

# Hello OpenGL ES 2

这是一个OpenGL ES 2的“hello World”程序，用来介绍以下几个在渲染中必要的一些重要对象。为保证对概念的理解正确，不对重要对象进行翻译，直接使用OpenGL ES 官方文档中的英文单词，以免不同书籍对同一次的不同翻译对理解产生负面影响。

- 请参照示例代码中01\_HelloOpenGLESL2\jni目录下的HelloContext和HelloContext2理解本章内容。

---

## OpenGL ES 2渲染管线中的重要对象

### Shader

Shader是渲染中不可或缺的东西，它是用来对数据进行处理程序，把输入的数据最后处理为什么样的图片都靠它来做。

- Shader有两种：一种叫VertexShader，用来处理顶点数据；另一种叫FragmentShader，用来进行颜色填充。
- 一个VertexShader和一个且只能和一个FragmentShader组成一个被渲染硬件执行的ShaderProgram；一次渲染动作只能使用一个Program对指定的一批输入数据进行处理。
- 每个Shader有一个约定的入口函数，即main函数，渲染时从此函数入口开始执行Shader指令对一个数据进行处理。
- 每个Shader有一个约定的数据输出地址，为Shader代码中约定的全局变量，硬件通过此变量获得Shader处理的结果然后交给下一个处理单元进行下一步的处理。
- 一次渲染过程同时有多个Program的实例在同时运行，实例的个数的上限为硬件中渲染单元的个数，此次渲染的一批数据分散在多个渲染单元上被并行处理。

### Attribute

这就是准备让OpenGL ES来处理的数据，一般为顶点数据、纹理坐标数据、法线坐标数据等。

- 一次渲染需要准备一批Attributes，每个Attribute对应一个顶点的各种属性。
- 一次渲染中，一个Program的运行实例被执行一次可以处理一个Attribute。对于一次渲染的Program，每个实例、或每个实例的执行，输入的Attribute都是不同的。它类似于函数被调用时输入的参数。
- 一批Attributes有两种数据组织方式，不同的组织方式需要调用不同的处理函数：一种是每个Attribute的各个属性相邻，多个Attributes组成一个Array，使用glDrawArray来处理；另一种是把Attribute的各种属性分开，每种属性组成一个Array，使用glDrawElement来处理。

### Uniform

用来作用到每个Attribute上，用来实现各种效果，比如移到、变形、光照等等，各种特效都是设定合适的Uniforms通过适当的算法作用到Attribute上来实现的。

- 一次渲染可以使用多的Uniform，Uniforms之间不需要有任何关系。
- 一次渲染中，一个Program的所有实例以及每个实例的每次执行，都使用相同的值，在渲染前通过glUniform开头的函数指定。
- 在Shader执行时，不可改变Uniform的值。

### Varying

Varying是VertexShader到FragmentShader的桥梁, 它可以把VertexShader中的数据传给FragmentShader, 常见用它来传递纹理坐标数据。

- 在VertexShader和FragmentShader中, 需要声明同样名称的Varying变量才能正确传送数据。

---

## OpenGL ES 2 的渲染步骤

### 准备工作

- 在开始绘制一帧图像前, 调用glClear函数初始化渲染表面。
- 编绘制需要的ShaderProgram预先编译好, 准备绘制时使用。

### 一次渲染动作

绘制一帧图像可以执行多次渲染动作, 每次渲染都需要下面几个步骤:

- 指定使用的ShaderProgram
- 把渲染使用的Uniform的值传给Program
- 指定渲染输入的Attribute数据
- 使用合适的方式开始渲染
- 渲染完成后清理上下文