



逻辑教育
Logic education

Hello CC

OpenGL 主题 141

视觉班—OpenGL 向量.矩阵应用

课程研发:CC老师

课程授课:CC老师

转载需注明出处,不得用于商业用途.已申请版权保护



逻辑教育
Logic education

课堂目标:

1. 向量与矩阵
2. 如果使用矩阵/向量来移动几何图形
3. OpenGL 对于模型视图/投影矩阵的约定
4. 点光源存储着色器使用

课程研发:CC老师

课程授课:CC老师



逻辑教育
Logic education

课后作业:

1. 请根据今天的课程内容,在个人博客上更新一篇博文

要求:

01.将课程内容加上自己的理解

02.更新的博客地址发送到讨论群.互相学习

课程研发:CC老师

课程授课:CC老师



3D数学.在图像图形开发中的充当了什么角色?

对于学习OpenGL 有一个误区, 就是大家认为如果不能精通那些3D图形数学知识, 会让我们寸步难行, 其实不然。就像我们不需要懂得任何关于汽车结构和内燃机方面的知识也能每天开车。但是, 我们最好能对汽车有足够的了解, 以便我们意识到什么时候需要更换机油、定期加油、汽车常规保养工作。同样要成为一名可靠和有能力的OpenGL程序员, 至少需要理解这些基础知识, 才知道能作什么? 以及那些工具适合我们要做的工作。初学者, 经过一段时间的实践, 就会渐渐理解矩阵和向量。并且培养出一种更为直观的能力, 能够在实践中充分利用所学的内容。

课程研发:CC老师

课程授课:CC老师



即使大家现在还没有能力在脑海中默算出2个矩阵的乘法、也要明白矩阵是什么？以及这些矩阵对OpenGL意味着什么？

GLTools 库中有一个组件叫Math3d, 其中包含了大量好用的OpenGL 一致的3D数学和数据类型。虽然我们不必亲自进行所有的矩阵和向量的操作，但我然知道它们是什么？以及如何运用它们。

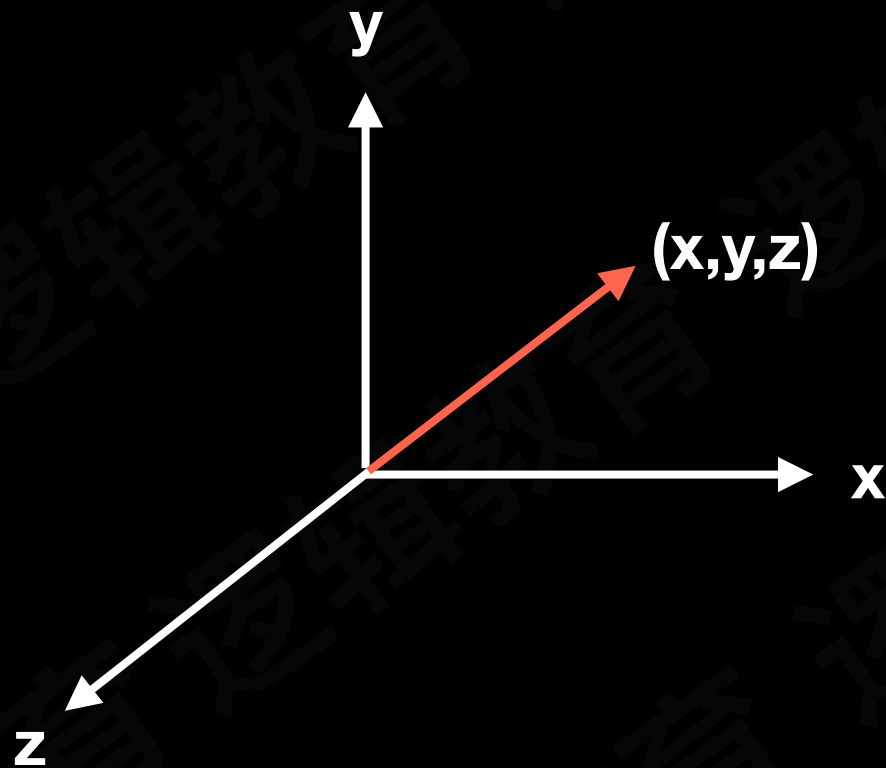
关于3D数学，是大家触碰OpenGL、甚至你们接触的AR Kit 框架以及Unity3D、游戏开发都是必须学习的一块

在开发过程我们涉及到的图形变换,就会涉及到矩阵/向量的计算.例如大家在使用CAnimation 实现仿射变换,就使用了OpenGL渲染技术.



OpenGL里的矩阵/向量使用

- 3个值 (x 、 y 、 z) 组合起来表示2个重要的值，方向和数量



课程研发:CC老师

课程授课:CC老师



OpenGL math3d库

- `math3d`库，有2个数据类型，能够表示一个三维或者四维向量。`M3DVector3f`可以表示一个三维向量 (x, y, z) ，而`M3DVector4f`则可以表示一个四维向量 (x, y, z, w) 。在典型情况下， w 坐标设为1.0。 x, y, z 值通过除以 w ，来进行缩放。而除以1.0则本质上不改变 x, y, z 值。

```
typedef float M3DVector3f[3];  
typedef float M3DVector4f[4];
```

声明一个三分量向量操作：

```
M3DVector3f vVector;
```

类似，声明一个四分量的操作：

```
M3DVector4f vVectro= {0.0f,0.0f,1.0f,1.0f};
```

声明一个三分量顶点数组，例如生成一个三角形

```
M3DVector3f vVerts[] = {  
    -0.5f,0.0f,0.0f,  
    0.5f,0.0f,0.0f,  
    0.0f,0.5f,0.0f  
};
```

课程研发:CC老师

课程授课:CC老师



向量/矩阵 点乘



//实现点乘方法:

//方法1: 返回的是-1, 1之间的值。它代表这个2个向量的余弦值。

```
float m3dDotProduct3(const M3DVector3f u,const  
M3DVector3f v);
```

//方法2: 返回2个向量之间的弧度值。

```
float m3dGetAngleBetweenVector3(const M3DVector3f  
u,const M3DVector3f v);
```

•

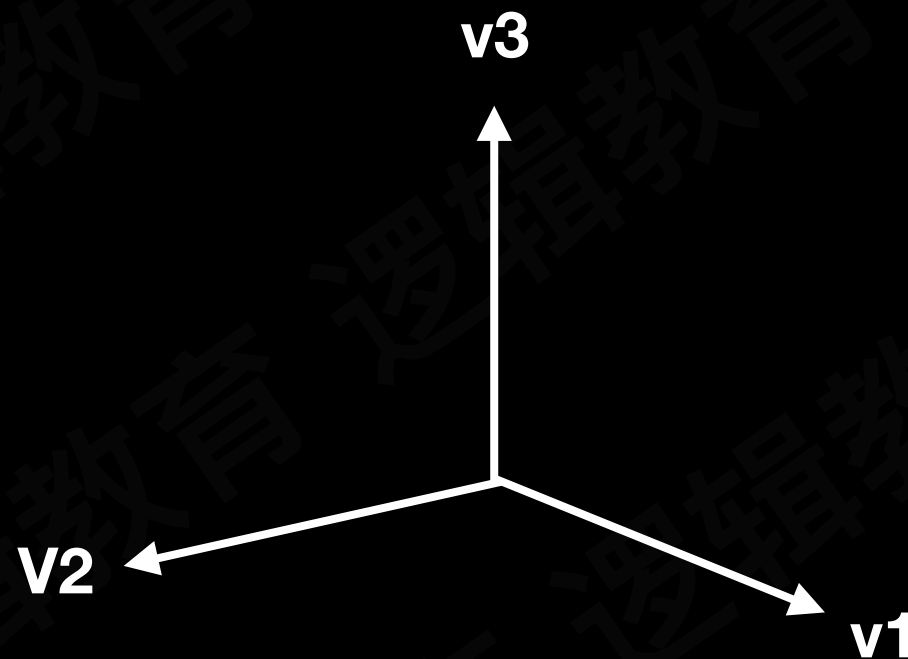
课程研发:CC老师

课程授课:CC老师



逻辑教育
Logic education

向量/矩阵 叉乘



```
void m3dCrossProduct3(M3DVector3f result,const M3DVector3f  
u ,const M3DVector3f v);
```

课程研发:CC老师
课程授课:CC老师



OpenGL 下的矩阵

```
typedef float M3DMatrix33f[9];  
typedef float M3DMatrix44f[16];
```

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \quad \begin{bmatrix} 0 & 42 \\ 1.5 & 0.877 \\ 2 & 14 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

三个矩阵



理解变化

变换	应用
视图	指定观察者位置
模型	在场景中移动物体
模型视图	描述视图/模型变换的二元性
投影	改变视景物大小和设置它的投影方式
视口	伪变化,对窗口上最终输出进行缩放

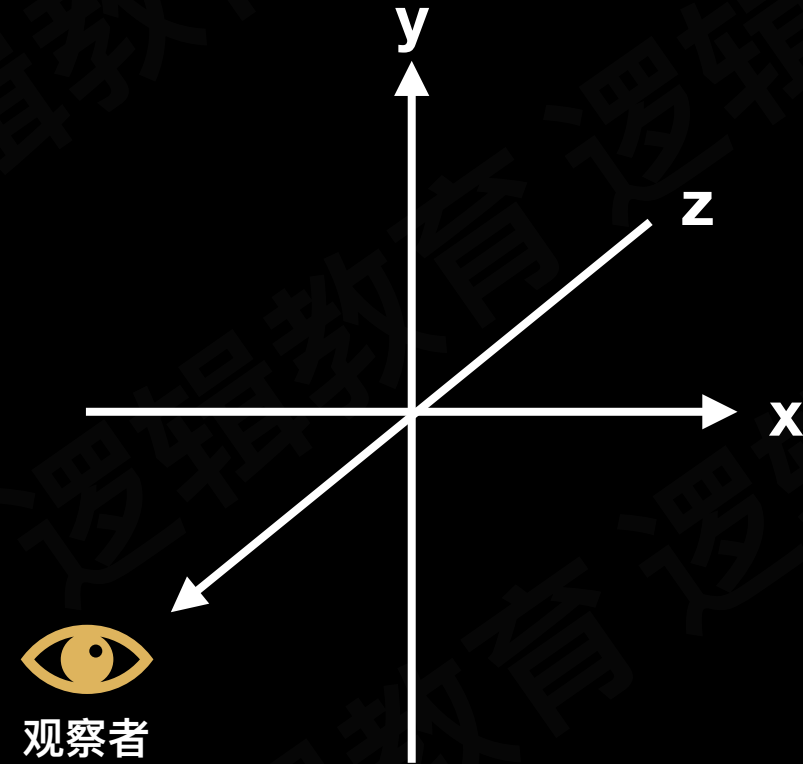
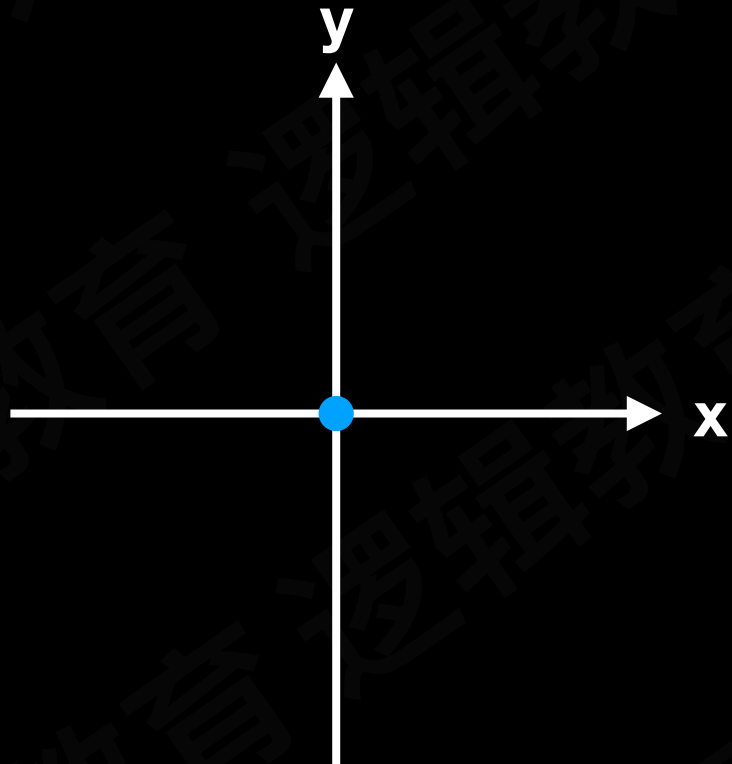
课程研发:CC老师

课程授课:CC老师



逻辑教育
Logic education

视觉坐标



课程研发:CC老师
课程授课:CC老师



逻辑教育
Logic education

课程研发:CC老师
课程授课:CC老师

转载需注明出处,不得用于商业用途.已申请版权保护