



计算机图形学小白入门

——从0开始实现OpenGL

几何顶点架构设计-VAO/EBO

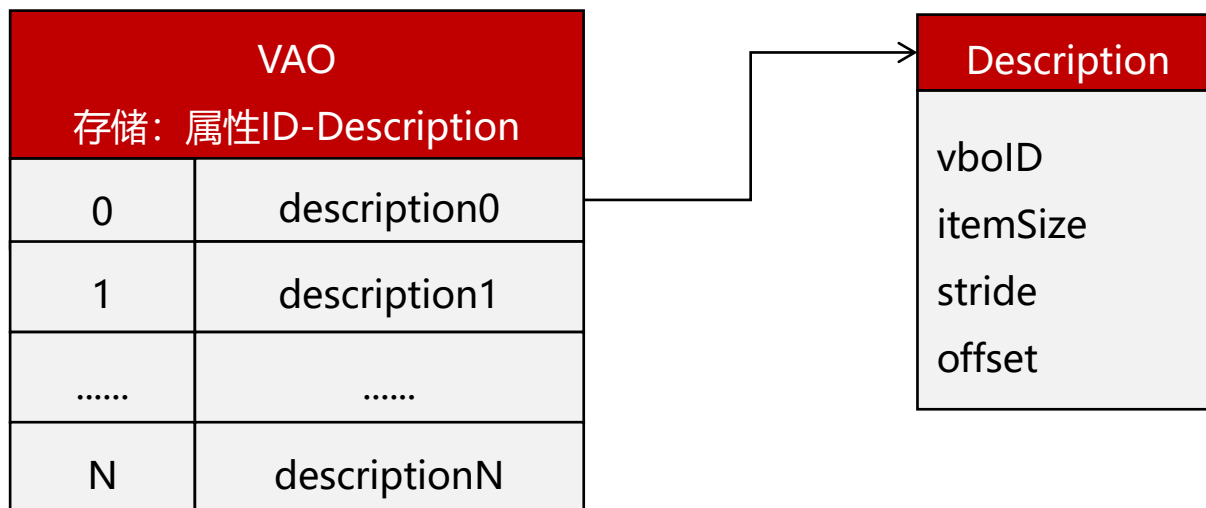


授课：赵新政
资深三维工程师

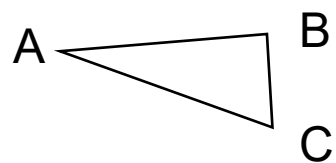
专注3D图形学技术
教育品牌

顶点数据组织-VAO

- 绘制一个**三角形**的时候，在每次VertexShader当中，会处理顶点所有的属性（Attribute），所以需要知道如下问题：
 - 当前处理第几个顶点？
 - 当前顶点position信息从哪里获取？
 - 当前顶点color/uv信息从哪里获取？
- 对于**本三角形绘制**，我们把**属性信息**赋与**编号**（position-0，color-1，uv-2），做成映射表格，即**VertexArrayObject (VAO)**，每个三角形对应一个VAO描述结构
- VertexShader里再使用属性编号讲顶点属性回读出来即可

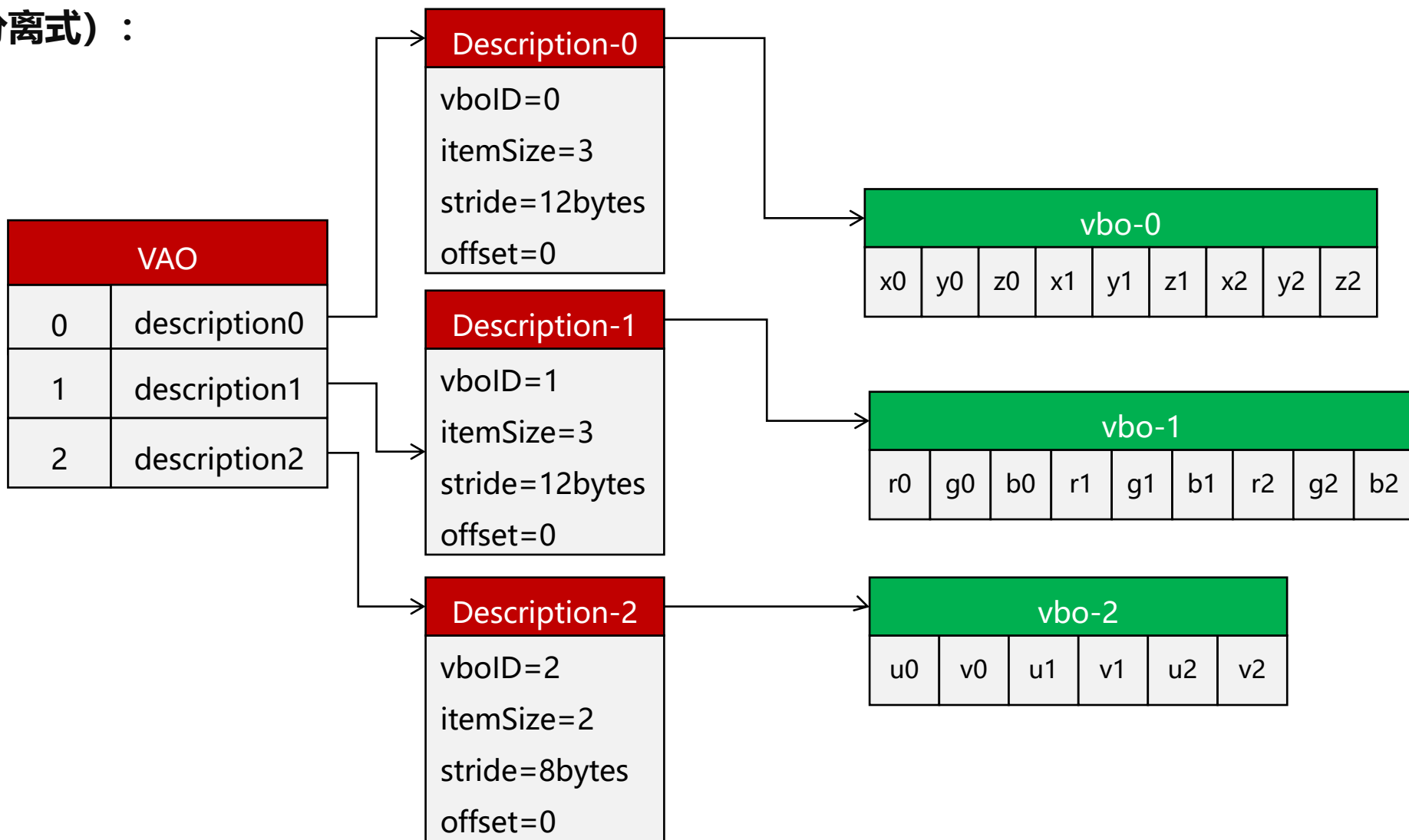


顶点数据组织举例（分离式）：

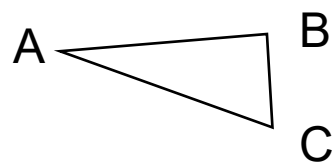


属性编号：
position 编号为0
color 编号为1
uv 编号为2

顶点编号：
A顶点为0
B顶点为1
C顶点为2



顶点数据组织举例（交叉式）：



属性编号：

position 编号为0

color 编号为1

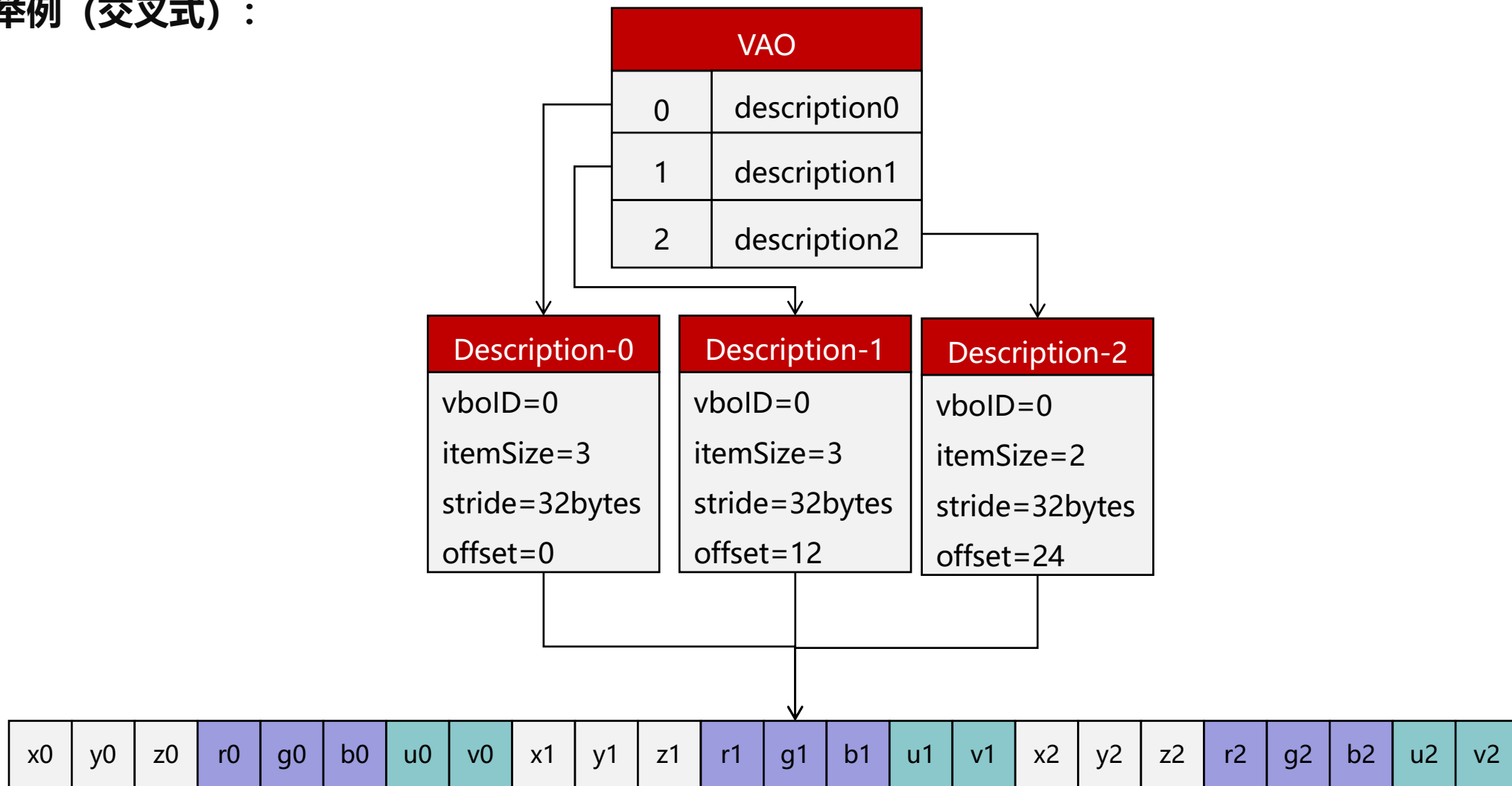
uv 编号为2

顶点编号：

A顶点为0

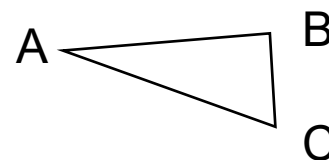
B顶点为1

C顶点为2



程序结构设计:

class GPU			
VBO Map 存储: id-vbo		VAO Map 存储: id-vao	
0	vbo0	0	vao0
1	vbo1	1	vao1
.....
N	vboN	N	vaoN

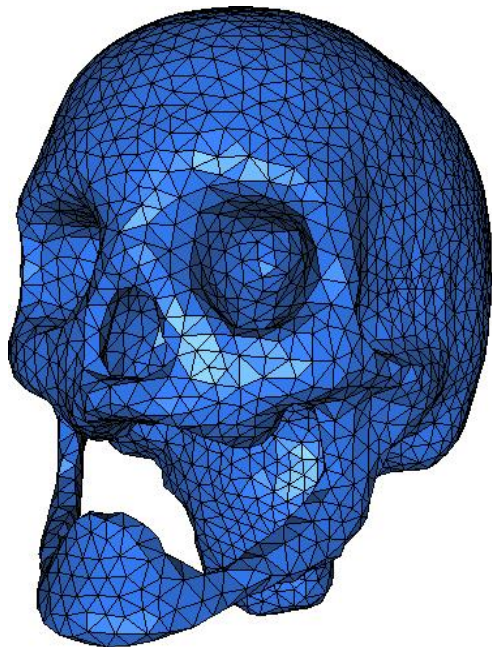


- 对于一个三角形ABC的数据信息
- 可以如下描述:
 - 一个交叉的VBO
 - 多个独立的VBO
- 可以如下管理:
 - 分配一个VAO
 - 设置VAO所有属性的描述信息
- 绘制的时候, 只需要传入需要绘制的三角形对应**VAO**即可

真的足够了么?

Mesh

- 由多个三角形拼接而成的曲面，称为一个Mesh



特点：

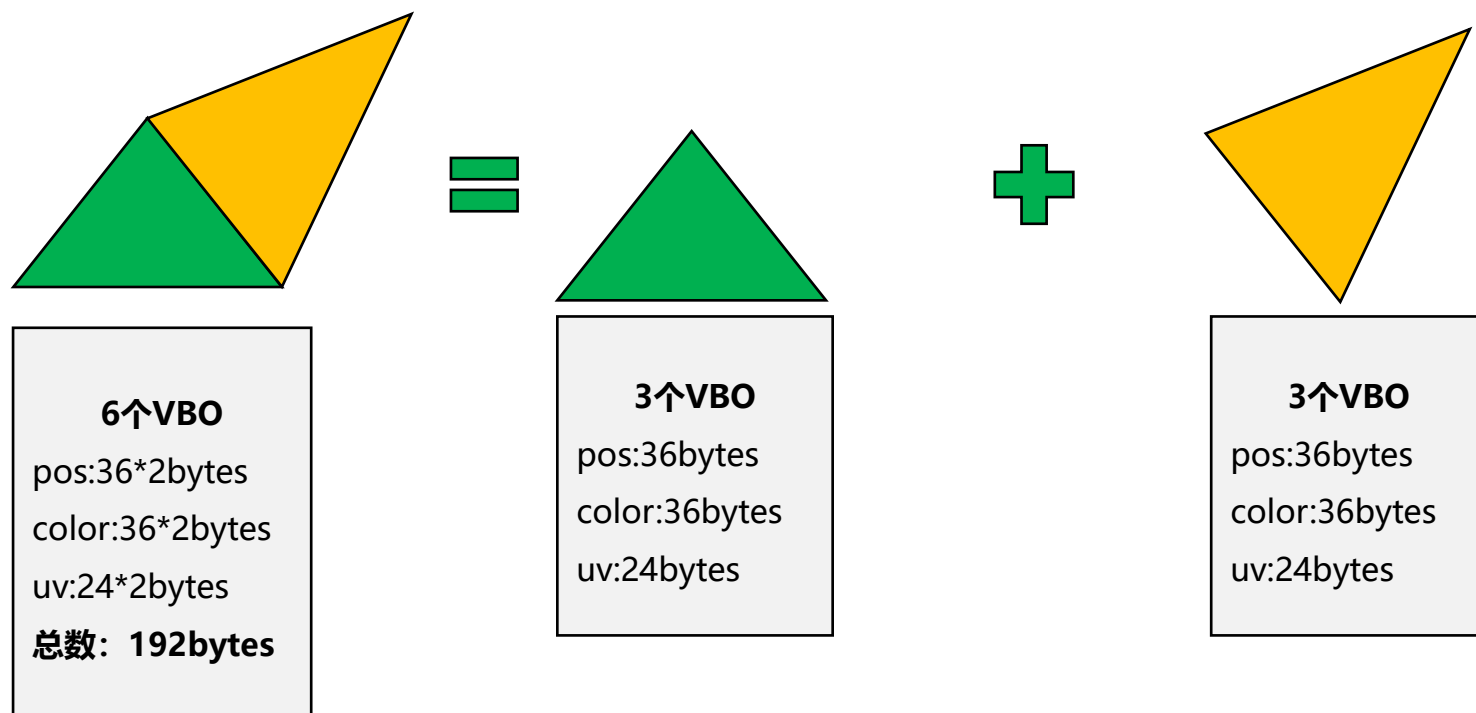
- 由多个三角形拼接而成
- 每个三角形都可能与邻接三角形共用顶点

问题：

- 共用顶点可否只描述一次，不用重复存储？
- 可否把所有顶点打包组织成多个VBO，整个Mesh做成一个VAO？

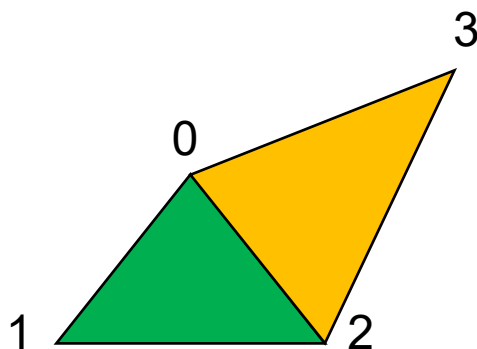
Mesh-顶点共用

- 两个三角形各自存储顶点数据，需要192bytes，也造成了一定重复数据
- 需要调用两次绘制函数



Mesh-顶点共用

- 将Mesh所有顶点属性合并存放：



3个VBO
pos:48bytes
color:48bytes
uv:32bytes
1个EBO
index: 24bytes
总数: 152bytes

x0	y0	z0	x1	y1	z1	x2	y2	z2	x3	y3	z3
----	----	----	----	----	----	----	----	----	----	----	----

r0	g0	b0	r1	g1	b1	r2	g2	b2	r3	g3	b3
----	----	----	----	----	----	----	----	----	----	----	----

u0	v0	u1	v1	u2	v2	u3	v3
----	----	----	----	----	----	----	----

0	1	2	2	3	0
---	---	---	---	---	---

- 使用顶点位置索引，描述三角形构成
- 用于存放顶点索引的缓存，类型为**int***型
- Element Buffer Object (EBO)**
- EBO 由Description描述，作为特殊字段加入VAO**

总结

- 对于一个**Mesh**，系统使用多个（或者一个）VBO存储所有顶点数据，而且可以避免顶点数据重复；使用EBO来描述三角形构成；使用VAO来表示整个模型属性构成

