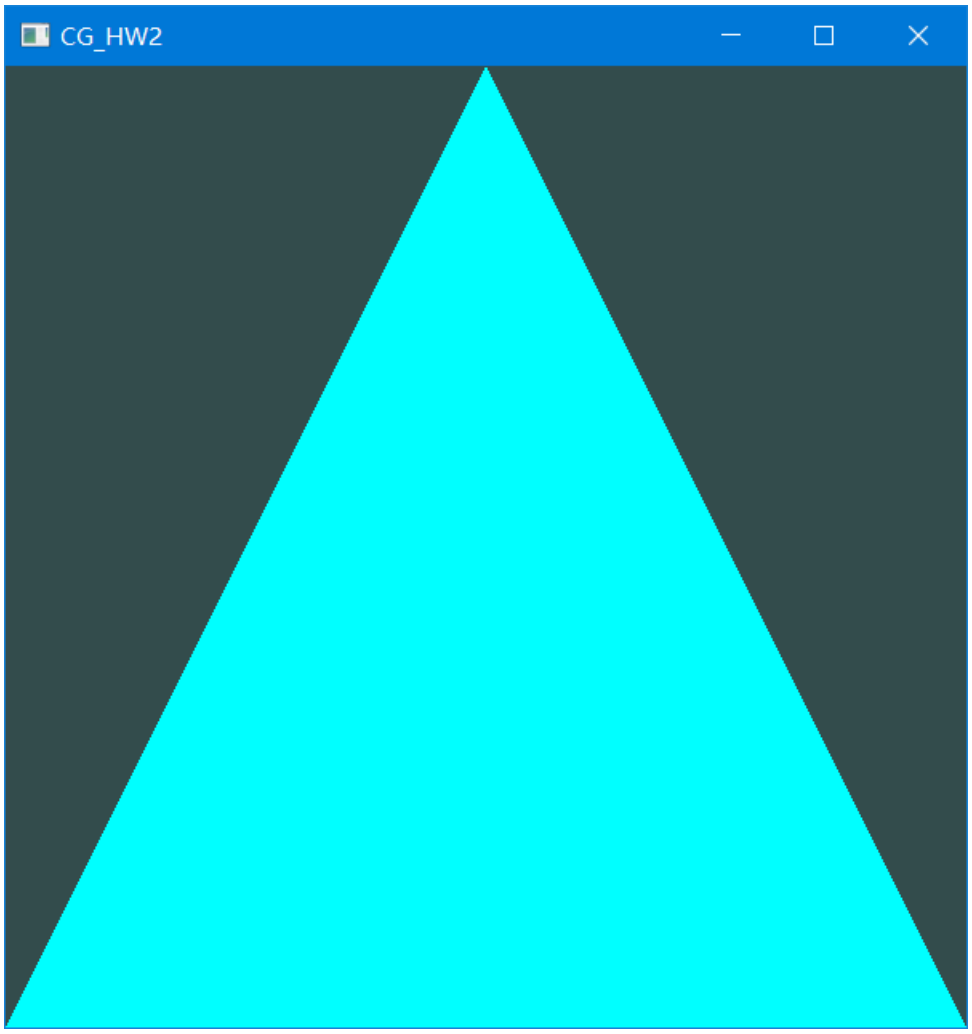
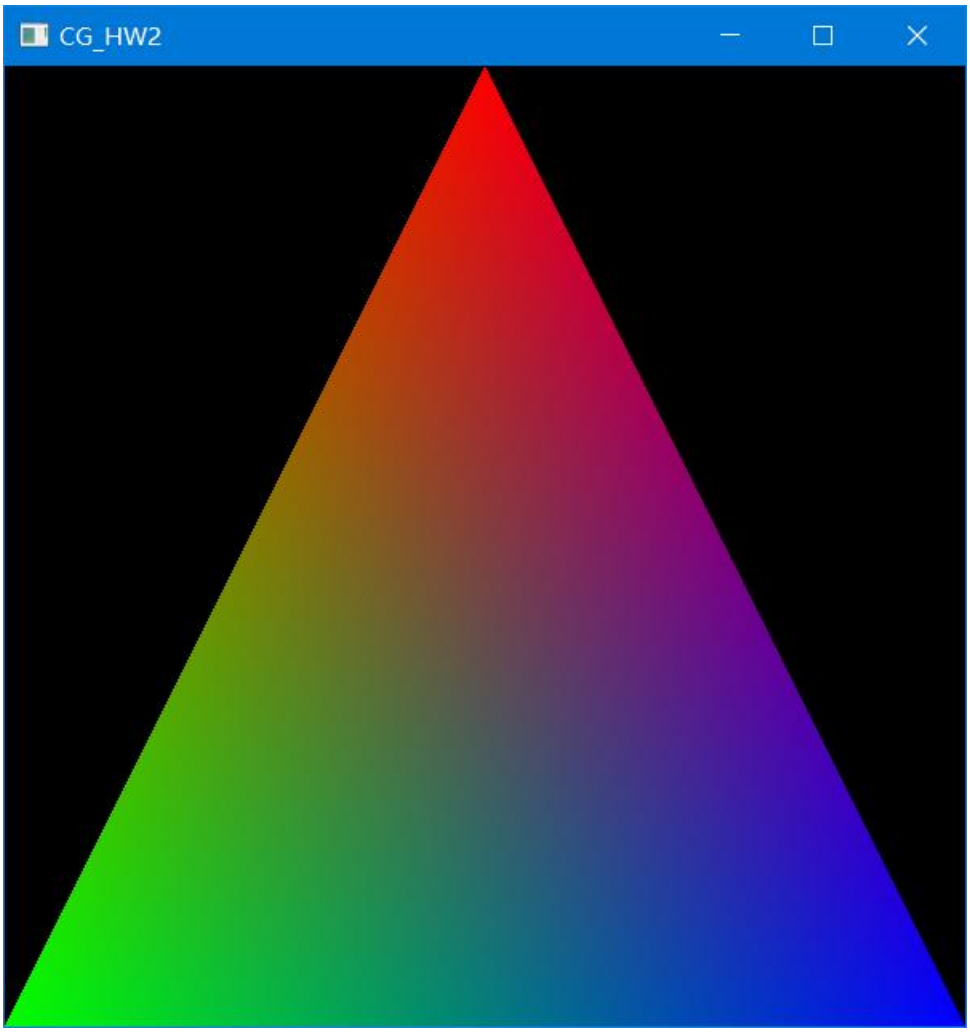
Homework2 Report

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



① 在 main 函数开头使用 glfwInit();实例化 GLFW 并调参，通过函数 glfwCreateWindow(600, 600, "CG_HW2", NULL, NULL);创建 GLFW 窗口并调节参数。
② 初始化 GLAD 以及调节视口，并添加渲染循环。
③ 初始化顶点坐标、VBO 和 VAO，绑定 VAO，通过 glBufferData(GL_ARRAY_BUFFER, sizeof(vertices), vertices, GL_STATIC_DRAW);将定点数据复制到 VBO 中。
④ 编写、编译定点着色器和片段着色器，并附着、链接成一个着色器程序对象。
⑤ 在渲染循环中调用 glUseProgram(shaderProgram);运行着色器程序，绑定 VAO，调用 glDrawArrays(GL_TRIANGLES, 0, 3);绘制图元。

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。



① 修改 vertices[]数组的值，使三个点坐标对应作业要求的颜色。

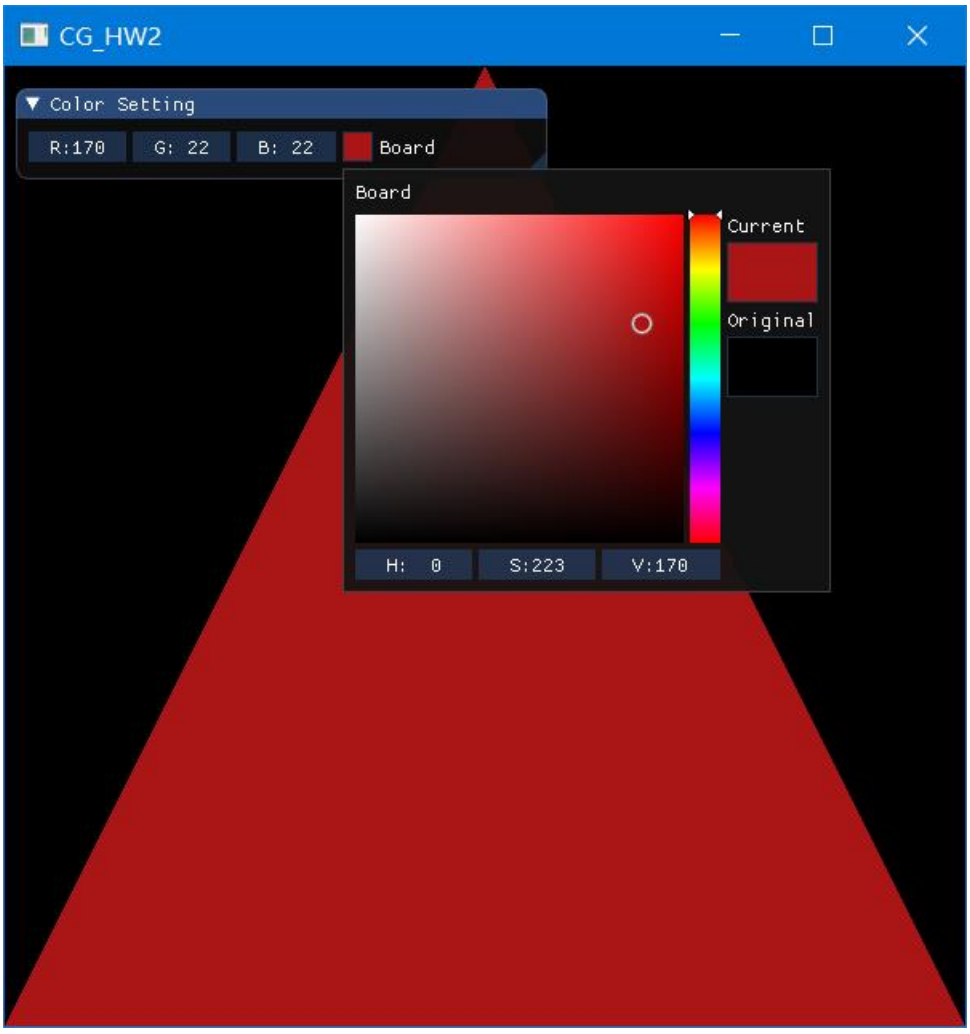
② 修改顶点着色器，增加颜色变量使其能输入和输出 RGB 值，输出到片段着色器内。

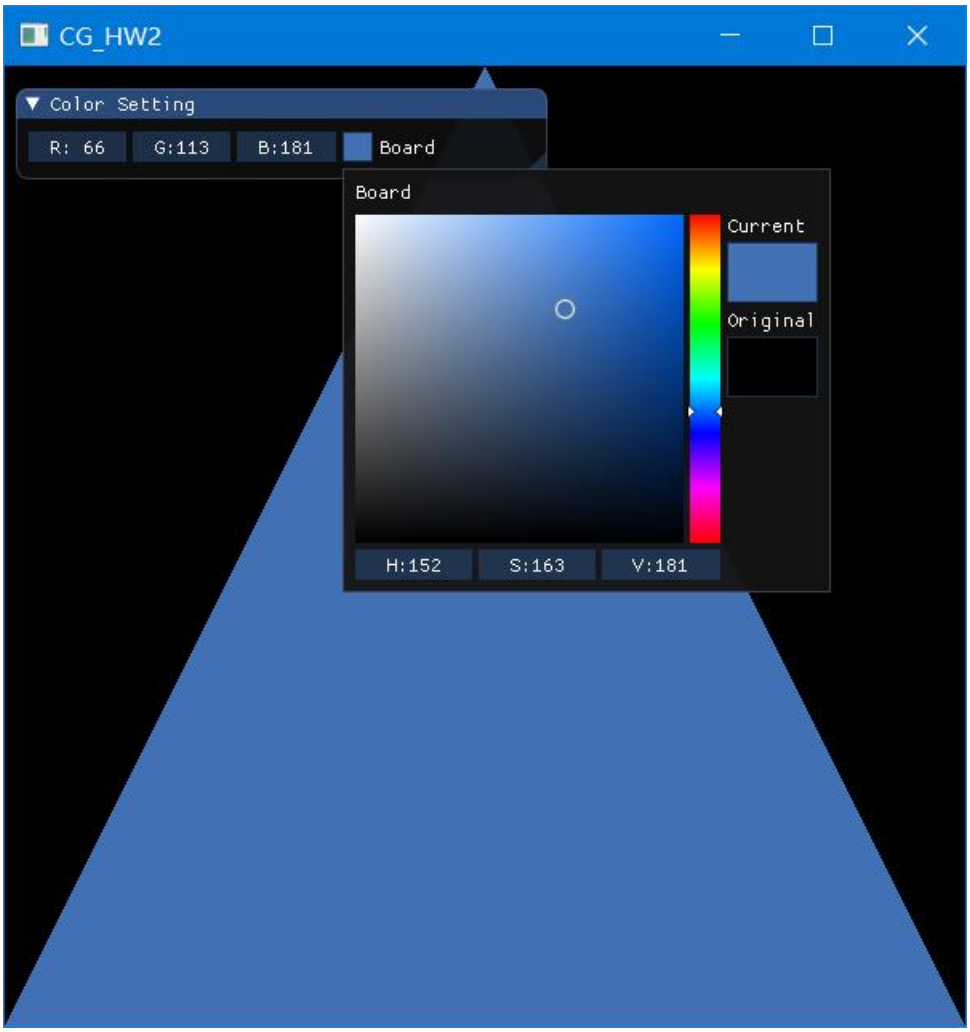
③ 分别调用 glVertexAttribPointer 函数两次，对顶点着色器和片段着色器更新，调整偏移和步长对应数组的值。步长皆为 6 个 float 的内存空间，位置属性和颜色属性的起始点分别为 0 和 3 个 float。

④ 原因解释:

首先，OpenGL 中的一个片段是 OpenGL 渲染一个像素所需的所有数据。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。

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颜色。


CG_HW2
Color Setting
R:170
G: 22
B: 22
Board
Board
Current
Original
H: 0
S:223
V:170

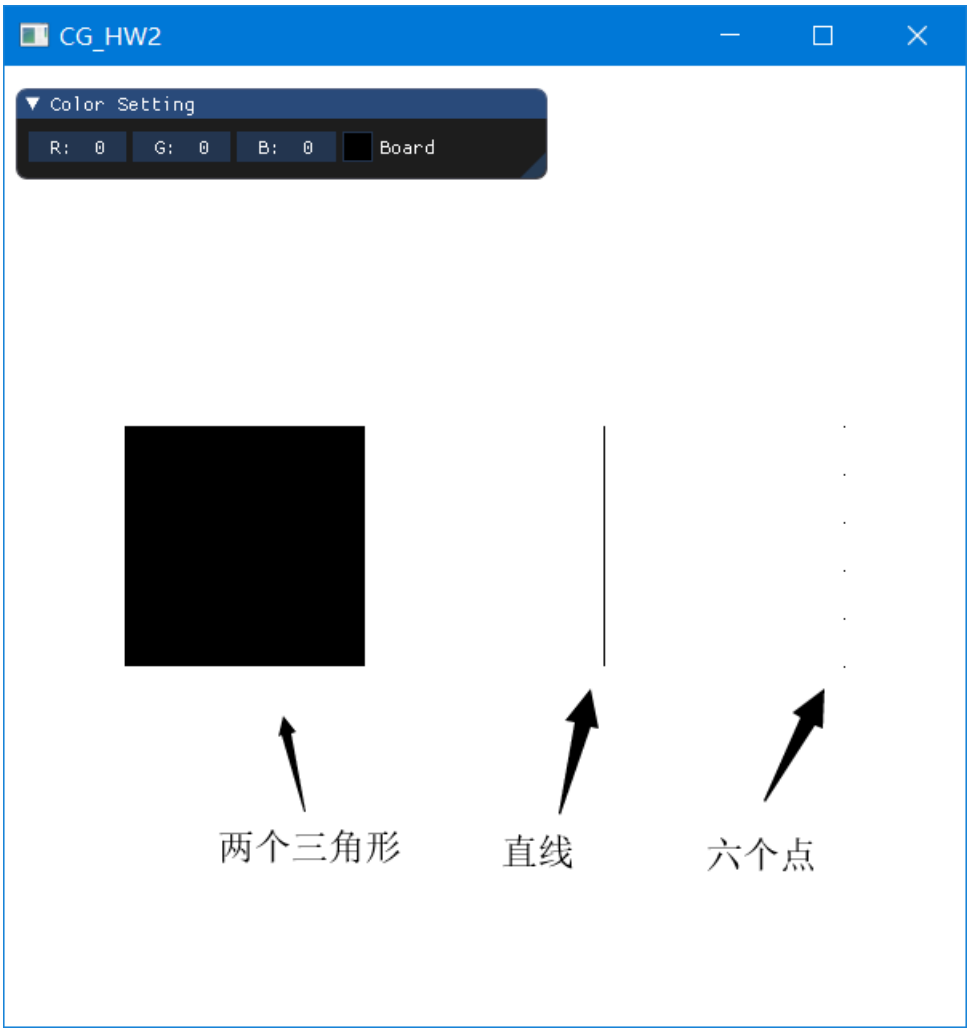


① 创建 IMGUI 并绑定 window（使用 glad 需要修改 imgui-impl_opengl3.h 中第 34 行的 `default OpenGL3 loader` 为 GLAD）

② 创建 IMGUI 窗口并添加元件 `ImGui::ColorEdit3("Board", (float*)&color);`作为选色板。

③ 在 while 中的 ImGui 窗口设置后面通过上述函数的引用参数修改颜色信息（数组），并重新设置指针，使顶点着色器的参数更新。

4. （Bonus）绘制点、线，使用 EBO 绘制多个三角形。



① 在数组中声明两个三角形要用的四个顶点，在另一数组声明两个三角形对应顶点索引。有两个点复用因此只需要存储四个点。

② 绑定 EBO，调用 `glDrawElements(GL_TRIANGLES, 6, GL_UNSIGNED_INT, 0)`;绘制 EBO。

③ 顶点数组增加顶点坐标和属性，增加、修改 `glDrawArrays` 函数参数绘制线和点。

作业提交的 src/中 4 个 main 分别对应以上四道题的代码。