

浙江大学实验报告

专业： 计算机科学与技术
姓名： 方彦祺
学号： 3220102829
日期： 2024-10-23
地点： 无

课程名称： 计算机图形学 指导老师： 童若锋 成绩：
实验名称： OpenGL 绘制五星红旗 实验类型： 基础实验 同组学生姓名： 无

一、实验目的和要求

学会配置 OpenGL 开发环境并使用图形 API 绘制五星红旗。



二、实验内容和原理

本次实验需要我们完成的部分是绘制一颗标准的正五角星的过程，以及调整参数并调用这个绘制正五角星的函数来在国旗上绘制特定位置的五颗星星。

查看绘制正五角星的函数，发现使用了常量 `GL_TRIANGLES`，这表示加入 `vertices` 变量的参数坐标每三个点绘制成一个三角形，因此我们这样分割一个正五角星：



即我们取上图蓝色等腰三角形的三个点为一组，共十五个点，绘制五个等腰三角形，即可组成一颗正五角星。得出第一个等腰三角形后，之后的每个三角形相当于绕正五角星的中心旋转 72° 。**Position** 中存储的 x, y 坐标值表示正五角星的中心的坐标，**radius** 变量存储这个正五角星外接圆的半径，也即正五角星中心到外点的距离，这是正五角星到内点距离的三倍，这个变量用以控制正五角星的大小。**Rotation** 变量代表整个正五角星在背景上旋转的角度，因此在将绘制三角形的坐标参数中加上角度 **rotation**，即可实现旋转。**Aspect** 参数是输出窗口从长宽比，为使输出图不产生形变，我们需要将具体坐标的 y 值距离 **position** 原点 y 值的值乘上 **aspect**，这样就能消除形变。

由此，我们得到了各个等腰三角形的三个顶点的表示坐标：

```
1. (_position.x + cos(i * 2.0f * M_PI / 5.0f + M_PI / 10.0f + _rotation) *  
    _radius, _position.y + sin(i * 2.0f * M_PI / 5.0f + M_PI / 10.0f + _ro  
    tation) * _radius * aspect)  
2. (_position.x + cos(i * 2.0f * M_PI / 5.0f + M_PI / 10.0f + _rotation +  
    M_PI * 3.0f / 5.0f) * _radius / 3.0f, _position.y + sin(i * 2.0f * M_PI  
    / 5.0f + M_PI / 10.0f + _rotation + M_PI * 3.0f / 5.0f) * _radius / 3.0  
    f * aspect)  
3. (_position.x + cos(i * 2.0f * M_PI / 5.0f + M_PI / 10.0f + _rotation +  
    M_PI * 7.0f / 5.0f) * _radius / 3.0f, _position.y + sin(i * 2.0f * M_PI  
    / 5.0f + M_PI / 10.0f + _rotation + M_PI * 7.0f / 5.0f) * _radius / 3.0  
    f * aspect)
```

然后我们使用这个函数来在红旗上绘制五颗星星，查阅相关资料，得到各个星星中心所处的位置、大小比例、旋转角度，在此不详细列出，需要注意的是，整个输出窗口的 x, y 分别取[-1,1]，这是由 ndc 坐标系决定的。将相关参数设置并传入 **Star** 函数，这由以下语句实现：

```
1. _stars[i].reset(new Star(ndc_position[i], rotation_in_radians[i], size_of_star, aspect_of_the_window));
```

四个参数分别代表正五角星中心点的坐标，星星的旋转角度，外接圆半径，屏幕长宽比。

三、主要仪器设备

Win11 系统的电脑，Visual Studio

四、操作方法和实验步骤

依据实验原理中的内容，完成代码，补充的代码包括

Star.cpp:

```
_vertices.reserve(15);

for (int i=0;i<5;i++){
    float outPoint = i * 2.0f * M_PI / 5.0f + M_PI / 10.0f + _rotation;
    _vertices.push_back(glm::vec2(_position.x + cos(outPoint) * _radius, _position.y + sin(outPoint) * _radius * aspect));

    float inPoint_1 = outPoint + M_PI * 3.0f / 5.0f;
    float inPoint_2 = outPoint + M_PI * 7.0f / 5.0f;
    float inRadius = _radius / 3.0f;
    _vertices.push_back(glm::vec2(_position.x + cos(inPoint_1) * inRadius, _position.y + sin(inPoint_1) * inRadius * aspect));
    _vertices.push_back(glm::vec2(_position.x + cos(inPoint_2) * inRadius, _position.y + sin(inPoint_2) * inRadius * aspect));
}
```

Render_flag.cpp:

```
float aspect_of_the_window = static_cast<float>(options.windowWidth) / static_cast<float>(options.windowHeight);
float rotation_in_radians[5];
glm::vec2 ndc_position[5];
float size_of_star = 0.05f;

rotation_in_radians[0] = M_PI / 10.0f + atan(0.6);
ndc_position[0] = glm::vec2(-1.0f / 3.0f, 0.8f);

rotation_in_radians[1] = M_PI / 10.0f + atan2(1, 7);
ndc_position[1] = glm::vec2(-0.2f, 0.6f);

rotation_in_radians[2] = M_PI / 10.0f - atan2(2, 7);
ndc_position[2] = glm::vec2(-0.2f, 0.3f);

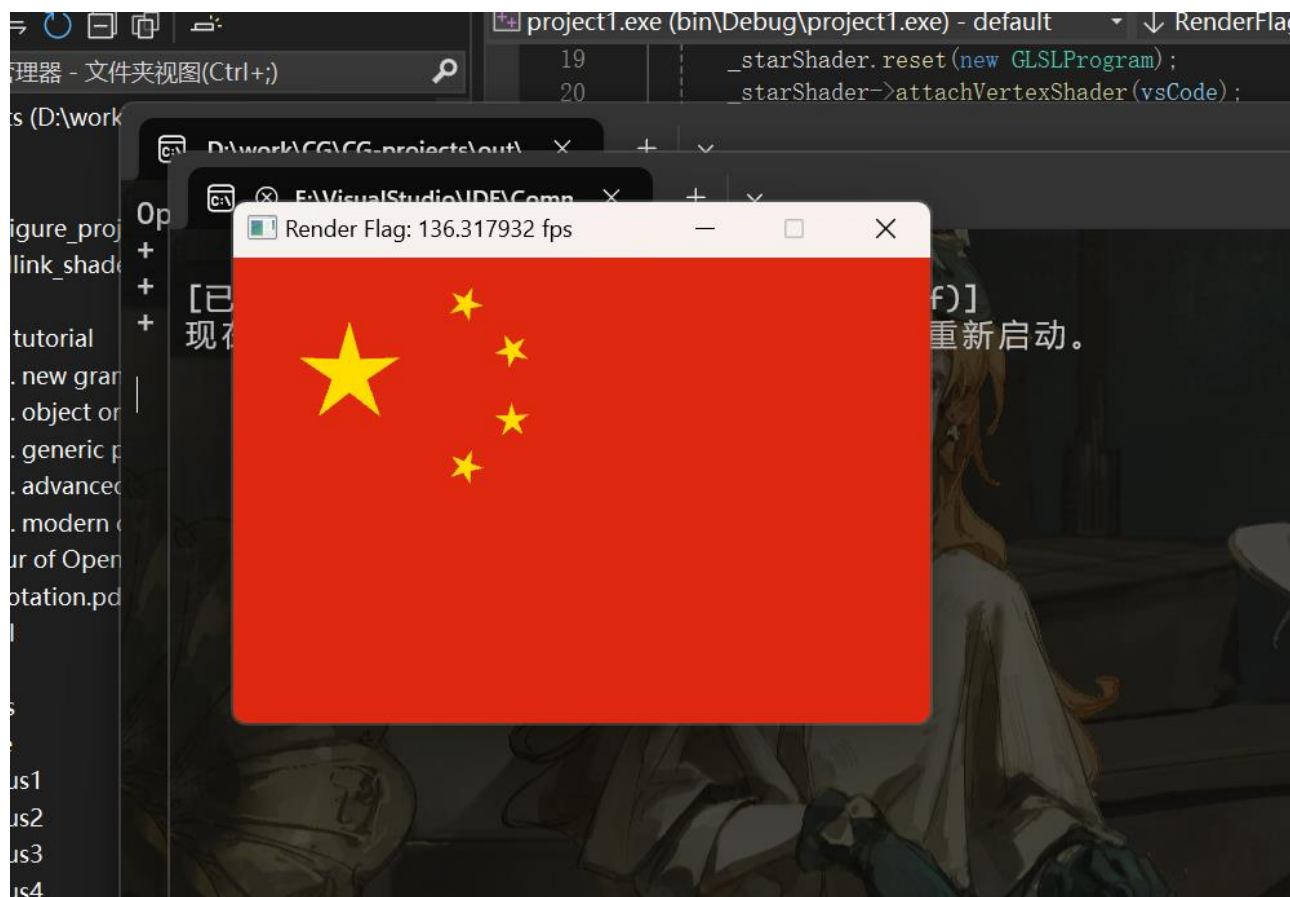
rotation_in_radians[3] = M_PI / 10.0f - atan(0.8);
ndc_position[3] = glm::vec2(-1.0f / 3.0f, 0.1f);

rotation_in_radians[4] = 0;
ndc_position[4] = glm::vec2(-2.0f / 3.0f, 0.5f);

for (int i = 0; i < 4; i++)
    _stars[i].reset(new Star(ndc_position[i], rotation_in_radians[i], size_of_star, aspect_of_the_window));
_stars[4].reset(new Star(ndc_position[4], rotation_in_radians[4], size_of_star * 3.0f, aspect_of_the_window));
```

五、实验结果与分析

运行工程，得到输出：



六、讨论、心得

本次实验较为简单，个人认为重点在于理解程序使用的是使用三角形绘制图形，以及查阅五星红旗上五颗星星的相关参数，此外就只需要自己在坐标系上思考理解怎样表示各个点即可。

七、参考链接

- [1] [OpenGL - LearnOpenGL CN \(learnopengl-cn.github.io\)](https://learnopengl-cn.github.io/)
- [2] [创建窗口 - LearnOpenGL CN \(learnopengl-cn.github.io\)](https://learnopengl-cn.github.io/)
- [3] [你好，窗口 - LearnOpenGL CN \(learnopengl-cn.github.io\)](https://learnopengl-cn.github.io/)
- [4] [你好，三角形 - LearnOpenGL CN \(learnopengl-cn.github.io\)](https://learnopengl-cn.github.io/)
- [5] [着色器 - LearnOpenGL CN \(learnopengl-cn.github.io\)](https://learnopengl-cn.github.io/)
- [6] [坐标系统 - LearnOpenGL CN \(learnopengl-cn.github.io\)](https://learnopengl-cn.github.io/)