**opengl的VAO**

VBO在渲染阶段才指定数据位置和顶点信息，然后根据此信息去解析缓冲区里的数据，联系着两者中间的桥梁是GL-Context。GL-Context整个程序一般只有一个，所以如果一个渲染流程里有两份不同的绘制代码，GL-Context就负责在他们之间进行状态切换、这也是为什么在渲染过程中，在每份绘制代码之中有glBindBuffer/glEnableVertexAttribArray/glVertexAttribPointer。那么优化方法就来了---把这些都放在初始化时候完成吧！--这样做的限制条件是“负责记录状态的GL-Context一般只有一个”，那么就不直接用GL-Context记录，用别的东西做记录吧--这个东西针对“每份绘制代码”有一个，记录该次绘制所需要的所有VBO所需信息，把它保存到GPU的特定位置，绘制的时候直接在这个位置取信息绘制。于是，VAO诞生了。

VAO的全名是Vertex Array Object，首先，它不是Buffer-Object，所以不用做存储数据；其次，它针对“顶点”而言，也就是说它跟“顶点的绘制”息息相关。它记录数据的存储和如何使用的细节信息。使用VAO的优势在于，如果有多个物体需要绘制，那么设置一次绘制物体需要的顶点数据、数据解析方式等信息，然后通过VAO保存起来后，后续的操作不再需要重复这一过程，只需要将VAO设定为当前的VAO，那么opengl则会使用这些状态信息，当场景中物体较多时，优势十分明显。