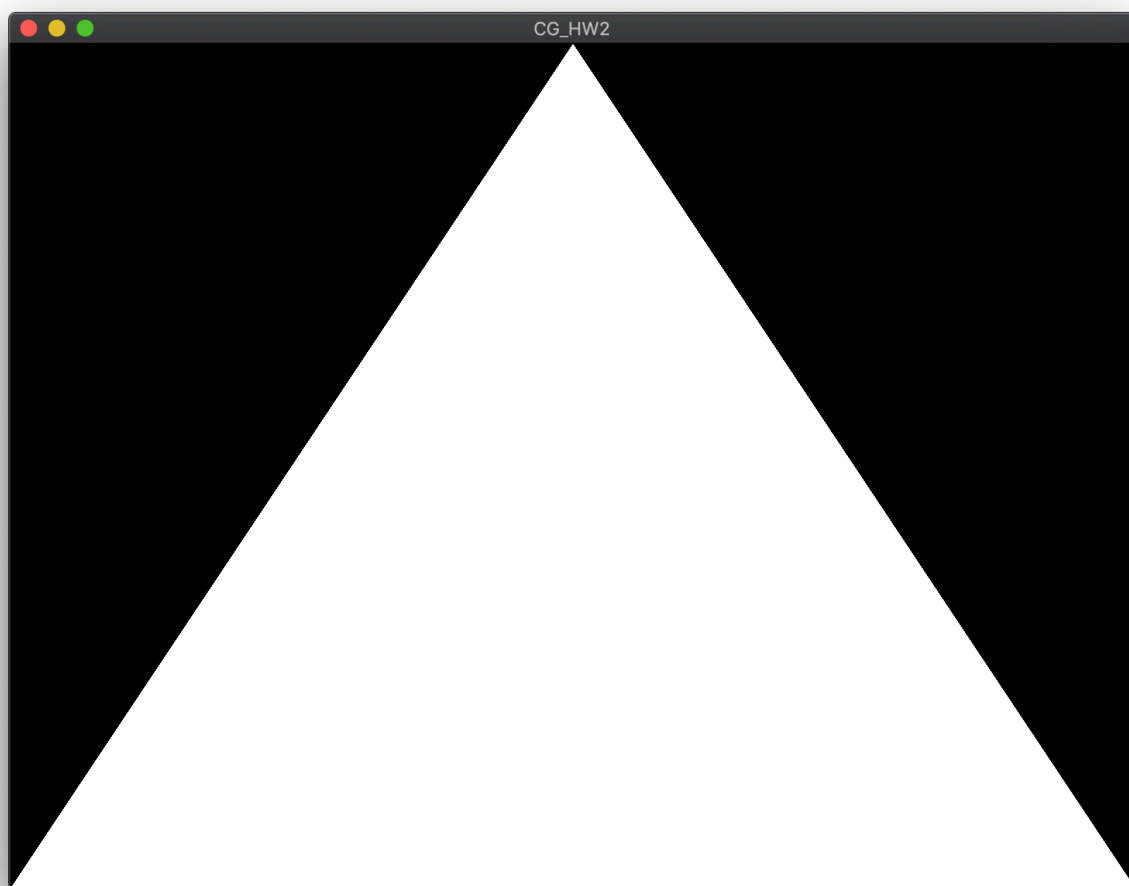


Basic:

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

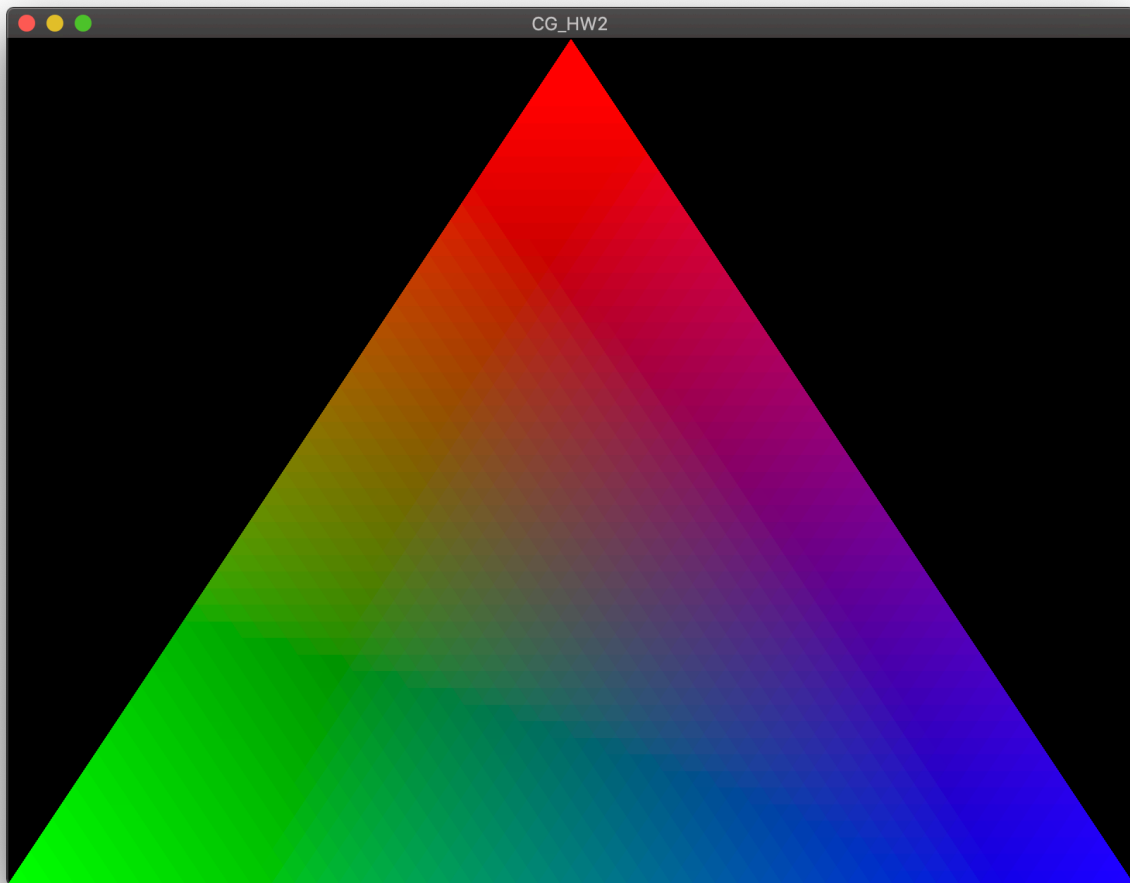


glfwCreateWindow创建一个窗口对象，glfwMakeContextCurrent将这个窗口的上下文设置为当前线程的主上下文，glGenVertexArrays创建顶点数组对象VAO，glGenBuffers创建顶点缓冲对象VBO，glGenBuffers创建索引缓冲对象EBO，glBindVertexArray绑定VAO，glBindBuffer绑定VBO，创建一个顶点属性（坐标）数组，指定三角形三个顶点的坐标，并通过glBufferData复制到VBO中，glBindBuffer绑定EBO，创建一个顶点索引数组，指定顶点的顺序，并通过glBufferData复制到EBO中，glVertexAttribPointer设定对顶点属性的解析创建顶点属性指针并glEnableVertexAttribArray生效，用GLSL语言编写顶点着色器代码，使用in关键字声明顶点属性输入，与VBO对应，gl_Position使其确定位置，再用glCreateShader、glShaderSource、glCompileShader生成顶点着色器，编写片段着色器代码，用out关键字声明输出变量，指定颜色为白色，再用glCreateShader、glShaderSource、glCompileShader生成片段着色器，glCreateProgram创建着色器程序，glAttachShader添加顶点着色器和片段着色器，再用glLinkProgram链接，在渲

染循环中glUseProgram激活这个着色器程序，再glBindVertexArray绑定VAO，glDrawElements指定图元和顶点数根据EBO进行图形绘制。

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。

修改1.，为glBindBuffer绑定VBO后创建的顶点属性数组的每个点加上对应颜色属性红绿蓝，修改glVertexAttribPointer设定对顶点属性的解析步长，glEnableVertexAttribArray设置为0表示位置属性，增加一个glVertexAttribPointer设置偏移量，glEnableVertexAttribArray设置为1表示颜色属性，修改顶点着色器代码，使用in关键字声明，区分输入属性位置值0和1，gl_Position使用位置0属性确定位置，使用out关键字为片段着色器输出位置1属性确定颜色，修改片段着色器代码，用in关键字声明输入变量接收再用out将其输出，不同顶点就得到各自的颜色。



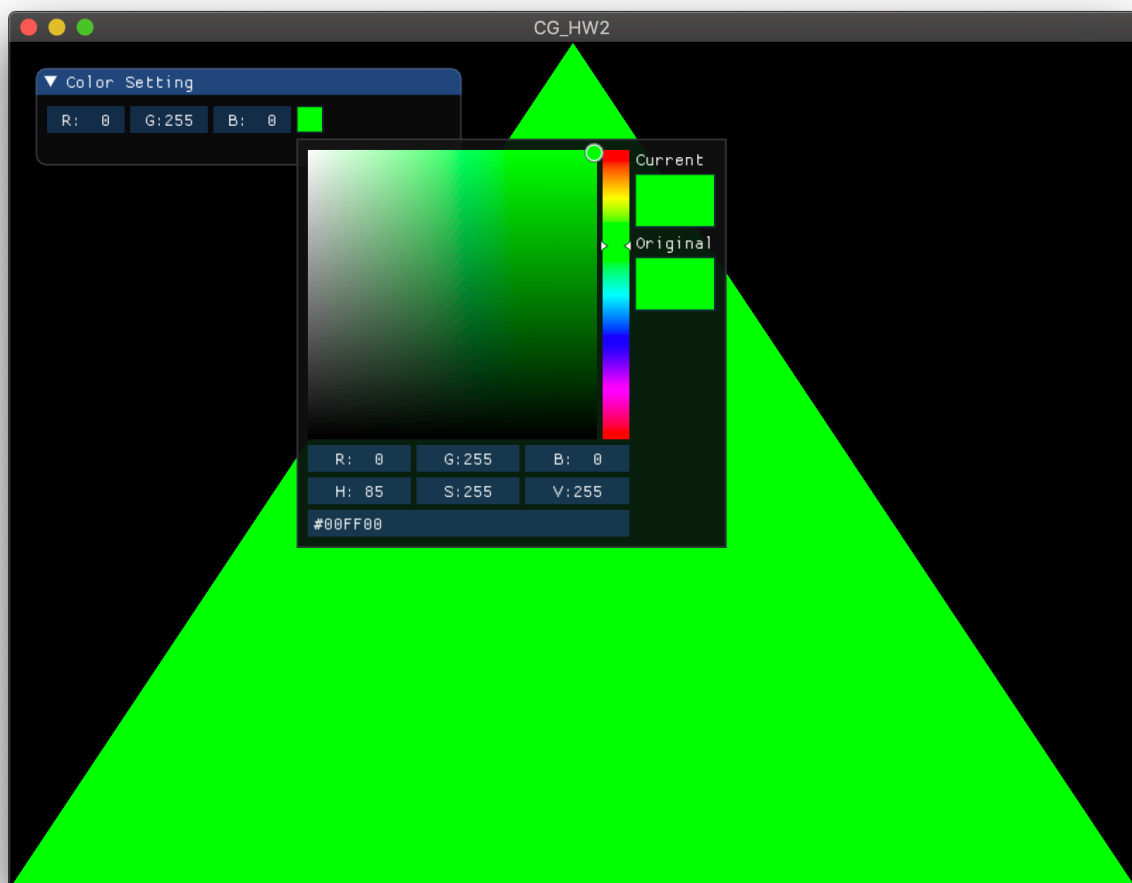
因为只为片段着色器提供了三角形三个顶点的颜色属性，片段着色器会为三角形的三个顶点外的其他区域的进行片段插值，在渲染时，光栅化阶段通常会造成比原指定顶点更多的片段，会根据每个片段在三角形形状上所处相对位置决定这些片段的位置，并对这些位置插值所有片段着色器的输入变量，所以会出现颜色渐变过渡的结果。

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。

为了使着色器代码容易管理，并支持从外部文件读取GLSL代码，使用Shader类将文件读取、编译、着色器程序的使用、uniform赋值进行封装。

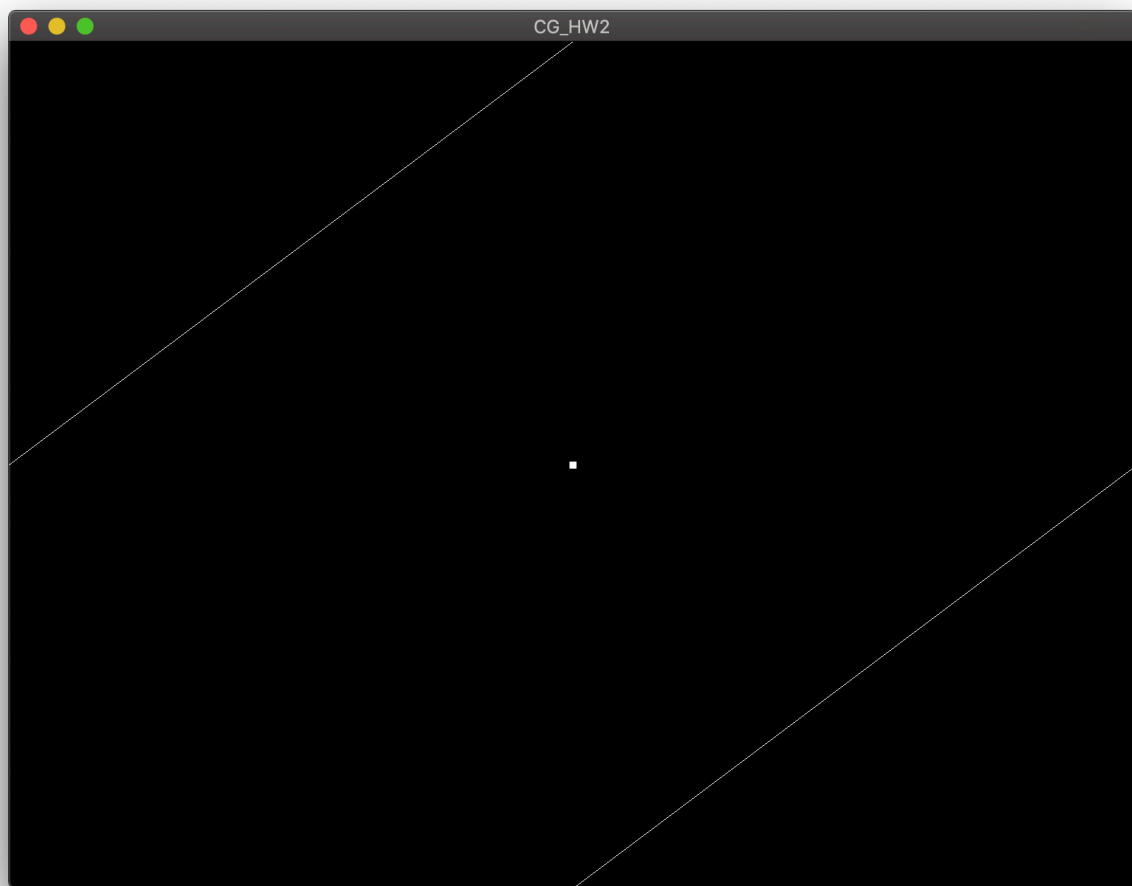
修改1.，修改片段着色器代码，用uniform关键字声明支持从CPU向GPU着色器发送数据的变量用于交互，再用out将其输出，引入ImGui，ImGui::CreateContext创建上下文，声明一个ImVec4变量用于与uniform变量交互，在渲染循环中加入

ImGui_ImplOpenGL3_NewFrame、ImGui_ImplGlfw_NewFrame、ImGui::NewFrame创建窗口，ImGui::ColorEdit3添加颜色选择控件用于控制ImVec4变量的值，再用glUniform4f为uniform变量赋值ImVec4变量的值，ImGui::Render、ImGui_ImplOpenGL3_RenderDrawData渲染ImGui窗口。



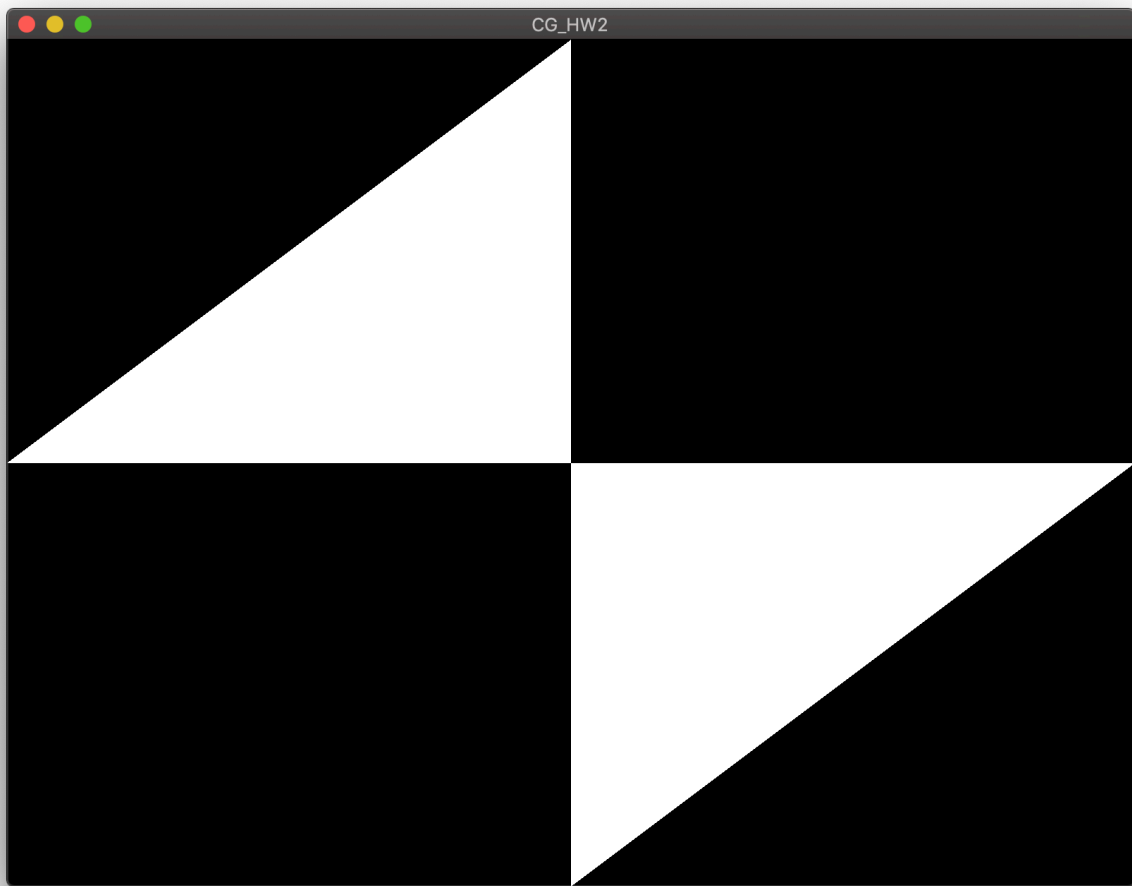
Bonus:

1. 绘制其他的图元，除了三角形，还有点、线等。



修改**Basic:1.**，在glBindBuffer绑定VBO后的顶点属性（坐标）数组添加多几个顶点的坐标，修改在glBindBuffer绑定EBO后的顶点索引数组，指定顶点的使用顺序，渲染循环中修改glDrawElements的图元为GL_LINE_STRIP用于画线，指定使用顶点个数为2，再添加一个图元为GL_POINTS的glDrawElements并设置其顶点数为1、偏移量为GLvoid*)(2 * sizeof(uint)，使其从索引数组的第三个点开始画点，glPointSize设置点的宽度使这个点容易观察，再添加一个glDrawElements画另一条线。

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



修改**Bonus:1.**，修改在glBindBuffer绑定EBO后的顶点索引数组，指定顶点的使用顺序，渲染循环中只用一个图元为GL_TRIANGLES的glDrawElements，并设置其顶点数为6即可画两个三角形。