作业一 绘制姓名首字母

18340012 陈晨

1 实验题目

使用三角面片作为图元绘制姓名首字母，可使用的 OpenGL 图元类型包括：GL_TRIANGLES、GL_TRIANGLE_STRIP 以及 GL_TRIANGLE_FAN。例如：黄小明，名字首字母为 HXM，因此需要绘制 HXM 三个字母，下图为 H 的 demo。在书面报告中，需要明确说明每个字母所需的语句数（glBegin, glEnd, glVertex）的数量，循环调用的需要重复计算（即循环体内 glVertex 等需要乘循环次数）。请尽量精简你的实现！

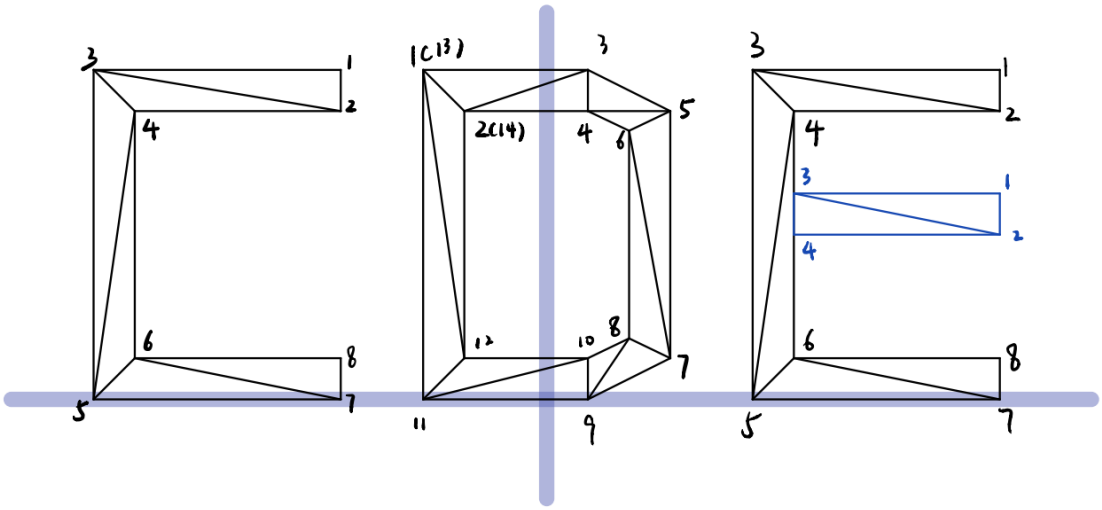
注：为了公平性，每人需绘制三个不同的字母，即遵循以下三条规则:

- 名字大于三个字的只需绘制前三个字的首字母。（若此时有相同字母，见规则 3）
- 名字少于三个字的后面多加一个字母，此字母为第二个字母的后一个字母。如 HX，则需绘制 HXY。（若此时有相同字母，见规则 3）
- 若有相同字母，则第二第三个相同字母依次往后推，如 AAA=>ABB=>ABC， ACC=>ACD，AAB=>ABB=>ABC

2 实验过程和核心代码

根据规则，我需要绘制CDE三个字母。

由于只能用三角形进行绘制，并且字母的图形其实是由相邻的三角形组成的。考虑采用GL_TRIANGLE_STRIP节省glVertex的数量。根据STRIP的画图规则，画出草图如下，其中数字为用glVertex声明顶点坐标的顺序。粗蓝线为坐标轴。



设边长为10，则根据上图可以推出每个点的坐标。完成scene_1函数如下：

```
void MyGLWidget::scene_1()
{

    glClear(GL_COLOR_BUFFER_BIT);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(0.0f, width(), 0.0f, height(), -1000.0f, 1000.0f);

    glMatrixMode(GL_MODELVIEW);
```

```
    glLoadIdentity();
    glTranslatef(0.5 * width(), 0.5 * height(), 0.0f);

    //your implementation here, maybe you should write several glBegin
    glPushMatrix();
    glColor3f(0.839f, 0.153f, 0.157f);
    //your implementation
    //C: 1 + 8 + 1 = 10
    glBegin(GL_TRIANGLE_STRIP);
    glVertex2f(-50.0f, 80.0f);
    glVertex2f(-50.0f, 70.0f);
    glVertex2f(-110.0f, 80.0f);
    glVertex2f(-100.0f, 70.0f);
    glVertex2f(-110.0f, 0.0f);
    glVertex2f(-100.0f, 10.0f);
    glVertex2f(-50.0f, 0.0f);
    glVertex2f(-50.0f, 10.0f);
    glEnd();

    //D: 1 + 14 + 1 = 16
    glBegin(GL_TRIANGLE_STRIP);
    glVertex2f(-30.0f, 80.0f);
    glVertex2f(-20.0f, 70.0f);
    glVertex2f(10.0f, 80.0f);
    glVertex2f(10.0f, 70.0f);
    glVertex2f(30.0f, 70.0f);
    glVertex2f(20.0f, 65.0f);
    glVertex2f(30.0f, 10.0f);
    glVertex2f(20.0f, 15.0f);
    glVertex2f(10.0f, 0.0f);
    glVertex2f(10.0f, 10.0f);
    glVertex2f(-30.0f, 0.0f);
    glVertex2f(-20.0f, 10.0f);
    glVertex2f(-30.0f, 80.0f);
    glVertex2f(-20.0f, 70.0f);
    glEnd();

    //E: (1 + 8 + 1) + (1 + 4 + 1) = 16
    glBegin(GL_TRIANGLE_STRIP);
    glVertex2f(110.0f, 80.0f);
    glVertex2f(110.0f, 70.0f);
    glVertex2f(50.0f, 80.0f);
    glVertex2f(60.0f, 70.0f);
    glVertex2f(50.0f, 0.0f);
    glVertex2f(60.0f, 10.0f);
    glVertex2f(110.0f, 0.0f);
    glVertex2f(110.0f, 10.0f);
    glEnd();
    glBegin(GL_TRIANGLE_STRIP);
    glVertex2f(110.0f, 50.0f);
    glVertex2f(110.0f, 40.0f);
    glVertex2f(60.0f, 50.0f);
    glVertex2f(60.0f, 40.0f);
    glEnd();

    glPopMatrix();
}
```

由上述代码可知，仅考虑glBegin，glEnd，glVertex，C的语句数为10，D的语句数为16，E的语句数为16，**共计42。**

除此之外，需要修改默认显示为scene_1：

```
MyGLWidget::MyGLWidget(QWidget* parent)
    :QOpenGLWidget(parent),
    scene_id(1)
{
}
```

3 实验结果

结果如上图所示，画面显示出了CDE三个字母。

4 实验感想

本实验主要的问题在于如何精简自己的实现。对于字母图形，由于三角形需要相邻拼接，采用GL_TRIANGLES显然会多出重复边的顶点声明。采用GL_TRIANGLE_STRIP则可以节省一部分重复的顶点声明。

对于C，glVertex数等于图形的顶点数，且只需要声明一次glBegin和glEnd，是最好的实现方式。

对于D，由于图形是一个闭环，会有两个重复声明的顶点。考虑用红色的大多边形叠上白色的小多边形来实现，虽然glVertex数等于图形的顶点数，但是会多出一组glBegin和glEnd，故数量没有发生改变，最终没有采用。

对于E，图形本身的顶点数为12，但是在一个STRIP之内，无法通过排列12个点的声明顺序画出E的图形。可以解决的方法是多引入两个点或者中间的横单独画，最后都是多出2条语句，最后采用了后者。