

十二、VS 和 FS 和简单特效

在 OpenGL 和 DirectX 等图形库中，有 VS 和 FS 的概念，他们分别是顶点着色器(Vertex Shader)和片元着色器(Fragment Shader)。

1. VS

VS 负责将顶点转换到 CS(裁剪空间)，包括仿射变换和投影等变换。

2. FS

FS 负责在目标设备的片元上完成最终的颜色计算，通常的片元都是指单个像素，但是假如绘制目标设备不是屏幕，那么片元就不是像素了，比如我们把绘制的目标设备指向一个纹理，然后把这个纹理保存成图片，那么这里的片元就指的是纹素(当然也和最终图片上面的像素对应)。

3. OpenGL 绘制的一个大概流程

3.1 首先我们需要先准备好需要渲染的顶点数据，比如一个模型的所有顶点和每个顶点的纹理坐标等数据，然后将其从内存复制到显存中

3.2 GPU 依次从显存中使用三个顶点(一个三角形)的数据参与 VS 运算，VS 代码由用户编写，完成模型的变换(包含旋转、缩放、投影等变换)

3.3 GPU 将经过 VS 处理之后的顶点在裁剪空间中裁剪

3.4 GPU 使用透视校正插值对顶点的每个属性进行插值

3.5 GPU 将插值之后的结果传递给 FS，FS 代码由用户编写，FS 对每个片元(可能是一个像素)计算，通常得到一个颜色值

3.6 GPU 将 FS 的计算结果输出到帧缓冲区

3.7 在某个适合的时刻，CPU 将帧缓冲区的内容输出到目标设备(屏幕或者纹理)

4. GPU 的并行工作

实际上如果我们认真分析我们之前的代码可以发现，主要的运算资源都是消耗在三角形的填充上面，而三角形填充的主要工作就是对每个像素进行计算(插值、取纹理、输出到屏幕)，而 GPU 在设计之初就可以进行大量的并行运算，GPU 通常在渲染片元的时候并行执行，而不像我们是在一个循环中依次对的那个像素渲染。实际上我们在编程中可以发现，每个像素的渲染结果基本上是和其他像素无关的，这也从侧面说明了片元的渲染是可以并行化的。

5. 我们可以尝试将我们之前写的渲染器进行改造，使其更接近上面所说的 OpenGL 程序结构。