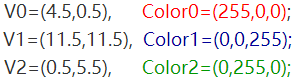
<https://blog.csdn.net/patient16/article/details/50194993>

一个例子：

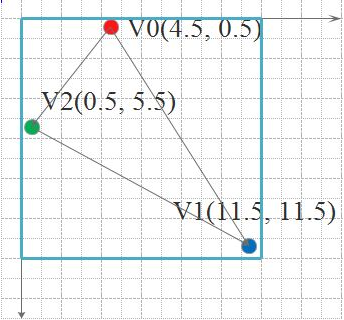
这里假设：

1. 输入的顶点是屏幕坐标（严格讲，Raster还需要做透视出发及viewport变换才能得到屏幕坐标，可认为shader里完成了这两个步骤），且每个顶点只有Color一种属性
2. 只开启了顶点着色器和像素着色器，并且对于输入的Color，着色器直接将之输出

以一个简单的三角形为例子，其中：



如下图：



**1）判断三角形是顺时针还是逆时针**

通过三角形两条边的叉乘符号可以得到，如上面的例子

V0v1 = (x1 – x0, y1 – y0)

V0v2 = (x2 – x0, y2 – y0)

V0v1 x v0v2 = (7.0, 11.0) x (-4.0, 5.0) = 7 \*5 – (-4 \* 11) = 79 > 0表示v0v1v2是顺时针

V0v2 x v0v1 = (-4.0, 5.0) x (7.0, 11.0) = (-4 \* 11) – 7 \* 5 = -79 < 0 表示v0v2v1是逆时针

**2）判断像素是否在三角形内**

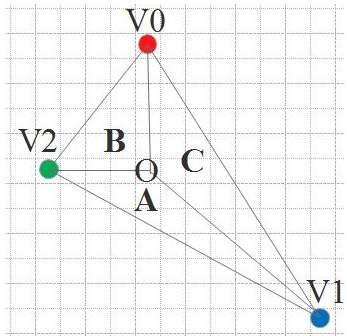
可以通过判断该位置是否在三角形三条边的同一侧

**3）确定每个像素的颜色**

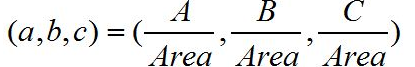
重心法，通过面积得到权重（求三角形的面积通过向量的叉乘，得到的是三角形面积的两倍）：

D3D和opengl去的定像素颜色的方法是相同的：以三个顶点的颜色及重心坐标作为权重系数插值得到任意像素的颜色

对于任意像素O，设三角形V2OV1，V1OV2，V1OV0的面积是A，B，C



则重心坐标：



对于像素O：



最终得到的结果：

