

计算机图形学 作业2

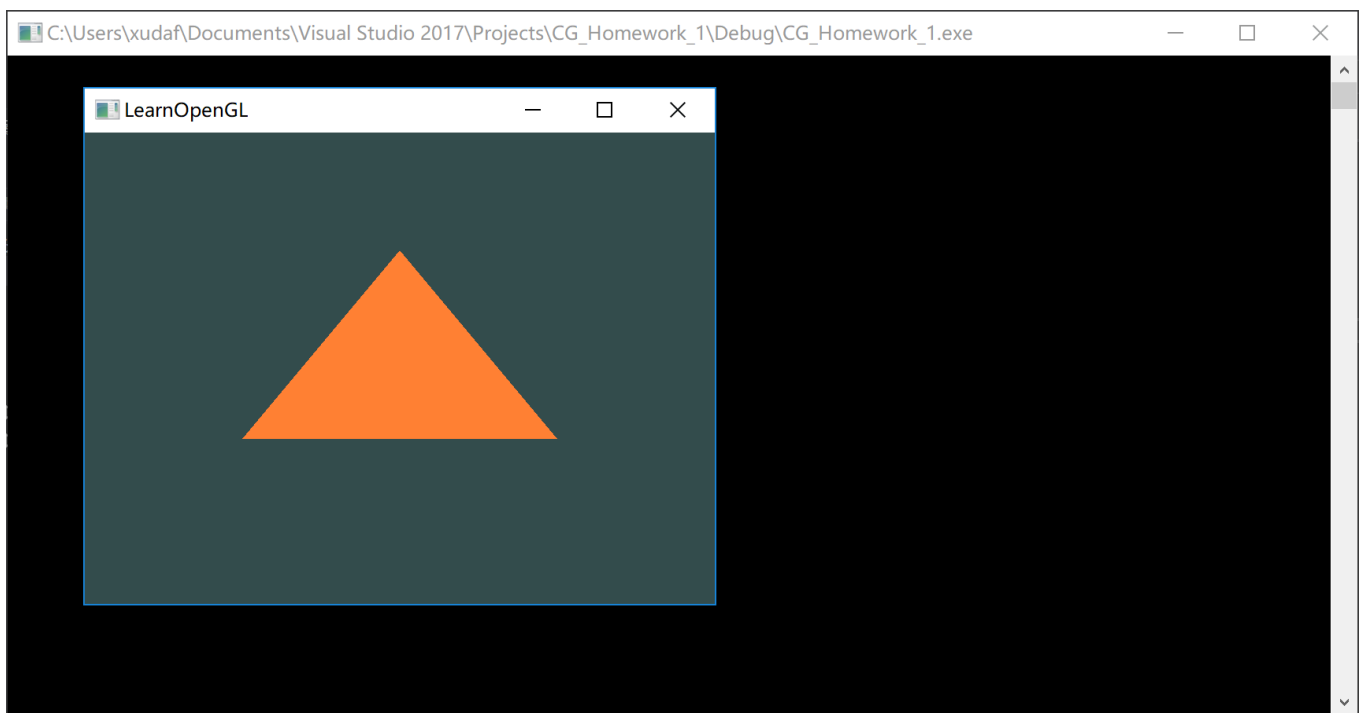
徐达烽

16340260

Problem 1

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

实验结果



实现思路

参照教程，写好两个shader，连接好。我把shader的代码存放在txt文件中，程序运行时先读取进来。

然后定义顶点数组，VAO和VBO。

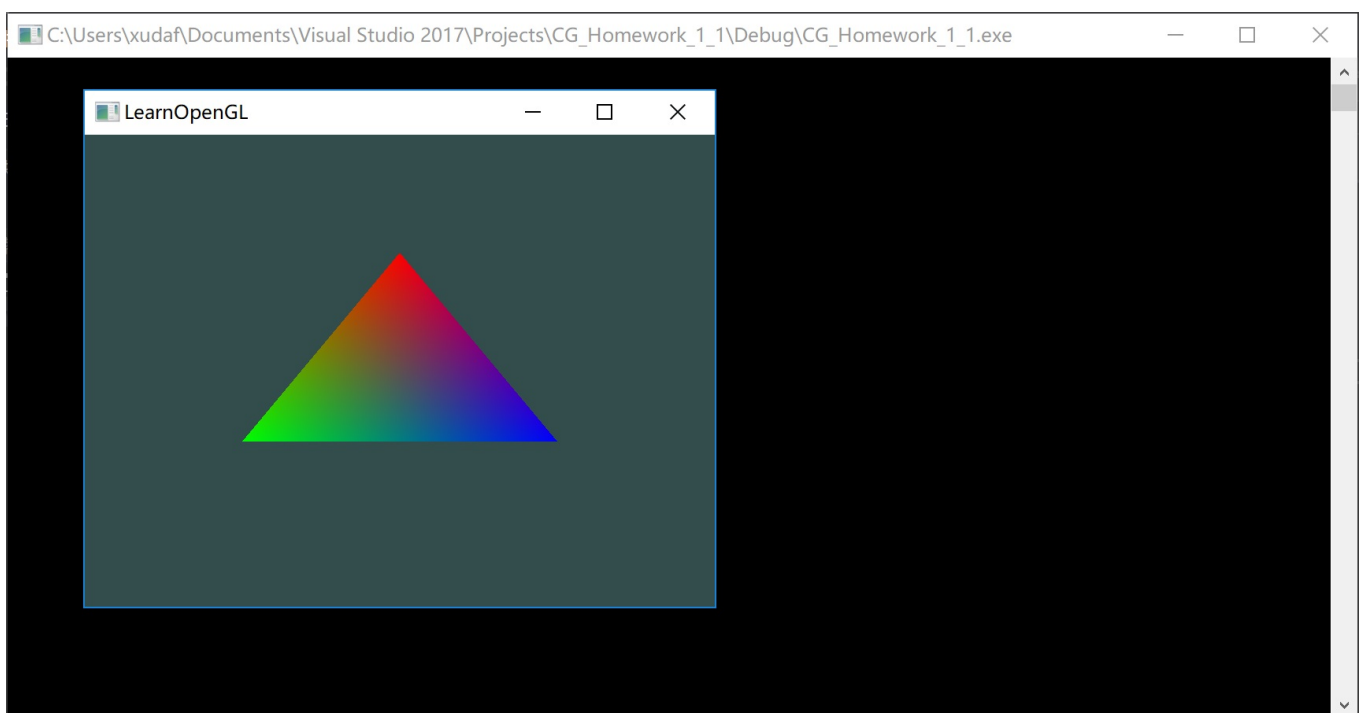
渲染循环中，这样实现画三角形：

```
1. glUseProgram(shaderProgram);
2. glBindVertexArray(VAO); // seeing as we only have a single VAO there's
   no need to bind it every time, but we'll do so to keep things a bit mo
   re organized
3. glDrawArrays(GL_TRIANGLES, 0, 3);
```

Problem 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。

实验结果



实现思路

修改vertex_shader和fragment_shader, 使其能够传进颜色作为属性。

```
1. // vertex_shader.txt
2. #version 330 core
```

```

3.  layout (location = 0) in vec3 aPos;
4.  layout (location = 1) in vec3 aColor;
5.  out vec3 ourColor;
6.  void main()
7.  {
8.      gl_Position = vec4(aPos, 1.0);
9.      ourColor = aColor;
10. }
11.
12. // fragment_shader.txt
13. #version 330 core
14. out vec4 FragColor;
15. in vec3 ourColor;
16. void main()
17. {
18.     FragColor = vec4(ourColor, 1.0f);
19. }

```

然后修改顶点数组，加入颜色坐标，并且在VAO中添加属性：

```

1.  glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 6 * sizeof(float), (void*)0);
2.  glEnableVertexAttribArray(0);
3.  glVertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE, 6 * sizeof(float), (void*)(3 * sizeof(float)));
4.  glEnableVertexAttribArray(1);

```

这样就做完了。为什么会出现这样的效果呢？因为在片段着色器中，会进行片段插值。

Problem 3

实验结果

见Problem_3_Result.gif 动图

实现思路

导入ImGui后，参照ImGui的API 文档（在ImGui.h的注释里），就可以完成了。

这样生成一个取色器：

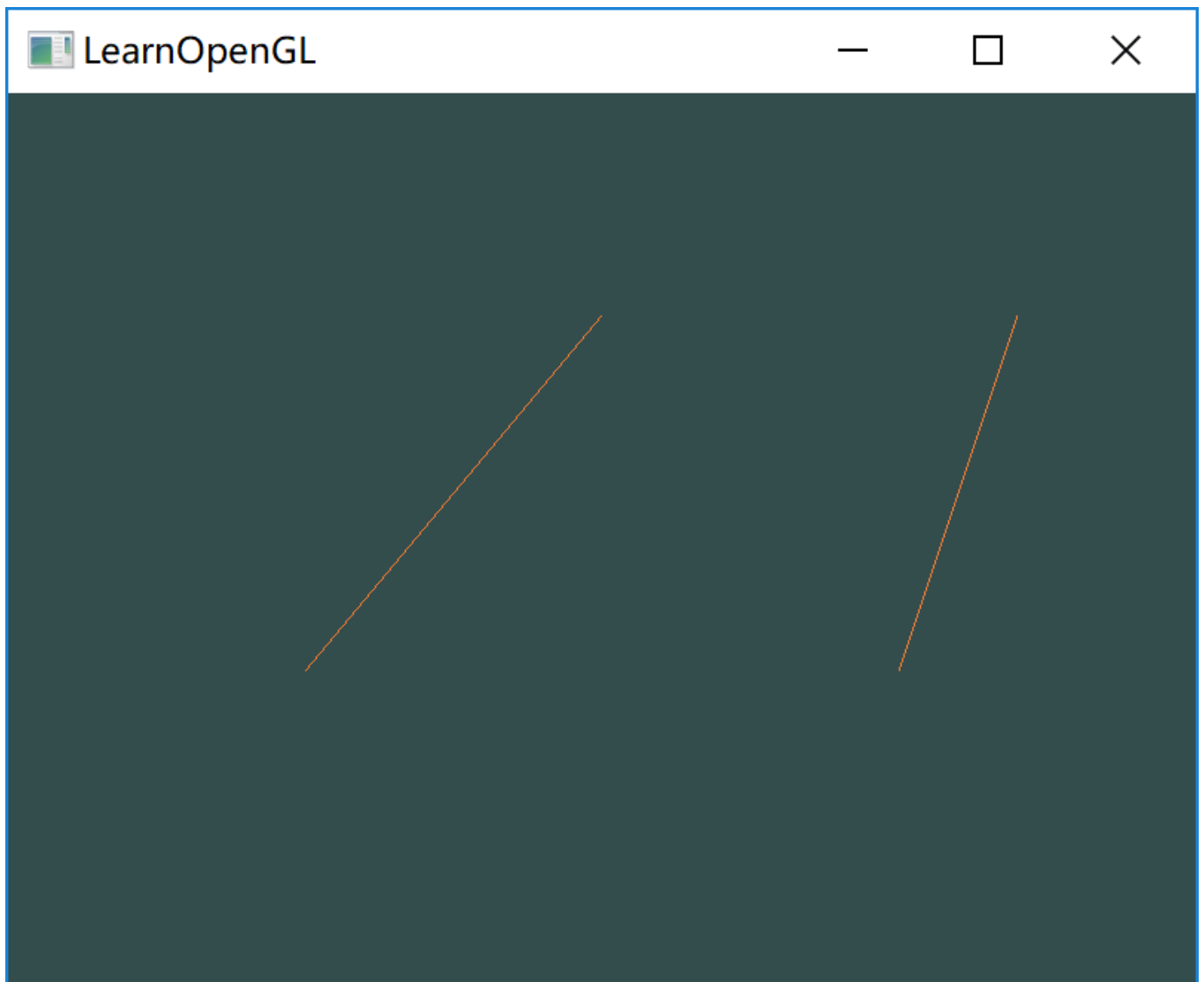
```
1.   ImGui::Begin("Hello, world!");
2.   ImGui::ColorEdit3("color picker", (float*)&color); // Edit 3 floats re
    presenting a color
3.   ImGui::End();
```

并且在fragment_shader中，设置ourColor为uniform变量，主函数中这样更新它：

```
1.   glUniform4f(vertexColorLocation, color.x, color.y, color.z, 1.0f);
```

Problem 4

实验结果



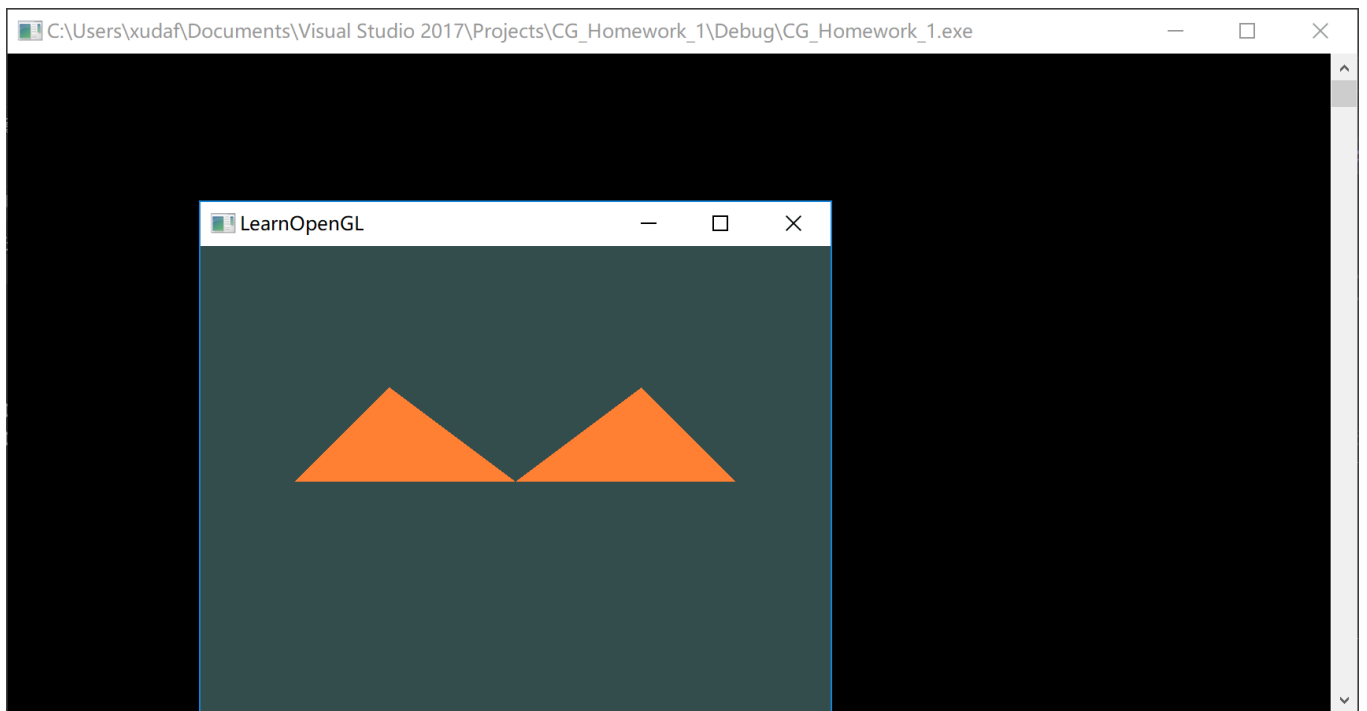
实验思路

使用GL_LINES即可画出直线。

```
1. glDrawArrays(GL_LINES, 0, 4);
```

Problem 5

实验结果



实现思路

依照教程设置好EBO, 并且渲染时用 `glDrawElements` 即可画出两个三角形。