



逻辑教育
Logic education

Hello CC

OpenGL 主题 [5]

视觉班—OpenGL 基础变化

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

课程目标:

- 矩阵绘图
- 模型视图矩阵
- 三角形批次类绘制图形
- 使用照相机和角色帧 进行移动

课程研发:CC老师

课程授课:CC老师

课程研发:CC老师

课程授课:CC老师

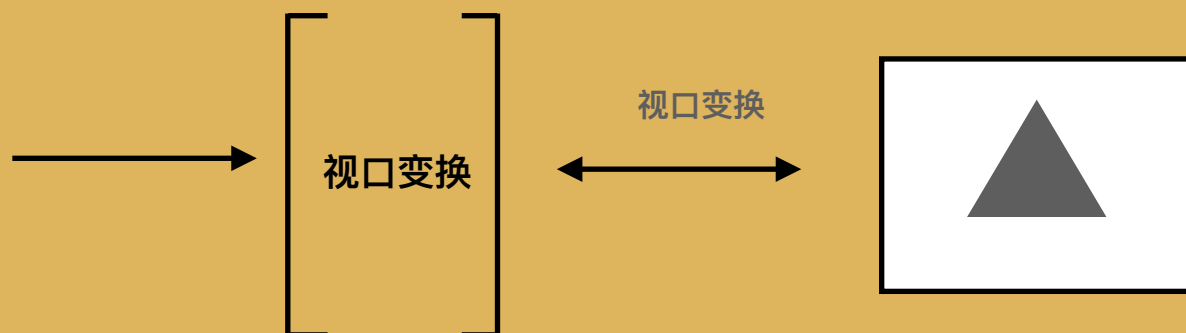
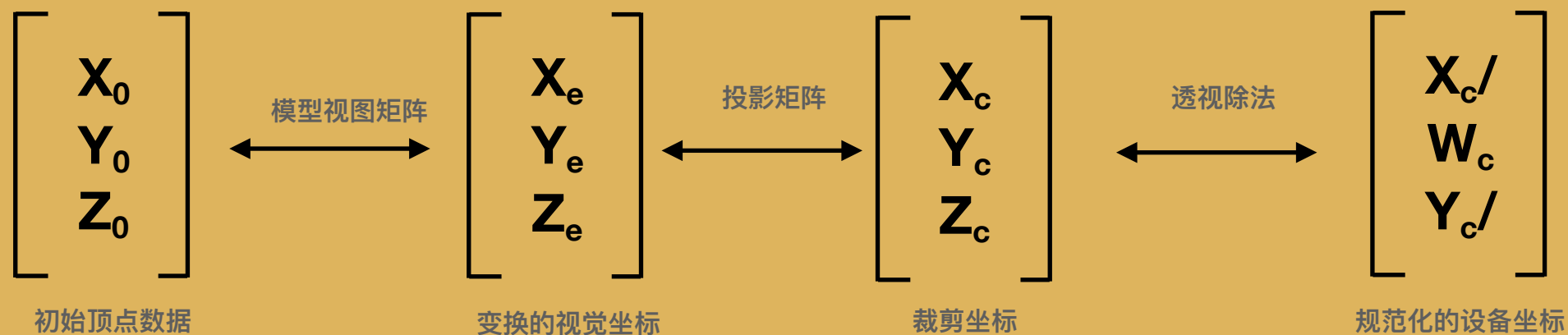
转载需注明出处,不得用于商业用途,已申请版权保护
转载分享需备注出处,不得用于商业用途



逻辑教育
Logic education

一. 顶点变换管线

顶点变换管线



课程研发:CC老师
课程授课:CC老师



1.2 使用矩阵堆栈

//类型

```
GLMatrixStack::GLMatrixStack(int iStackSize = 64);
```

//在堆栈顶部载入一个单元矩阵

```
void GLMatrixStack::LoadIdentity(void);
```

//在堆栈顶部载入任何矩阵

//参数: 4*4矩阵

```
void GLMatrixStack::LoadMatrix(const M3DMatrix44f m);
```

//矩阵乘以矩阵堆栈顶部矩阵, 相乘结果存储到堆栈的顶部

```
void GLMatrixStack::MultMatrix(const M3DMatrix44f);
```

//获取矩阵堆栈顶部的值 GetMatrix 函数

//为了适应GLShaderManager的使用, 或者获取顶部矩阵的副本

```
const M3DMatrix44f & GLMatrixStack::GetMatrix(void);
```

```
void GLMatrixStack::GetMatrix(M3DMatrix44f mMatrix);
```

课程研发:CC老师

课程授课:CC老师



1.3 压栈.出栈

//将当前矩阵压入堆栈(栈顶矩阵copy 一份到栈顶)

void GLMatrixStack::PushMatrix(void);

//将M3DMatrix44f 矩阵对象压入当前矩阵堆栈

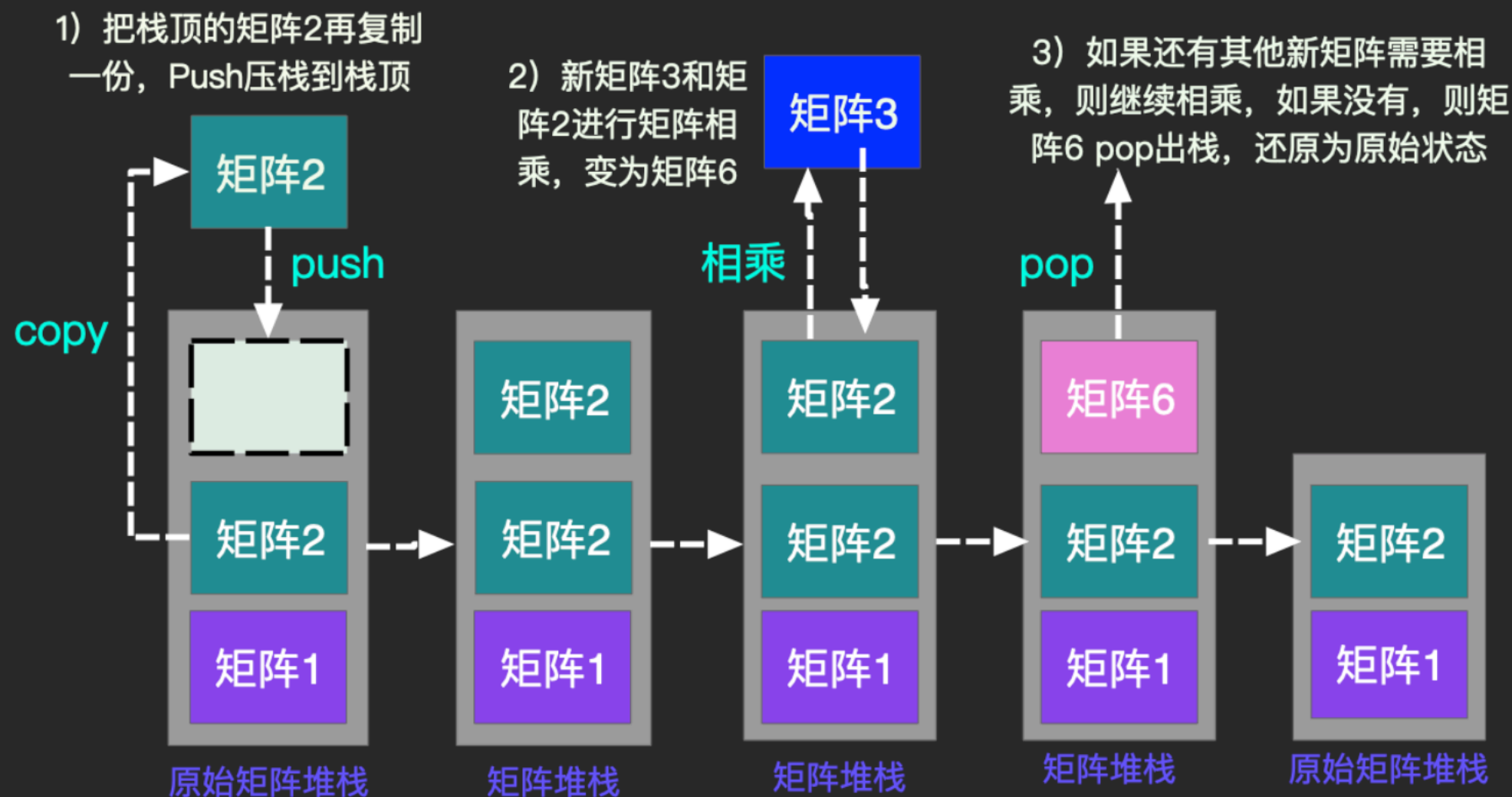
void PushMatrix(const M3DMatrix44f mMatrix);

//将GLFame 对象压入矩阵对象

void PushMatrix(GLFame &frame);

//出栈 (出栈指的是移除顶部的矩阵对象)

void GLMatrixStack::PopMatrix(void);



《矩阵入栈、相乘、出栈流程》

绘图by 简书: 凡几多

2019.5.6

如需转载请注明出处

此图: 出自于凡几多同学 视频学习总结. 此图解释的非常好. 可以关注他哦. 🍊🍊

课程研发: CC老师

课程授课: CC老师



1.4 仿射变换

//Rotate 函数angle参数是传递的度数，而不是弧度

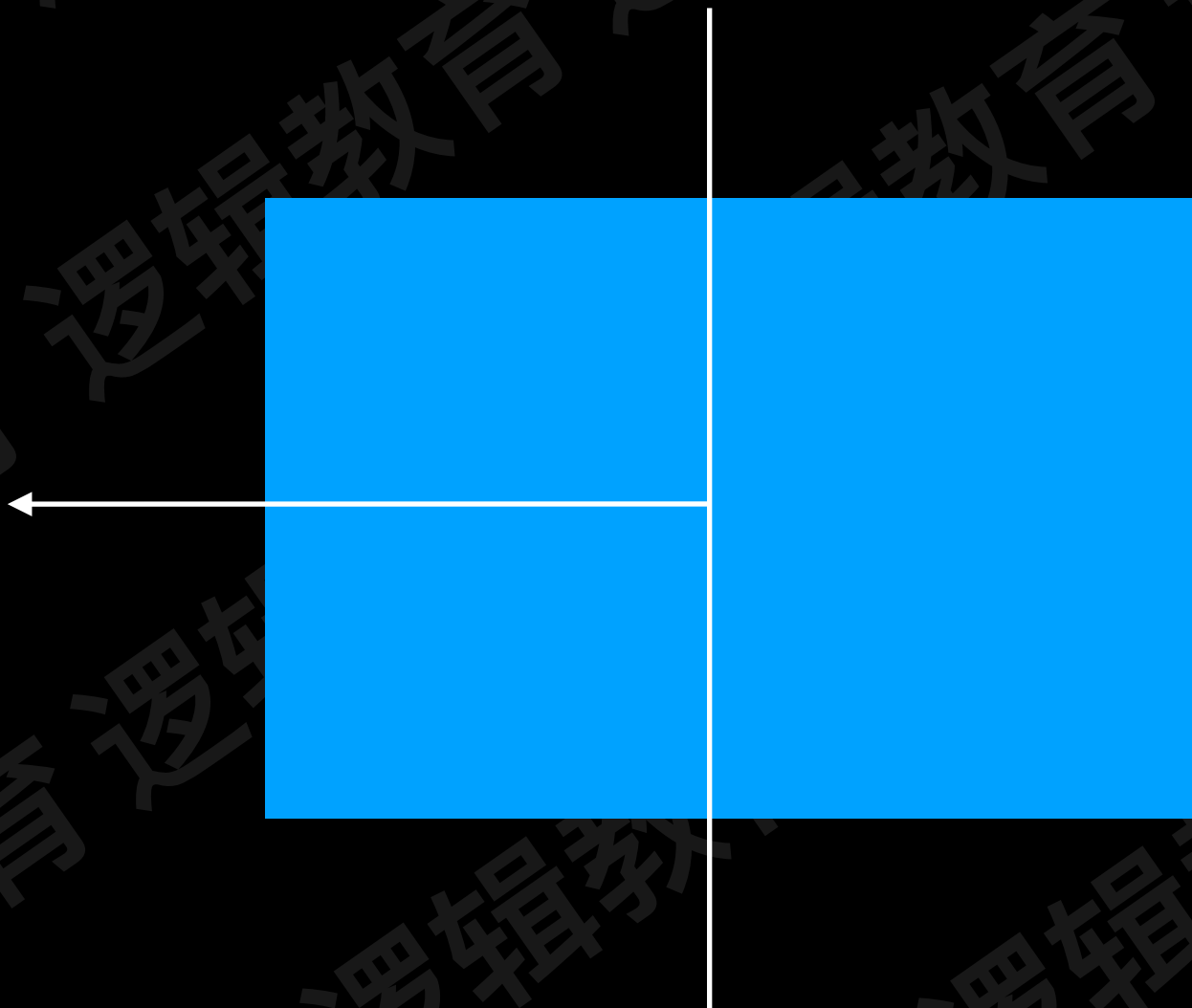
```
void MatrixStack::Rotate(GLfloat angle,GLfloat x,GLfloat y,GLfloat z);
```

```
void MatrixStack::Translate(GLfloat x,GLfloat y,GLfloat z);
```

```
void MatrixStack::Scale(GLfloat x,GLfloat y,GLfloat z);
```



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



2.1 使用照相机(摄像机) 和 角色帧 进行移动

```
class GLFrame
{
protected:
    M3DVector3f vOrigin;    // Where am I?
    M3DVector3f vForward;   // Where am I going?
    M3DVector3f vUp;        // Which way is up?
}
```



2.2 矩阵分析

X 轴方向 Y 轴方向 Z 轴方向 交换、位置

↓ ↓ ↓ ↓

$$\begin{bmatrix} X_x & Y_x & Z_x & T_x \\ X_y & Y_y & Z_y & T_y \\ X_z & Y_z & Z_z & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

一个4*4矩阵是如何在3D空间中表示一个位置和方向的

列向量进行了特别的标注：矩阵的最后一行都为0，只有最后一个元素为1

课程研发:CC老师

课程授课:CC老师



2.3 GLFrame

//将堆栈的顶部压入任何矩阵

```
void GLMatrixStack::LoadMatrix(GLFrame &frame);
```

//矩阵乘以矩阵堆栈顶部的矩阵。相乘结果存储在堆栈的顶部

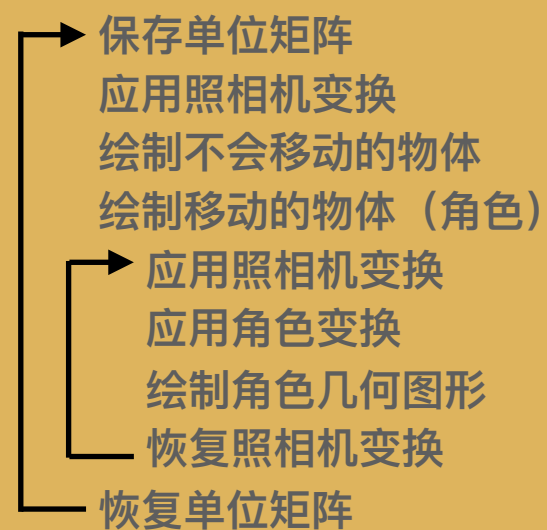
```
void GLMatrixStack::MultMatrix(GLFrame &frame);
```

//将当前的矩阵压栈

```
void GLMatrixStack::PushMatrix(GLFrame &frame);
```



2.4 照相机管理



3D环境中典型的渲染循环流程

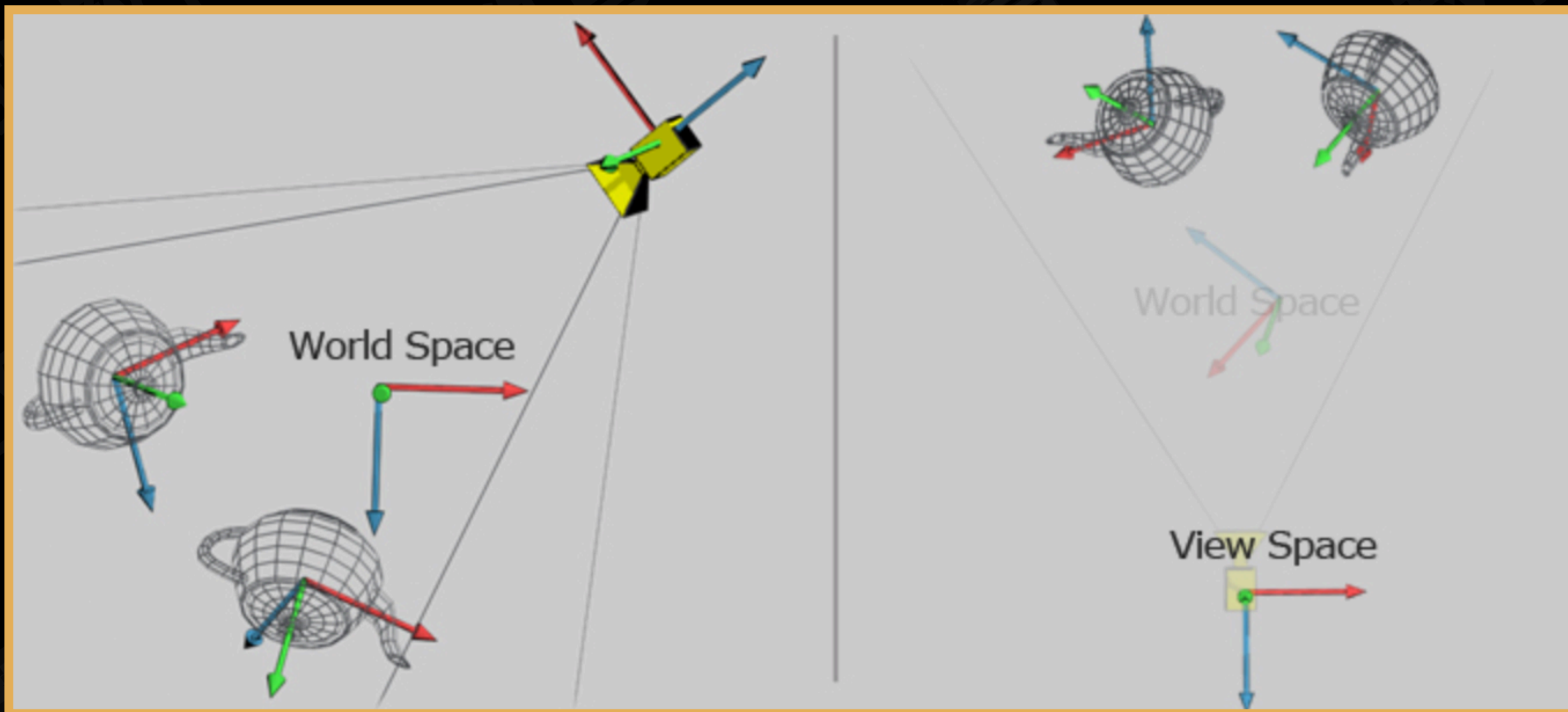
课程研发:CC老师

课程授课:CC老师



逻辑教育
Logic education

视变换



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

2.4 照相机管理

//GLFrame函数, 这个函数用来检索条件适合的照相矩阵

```
void GetCameraMatrix(M3DMatrix44f m, bool bRotationOnly = false);
```

课程研发:CC老师

课程授课:CC老师



逻辑教育
Logic education

**Thanks For Your Attention !
See You Next Time!**

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