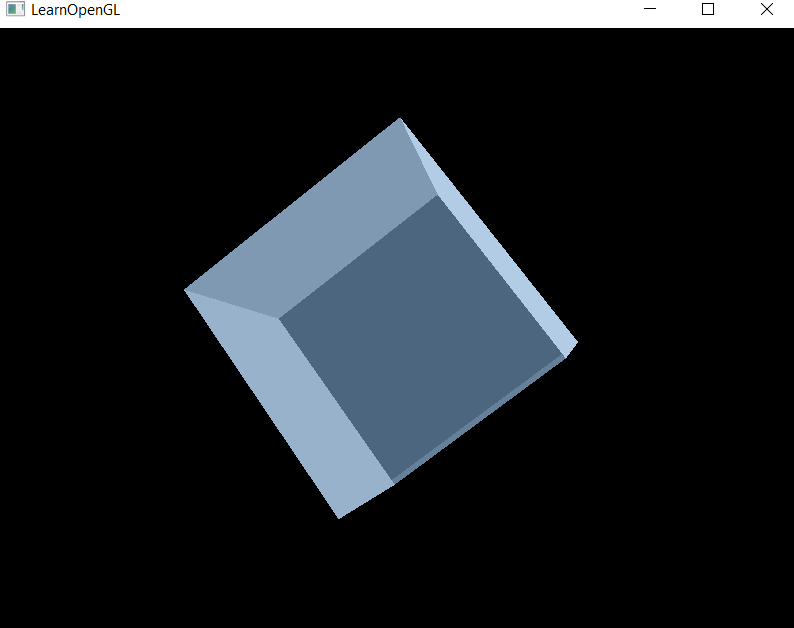
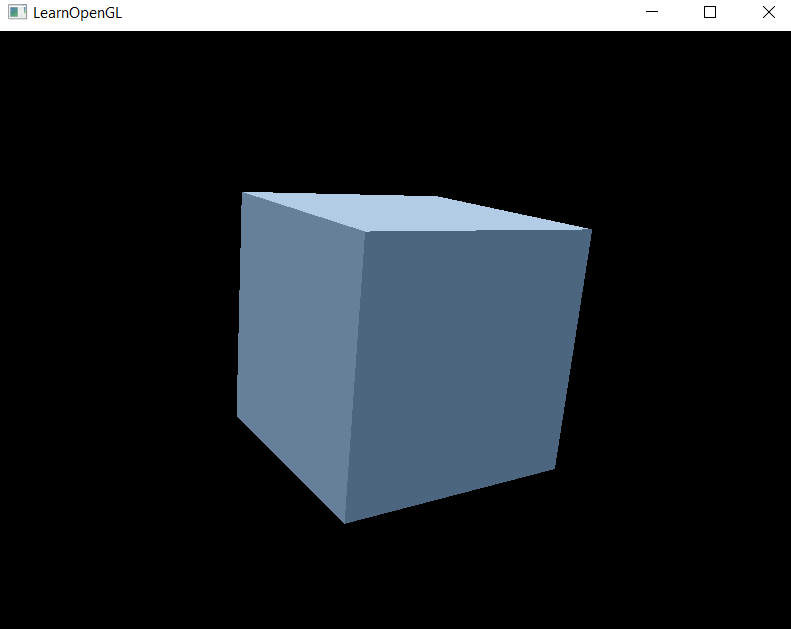
**关闭深度测试**

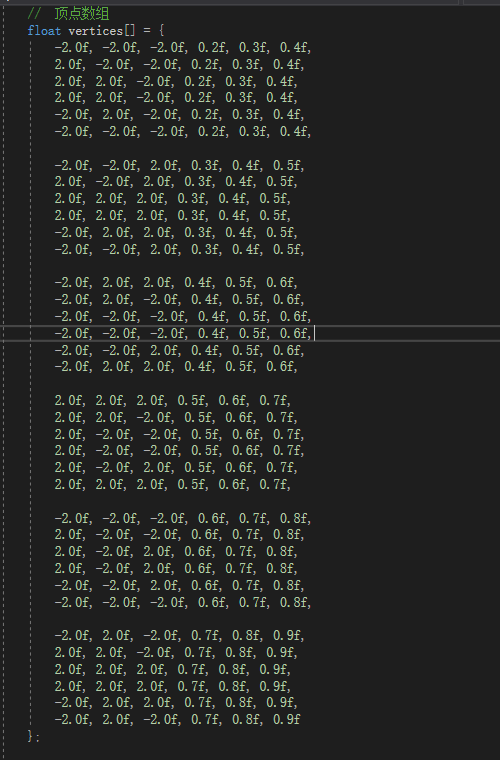


**开启深度测试**

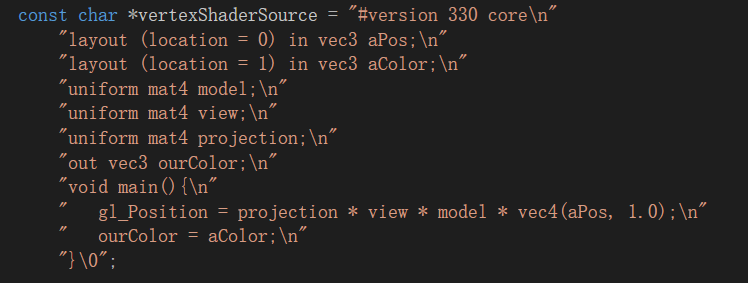


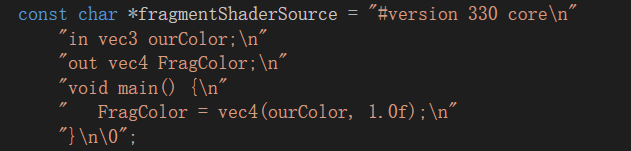
**实现思路:**

1. 立方体有6个面, 每个面2个三角形, 一个三角形三个顶点, 因此创建一个含有36个顶点的数组. 同时, 为了深度测试, 给每个面一个单独的颜色

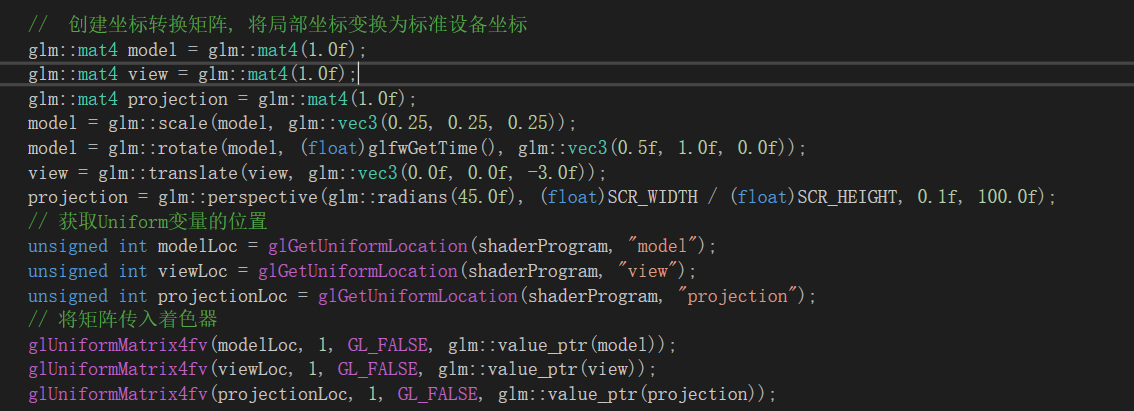


1. **编写顶点着色器和片段着色器, 并编译、链接**

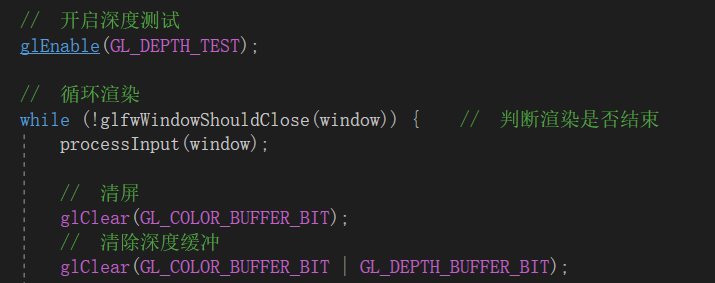




1. **创建从局部空间坐标->标准化设备坐标的转换矩阵, 并传入Shader中**



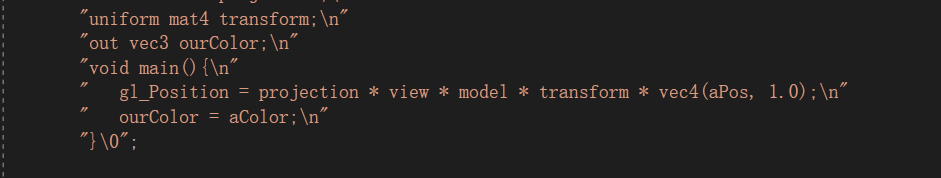
1. **开启深度测试, 并在每次循环渲染中清除深度缓冲**



1. **平移、旋转、放缩(结果见视频)**

**实现思路:**

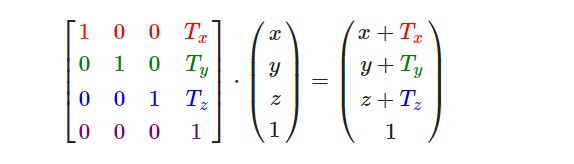
1. **在顶点着色器中定义transform矩阵, 并与顶点相乘**



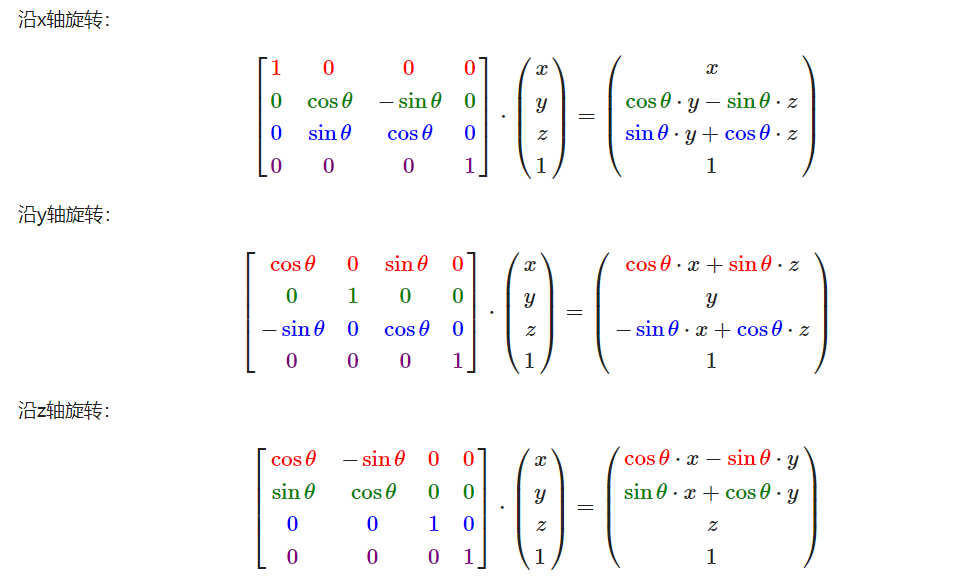
1. **根据ImGui中选择的变换类型, 给顶点着色器传入对应的矩阵**



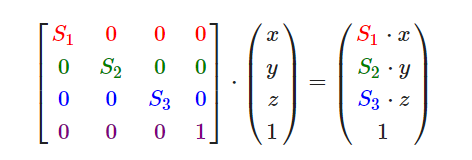
**对于平移, 变换矩阵为：**



**对于旋转, 变换矩阵为：**



**对于缩放, 变换矩阵为：**



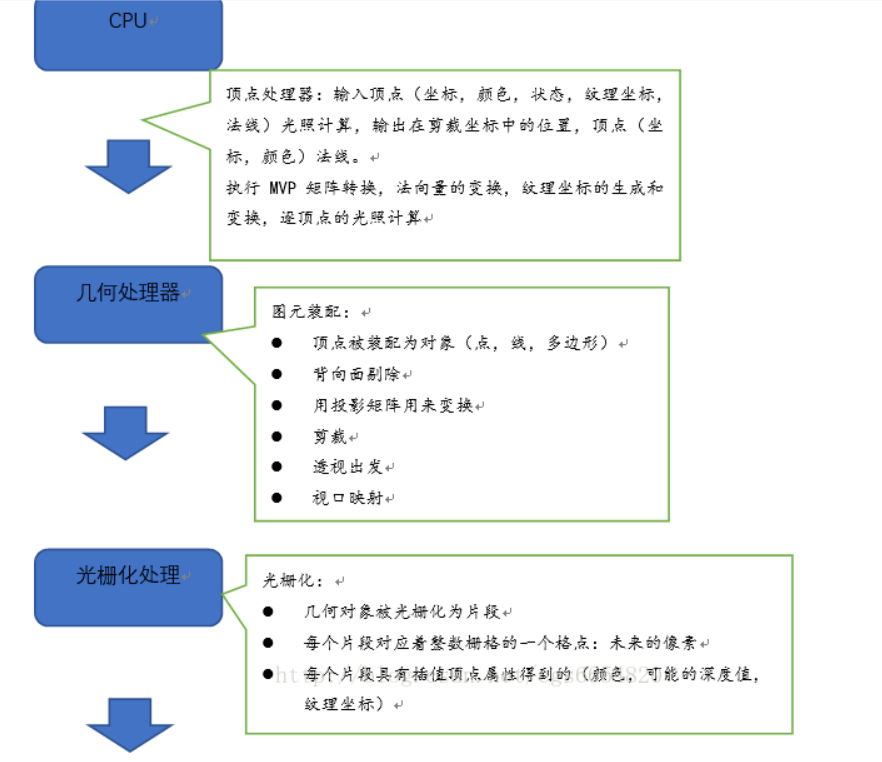
1. **结合Shader谈谈对渲染管线的理解**
2. **Shader:**

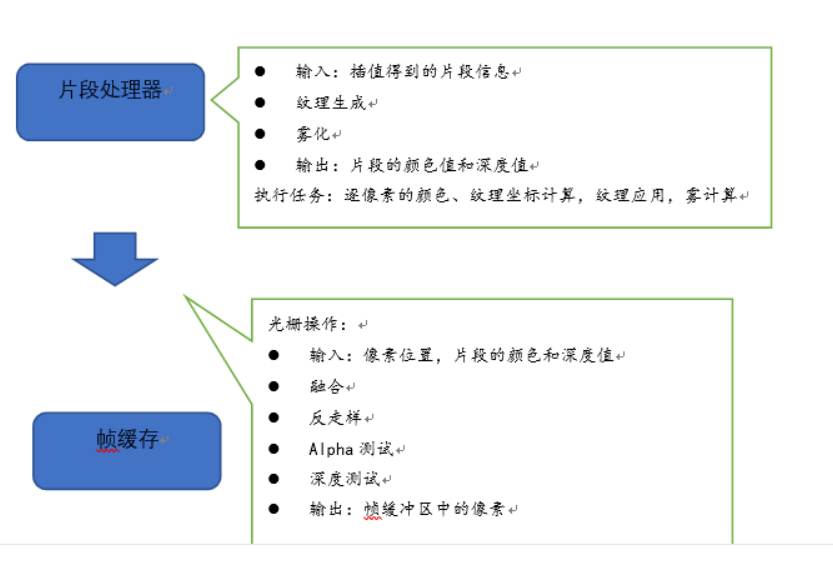
**即着色器. 它是一种用于实现图像渲染, 运行在GPU上的程序片段, 用来代替以往固定的渲染管线, 这样就可以方便地对GPU进行控制.**

**现在一般使用GLSL语言(OpenGL Shading Language)编写Shader程序.**

1. **渲染管线:**

**即实现图像渲染过程中执行的一系列流水线操作. 包括顶点着色器、图元装配、几何着色器、光栅化、片段着色器等步骤.**





1. **总结**

**Shader是具体的程序; 渲染管线是抽象的流程描述. Shader中的顶点着色器和片段着色器是渲染管线的一道流程**

1. **Bonus(结果见视频)**

**实现思路: 将三种变换矩阵相乘, 实现三种变换相结合**