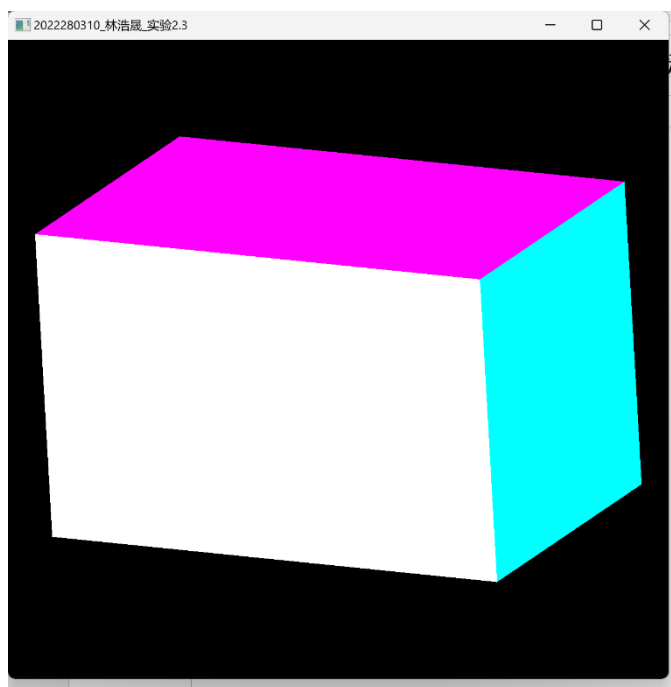
5. 实验结果

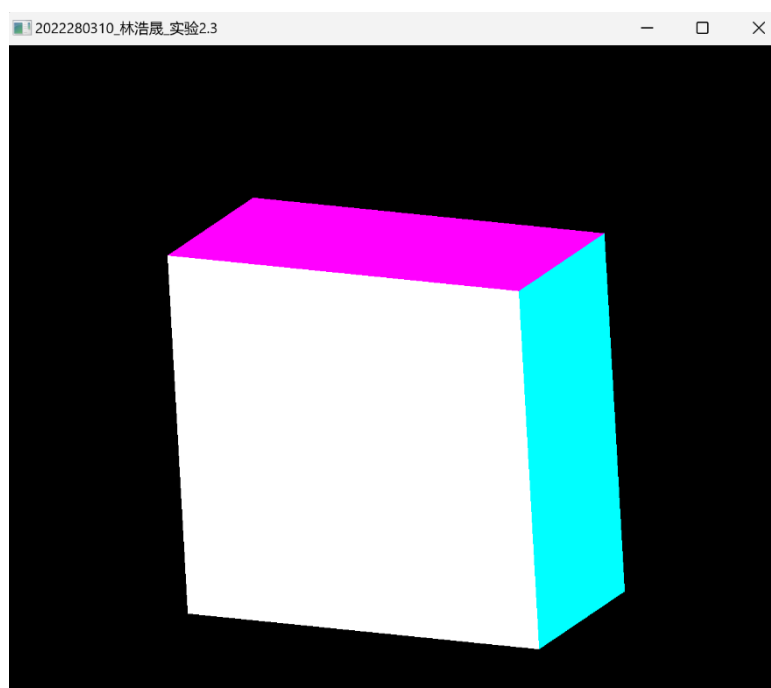
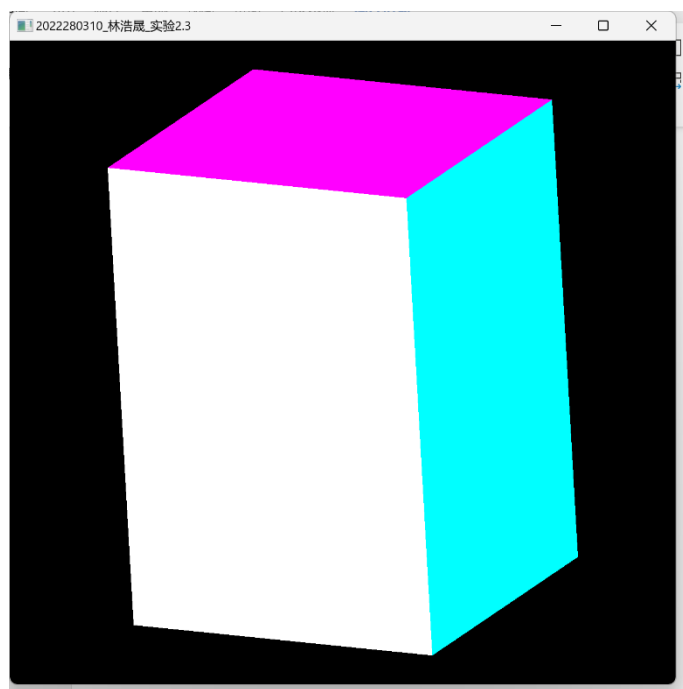
①将图形旋转至如下所示, 可知旋转功能完成

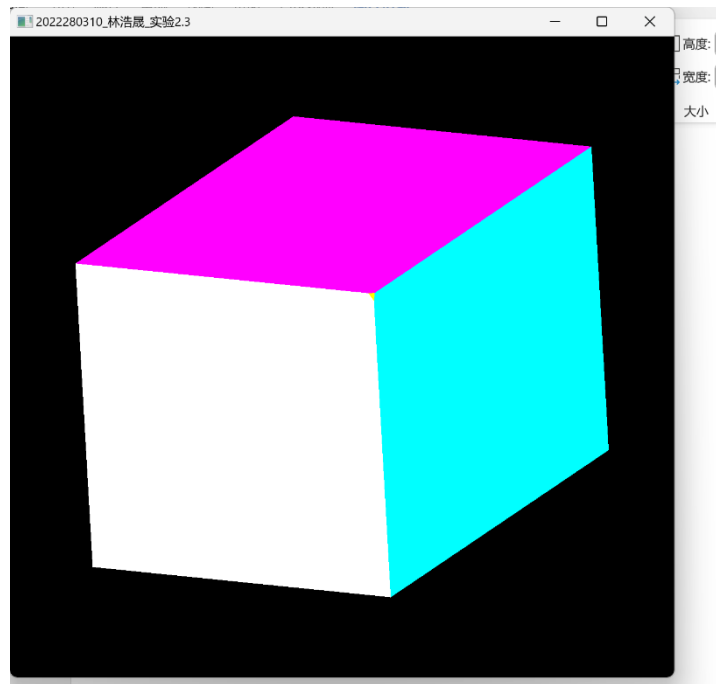


②x 轴方向缩小放大，y 轴方向缩小放大和 z 轴方向缩小放大如下所示









③x 方向向左移向右移，y 方向向左移右移，z 方向向左移右移如下所示

