计算机图形学 Homework 2 - GUI and Draw simple graphics

Basic：

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

   实现的方法为通过 VBO 传入三个顶点的位置和颜色，在这步中三个顶点的颜色相同，因此绘出的三角形中仅填充一种颜色。具体过程为：

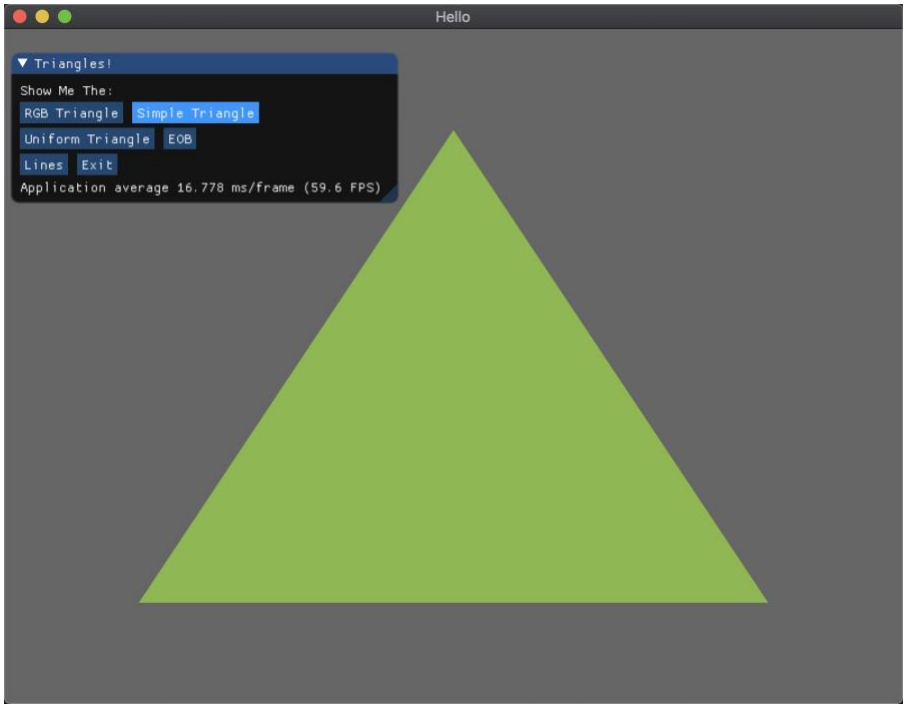
   首先通过 glGenVertexArrays 绑定一个生成的顶点数组对象，以存储顶点属性调用。通过 glBindBuffer 绑定一个生产的顶点缓冲对象，用于存储顶点数据，将顶点数据使用 glBufferData 传输至缓冲中。最后，通过 glVertexAttribPointer 和 glEnableVertexAttribArray 来设置顶点属性指针，即解释传入的数据应该如何被着色器读取。

   在这一步中，实现的顶点着色器的功能是将获得的位置数据赋给 gl_Position，以直接输出顶点位置，并将获得的颜色数据输出到 verColor 变量，这个变量将被输入至片段着色器。片段着色器的功能位将输入的 verColor 变量输出到一个 4 分量向量，以表示三角形渲染的颜色。

   提交的代码中包含了一个 shader 类，该类的作用是将传入的着色器代码路径读入文件并进行编译与连接至着色器程序。其他部分的代码中通过 shader 类的 use 方法来使用获得的着色器程序。

   在每个渲染循环中，调用 shader->use() 使用着色器程序，然后通过 glBindVertexArray 来绑定该项所对应的 VAO，同时可以使用到其绑定的 VBO。最后使用 glDrawArrays 来绘制出目标三角形。

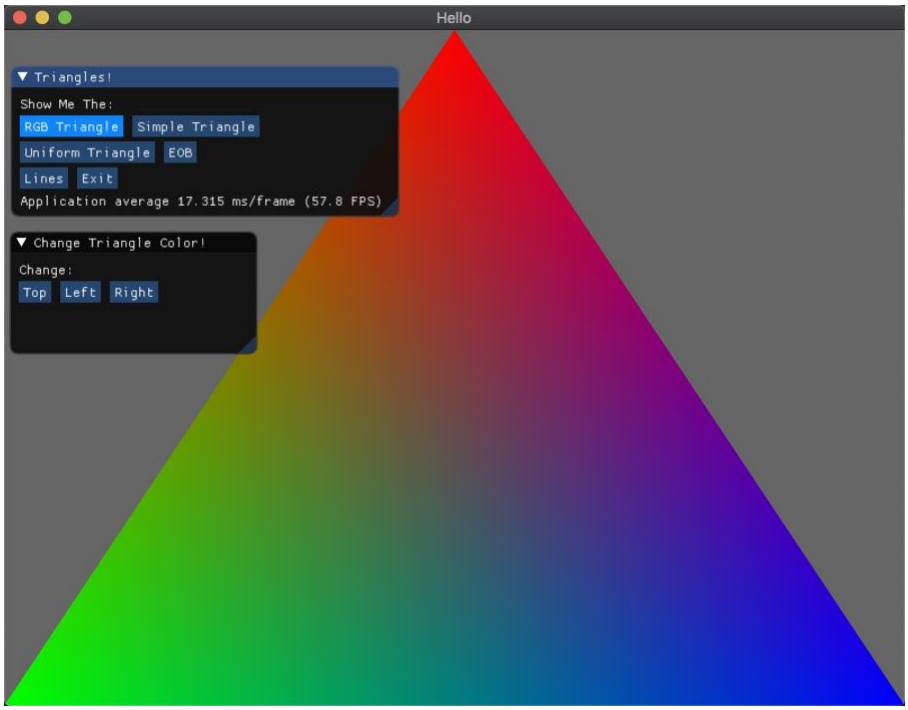
   最终的效果如下图：



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获得的结果。

这一步的实现方式与上一步的实现类似，区别在于输入的顶点数据中三个顶点的颜色不同。

效果如下：

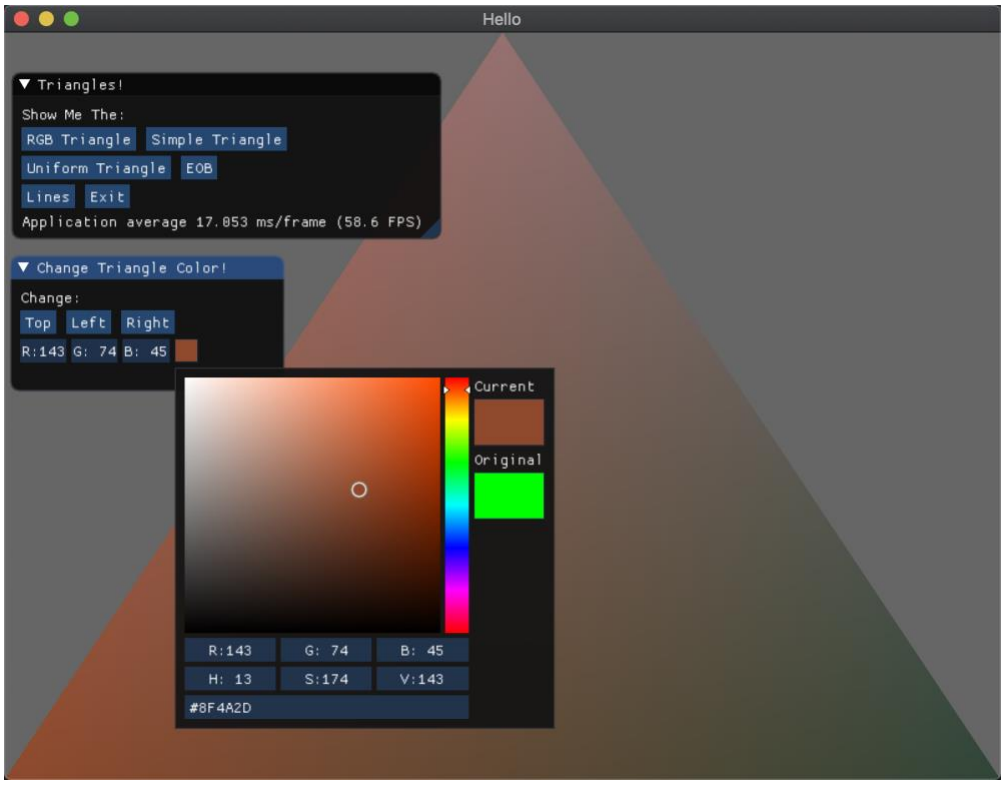


出现获得的效果的原因是当三个顶点的颜色不相同时，OpenGL 在渲染过程中在片段着色器中的片段差值过程，对于三角形的各个位置输入片段着色器的变量进行插值，从而获得上图中三角形各个部分的颜色线性变化的效果。

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。

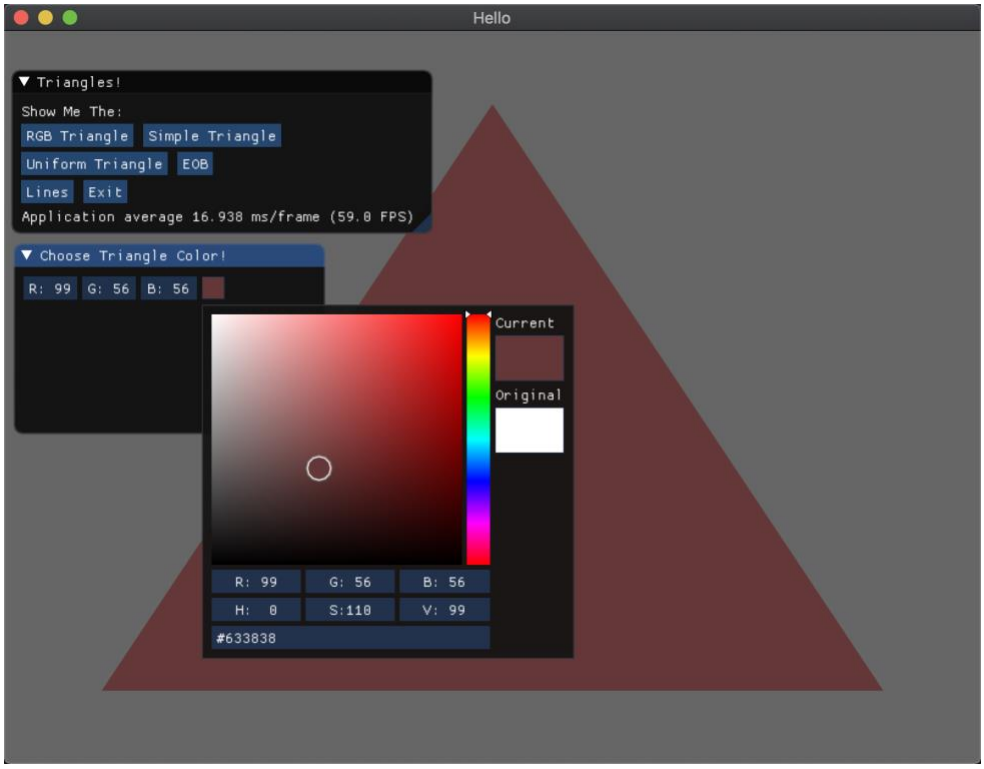
在这一步中我使用了 ImGui 库，首先创建了一个使用按钮选择显示哪一步的效果的 GUI 窗口。切换不同显示内容的方法是根据需要显示的内容切换绑定的 VAO 并进行内容的绘制。

然后在上一步的三角形的基础上添加一个可以更改三角形三个顶点颜色的窗口，窗口中使用 ImGui::ColorEdit3 来进行颜色的选择，然后在每一次渲染循环中通过 glBufferData 来将新的颜色数据传入顶点缓存。实现的效果如下：



对于单色的三角形，我选择了使用另一个片段着色器，该着色器接收一个 uniform 变量的输入。在每轮渲染循环中通过 glGetUniformLocation 获取该变量的位置，然后通过 glUniform4f 传递新的颜色至着色器以实现三角形颜色的更改。

效果如下：

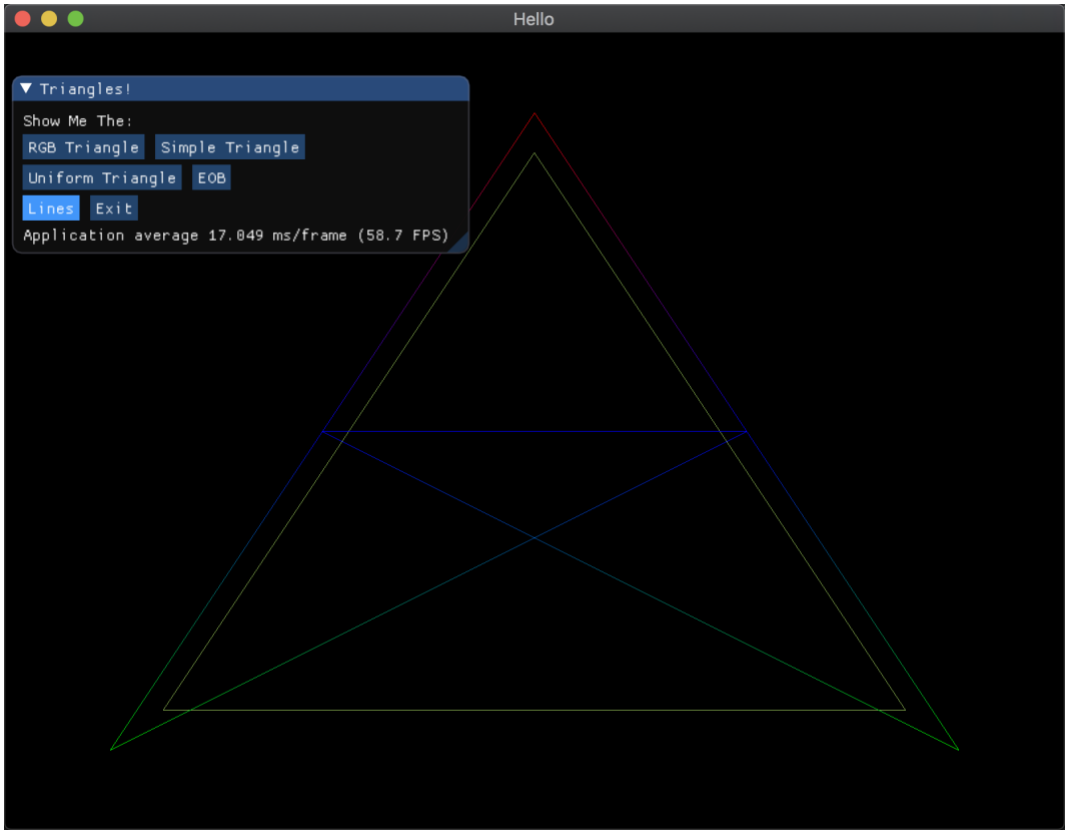


Bonus

1. 绘制其他的图元，除了三角形，还有点、线等。

   glDrawArrays 和 glDrawElements 接收的第一个参数为指定渲染的图元类型，可支持点、线、三角形等多种图元。在这里通过更改该参数使用 GL_LINE_LOOP 和 GL_LINES 进行了线的绘制，分别描出了前面的简单三角形的边缘线条以及通过指定的顶点和索引绘制了自定义的线条图形。

   效果如下：

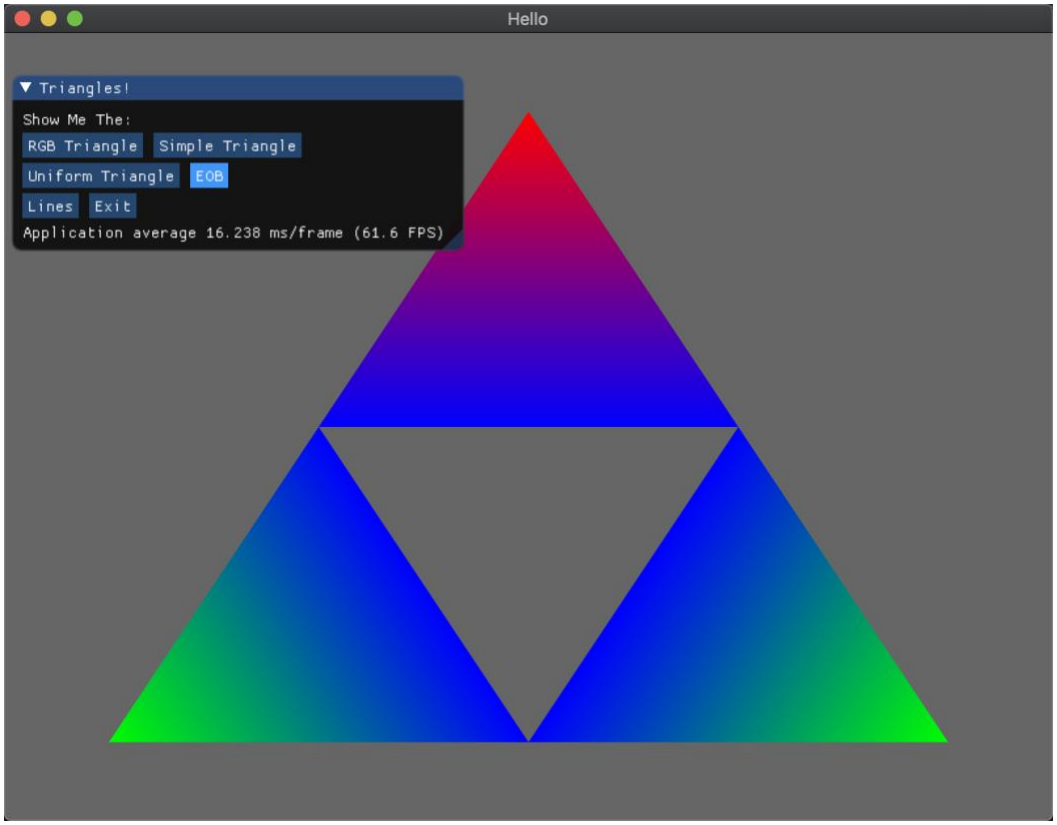


2. 使用 EBO(Element Buffer Object)绘制多个三角形。

EBO 为索引缓冲对象，缓冲类型为 GL_ELEMENT_ARRAY_BUFFER，创建和绑定的方法都与 VBO 类似。其作用是在渲染时指定绘制顶点的顺序，从而减少需要存储的顶点数目。

在这里，我 添加了 6 个顶点的数据，通过 9 个索引绘制出了三个三角形，通过这种方式可以避免额外的三个顶点的开销。绘制的时候使用 glDrawElements(GL_TRIANGLES, 9, GL_UNSIGNED_INT, 0);进行绘制，其中 9 代表绘制 9 个顶点，即 3 个三角形。

效果如下：


Hello
Triangles!
Show Me The:
RGB Triangle
Simple Triangle
Uniform Triangle
EOB
Lines
Exit
Application average 16.238 ms/frame (61.6 FPS)