

浙江大学实验报告

专业： 计算机科学与技术

姓名： 余启航

学号： 3190103324

日期： 2021/10/20

地点： 玉泉曹光彪西 503

课程名称： 计算机图形学 指导老师： 童若锋 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：



图 1 五星红旗范例

三、主要仪器设备

Visual Studio C++2017

Glut 压缩包

Ex1 工程

四、操作方法和实验步骤

1. 观察已有条件

观察已经给出的代码，发现已经给出了框架，只需要自己实现绘图的函数。从给出的四边形的绘制代码中可以看出，在绘制图形之前，需要给定其颜色，接着调用 `glBegin()` 表明开始绘制，需要给一个参数，说明是什么图形。接着给出每个点的坐标，最后调用 `glEnd()` 表明绘制结束。

2. 分解绘图过程

需要绘制五个五角星，那么就可以先实现五角星的绘制，由于每个五角星的大小、旋转角度和中心位

置都不同，所以这三个是需要传入绘图函数的参数。

1. `void drawstar(GLfloat center[3], GLfloat rate, GLfloat angle);`

观察五角星，可以得知这是一共凹多边形，可以将其拆成凸多边形进行绘制，我选择将其拆成三角形，通过绘制如下图的三个有重叠部分的三角形可以得到五角星



图 2 五角星分解三角形

我们可以看出，三个三角形的形状是完全相同的，且三角形的绘制只需给出三个顶点坐标即可，因此，三角形的绘制只需要传入顶点坐标。

1. `void drawtriangles(GLfloat a[3], GLfloat b[3], GLfloat c[3]);`

3. 实现绘图函数

(1) 实现三角形绘制

首先设置颜色，这里由于是绘制五角星，所以选用国旗的黄色。接着需要表明开始绘制，并说明绘制的是三角形。然后给定三个坐标，最后说明绘制结束。代码如下：

```
1. void drawtriangles(GLfloat a[3], GLfloat b[3], GLfloat c[3])
2. {
3.     glColor3f(0.98, 0.957, 0.031); //黄色的参数
4.     glBegin(GL_TRIANGLES);          //说明开始绘制，且绘制的是三角形
5.     glVertex3f(a[0], a[1], a[2]);
6.     glVertex3f(b[0], b[1], b[2]);
7.     glVertex3f(c[0], c[1], c[2]);
8.     glEnd();                        //结束绘制
9. }
```

(2) 实现五角星绘制

五角星需要绘制三个三角形，先假设没有旋转，我们选择图二中的第一个三角形为初始值根据比例给出初始坐标，之后的绘制都是基于此坐标进行的变换。

1. `GLfloat v[3][3] = { 0.2, 0.067, 1.0, -0.2, 0.067, 1.0, 0.0, -0.067, 1.0 };`

下面考虑五角星的大小和旋转以及中心位置，分别调用函数进行矩阵变换

```
1. glTranslatef(center[0], center[1], center[2]);
2. glRotatef(angle, 0.0, 0.0, 1.0);
3. glScalef(rate, rate, 1);
```

`glTranslatef()`是将坐标进行平移，传入的参数为基于三个坐标系的移动的量；`glRotatef()`函数是对坐标进行旋转，传入的第一个参数为右手系内逆时针旋转角度，后面三个参数为 x, y, z 值的向量为旋转轴；`glScalef()`函数是对坐标进行缩放，传入三个参数为 x, y, z 方向缩放比。变换之后调用三角形绘制函数绘制三角形。绘制结束调用 `glLoadIdentity()`，将变换矩阵单位化，方便下次变换。同理绘制三个三角形即完成五角星绘制。

(3) 补充绘制函数 display()

确定五个五角星的中心位置和各自的旋转角度以及和最大五角星相比的缩放比，调用绘制五角星的函数分别进行绘制。

```
1.  GLfloat angle[4] = { 51.33,71.5,99.5,120 }; //小五角星旋转角度
2.  GLfloat GCenter[3] = { -0.667, 0.333, 0.0 }; //大五角星中心
3.  //小五角星中心
4.  GLfloat GCenter1[3] = { -0.333, 0.067, 0.0 };
5.  GLfloat GCenter2[3] = { -0.2, 0.2, 0.0 };
6.  GLfloat GCenter3[3] = { -0.2, 0.4, 0.0 };
7.  GLfloat GCenter4[3] = { -0.333, 0.533, 0.0 };
8.  GLfloat rate = 1.0; //缩放比
9.  drawstar(GCenter, rate, 0);
10. rate = 0.333;
11. drawstar(GCenter1, rate, angle[0]);
12. drawstar(GCenter2, rate, angle[1]);
13. drawstar(GCenter3, rate, angle[2]);
14. drawstar(GCenter4, rate, angle[3]);
```

(4) 避免窗口比例变化对显示的影响

窗口的大小变化，由于绘图根据的窗口大小，会发生拉伸，此时可以根据注册回调函数进行控制：

```
1.  glutReshapeFunc(ReSize);
    它在窗口变化时调用的括号内函数，括号内函数如下：
2.  void ReSize(GLsizei w, GLsizei h)
3.  {
4.      GLfloat Ratio;
5.      // 防止被 0 所除
6.      if (0 == h) {
7.          h = 1;
8.      }
9.      glViewport(0, 0, w, h); // 设置视口为窗口的大小
10.     glMatrixMode(GL_PROJECTION); // 选择投影矩阵，并重置坐标系
11.     glLoadIdentity();
12.     Ratio = (GLfloat)w / (GLfloat)h; // 计算窗口的纵横比（像素比）
13.     GLfloat rt = 1.0; // 定义裁剪区域（根据窗口的纵横比，并使用正投影）
14.     if (w <= h) { // 宽 < 高
15.         glOrtho(-rt, rt, -rt / Ratio, rt / Ratio, 1.0, -1.0);
16.     }
17.     else { // 宽 > 高
18.         glOrtho(-rt * Ratio, rt * Ratio, -rt, rt, 1.0, -1.0);
19.     }
20.     // 选择模型视图矩阵，并重置坐标系
21.     glMatrixMode(GL_MODELVIEW);
22.     glLoadIdentity();
23. }
```

4. 编译运行

对项目进行设置，主要是如下两项，使得文件可以编译

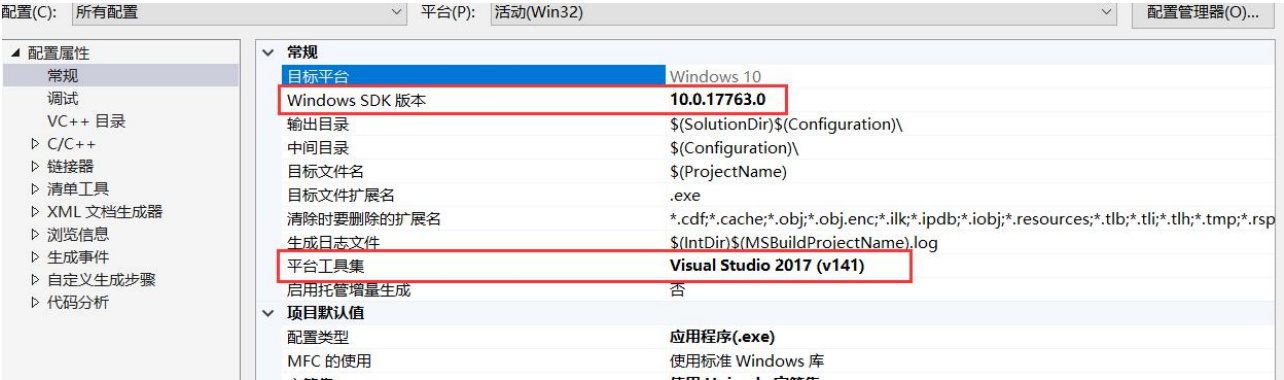


图 3 编译配置

每次运行除了出现五星红旗窗口，还会出现无用的终端窗口，可以根据如下设置去除终端窗口



图 4 去除窗口设置 1

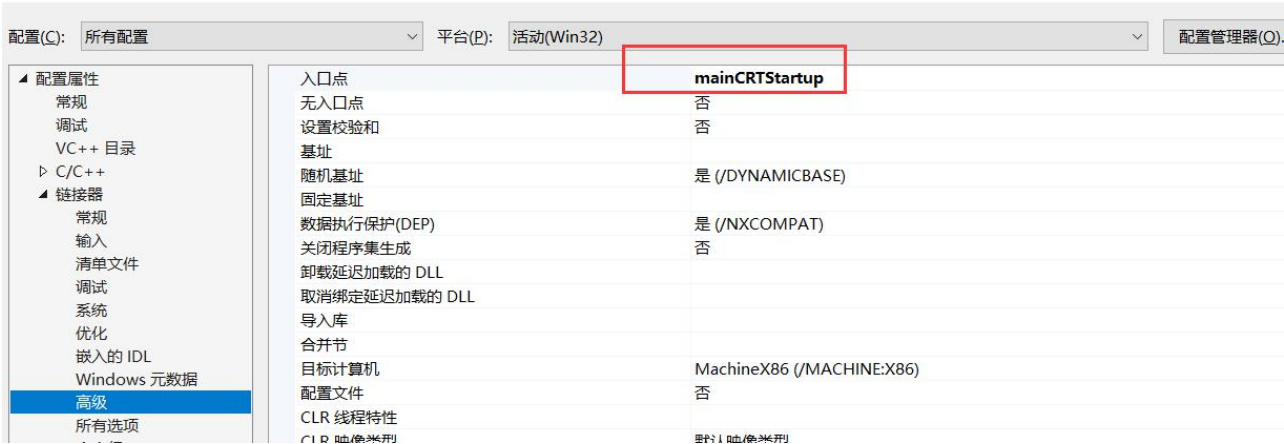


图 5 去除窗口设置 2

五、实验数据记录和处理

本实验无数据需要记录处理

六、实验结果与分析

1. 实验结果

项目运行得到了绘制完成的五星红旗，改变窗口大小不会影响红旗比例



图 6 五星红旗 1



图 7 五星红旗 2



图 8 五星红旗 3

2. 分析

程序能绘制出准确的五星红图案并且稳定显示，不会随窗口的变化而发生错误的 拉伸等变换。

七、讨论、心得

1. 实验过程中体会最深的还是对于 OpenGL 的不了解，不知道有哪些功能以及对于一些函数的功能不清晰。导致实验刚开始优点手足无措。

2. 本次实验的思路来自上课时候讲到的问题的化简分解，将五星红旗变成五角星的旋转缩放和平移，将五角星变成三角形的组合。
3. 在刚开始实验时，我把窗口设置成长宽不同的尺寸，然后测试三角形基于原点的旋转，发现三角形会由于长宽尺寸的不同但是比例不变而发生变形，于是我将窗口设置为长宽相同的尺寸，在绘制时只取其中一块。到实验结束后，发现窗口会变化导致红旗依然不成比例，于是查询得到了视口和投影的相关函数，问题得到了解决。
4. 一开始没理解到变换矩阵实在栈里的的意义，在进行变换时候，没有注意变换函数的顺序，导致显示奇形怪状，后来讨论得知顺序会产生影响，才正确选择了顺序。
5. 本次实验作为第一个实验，难度不大，极大激发了我的兴趣，非常喜欢。