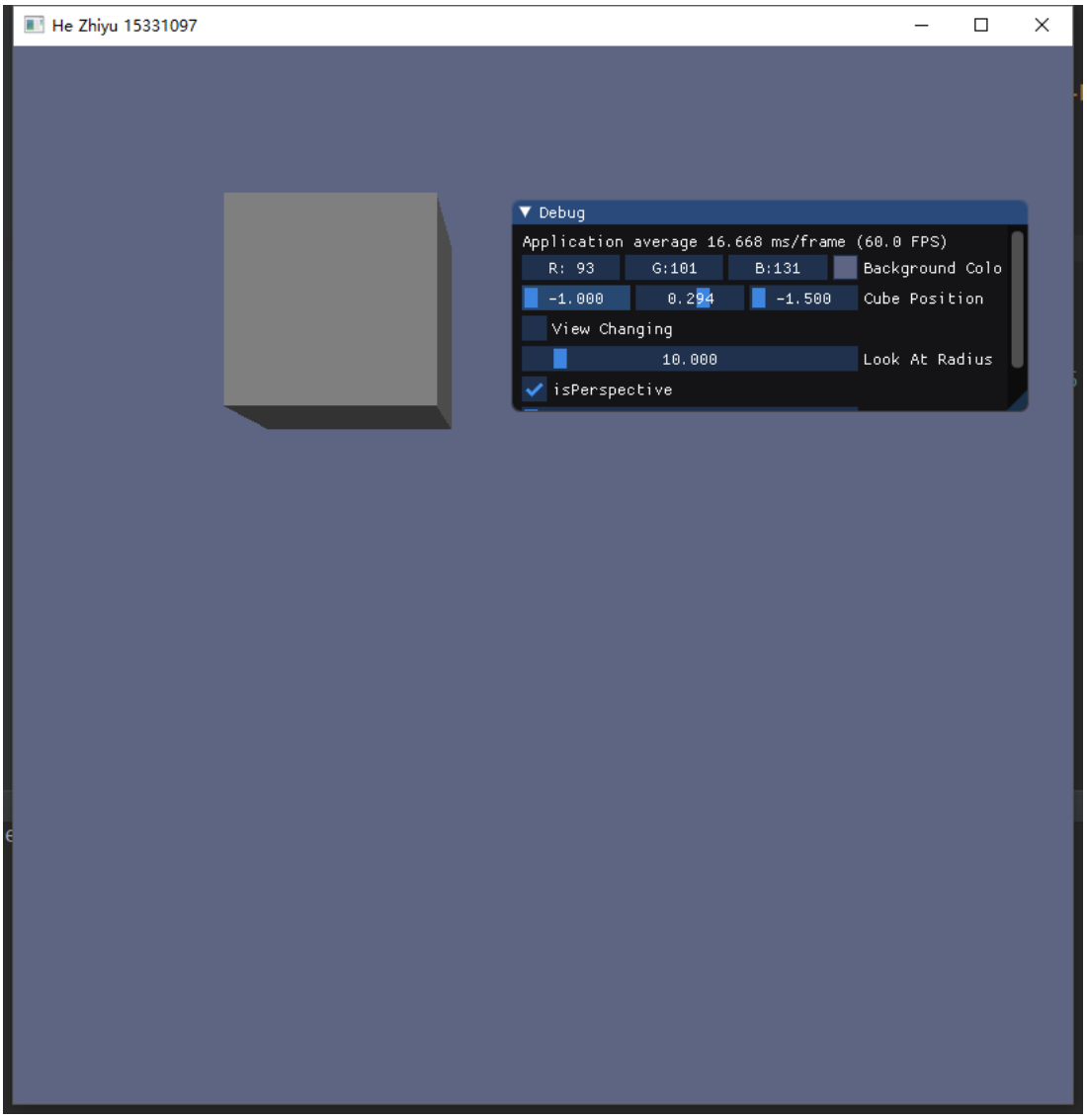
CG – HW5 – Camera
15331097，何志宇

如何运行

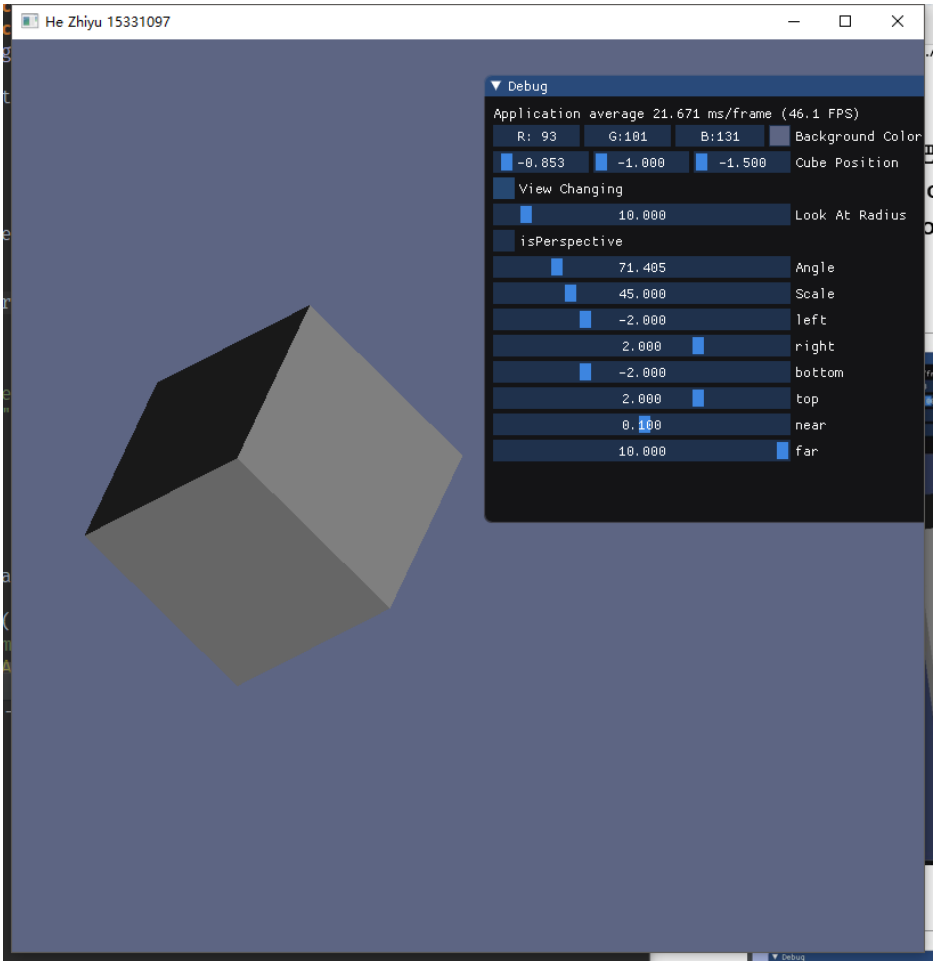
windows 环境下，打开./cmake-build-debug/hw3.exe 即可运行。若无此文件夹。可以根据 CMakeLists.txt 直接编译即可。

1. 投影(Projection): 把上次作业绘制的 cube 放置在(-1.5, 0.5, -1.5)位置，要求 6 个面颜色不一致 正交投影(orthographic projection)：实现正交投影，使用多组(left, right, bottom, top, near, far)参 数，比较结果差异 透视投影(perspective projection)：实现透视投影，使用多组参数，比较结果差异。

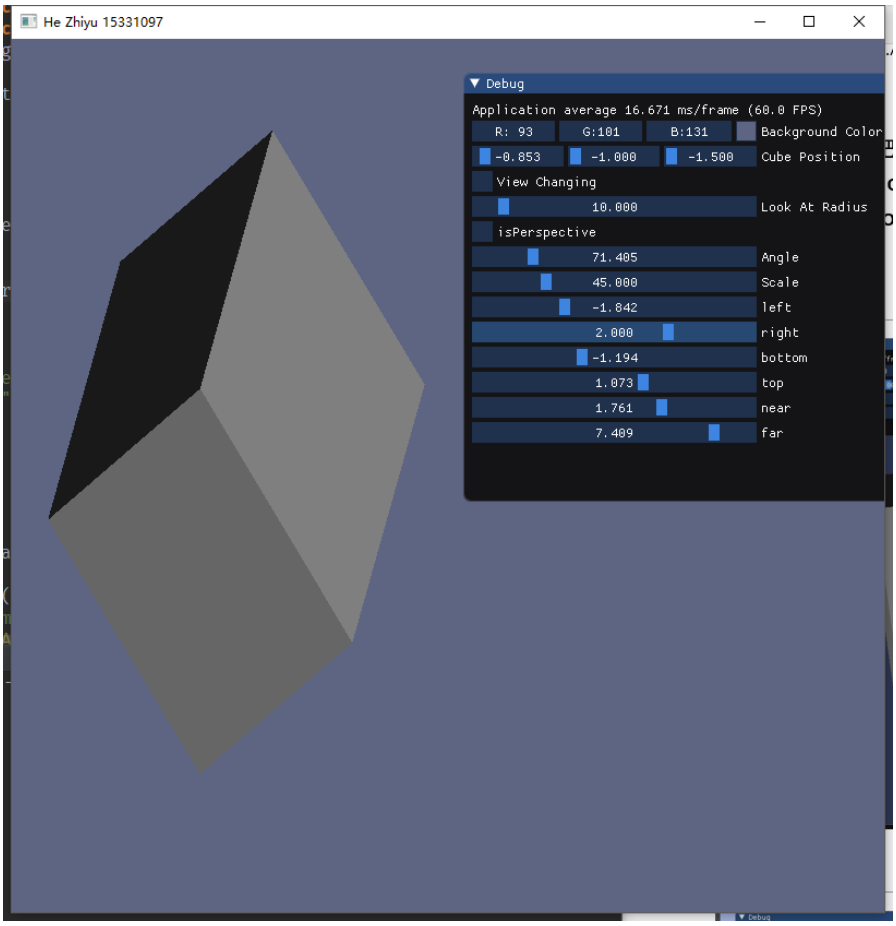
透视投影：



正交投影：



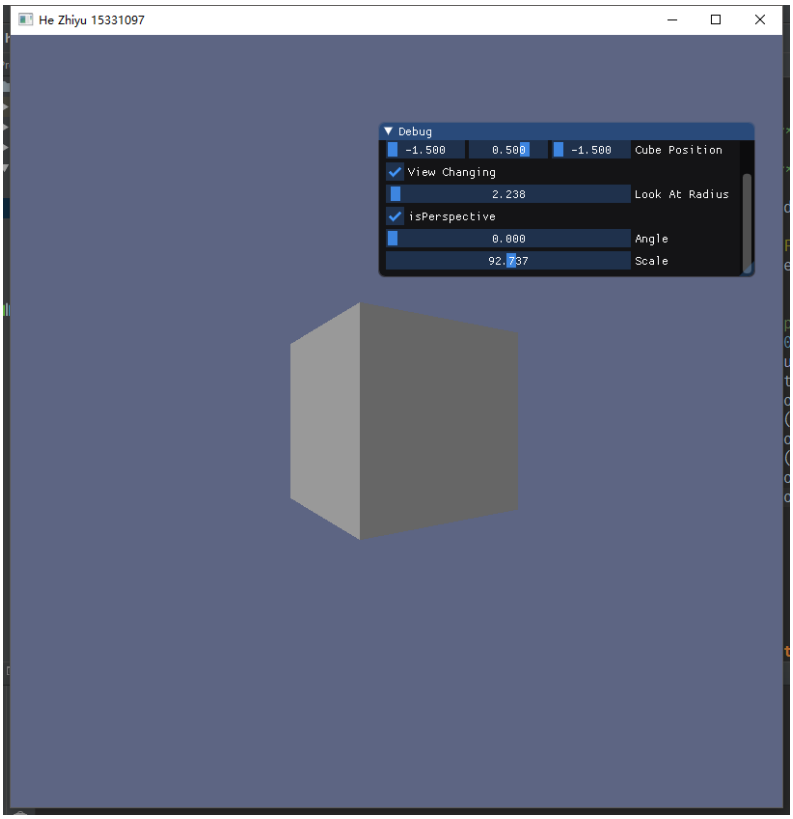
正交投影参数变化：



这些参数主要是影响正交投影坐标系的投影变化。比较直观。

视角变换(View Changing)：把 cube 放置在(0, 0, 0)处，做透视投影，使摄像机围绕 cube 旋转，并且时刻看着 cube 中心

透视投影变化：



实现思路

立方体绘制跟 hw4 一致。主要是摄像头的视图矩阵变换：

```
// transformations
glm::mat4 view;
if (isAnimate) {
    auto camX = static_cast<float>(sin(glfwGetTime()) * lookAtRadius);
    auto camZ = static_cast<float>(cos(glfwGetTime()) * lookAtRadius);
    view = glm::lookAt(glm::vec3(camX, 0.0f, camZ), glm::vec3(0.0f, 0.0f, 0.0f),
glm::vec3(0.0f, 1.0f, 0.0f));
} else {
    view = glm::translate(view, glm::vec3(0.0f, 1.0f, -5.0f));
}


glm::mat4 projection;
if (isPerspective) {
    projection = glm::perspective(glm::radians(scale), (float) SCREEN_WIDTH /
(float) SCREEN_HEIGHT, 0.1f,
                                  100.0f);
} else {
    projection = glm::ortho(left, right, bottom, top, nearArgs, farArgs);
}
```

以上是核心代码，主要是启用动画时，isAnimate == true，计算圆坐标方程，通过关于时间的正弦余弦函数来确定，使用 glm 中的 lookAt 的方法创造视图矩阵，摄像头的方向就是摄像头的位置指向原点位置。