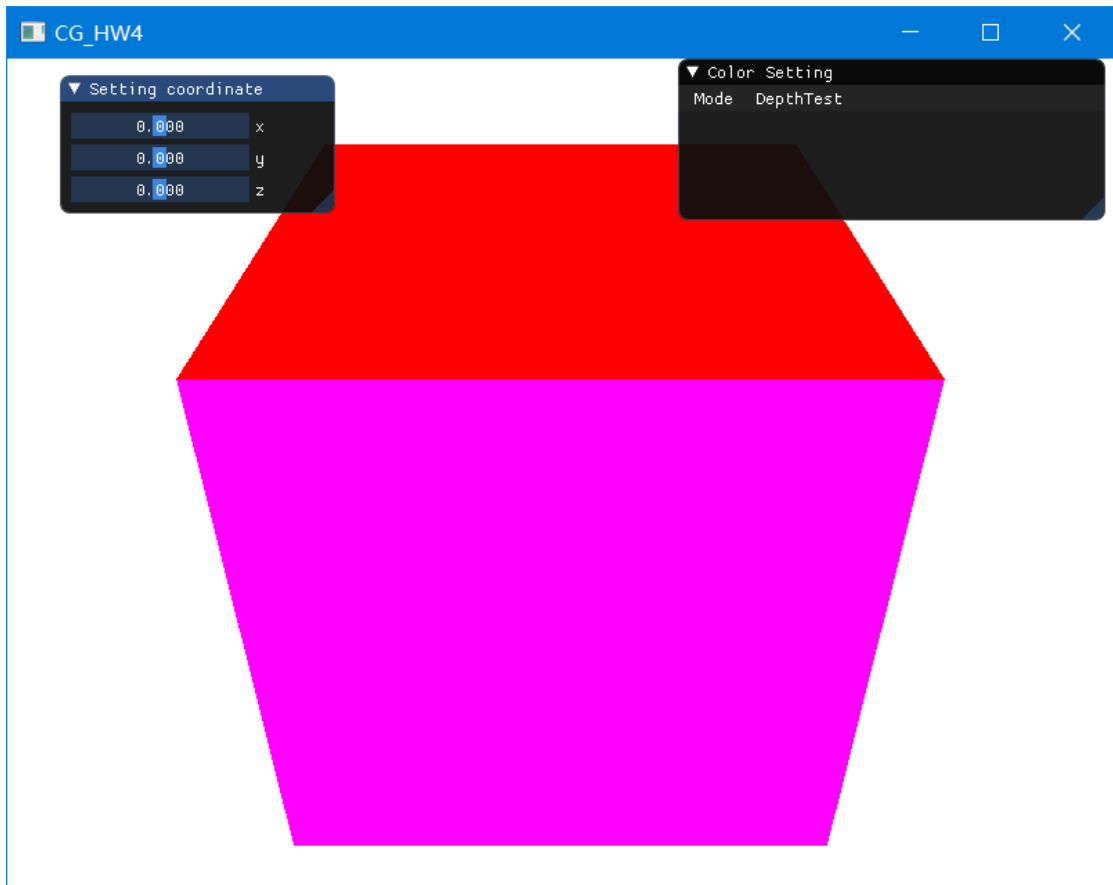
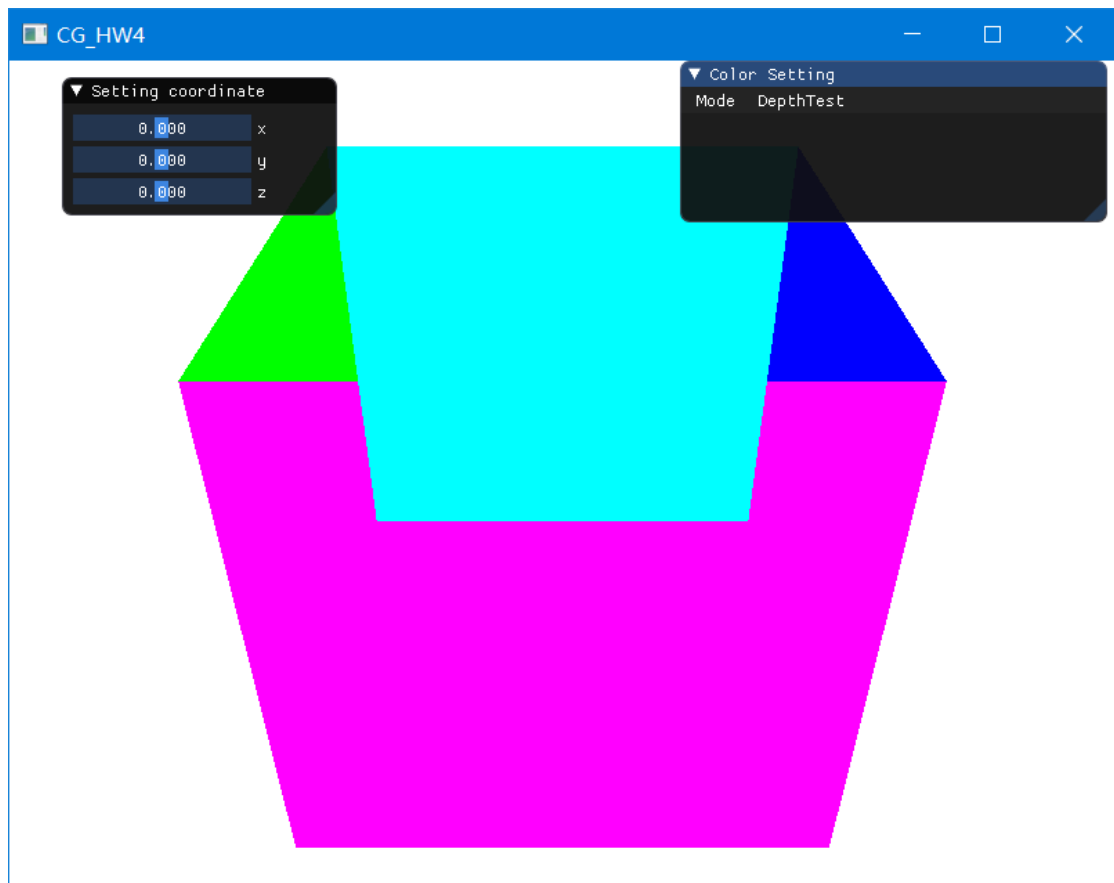


Howework 4

1. 画一个立方体(cube): 边长为 4, 中心位置为(0, 0, 0)。分别启动和关闭深度测试
`glEnable(GL_DEPTH_TEST)` 、 `glDisable(GL_DEPTH_TEST)` , 查看区别, 并分析原因。
启动深度测试:



关闭深度测试:

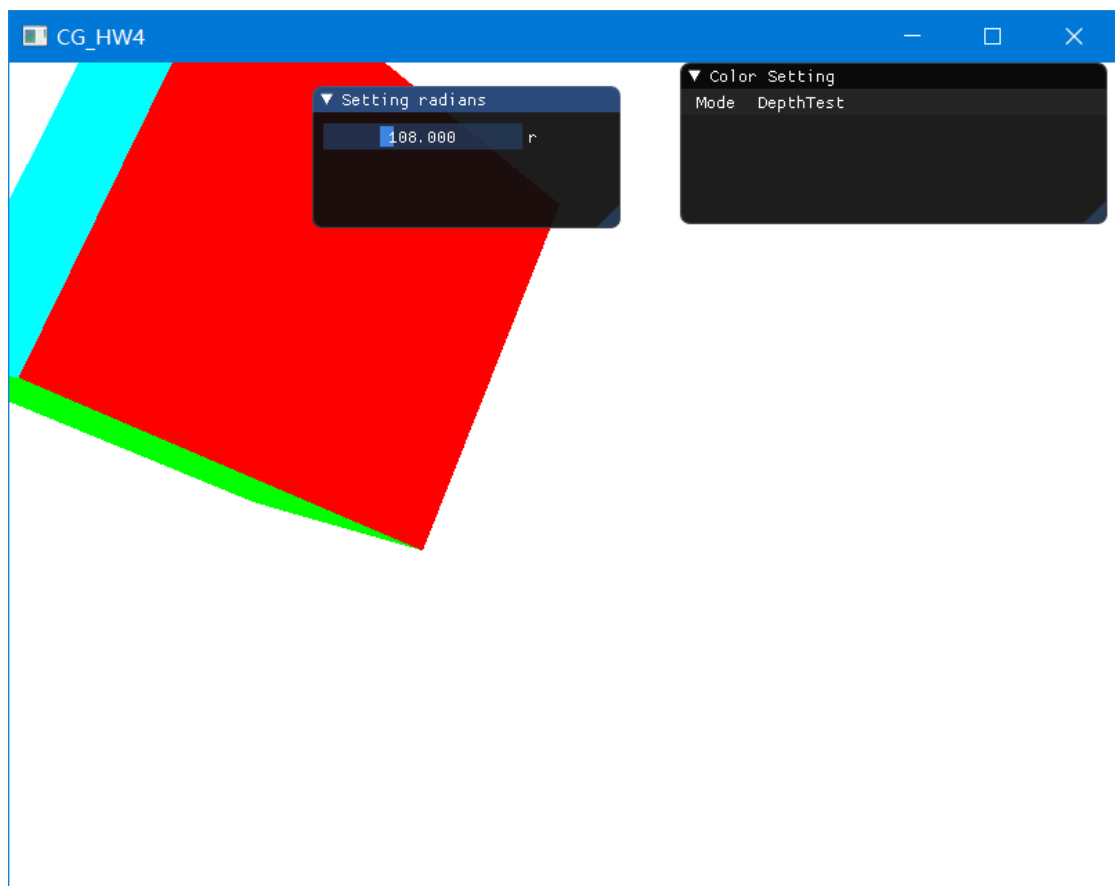
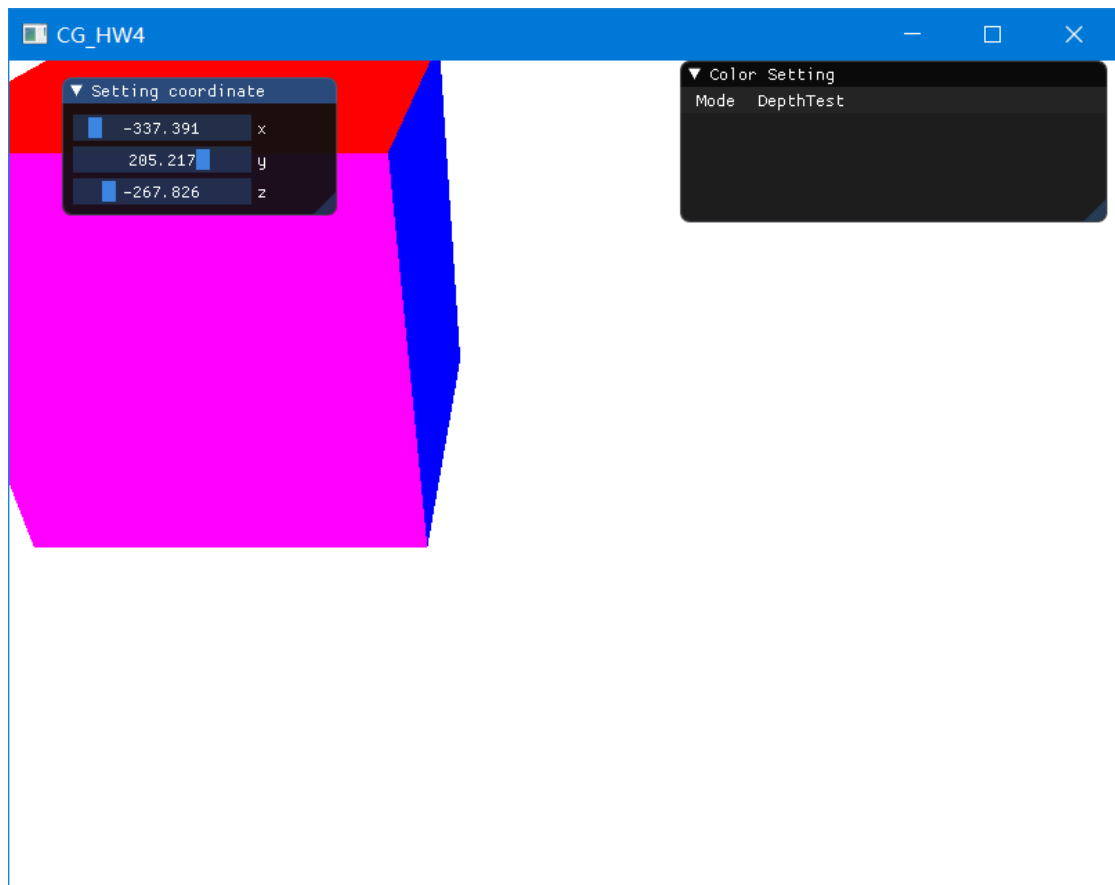


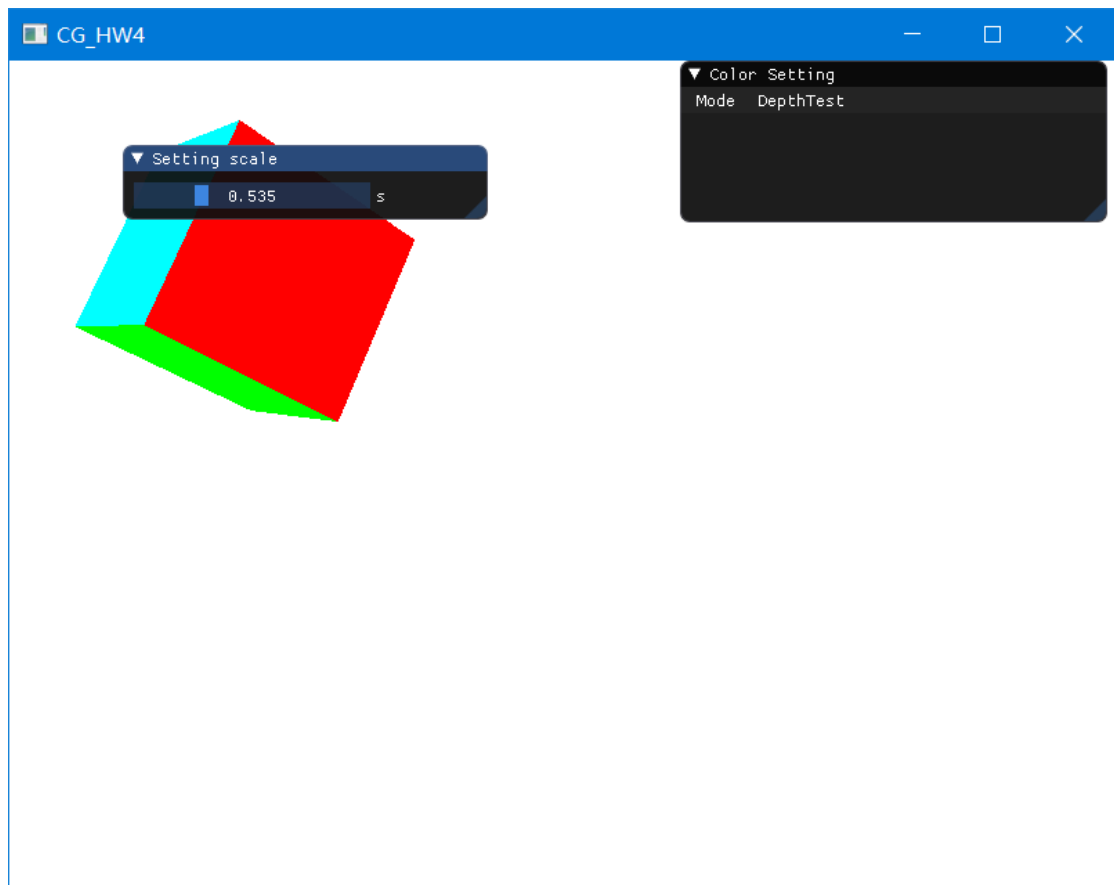
关闭深度测试后，相对于摄像机位于正方体背面的本应看不到的面覆盖在了应看到的面上。开启深度测试则正常显示。在不开启深度测试时，OpenGL 是一个三角形一个三角形地来绘制立方体的，即按照数组里点的顺序绘制。因此之后绘制的三角形将覆盖先前绘制的三角形。而开启深度测试之后，将对片段的深度值与 Z 缓冲的深度值来决定是否绘制以及更新 Z 缓冲，以此避免了不合理的图形覆盖。

2. 平移(Translation): 使画好的 cube 沿着水平或垂直方向来回移动。
3. 旋转(Rotation): 使画好的 cube 沿着 XoZ 平面的 x=z 轴持续旋转。
4. 放缩(Scaling): 使画好的 cube 持续放大缩小。
5. 在 GUI 里添加菜单栏，可以选择各种变换。

```
glm::mat4 model;  
model = glm::translate(model, glm::vec3(x/100, y/100, z/100));  
model = glm::rotate(model, glm::radians(-55.0f), glm::vec3(1.0f, 0.0f, 0.0f));  
model = glm::rotate(model, glm::radians(r), glm::vec3(1.0f, 0.0f, 1.0f));  
model = glm::scale(model, glm::vec3(s, s, s));
```

三个物体行为可以用 3 个函数完成。由于矩阵相乘不符合交换律，因此为了不产生不利影响，先进行缩放操作，然后是旋转，最后是位移。而在代码中则是相反的顺序读取函数（如同矩阵相乘从右开始读一样）。此处关于题目旋转的理解我是绕自身自转。假如是绕世界空间的轴旋转的公转的话，则需要进行变换，先将自转轴与物体同时旋转到与某一坐标轴重合后，执行公转，再逆向返回最初自转轴位置。





6. 结合 Shader 谈谈对渲染管线的理解

渲染管线是一系列变化处理的组合，它的任务是将一系列 3D 坐标的信息转换成 2D 像素再在屏幕上显示。Shader 则是渲染管线的部分组成部分。部分 Shader 是可编程的，于是可以通过对 Shader 进行设置已获得希望的显示效果，如在 Shader 中实现对坐标、颜色的变换等。因为图形必须经过渲染管线来绘制，因此通过对渲染管线的操作（指对 Shader 内部的设置）进行变换来对整个图形进行修饰将更容易，消耗也更低，因为我们不需要对每个顶点在 CPU 进行操作，而是直接在管线中通过时统一利用 GPU 操作。