# **UnityShader-模仿炉石传说卡牌动态效果(二)**

作者：[枸杞忧天](http://gad.qq.com/user/index?id=1075523)

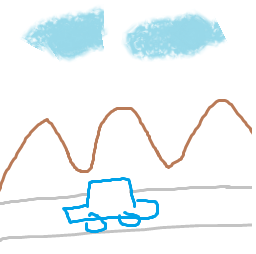
链接：http://gad.qq.com/article/detail/286351

在第一部分《[Unity Shader-模仿炉石传说卡牌动态效果(一)](http://gad.qq.com/article/detail/286350)》中，鄙人简单介绍了材质编辑器的组成及特点，基本使用方法及“扰动”效果的实现思路。这第二部分，将总结“**移动**”、“**旋转**”效果的实现及两种混合效果的显示差异。

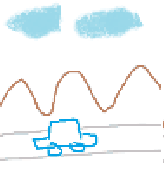
那就开车吧。

**移动**

移动是最容易实现的特效，最简单粗暴的方法就是采样时让UV坐标不断的增加或者减小，通常用时间 \* 速度来控制移动效果，如：

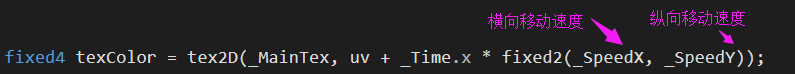


↑ 一辆静止的小汽车



↑ U不断增加后的效果，快乐奔跑的小汽车

Shader写起来大概像这个样子：

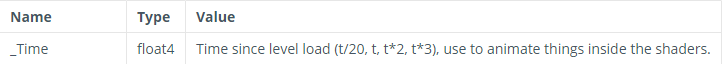


**需要注意的几点是：**

1、选择了正确的循环模式

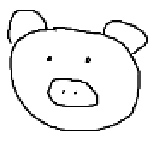


2、选择合适的时间

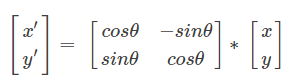


**旋转**

旋转效果本质上是平铺在X、Y轴上的2D图片，绕Z轴旋转，通常用时间 \* 旋转速度来控制旋转效果。因此只要用旋转角度构建一个旋转矩阵，再用UV坐标与这个矩阵做乘法，即可，如：



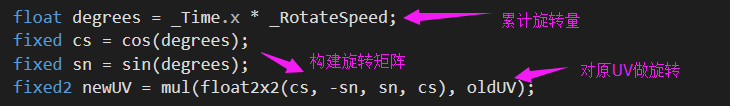
↑一只旋转的小猪



↑旋转矩阵

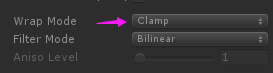
因为缩放、旋转、平移矩阵的推导网上多的是，而这些又是图形学的入门知识，因此在这就不再赘述了(其实是因为我忘了，以后有机会再总结一下吧)。

Shader写起来大概像这个样子：

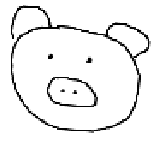


**需要注意的几点是：**

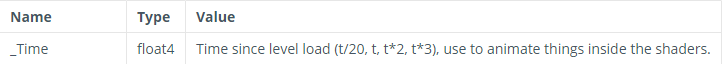
1、选择了正确的循环模式



如果选择了Repeat，那么效果就会变成：

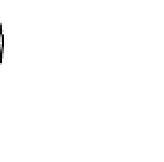


2、选择合适的时间



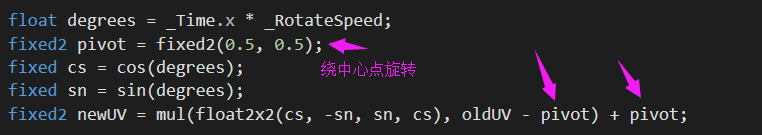
3、合理的锚点

如果不修正旋转锚点的话，猪是以左下角(0,0)为锚点旋转的，如：



↑绕(0,0)点旋转的小猪

微调一下即可：



**稍加延伸：移动 + 旋转 or 旋转 + 移动**

旋转一定角度基础上的**移动**：

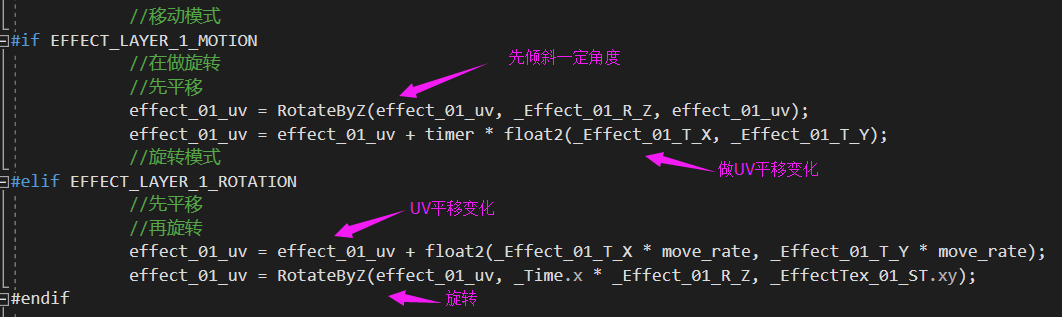


↑通往天国的小汽车

在非中心点的地方**旋转**：



↑在左下角旋转的小猪



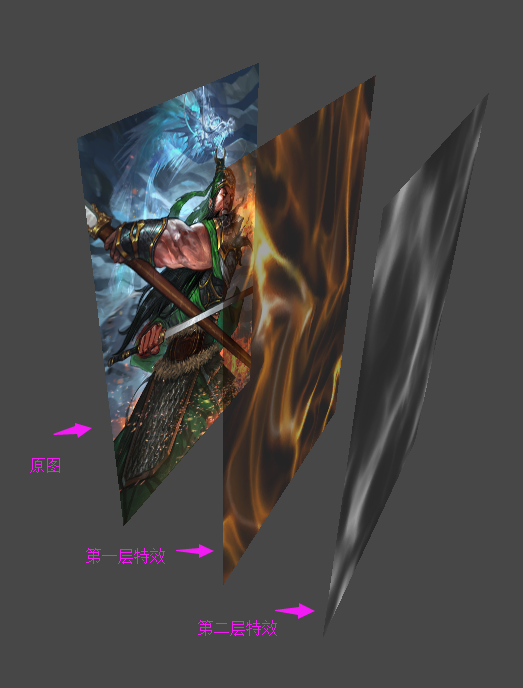
↑ 对应的shader

我个人的经验：

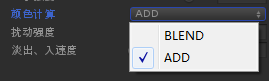
我从不记缩放、旋转、平移的先后顺序(因为我也记不住)，试几次对了就行了。

**混合效果差异**

这里所指的**混合效果**，并非直接用shader中的**Blend**，因为Blend是指**片段着色器的返回颜色**和**当前颜色缓冲区颜色**进行计算的逻辑，而我们的片段着色器返出的颜色，是将各个特效层颜色混合结果进行混合(有点绕)。换言之：此提及的“混合”只是将**每层特效的颜色**(之前提及的三层)，通过某种方式叠加在一起的逻辑。



↑多层效果混合示意



我只用了两种混合模式，BLEND和ADD，名字没起好(因为BLEND就是混合的意思...)，直接看公式吧：

**BLEND：**

混合后混合区颜色 = 当前特效图颜色 \* 当前特效图Alpha +  当前混合区颜色 \* (1 - 当前特效图Alpha)

**ADD：**

混合后混合区颜色 = 当前特效图颜色 +  当前混合区颜色

这介绍的有点**水**了，因为这原理跟Shader的Blend没什么差别，分别对应了：

**Blend SrcAlpha OneMinusSrcAlpha**

和 **Blend One One**

只是将颜色缓冲区的概念换成了“当前混合区颜色”，而这种混合模式，也可完全照搬Shader中支持的各种混合模式，**只要特效妹子开心就好**。

**其实我想强调的是：**

1、心里要对当前混合区颜色和效果图采样颜色有个概念。

2、处理Alpha时要稍加留意，你使用的纹理有没有Alpha通道你要知道，没有Alpha通道是让Unity按灰度值导入为Alpha通道还是自己算得清楚。

3、采样出的RGB是否要先被Alpha处理一下心里得有数。

4、没什么绝对正确或错误的混合，只要看起来是对的，它就是对的。

**总结**

**片段着色器返出的颜色 = 原图颜色 op 特效1 op 特效2 op 特效3**

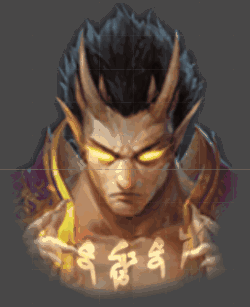
**特效x = 特效图颜色 \* 特效混合色 \* 遮罩图对应通道**

**特效图颜色 = tex2D(特效图, uv + 偏移量 + 扰动量)**

讲到这里其实整个儿Shader的思路都已经总结的差不多了，至于真正的什么哪一层怎么处理就出什么效果？应该用几层？什么参数？没有什么太多意义，交给特效同学多配制几次就好了。

**小彩蛋**

其实，“移动”效果还有相对高级一点的用法，它的效果看起来像这样：



如果你用心眼去看的话，会发现这哥们手中的字儿是呈**弧形**的，而我们的图是这样的：



其实原理也很简单，只是用了一张**Motion贴图来模拟uv**，看一下他在PS中的样子你估计就明白了：



**一句话总结就是：用渐变的颜色组成一张画布，这张画布就是渲染时UV采样的参照(0~1)，扭曲画布，绘制效果自然也就扭曲了。**

如果你用了Motion贴图，**一定记得：要狠狠的减小他的尺寸，但不要压缩它的品质，更不要用Point过滤。因为：插值出来的“UV”比你在PhotoShop中通过“渐变”得来的“UV”平滑且靠谱的多。**

