



中山大學
SUN YAT-SEN UNIVERSITY

《计算机图形学作业》 实验报告

(作业二)

学 院 名 称 : 数据科学与计算机学院

学 生 姓 名 : 姚雪辉

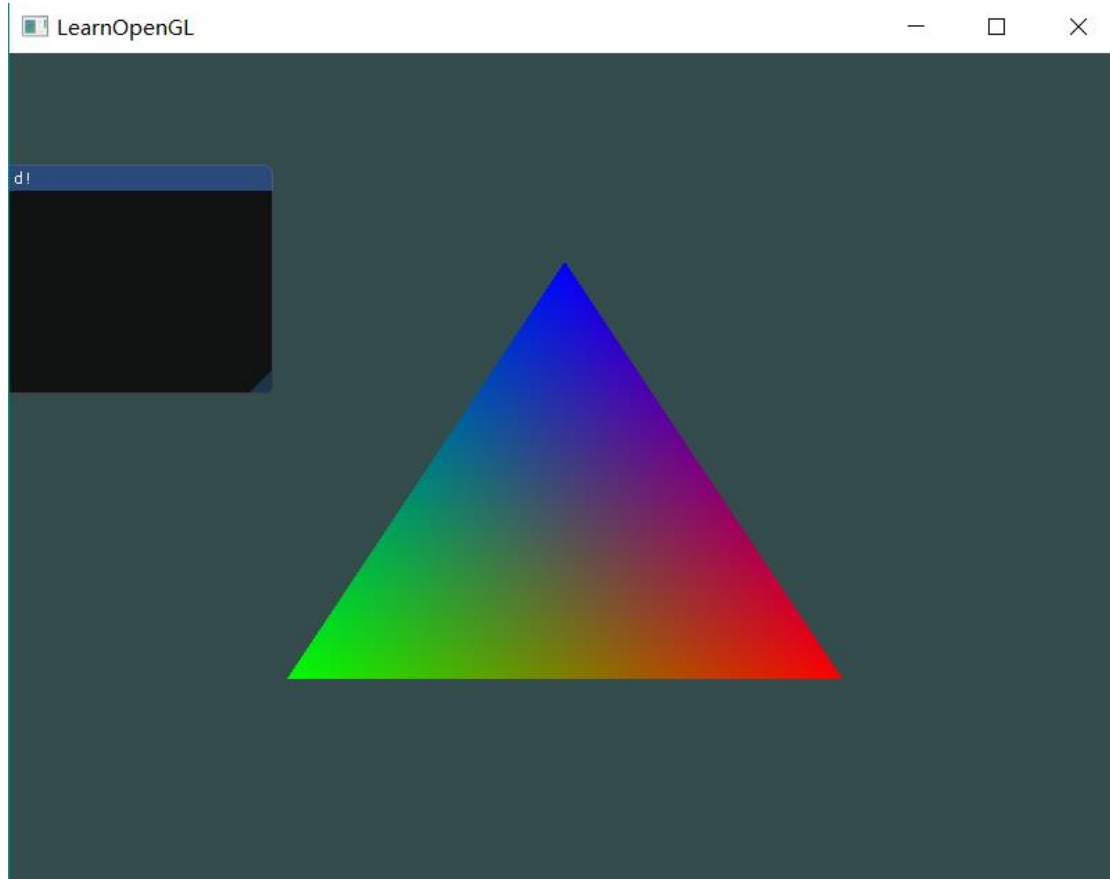
学 号 : 15355119

专业 (班级) : 16 软件工程四 (7) 班

时 间 : 2019 年 3 月 11 日

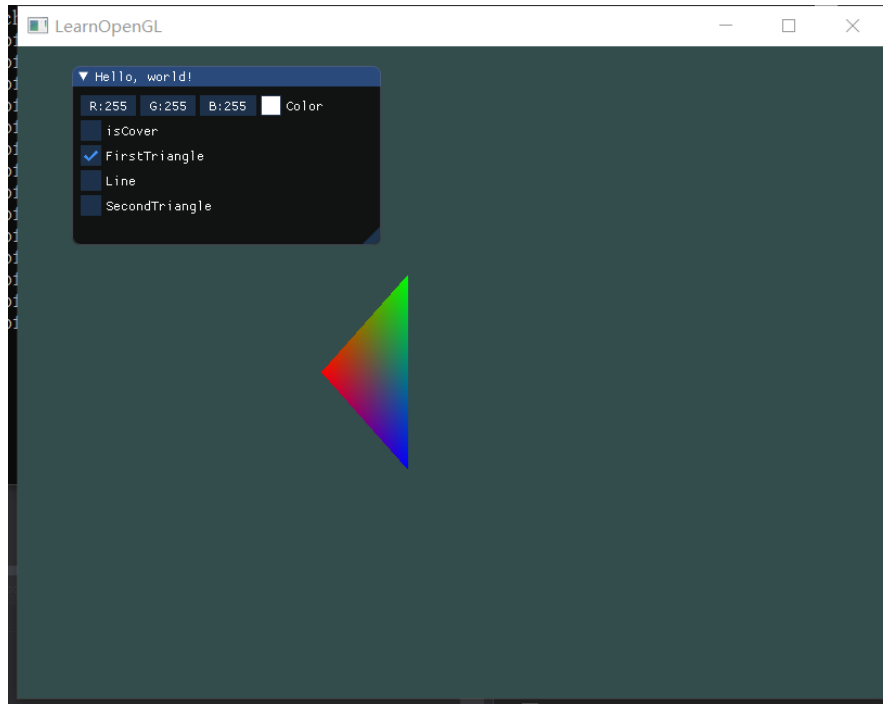
第一题：

当渲染一个三角形时，光栅化阶段通常会造成比原指定顶点更多的片段。光栅会根据每个片段在三角形形状上所处相对位置决定这些片段的位置。基于这些位置，它会插值所有片段着色器的输入变量。

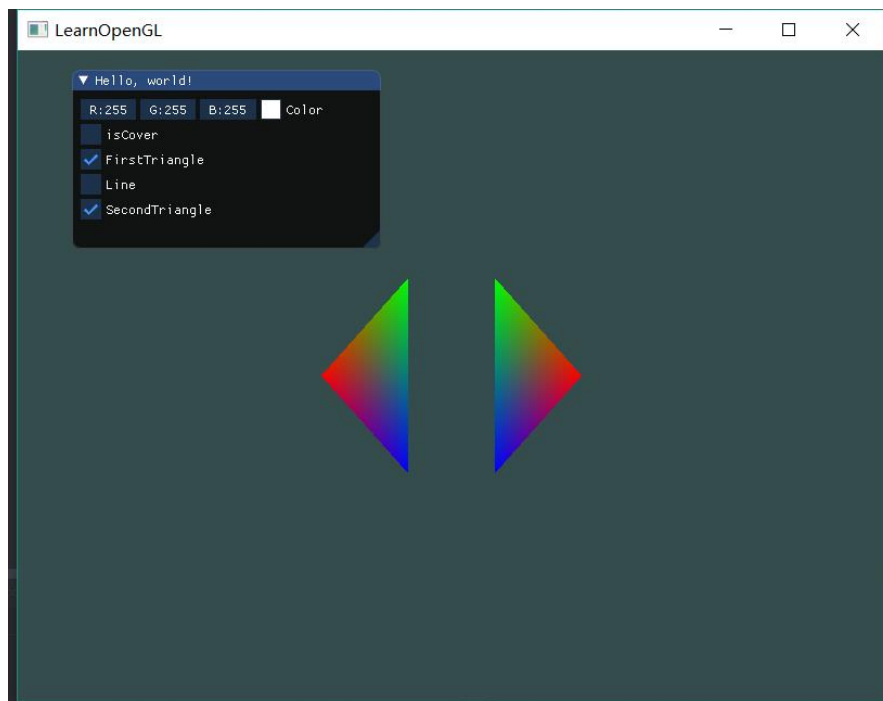


第二题

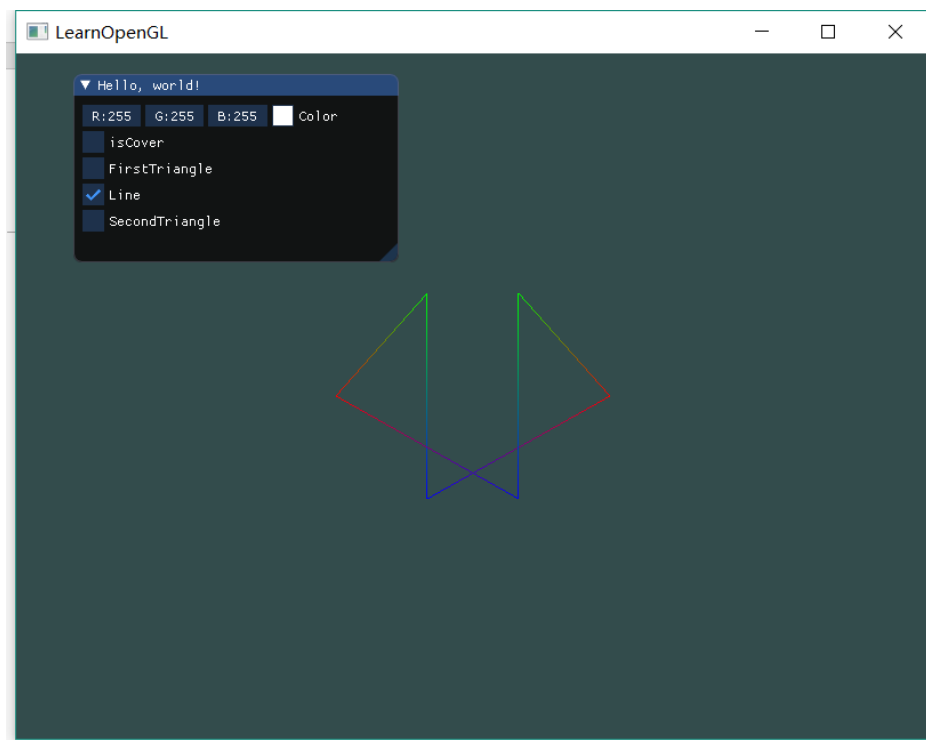
默认状态是只显示一个三角形，在 isCover 为 false 的情况下，颜色的构成是只有一个颜色，但是颜色分给三个顶点，左边的点是颜色 x 分量，其他用 0 填充如图是(225,0,0)，类似的上面的点是(0,255,0),下面的点是(0,0,255);



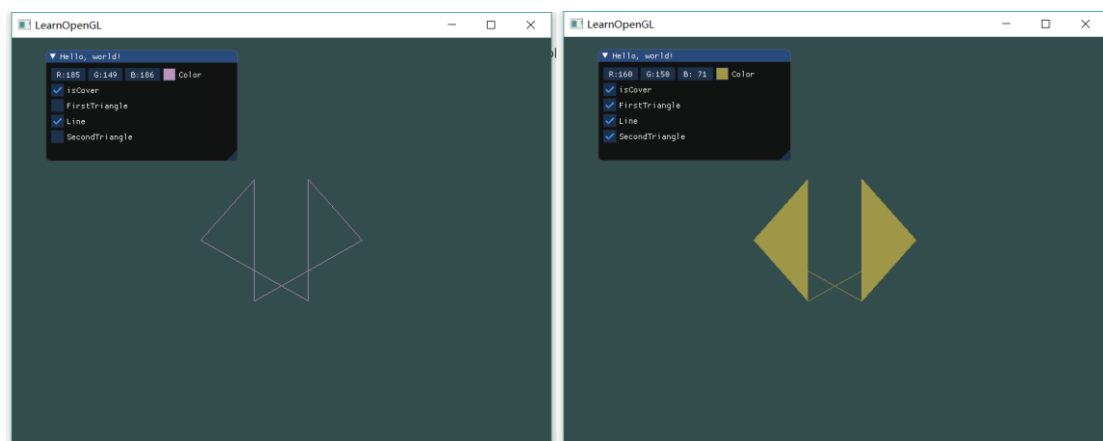
当点击 SecondTriangle 时会以中间为对称轴镜像出一个三角形



单独画出一个线段，用的是闭合属性 `GL_LINE_LOOP`



这时候点击 `isCover`,则所有的颜色都会被替换成原本应该显示的颜色，和 `imgui` 的 `Color` 属性匹配



实现思路

第一题参考 [github](#) 的代码,

- 生成顶点着色器
- 将着色器源代码赋值给顶点着色器
- 编译着色器源代码
- 判断是否编译正确
- 生成片段着色器
- 将着色器源代码赋值给片段着色器

- 编译着色器源代码
- 判断是否编译正确
- 创建着色器程序
- 链接着色器
- 删除顶点着色器
- 删除片段着色器

■ 初始化 ImGui

在循环里面

- 建立顶点信息
- 利用 ID 创建 VAO，VBO 和 EBO
- 先绑定 VAO
- 再绑定 VBO
- 把顶点信息复制到缓存区中
- 绑定 EBO
- 将索引信息复制到索引缓冲中
- 链接顶点属性
- 启动顶点属性

■ 绑定 ImGui 的参数

画图的时候

- 使用着色器程序
- 绑定 VAO
- 画图

画完图

- 释放 VAO，VBO，EBO

退出循环

- 释放 ImGui

主要 function/algorithm 的解释

第二题只是加了 isCover 的功能，因为一开始建立顶点数据的时候采用如下方式，

```
float vertices[] = {  
    -0.3f, 0.0f, 0.0f,  vectexColor.x, Y, Z,  
    -0.1f, 0.3f, 0.0f,  X, vectexColor.y, Z,  
    -0.1f, -0.3f, 0.0f,  X, Y, vectexColor.z,  
    0.3f, 0.0f, 0.0f,  vectexColor.x, Y, Z,  
    0.1f, 0.3f, 0.0f,  X, vectexColor.y, Z,  
    0.1f, -0.3f, 0.0f,  X, Y, vectexColor.z  
};
```

因为既想显示第一题，又想显示第二题，所以部分顶点拆开了，再用 ImGui 控制 isCover 的属性，然后更改属性即可。

```
if (isCover) {  
    X = vectexColor.x;  
    Y = vectexColor.y;  
    Z = vectexColor.z;  
}  
else {  
    X = 0;  
    Y = 0;  
    Z = 0;  
}
```

线的实现是 `glDrawArrays(GL_LINE_LOOP, 0, 6);`

从第一个点开始，画完六个点，并且连成一个封闭图形。

另外网上有很多参考代码，但是都有一个问题，会爆内存，因为在循环内部，不断的生成 VAO, VBO 和 EBO，然后又不释放，所以虽然可以运行几秒，时间一长就会卡住不动。所以特地上面流程“画完图”和“退出循环”之间加上“释放掉 VAO, VBO, EBO”。

具体的 opengl 提供的函数就不解释了，官网上都有详细的介绍。