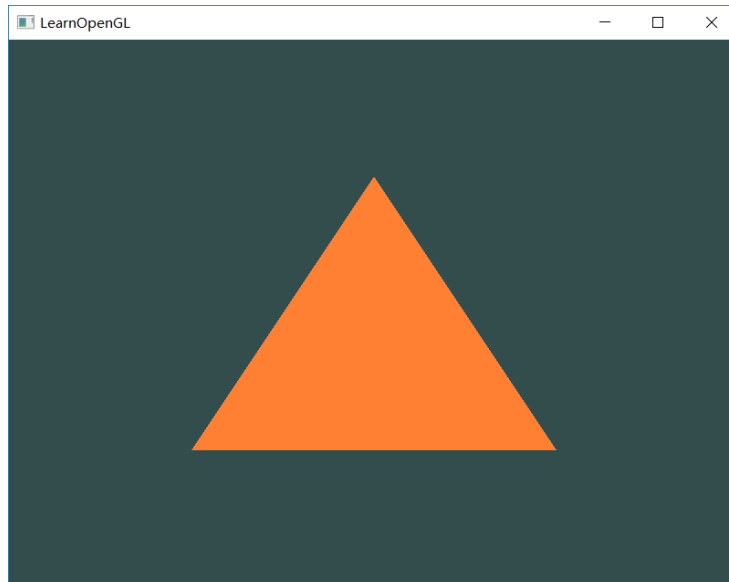


1. 使用 OpenGL(3.3 及以上)+GLFW 或 freeglut 画一个简单的三角形。

运行结果：

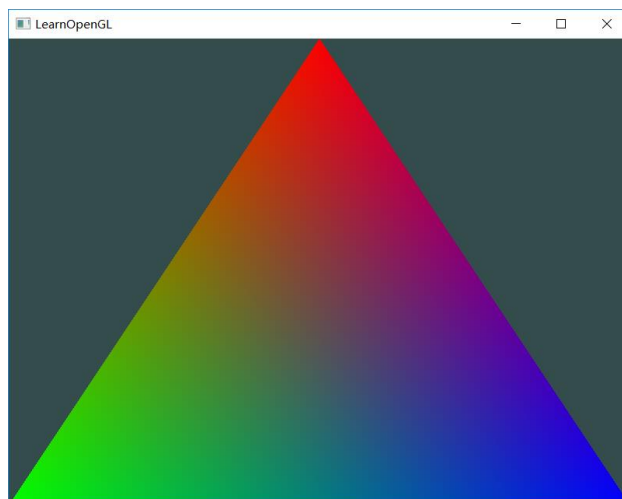


实现思路：

- (1) 初始化 GLFW，创建窗口对象，初始化 GLAD。
- (2) 使用 GLSL 编写顶点着色器和片段着色器代码，编译着色器代码并链接至着色器对象。
- (3) 创建 VAO, VBO，配置顶点属性。
- (4) 使用上述着色器程序和 VAO 进行循环渲染。

2. 对三角形的三个顶点分别改为红绿蓝，像下面这样。并解释为什么会出现这样的结果。

运行结果：



实现思路：

在第一题的基础上修改以下部分：

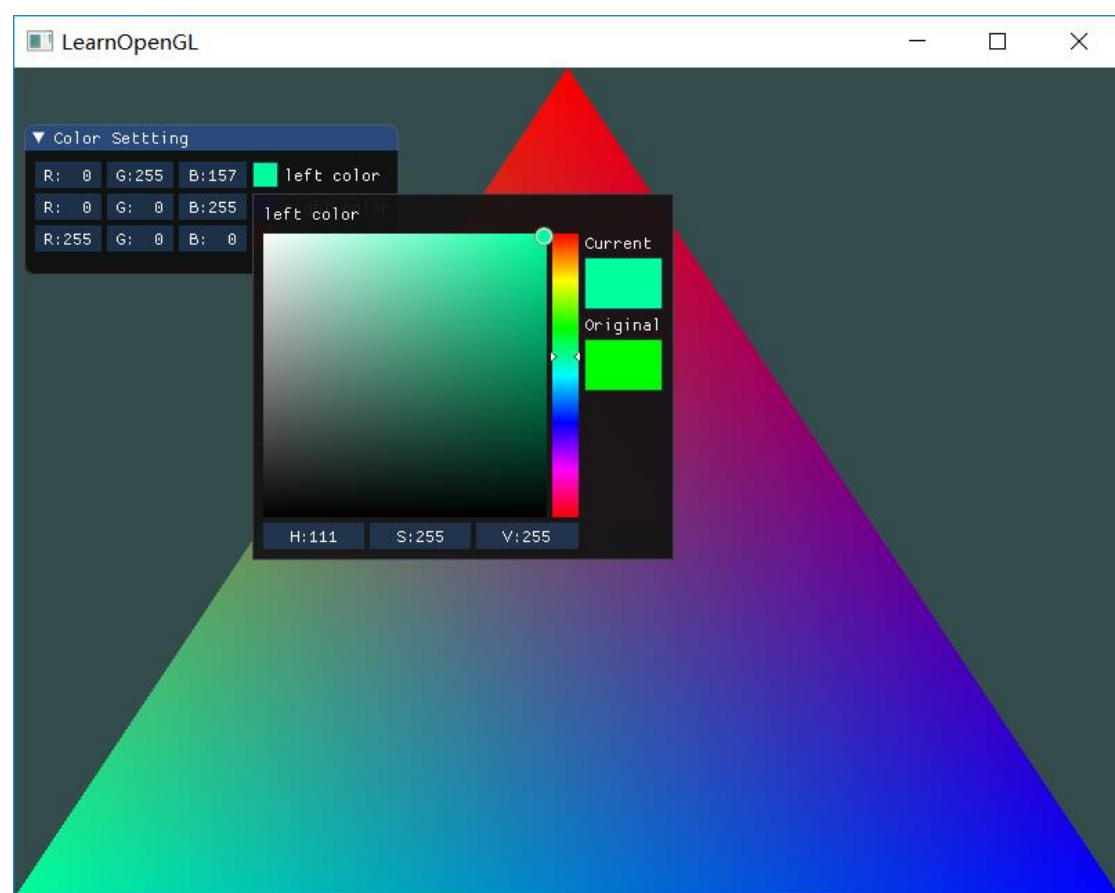
- (1) 给顶点着色器增加一个顶点属性输入 `aColor`，并增加一个输出变量 `ourColor`，将 `aColor` 赋值给 `ourColor`；给片段着色器增加一个输入变量 `ourColor` 用于接收顶点着色器的输出，将 `ourColor` 扩展为 `vec4` 作为片段着色器的输出。
- (2) 修改顶点数组，修改顶点坐标以及添加顶点对应颜色。
- (3) 配置顶点属性指针，将三角形顶点坐标配置至位置值为 0 的顶点属性，顶点颜色配置至位置值为 1 的顶点属性。

结果解释：

片段着色器进行了片段插值。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。因此，该三角形显示为一个调色板。

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。

运行结果：



实现思路：

在第二题的基础上增加以下内容：

- (1) 引入 `imgui` 库，初始化 `imgui`；

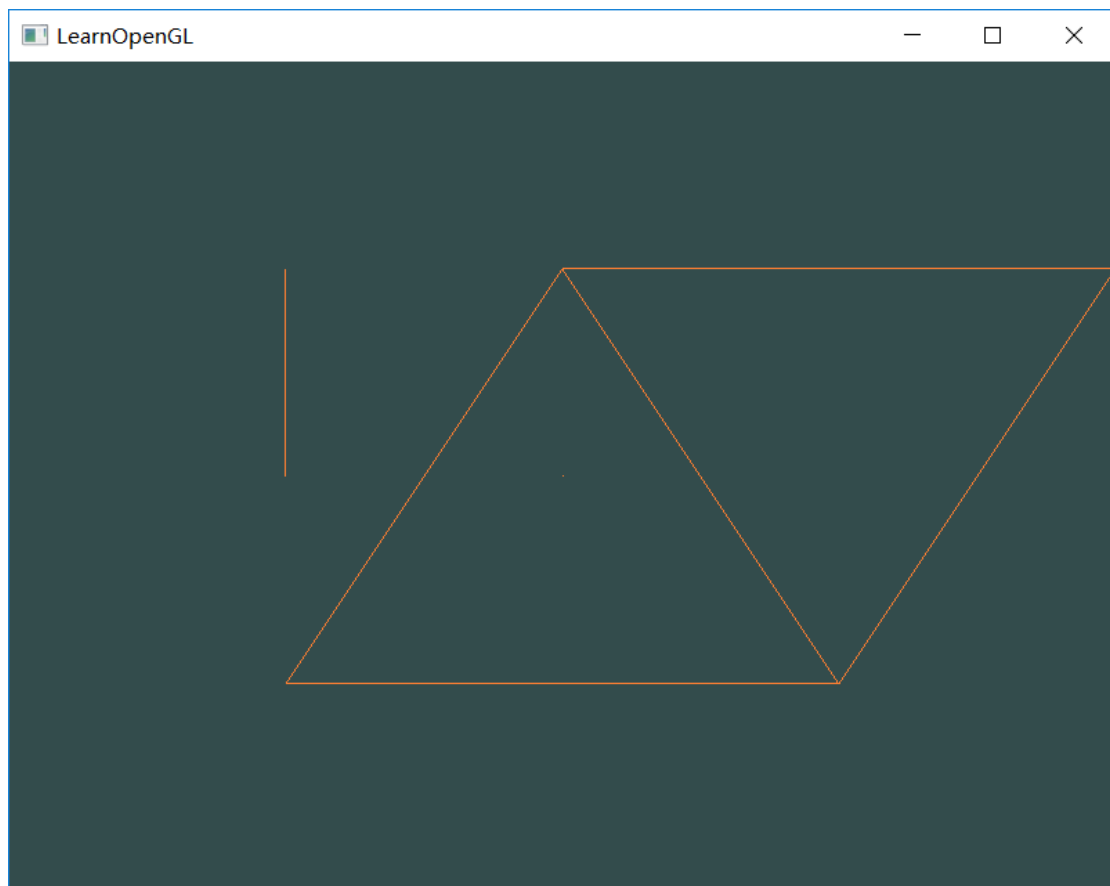
- (2) 创建 imgui 窗口, 包含 3 个 ColorEdit3, 用于控制三角形三个顶点的颜色; 添加 3 个变量用于接收 ColorEdit3 的颜色;
- (3) 在循环渲染中, 将 3 个变量的颜色赋值到 vertices 数组对应位置, 并调用 glBufferData, 将更改后的 vertices 的数据发送给 VBO。

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。

运行结果: 使用 EBO 在线框模式下绘制了两个三角形; 左侧绘制了一条直线; 中心绘制了一个点。



实现思路:

在第一题的基础上做以下修改:

- (1) 给 vertices 顶点数组增加右上角一个顶点, 创建索引数组, 使用 vertices 的这 4 个点的索引向索引数组添加 6 个索引项;
- (2) 创建 EBO, 在绑定完 VAO 后绑定 EBO, 通过 glBufferData 向 EBO 导入索引数组的数据;
- (3) 使用 glDrawElements 绘制出两个三角形;
- (4) 向 vertices 添加两个点作为直线的两个端点, 再添加一个点用于绘制一个点;

- (5) `glDrawArrays` 使用 `GL_LINES` 作为第一个参数, 绘制直线;
- (6) `glDrawArrays` 使用 `GL_POINTS` 作为第一个参数, 绘制点;