

## LSTG+ shader 的使用方法

这里假设我们已经有一系列的 shader 文件了，shader 文件是.fx 后缀的例如 shader1.fx, shader2.fx, shader3.fx.....。

### LSTG+使用 shader 的函数介绍

PushRenderTarget(texturename)

PopRenderTarget(texturename)

PostEffect(texturename, shadername,[shadervar1, shadervar2.....])

在 LSTG+中，要是用 shader，必须知道上述三个函数

#### (1). PushRenderTarget(texturename)

开辟一个名叫 texturename 的纹理渲染区域，能使调用这个函数以后的渲染行为均在该区域进行。

#### (2). PopRenderTarget(texturename)

关闭名叫 texturename 的纹理渲染缓冲区域，并输出一个名叫 texturename 的纹理。该纹理等价于 LSTG 中 loadtexture 得到的纹理。可以使用这块纹理进行后续的图象渲染。

#### (3). PostEffect(texturename, shadername,[shadervar1, shadervar2.....])

将名为 shadername 的 shader 作用于名为 texturename 的纹理上，该纹理应当为(1), (2)步骤之后的纹理（其实其加载的纹理能不能直接这么用我还没试过 XD）。后续的参数为 shader 使用时所需要的参数，这些必要的参数可以再.fx 文件中查找得到。

### LSTG+ shader 使用方法

函数(1)与函数(2)必须成对调用，函数(1)为开启渲染缓冲区域，函数(2)为结束渲

染区域，类似于 `do……end` 这样成对使用。函数(1)与函数(2)之间的部分为渲染行为，在函数(1)与(2)之间的渲染均在开辟的缓冲区内完成。所以函数(1)与(2)必须写在 `object` 的 `on render` 函数中。

例如：一个简单的例子

```
1
2  obj1=Class(object)
3
4  =function obj1:init()
5      .....
6      .....
7      .....
8      self.img='leaf'
9      .....
10 end
11
12 =function obj1:frame()
13     object.frame(self)
14 end
15
16 =function obj1:render()
17     SetViewMode'world'
18     .....
19     PushRenderTarget("test_texture") --开启渲染缓冲区
20     .....
21     Render(self.img,0,0)
22     .....
23     PopRenderTarget("test_texture") --结束渲染缓冲区，这样就得到了一个名为test_texture的纹理
24     |
25     PostEffect("test_texture","test_shader",.....)-- 将之前已经加载好的fx文件作用于该纹理。
26 end
```

这样便完成了一次 `shader` 的使用了。

## 结束语

这则简单的教程展示了 `LSTG+`中使用 `shader` 的基本函数与功能，可能会有小错误，请大家实践后告诉我，以便我改正。

——丽丽，2017 年 4 月 19 日星期三