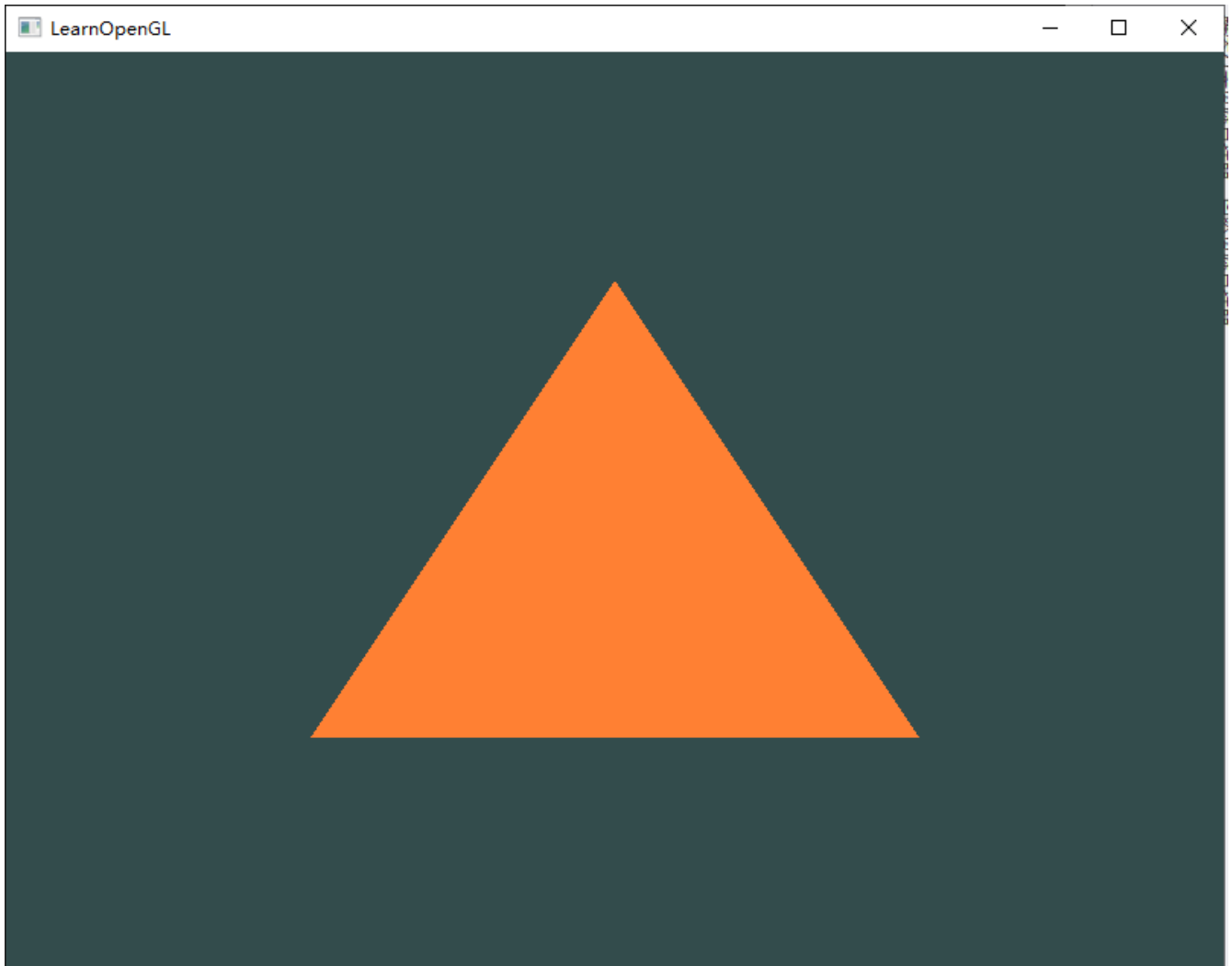


Homework 2

16340282 袁之浩

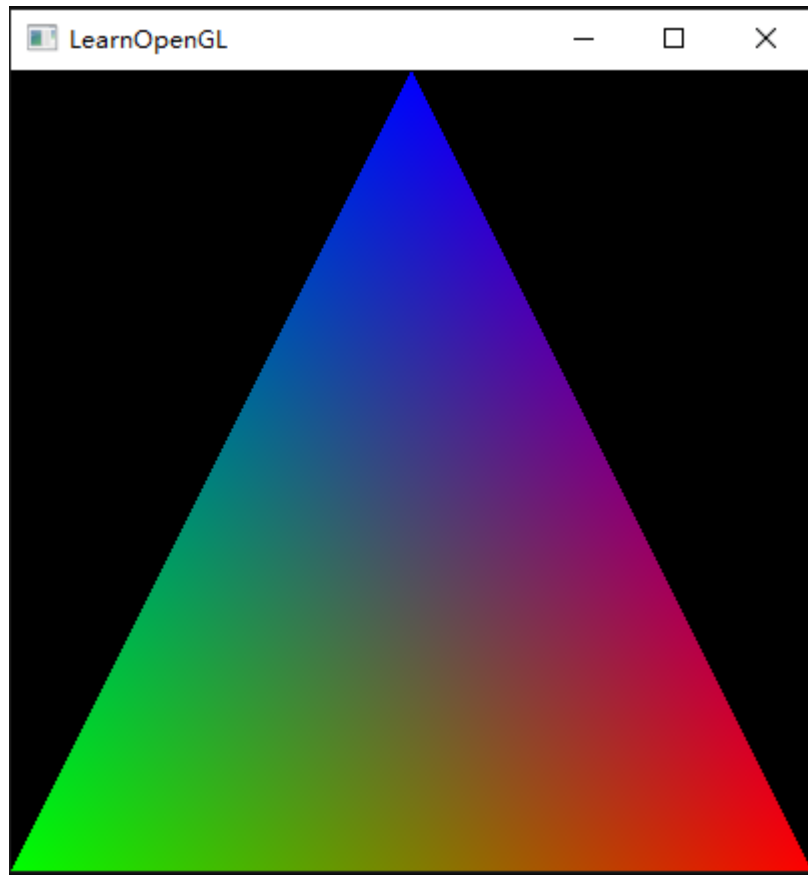
效果展示

1. 使用OpenGL(3.3及以上)+GLFW画一个简单的三角形。

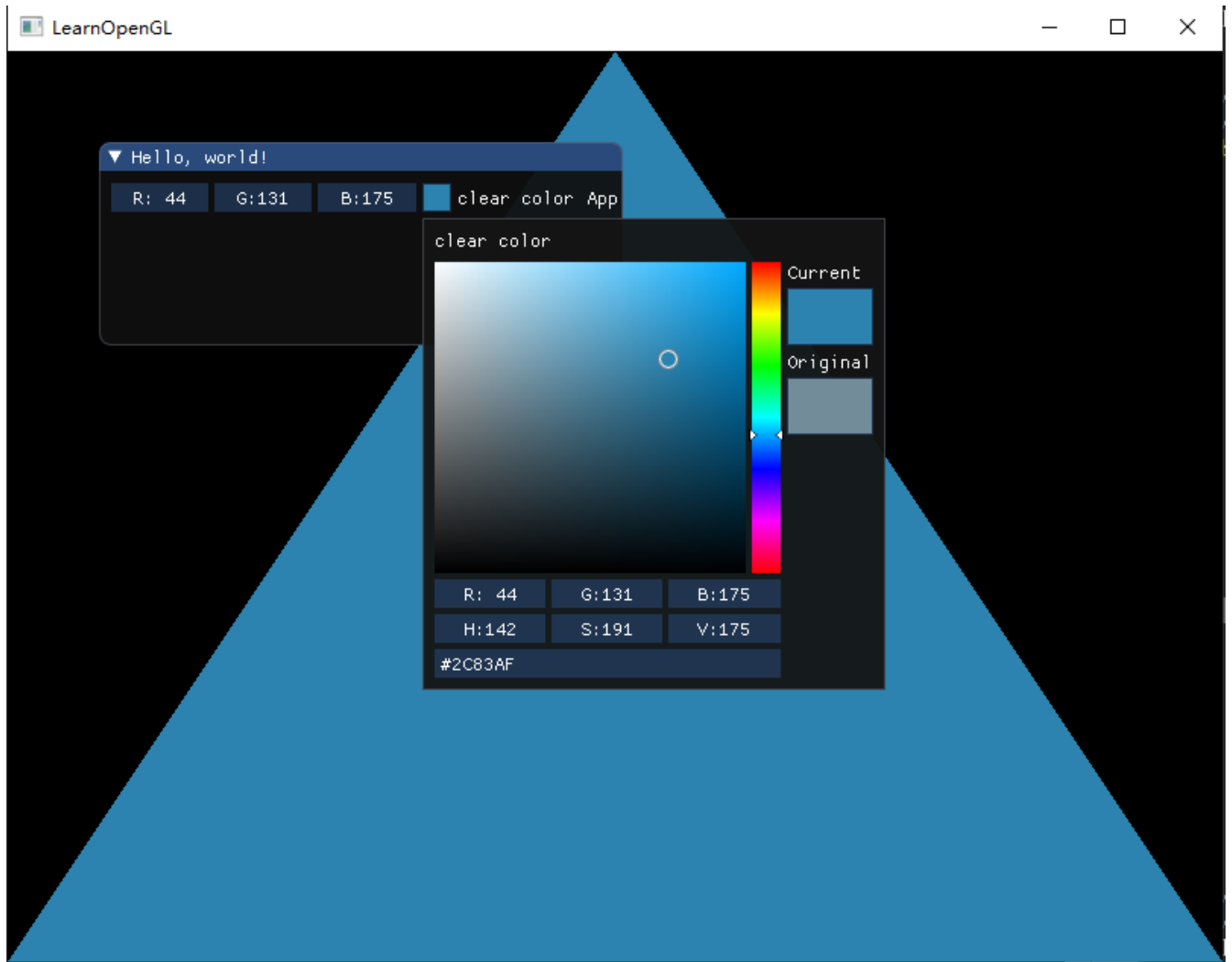


2. 对三角形的三个顶点分别改为红绿蓝，并解释为什么会出现这样的结果。

我们只提供了三种颜色，但是却出现了一个大调色版，因为在片段着色器中会进行片段插值(Fragment Interpolation)。当渲染一个三角形时，光栅化(Rasterization)阶段通常会造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(Interpolate)所有片段着色器的输入变量。片段插值会被应用到片段着色器的所有输入属性上。

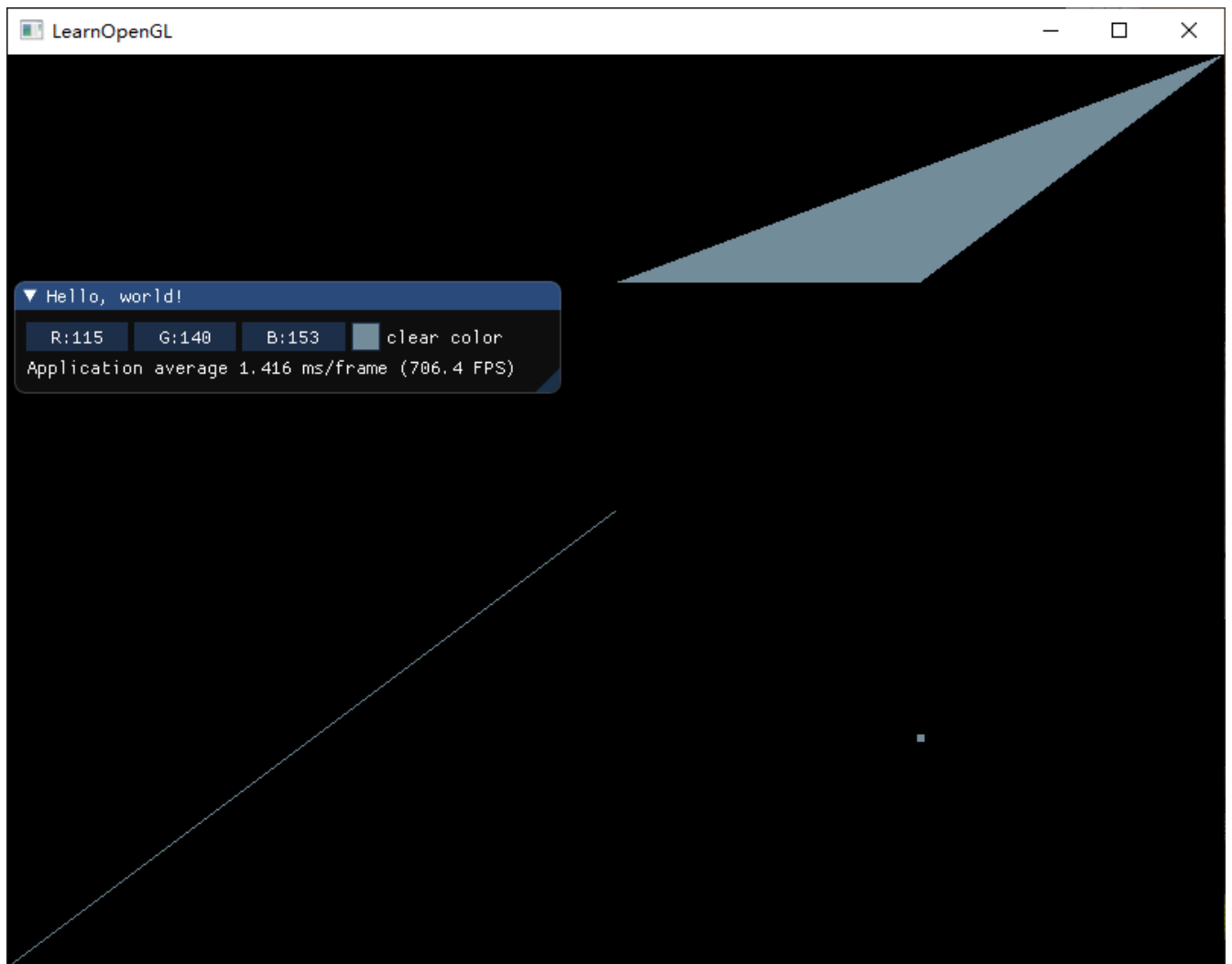


3. 给上述工作添加一个GUI，里面有一个菜单栏，使得可以选择并改变三角形的颜色。

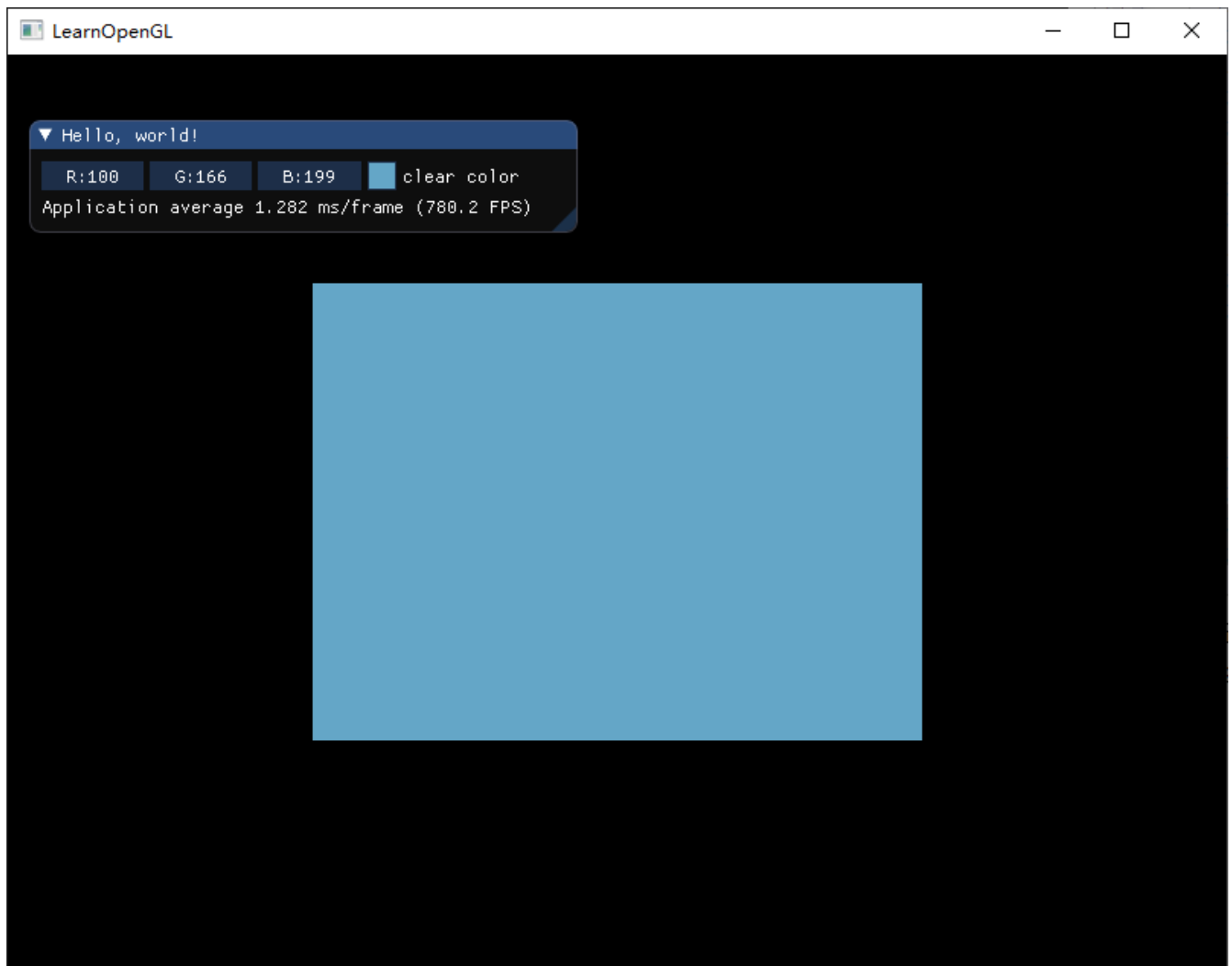


Bonus

1. 绘制其他的图元，除了三角形，还有点、线等。



2. 使用EBO(Element Buffer Object)绘制多个三角形。



实现思路

这次实验是我们第一次接触openGL编程，主要了解了openGL编程的框架，基本概念和基本元素的绘制。

绘制的基本流程为：

1. glfw初始化
2. glfw窗口创建
3. glad初始化
4. imgui初始化
5. 创建顶点数组，VBO
6. 渲染循环，创建GUI，绘制图形
7. 删除资源，结束程序

学习到的基本概念：

- 顶点缓冲对象(Vertex Buffer Objects, VBO)

它会在GPU内存（通常被称为显存）中储存大量顶点。使用这些缓冲对象的好处是我们可以一次性的发送一大批数据到显卡上，而不是每个顶点发送一次。

- 顶点数组对象(Vertex Array Object, VAO)

可以包含多个VBO，使在不同顶点数据和属性配置之间切换变得非常简单，只需要绑定不同的VAO就行了。刚刚设置的所有状态都将存储在VAO中。

- 索引缓冲对象(Element Buffer Object, EBO)

通过储存索引来避免重复点的绘制。

- 顶点着色器(Vertex Shader)

处理顶点数据。

- 片段着色器(Fragment Shader)

计算像素最后的颜色输出。