**Gfx2.0常用Shader的参数说明**

引擎部 马个

2014．1.14

本文档对Gfx2.0特效编辑器的8种常用Shader（.sdr脚本）的Shader参数进行了讲解说明，旨在帮助特效美术的同学更好地理解各Shader参数的含义和如何控制它们。

建议先阅读：《*GFX新增的自定义着色器功能说明与教程.docx*》

位于[\\sourceserver\InHouseTools\文档\](file:///\\sourceserver\InHouseTools\文档\)。

《*shaders功能运用.docx*》该文档从美术角度进行了各特效的Shader应用分析，所以本文档力求从程序角度带给美术同学新的理解。

1. *溶解。*
2. *溶解边缘半透。*
3. *溶解边缘颜色。*
4. *燃烧。*
5. *流动。*
6. *流动叠加。*
7. *波动和波动2。*
8. *Fur和Fur2。*

简单说明一些概念：

* Gfx2.0的自定义Shader通常使用2层贴图，第一层是原始贴图，比如特效的Diffuse纹理或者模型的原始贴图，在属性栏里设置：



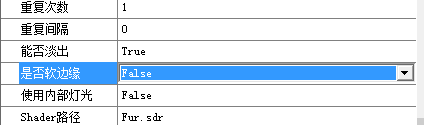
第二层是Shader专用贴图，开启自定义Shader后可能会用到，通常用来设置遮罩图（mask图），在属性栏设置：



* Shader参数的索引从0开始，依次命名为c0，c1，c2…

每个参数有4个分量x,y,z,w。我们取名为c0.x，c0.y，c0.z，c0.w。

* 注意，本文档只适用于Gfx2.0的Shader，不适用于Gfx2.2!前者的Shader系统基于DX8，后者基于DX9，后者的Shader系统更为高级和强大，有兴趣可以去了解下。举个例子，你会发现2.2的编辑器支持“是否软边缘”这个属性，但2.0没有：



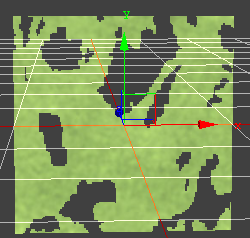
这个软边效果只在2.0的“溶解边缘半透”有支持，而在2.2中所有特效都支持。

1. 溶解。

这个效果是通过第2层遮罩图的alpha通道a和c0共同作用来表现消散的效果（如下图）。

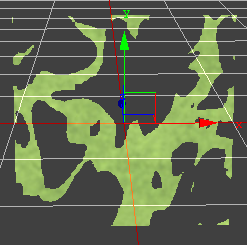
可控参数：c0.x，取值范围0到1。

含义：Shader会用a减去c0.x的结果作为最终alpha值。可以这样理解，c0.x从0变到1，则特效会逐渐消失。

 +  = 

*Original mask result*

说明：这里c0.x为0，则a-c0.x=a，所以最终结果只受mask图的alpha通道影响。

 +  = 

*Original mask result*

说明：这里c0.x为0.5，则a-c0.x<a，所以最终结果消失的部分更多了。

1. 溶解边缘半透。

该效果在溶解的基础上加入了软边效果，非常好理解。

可控参数：

c0，跟溶解的一样。

c1.x，取值范围大于等于0。用来控制硬边到软边的过渡范围。值越大，则半透的越多。当设置为0时，则效果与溶解的一样。

1. 溶解边缘颜色

该效果在溶解边缘半透的基础上可以通过c2控制边缘的颜色。

《*shaders功能运用.docx*》中已经介绍得很清楚，这里就不复述了。

1. 燃烧

该效果从Shader代码可以看出，几乎完全跟“溶解边缘颜色”相同。所以同样可以看作溶解效果的一个变种。不同的仅仅是在边缘叠加上了红色，仅此而已。

可控参数：与溶解边缘颜色相同。

总结，前4种效果都是“溶解”效果的一系列变种。

1. 流动。

该效果是第2层贴图进行纹理动画，然后与第1层贴图进行插值混合（不是叠加），呈现流动的效果。

可控参数：

c0，xyz值控制纹理动画的平移。注意，纹理动画只作用于第2层贴图。

c1.x，取值范围0到1，作为两层贴图的混合因子。等于0时只显示第1层贴图，等于1时只显示第2层贴图。

1. 流动叠加

该效果的Shader代码与“流动”只有1处不同，“流动”是两层贴图间的过渡，而“流动叠加”是两层贴图的叠加。

控制参数：

c0与流动的相同。

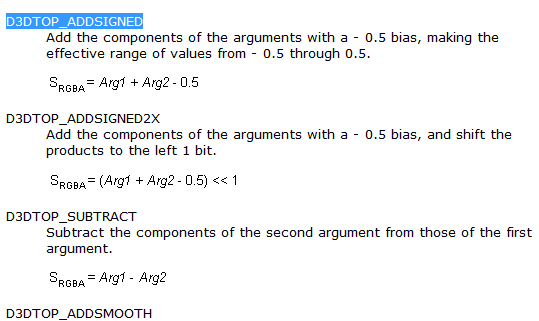
c1.x，取值范围0到1，控制第2层贴图的显示与否（如下公式）。

*最终颜色 = 第2层贴图 \* c1.x + 第1层贴图*

1. 波动和波动2。

这两个效果在表现上是一样的，只不过后者支持参数控制淡入淡出。

表现是两层贴图间的一个AddSigned方式的叠加（该术语来自DX，DX的固定管线的纹理叠加支持多种方式，如下图，AddSigned只是其中一种，进入灵活的Shader时代后，就可以任意自由的控制叠加等了，所以废弃了过时的固定管线）。



如图中公式所示，AddSigned叠加 *=第1层贴图+第2层贴图-0.5*。

控制参数：

波动无控制参数。

波动2：c0.x，取值范围0到1。用来控制第2层贴图的显示与否，以及最终颜色的alpha值。所以最终的颜色计算公式是：

*最终颜色rgb = 第1层贴图 + (第2层贴图 \* c0.x) - 0.5*

*最终颜色a = 第2层贴图a \* c0.x*

1. Fur和Fur2

这两个效果的表现也是一样的，只不过后者加入了淡入淡出的支持。

从Shader代码分析，该效果应该本来是用于毛皮材质的高光渲染，通过第2层高光图来实现，其alpha值越大表示该处高光效果越强，从而最终颜色受本身贴图的影响越来越小，受高光图影响越来越大。

可控参数：

Fur：c1（不知道为什么是c1而不是c0，Shader是这样写的）作为皮肤颜色用来调制第1层贴图（modulate，即与第1层贴图颜色值相乘）。比如为（1,1,1）则不改变第1层贴图的颜色。

Fur2：

c1与Fur相同。

c0.x，取值范围0到1，用来控制最终颜色的alpha值，起一个简单的淡入淡出的效果。