

# HW1 绘制姓名首字母

---

18364066 Yanzuo Lu

September 28, 2020

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Character "L"</b>	<b>2</b>
<b>3</b>	<b>Character "Y"</b>	<b>3</b>
<b>4</b>	<b>Character "Z"</b>	<b>4</b>
<b>5</b>	<b>Results</b>	<b>5</b>

## 1 Introduction

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。

## 2 Character "L"

第一个需要绘制的字母是"L"，该字母结构较为简单，我将其分解为 5 个相邻的三角形以后，标定了 7 个节点，利用 GL\_TRIANGLE\_STRIP 图元就可以轻易地绘制出来，分解的示意图如下所示。

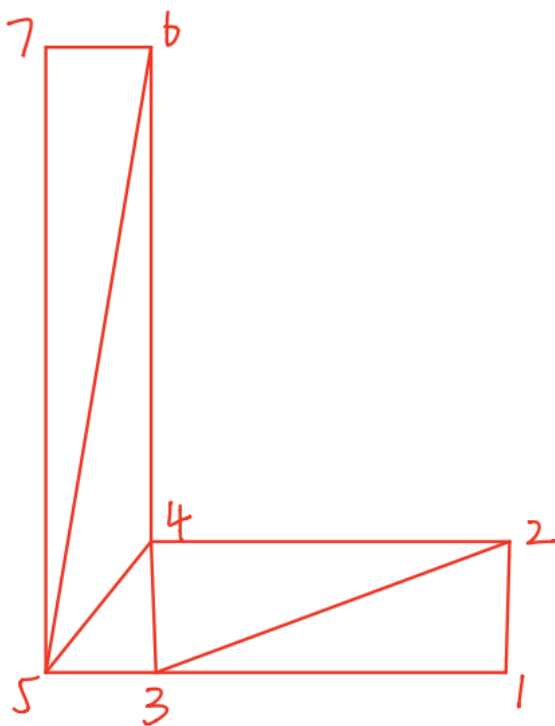


Figure 1: Character "L"

此外，我将字母"L" 放置在了距离坐标原点靠左的位置，利用了 glTranslatef 函数事先进行平移，然后调用 1 次 glBegin()+glEnd() 和 7 次 glVertex2f 即可绘制成功，核心代码如下所示。

```
// character "L"
glTranslatef(-150.0f, 0.0f, 0.0f);
glBegin(GL_TRIANGLE_STRIP);
glVertex2f(100.0f, 0.0f);
```

```

glVertex2f(100.0f, 20.0f);
glVertex2f(20.0f, 0.0f);
glVertex2f(20.0f, 20.0f);
glVertex2f(0.0f, 0.0f);
glVertex2f(20.0f, 150.0f);
glVertex2f(0.0f, 150.0f);
glEnd();

```

### 3 Character "Y"

第二个需要绘制的字母是"Y", 该字母结构相较于字母"L" 更加复杂, 因为涉及到斜向的笔画, 我同样对字母进行了若干个三角形的分解, 更具体地来说应该是三个矩形和一个三角形, 矩形可分作两个三角形来绘制。相同地可以使用 GL\_TRIANGLE\_STRIP 而不是 GL\_TRIANGLES 来精简代码, 分解的示意图如下所示, 其中括号内的数字表示另一个 glBegin()+glEnd() 内部的 glVertex2f 顺序。

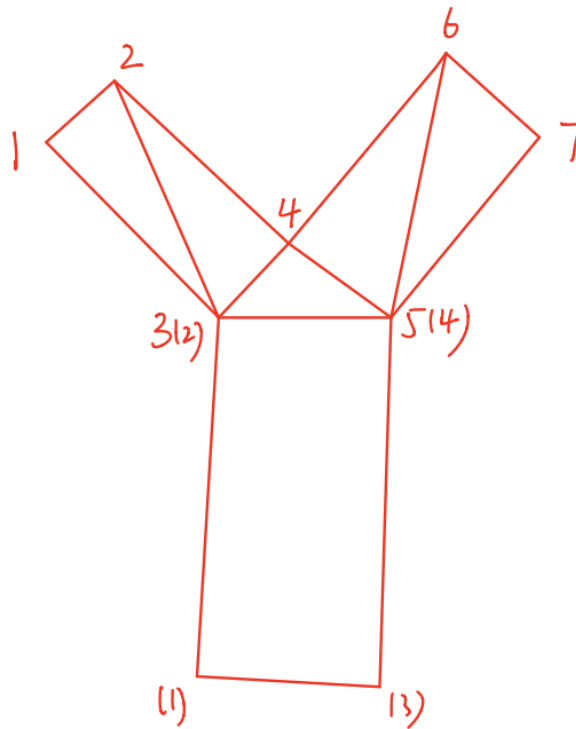


Figure 2: Character "Y"

字母"Y" 则放置在坐标系的中央, 因为事先在绘制字母"L" 的时候已经向左平移了一段距离, 在 glBegin() 之前需要平移回到合适的位置, 并且由于矩形较多, 无法利用 GL\_TRIANGLE\_STRIP

的同时只通过一次 glBegin()+glEnd() 完成绘制，所以总共调用了 2 次 glBegin()+glEnd() 和 11 次 glVertex2f 来完成字母”Y”，核心代码如下所示。

```
// character "Y"
glTranslatef(100.0f, 0.0f, 0.0f);
glBegin(GL_TRIANGLE_STRIP);
glVertex2f(0.0f, 130.0f);
glVertex2f(20.0f, 150.0f);
glVertex2f(35.85f, 100.0f);
glVertex2f(50.0f, 120.0f);
glVertex2f(64.14f, 100.0f);
glVertex2f(80.0f, 150.0f);
glVertex2f(100.0f, 130.0f);
glEnd();

glBegin(GL_TRIANGLE_STRIP);
glVertex2f(35.85f, 0.0f);
glVertex2f(35.85f, 100.0f);
glVertex2f(64.14f, 0.0f);
glVertex2f(64.14f, 100.0f);
glEnd();
```

## 4 Character ”Z”

最后一个需要绘制的字母是”Z”，该字母是三个字母中最为复杂的一个，因为涉及到横向和斜向笔画的连接，同理我在草稿中对其进行若干个三角形的分解，该分解可以只使用一次 glBegin()+glEnd() 完成绘制，分解示意图如下所示。

实现方面也和前两个字母没有太大的差异，主要是需要标定好坐标位置，还有一点就是事先需使用 glTranslatef 向右平移一段距离使得三个字母能够并排排列，总共调用了 1 次 glBegin()+glEnd() 和 10 次 glVertex2f 来完成字母”Z”，核心代码如下所示。

```
// character "Z"
glTranslatef(100.0f, 0.0f, 0.0f);
glBegin(GL_TRIANGLE_STRIP);
glVertex2f(100.0f, 0.0f);
```

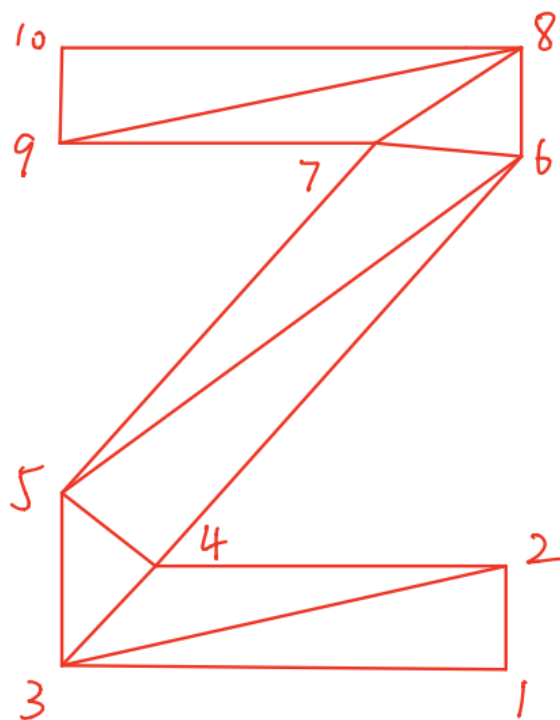


Figure 3: Character "Z"

```
glVertex2f(100.0f, 20.0f);
glVertex2f(0.0f, 0.0f);
glVertex2f(20.0f, 20.0f);
glVertex2f(0.0f, 30.0f);
glVertex2f(100.0f, 130.0f);
glVertex2f(80.0f, 130.0f);
glVertex2f(100.0f, 150.0f);
glVertex2f(0.0f, 130.0f);
glVertex2f(0.0f, 150.0f);
glEnd();
```

## 5 Results

将上述三个字母的代码组合到一起并放入 scene\_1 函数中，最后的演示效果如下图所示。



Figure 4: Qt Display