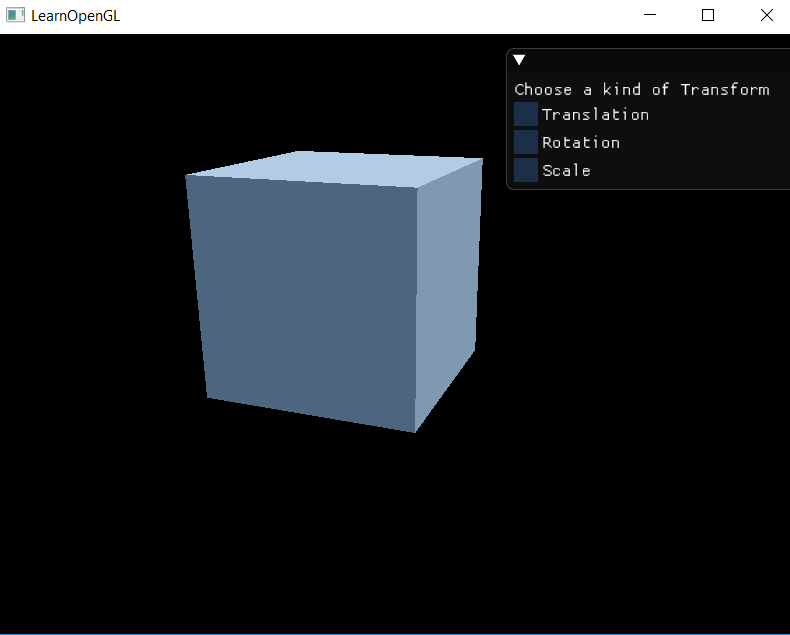
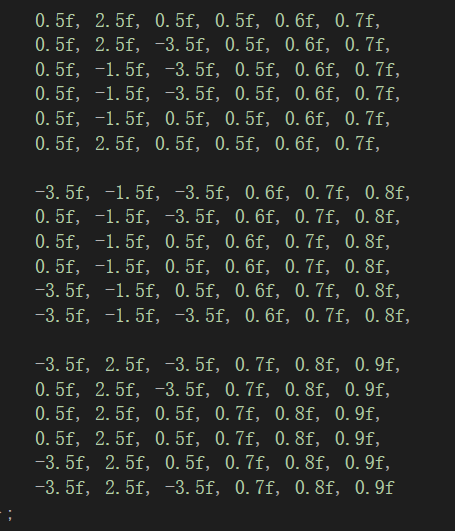
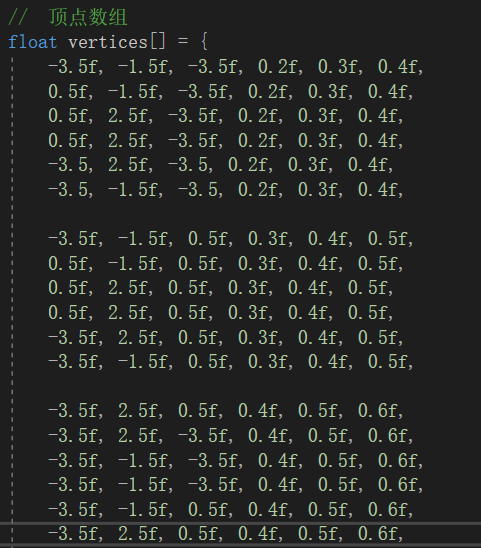
2. **将cube放置在(-1.5, 0.5, -1.5)位置**



**实现思路:**

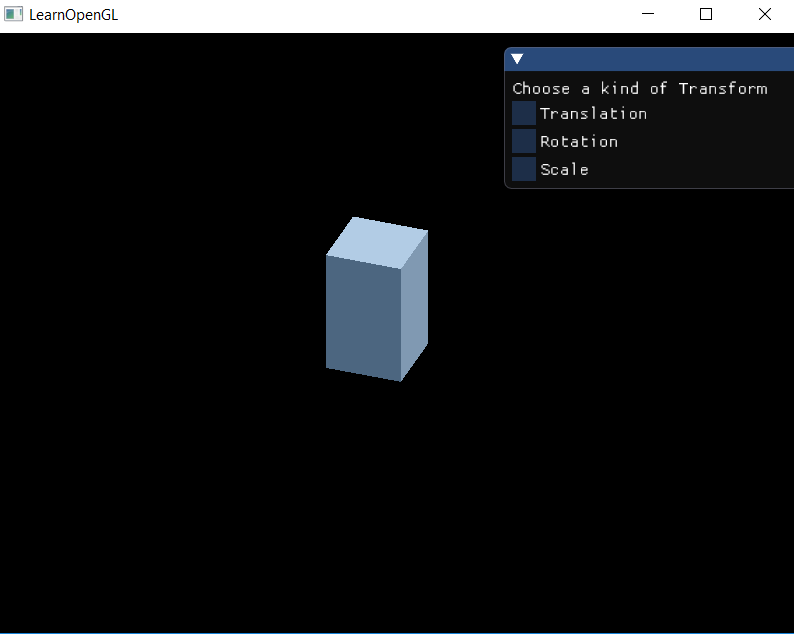
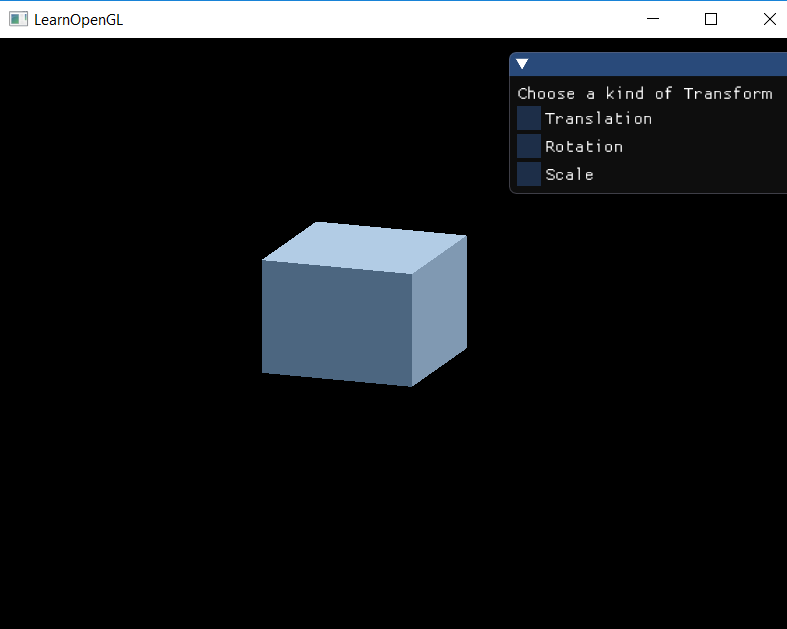
物体中心默认放在(0, 0, 0)的位置, 将顶点数组中每个顶点都加上(-1.5, 0.5, -1.5)即可将物体中心移动到(-1.5, 0.5, -1.5)



1. **正交投影**

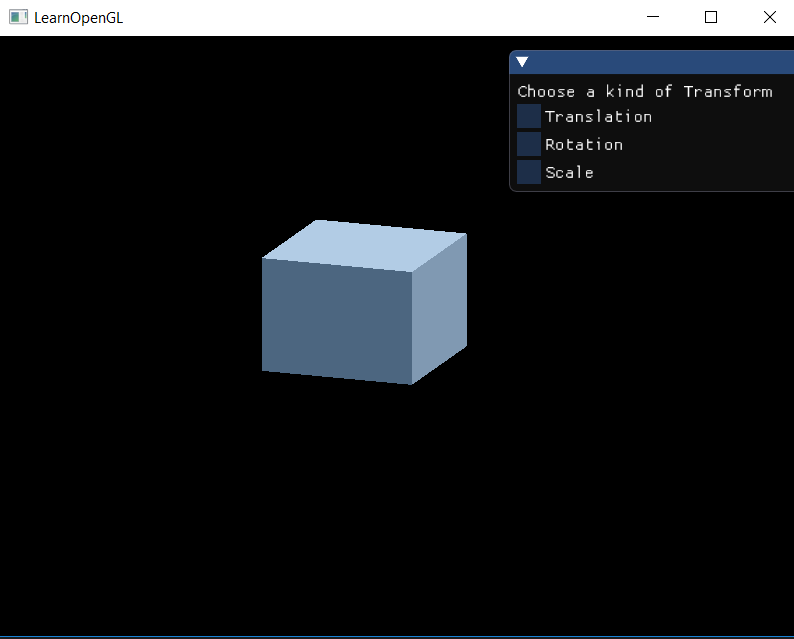
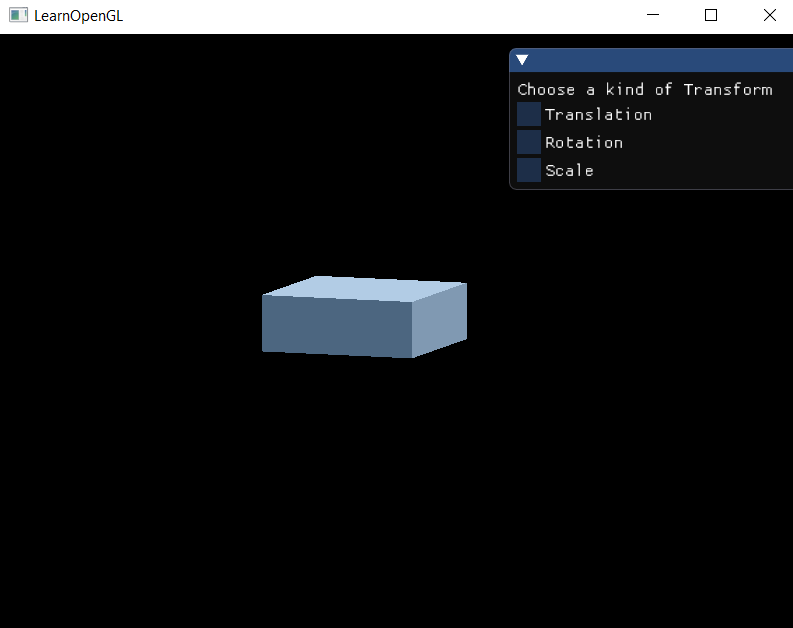
**平截头体的宽:**

**-10~10 -20~20**



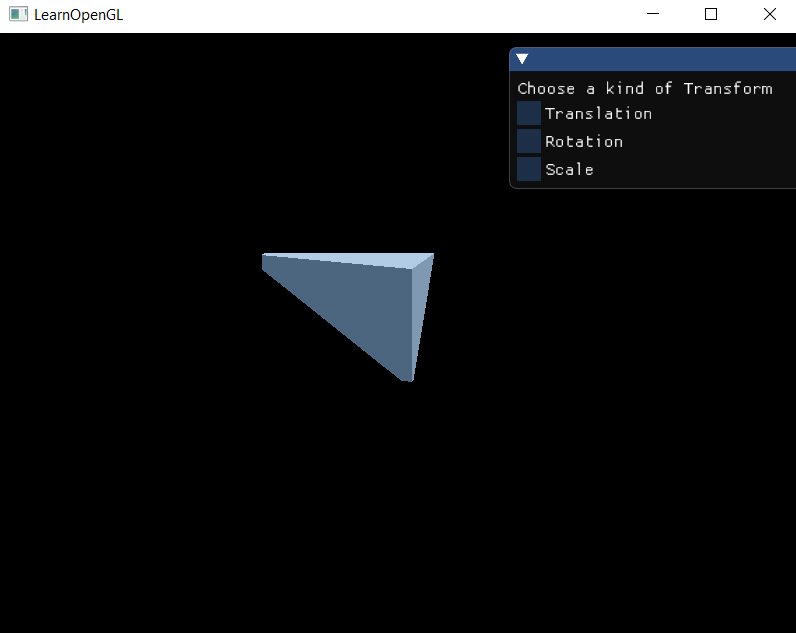
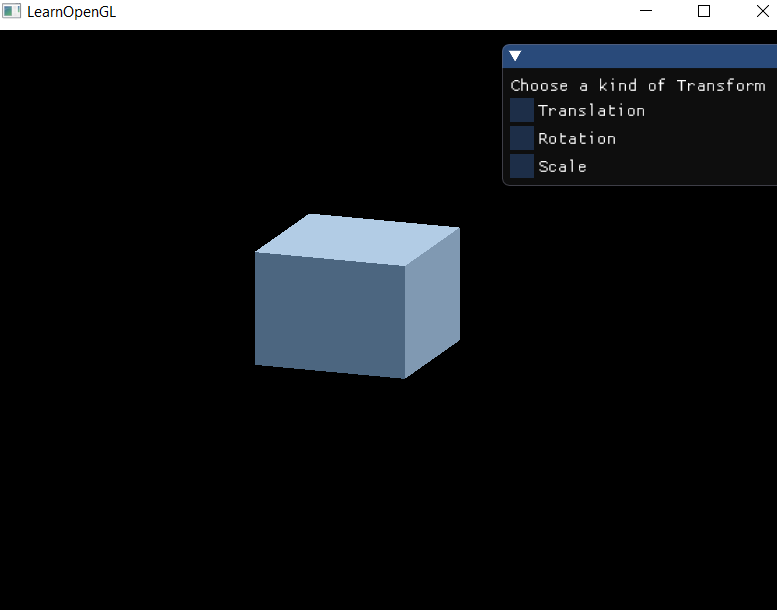
**平截头体的高:**

**-10~10 -20~20**

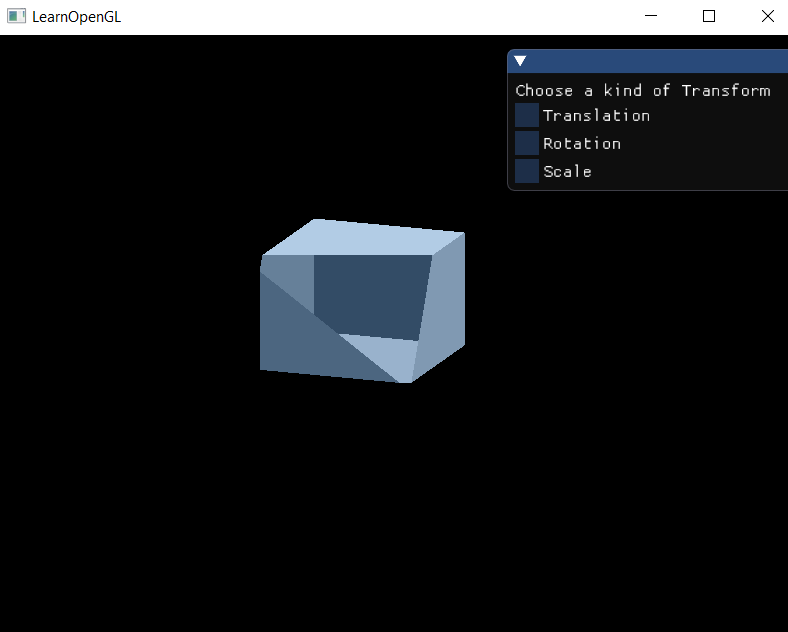
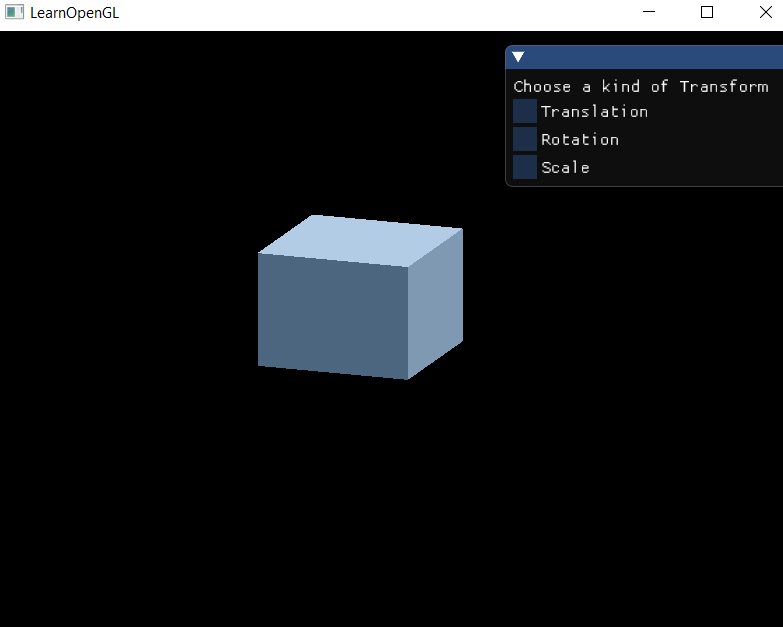
**平行头体的远平面:**

**100.0f 20.f**



**平行头体的近平面:**

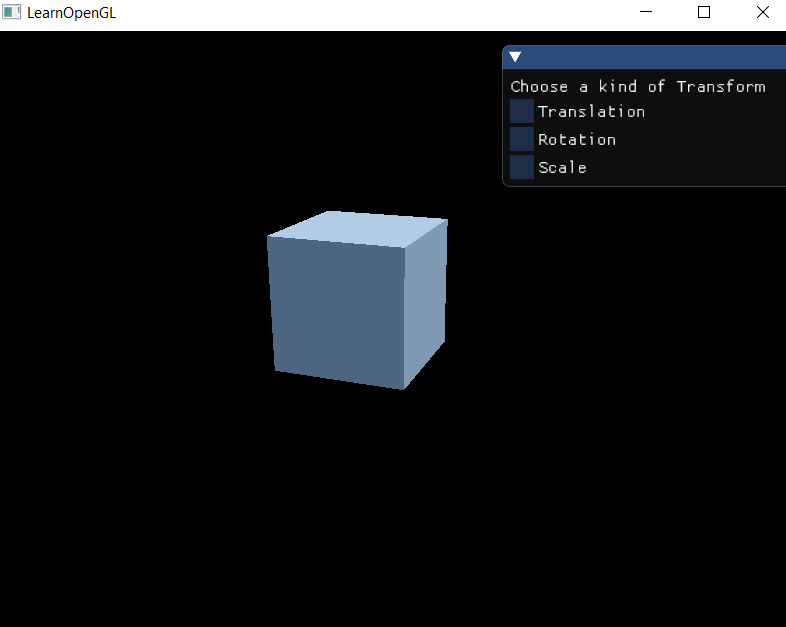
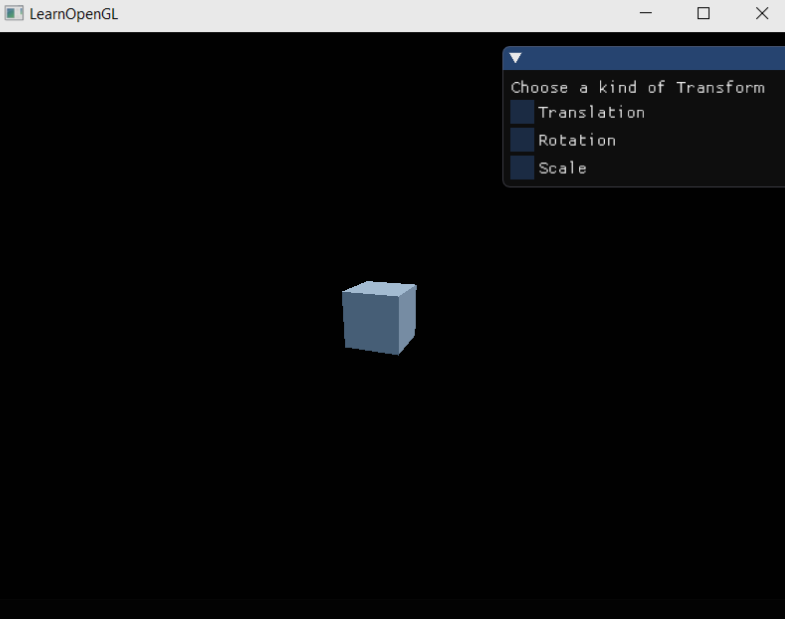
**0.1f 20.f**



1. **透视投影**

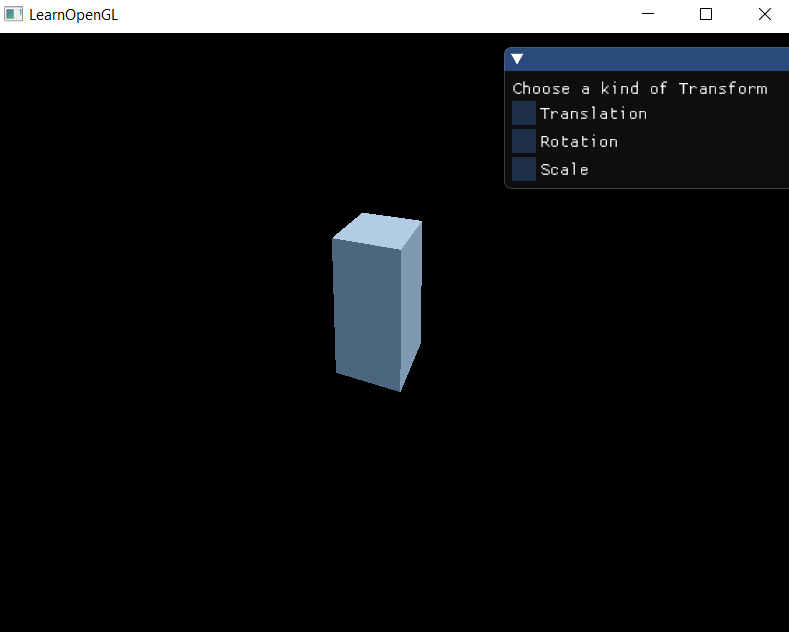
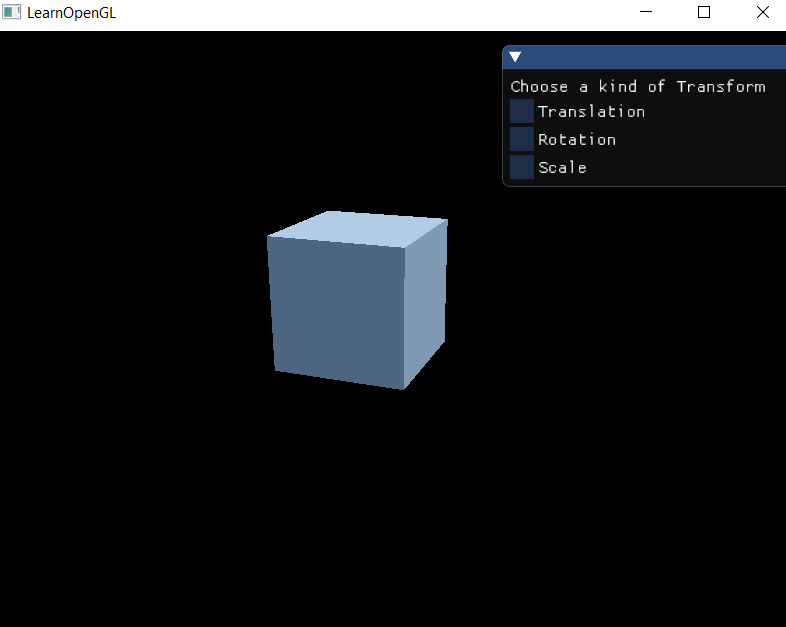
**观察空间大小**

**45° 90°**

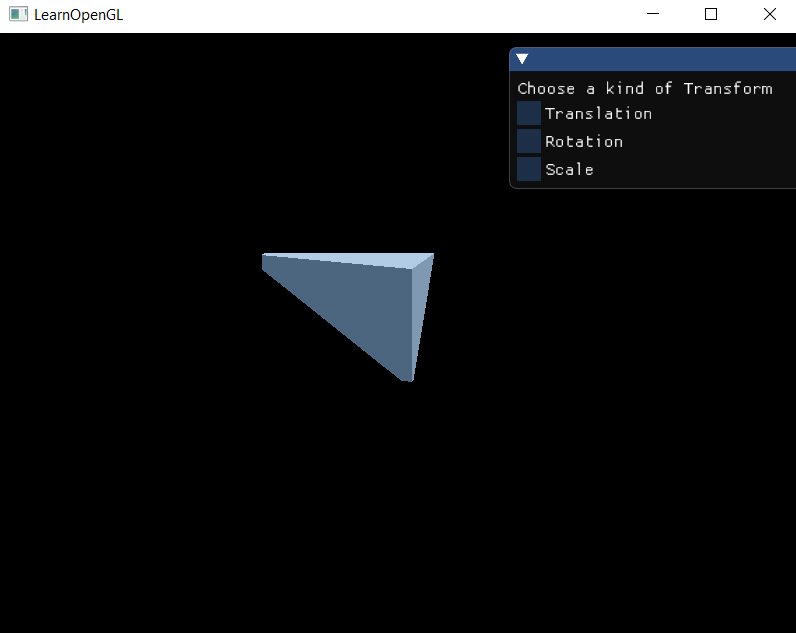
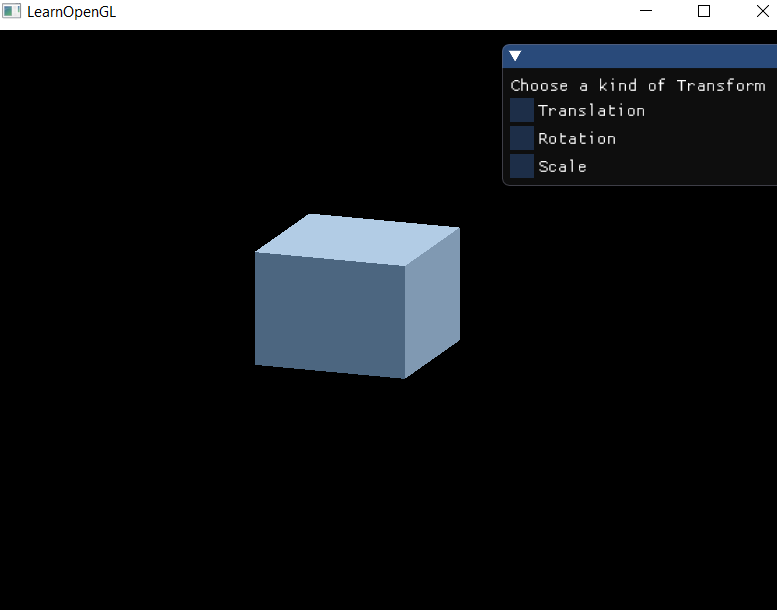
**宽高比**

**屏幕宽/屏幕高 2\*屏幕宽/屏幕高**



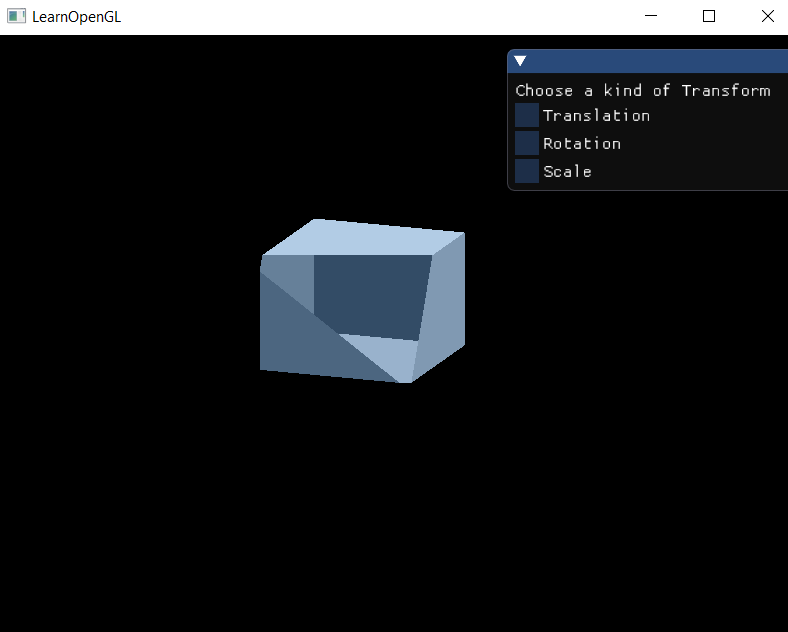
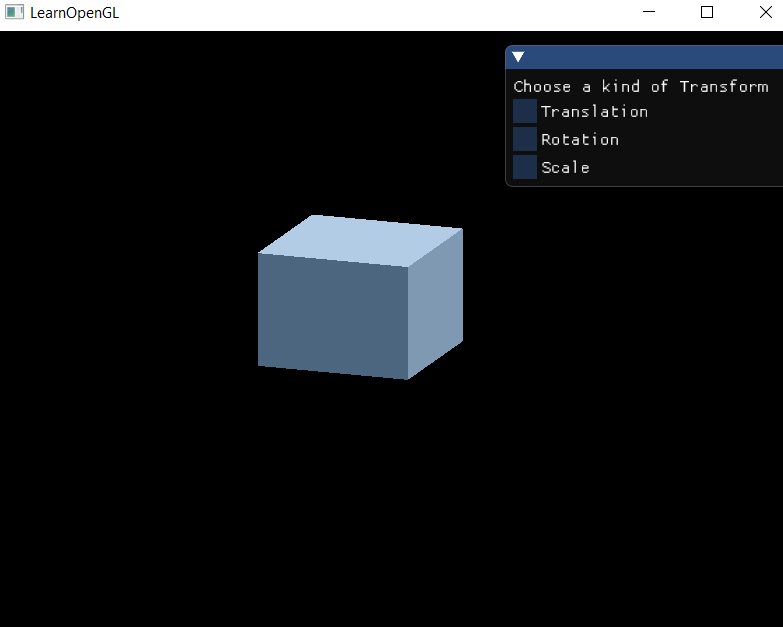
**平行头体的远平面:**

**100.0f 20.f**



**平行头体的近平面:**

**0.1f 20.f**





**见视频2**

**实现原理:**

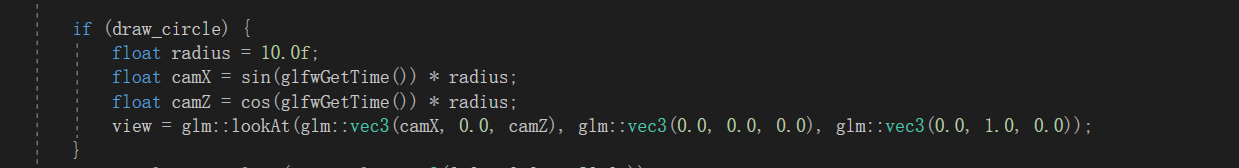


将相机绕着cube旋转, 等价于让相机位置不动, 物体旋转



**见视频3**

**实现原理:**



1. LookAt矩阵的作用是将任意向量转换为相机坐标空间下的向量.
2. 创建LookAt矩阵的参数分别是: 相机位置, 目标位置, 定义相机空间x轴正向用到的up向量
3. 因为(sinx \* r)^2 + (cosx^2 \* r) = r^2, 所以相机的位置会随着时间变化而在XOZ平面上画圆, 呈现给我们的效果就是相机不动, 物体围绕(0, 0, 0)画圆

**Question:**

**在现实生活中，我们一般将摄像机摆放的空间View matrix和被拍摄的物体摆设的空间Model matrix分开，但 是在OpenGL中却将两个合二为一设为ModelView matrix，通过上面的作业启发，你认为是为什么呢？在报 告中写入。（Hints：你可能有不止一个摄像机）**

因为, 在OpenGL中, 为了让物体看起来”运动”, 我们一般采用固定物体, 运动相机的方法. 所以, 当有多个相机时, 因为每个相机相对物体的角度、距离等因素不同, 当为了使物体看起来”运动”, 不同相机必须根据它与物体之间的实际情况, 采用不同的运动方式, 因此会将Model和View结合起来.

而在现实生活中, 多个相机是不动的, 只需要运动一个物体, 因此将Model和View分离开.