

OpenGL基本概念

03 November 2013

最近在做用OpenGL (<http://www.opengl.org/>)显示YUV数据的工作。完成之后，决定将学到的东西整理总结一下。在探索OpenGL的过程中，一篇英文的tutorial (<http://db-in.com/blog/2011/01/all-about-opengl-es-2-x-part-13/>)让我受益匪浅。我自己的总结也将主要取材于此。那篇tutorial的作者是巴西人，其英文虽然语法不甚精湛，但文章可读性极好。如果你英文还可以，强烈推荐阅读原文，比我这总结有用多了。

本文主要介绍OpenGL的基本概念。

OpenGL的组成

OpenGL主要包括三个要素：

- 原件 (Primitives)
- 缓冲区 (Buffers)
- 光栅化 (Rasterisation)

原件

原件包括点、线、三角形。有的OpenGL版本还包括四边形，但在OpenGL ES（见文末说明）中只有这三种原件。

缓冲区

包括三个概念：

- 帧缓冲区 (Frame Buffers)
- 渲染缓冲区 (Render Buffers)
- 缓冲区对象 (Buffer Objects)

帧缓冲区是GL储存渲染结果的地方。它可以包含多个图像，也就是包含多个渲染缓冲区。

渲染缓冲区就是一个图像，例如颜色图像、深度图像、模版图像（决定每个位置是否可见的掩膜）等。

缓冲区对象就是程序员提供给OpenGL的数据，分为结构类和索引类的。前者被称为“数组缓冲区对象”或“顶点缓冲区对象”（“Array Buffer Object”或“Vertex Buffer Object”），即用来描述模型的数组，如顶点数组、纹理数组等；后者被称为“索引缓冲区对象”（“Index Buffer Object”），是对上述数组的索引。

光栅化

光栅化的过程就是根据缓冲区对象里提供的数据经过渲染从3D模型得到2D图像的过程。得到的2D图像会根据帧缓冲区的配置来决定是直接送到屏幕显示或是做别的用处。

OpenGL的流水线

从2.0版本开始，OpenGL支持可编程的流水线。也就是说，程序员可以通过Shader（一种程序）来控制GPU渲染的过程。Shader分为Vertex Shader和Fragment Shader。这两种Shader是成对出现的，GPU先执行前者，后执行后者。

Vertex Shader

Vertex Shader对于3D模型网格的每个顶点执行一次，主要是确定该顶点的最终位置。Vertex Shader还会准备并输出一些变量，传给Fragment Shader使用。

Fragment Shader

Fragment Shader对于最终得到的2D图像的每个像素处理一次。3D物体的表面最终显示成什么样将由它决定。例如为模型的可见表面添加纹理，处理光照、阴影的影响等等，都在这里做。Fragment Shader的输出是一个RGBA格式的像素颜色值。

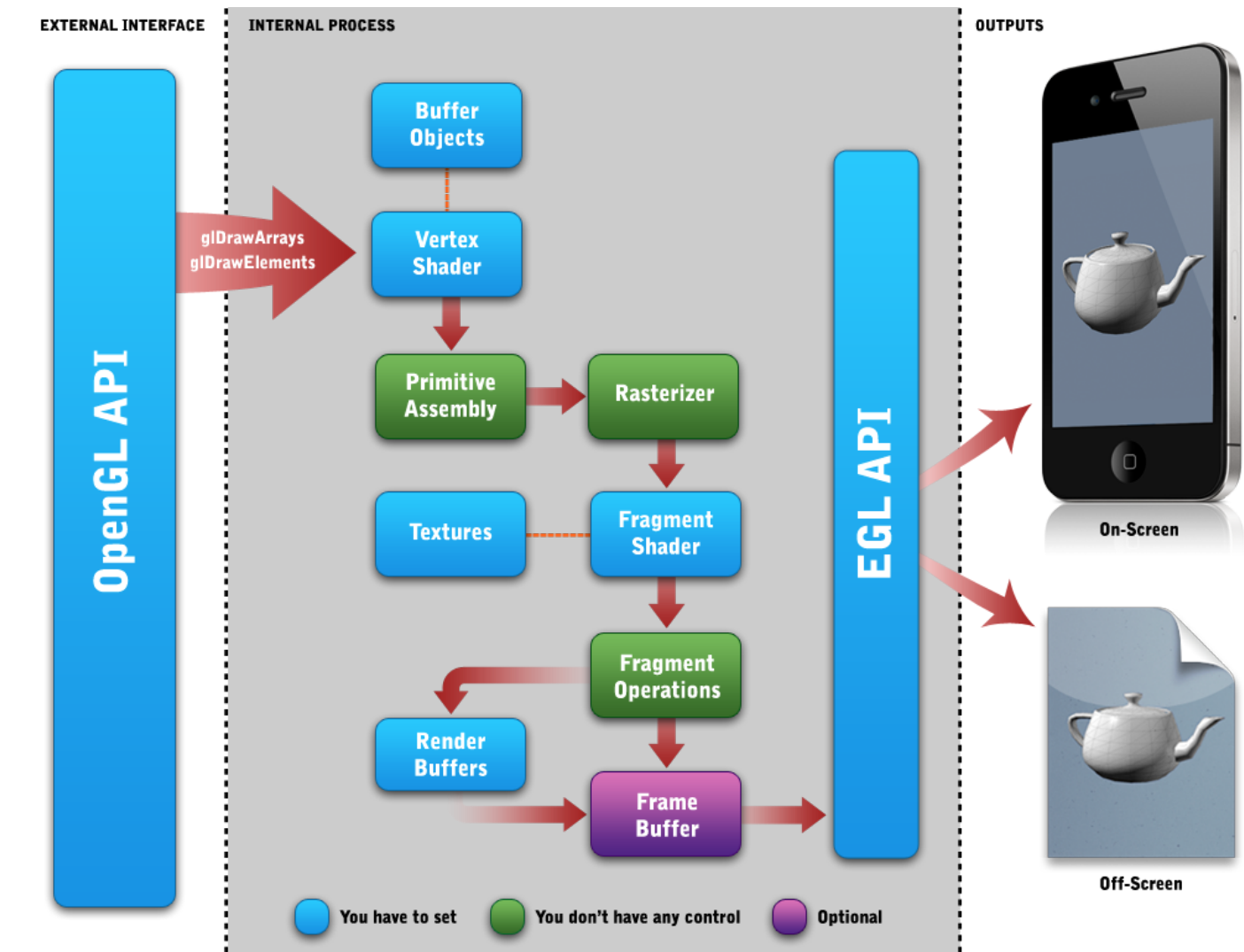
OpenGL，OpenGL ES，EGL和EAGL

OpenGL提供一组API，各显卡制造商和系统制造商来实现这组API。OpenGL ES (Embedded System)是OpenGL的嵌入式系统版本，用于手机等移动设备。可以简单认为OpenGL ES是OpenGL的一个子集。

EGL是另外一组API，主要由系统制造商实现。它负责OpenGL和原生窗口系统之间的接口，用于把OpenGL渲染的结果显示在屏幕上。EAGL是Apple公司对EGL做了修改，自己实现的一套屏幕显示接口，一般只用于iOS和OS X系统中。

OpenGL图示

最后，给出一幅包含上面这些概念的图。



(/images/2013-11-03-opengl-concepts.png)

技术 ⁸¹ (/categories.html#技术-ref)

编程 ²¹ (/tags.html#编程-ref)

OpenGL ² (/tags.html#OpenGL-ref)

← Previous (前一篇) (/posts/show-up-the-mobile-apps-i-published)

Archive (目录) (/archive.html)

Next (后一篇) → (/posts/embed-ppt-slides-in-webpage)

0条评论

最新 最早 最热

还没有评论，沙发等你来抢

社交帐号登录: 微博 QQ 人人 豆瓣 更多»



说点什么吧...

发布