

# 浙江大学实验报告

专业：建筑学

姓名：汪川淇

学号：3180104481

日期：03/22/2020

地点：浙江大学紫金港校区

课程名称：\_\_\_\_计算机图形学\_\_\_\_ 指导老师：\_\_\_\_唐敏\_\_\_\_ 成绩：\_\_\_\_

实验名称：\_\_\_\_GLUT 程序设计\_\_\_\_ 实验类型：\_\_\_\_基础实验\_\_\_\_ 同组学生姓名：\_\_\_\_

## 一、实验目的和要求

学会配置 GLUT 开发库并使用 Visual Studio C++ 开发 OpenGL 程序。

## 二、实验内容和原理

在 Windows 系统中，配置 GLUT 库：解压并打开文件夹 glut.zip，取出 glut.h，glut32.lib，glut32.dll。之后有两种配置方式，一是将以上 3 个文件分别放在系统盘的相应目录下；二是针对具体项目（本次实验给定项目 Ex1）进行配置。

开发 OpenGL 程序：编译运行项目 Ex1，确认无误后修改代码生成以下图形：



## 三、主要仪器设备

Visual Studio C++2008

Glut 压缩包

Ex1 工程

#### 四、操作方法和实验步骤

使用 visual studio2019、提供的 glad/glfw 框架，参考 Hello\_OpenGL 内容进行试验。

实验步骤分成三个部分：

- 1、确定单个五角星绘制方法，及相对顶点坐标；
- 2、实现 Star.h 中对五角星的缩放与旋转操作，实现 opengl 坐标位置与像素位置换算；
- 3、在 render\_flag 中实现对 5 个五角星的定位及绘制；

具体步骤上：

- 1、注意到 gl 绘制命令中：

```
glDrawArrays(GL_TRIANGLES, 0, sizeof(_vertices) / sizeof(glm::vec2));
```

采用十个三角形分别绘制并拼装的方式得到三角星（后尝试过 GL\_TRIANGLE\_FAN）并通过数学关系计算出相对（vec2）Position 的相对位置。

- 2、通过 star 中 aspect\_of\_the\_window 传入窗口的长宽比，作为系数作用在三角形半径上。
- 3、在三角形顶点位置加入 rotation（角度制）影响系数。
- 4、根据五星红旗上五星的位置，绘制五角星。

具体代码如下：

Render\_flag 中：

```
_flag = new Flag;
float aspect_of_the_window = (float)_windowWidth / (float)_windowHeight;
//std::cout << aspect_of_the_window;

_stars[0]=new Star(glm::vec2(-0.38,0.25), 90, 0.12,aspect_of_the_window);
_stars[1] = new Star(glm::vec2(-0.25, 0.4), 126, 0.05, aspect_of_the_window);
_stars[2] = new Star(glm::vec2(-0.21,0.3), 144, 0.05, aspect_of_the_window);
_stars[3] = new Star(glm::vec2(-0.21,0.2), 90 , 0.05, aspect_of_the_window);
_stars[4] = new Star(glm::vec2(-0.25, 0.1), 126, 0.05, aspect_of_the_window);
```

Star.Cpp 中：

```
const float PI = acos(0) * 2;
_rotation /= 180;
_rotation *= PI;

float r = _radius / 2.618f;
glm::vec2 L0 = _position;
glm::vec2 L1 = _position+glm::vec2(_radius * cos(_rotation)/aspect, _radius * sin(_rotation));
glm::vec2 L2 = _position+glm::vec2(r * cos(_rotation + 0.2* PI) / aspect, r * sin(_rotation + 0.2 * PI));
glm::vec2 L3 = _position+glm::vec2(_radius * cos(_rotation+0.4*PI) / aspect, _radius * sin(_rotation+ 0.4 * PI));
glm::vec2 L4 = _position + glm::vec2(r * cos(_rotation + 0.6 * PI) / aspect, r * sin(_rotation + 0.6 * PI));
glm::vec2 L5 = _position + glm::vec2(_radius * cos(_rotation + 0.8 * PI) / aspect, _radius * sin(_rotation + 0.8 * PI));
glm::vec2 L6 = _position + glm::vec2(r * cos(_rotation + 1. * PI) / aspect, r * sin(_rotation + 1. * PI));
```

```

glm::vec2 L7 = _position + glm::vec2(_radius * cos(_rotation + 1.2 * PI) / aspect, _radius * sin(_rotation + 1.2 * PI));
glm::vec2 L8 = _position + glm::vec2(r * cos(_rotation + 1.4 * PI) / aspect, r * sin(_rotation + 1.4 * PI));
glm::vec2 L9 = _position + glm::vec2(_radius * cos(_rotation + 1.6 * PI) / aspect, _radius * sin(_rotation + 1.6 * PI));
glm::vec2 L10 = _position + glm::vec2(r * cos(_rotation + 1.8 * PI) / aspect, r * sin(_rotation + 1.8 * PI));
const glm::vec2 _vertices[30]{
    L0,L1,L2,
    L0,L2,L3,
    L0,L3,L4,
    L0,L4,L5,
    L0,L5,L6,
    L0,L6,L7,
    L0,L7,L8,
    L0,L8,L9,
    L0,L9,L10,
    L0,L10,L1
};

```

## 五、实验数据记录和处理

结果：见 **render\_flag\_wcq.exe**

出现过的问题：

1、绘制的三角形初始位置为平躺方向。

处理方式：提供初始 **rotation=90**

2、五角星高宽方向变形

处理方式：提供 **aspect\_of\_the\_window** 系数

## 六、讨论、心得

其他尝试：

1、依照 **csdn** 上，实现了一个固定管线版本 **glew** 框架的长方形绘制。

2、依照网络教程，实现了一个 **obj** 读写（教程上使用的是 **win32** 的窗口框架）

3、尝试使用旋转矩阵完成旋转操作。（实现了一个固定管线版本，可编程版本还没试）

```

glPushMatrix();
glClear(GL_COLOR_BUFFER_BIT);
glScalef(1.0f, 1.0f, 1.0f);
glRotatef(-30.0f, 0.0f, 0.0f, 1.0f);
glTranslatef(1.0f, -1.0f, 0.0f);

glBegin(GL_POLYGON);

glColor4ub(0, 0, 255, 255);
glVertex3f(0.0f, 1.0f, -20.0f);
glColor4ub(255, 0, 0, 255);
glVertex3f(0.0f, -1.0f, -20.0f);

```

```
glColor4ub(0, 0, 255, 255);  
glVertex3f(2.0f, -1.0f, -20.0f);  
glColor4ub(255, 0, 0, 255);  
glVertex3f(2.0f, 1.0f, -20.0f);
```

```
glEnd(); //end draw  
glPopMatrix();
```

**4、RayTracing-OneWeekend** 进展到第七个章节，还在理解 **Diffuse Material** 的实现。

心得：

- 1、加深了 **OpenGL** 是一个状态机的理解。
- 2、增强了对于着色器语言的理解 **vs / fs**，但现在使用图形接口直接调用函数实现的，之后希望能进一步了解一下这个数据具体式怎么和显卡交互的（比如显存分配，以及 **drawcall**，以及着色语言里的 **Pass()**）。
- 3、理解了 **vao**，**vbo** 的作用。