



# 计算机图形学小白入门

——从0开始实现OpenGL

图形状态机



授课：赵新政  
资深三维工程师

专注3D图形学技术  
教育品牌

## 状态机

- 对象内部保存着一系列描述其状态的参数
- 对象的状态改变依赖于对外提供的接口
- 对象的行为依赖于其内部状态参数

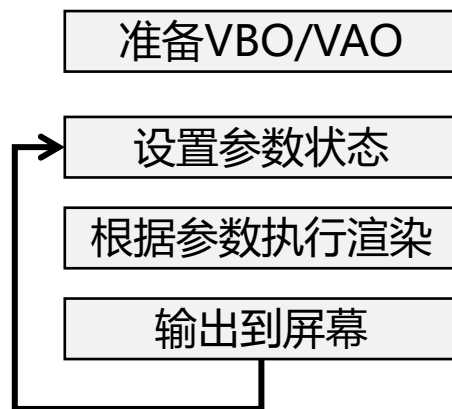
class MediaPlayer	
sound	18
speed	1.2
record	true

举例：音乐播放器

- sound：音量状态
- speed：播放倍速状态
- record：是否记历史录播放曲目

## 图形状态机

- OpenGL的API设计体系，本质来讲就是一个状态机；所有操作都是由API来对某个对象设置状态，最后执行渲染
- 举例：
  - 如果当下状态设置为绘制三角形，那么绘制时就会把顶点+索引的组合认为成三角形的描述
  - 如果当下状态设置为绘制直线，那么绘制时就会把顶点+索引的组合认为成直线的描述
- 图形状态机执行路径如下：



## 绑定-Bind

- OpenGL中，任何对象（例如VBO/VAO），用户只能够获取其内部分配的ID号
- 用户如果要对其进行操作，则必须先绑定当前操作对象ID，那么后面API执行结果才会影响到此对象
- VBO操作如下：

```
uint vbo=0;//即将分配的vbo编号
glGenBuffers(1,&vbo);//分配VBO, id给到vbo变量
glBindBuffer(GL_ARRAY_BUFFER, vbo)//绑定当前vbo, 下面操作都会针对它
glBufferData(GL_ARRAY_BUFFER, sizeof(vertices), vertices, GL_STATIC_DRAW);//更新数据
glBindBuffer(GL_ARRAY_BUFFER, 0);//解绑当前vbo
```

```
uint vao=0;//即将分配的vao编号
glGenVertexArrays(1, &vao);//分配一个VAO, id给到vao
glBindVertexArray(vao);//绑定当前vao, 下面操作都会针对它
glBindBuffer(GL_ARRAY_BUFFER, vbo);//绑定当前vbo, 下面操作都会针对它

//在当前的vao里, 设置当前mesh的0号属性数据, 数据源就在当前绑定的vbo中
glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 3 * sizeof(float), (void*)0);
glBindVertexArray(0);//解绑当前vao
```