

Unity Shader_3-固定函数着色器

主要参考：《Unity shader 入门精要 著冯乐乐》

作者博客 <http://blog.csdn.net/candycat1992/article/details/50167285>

浅墨博客 http://blog.csdn.net/poem_qianmo/article/details/49556461

风宇冲博客 http://blog.sina.com.cn/s/blog_471132920101d5kh.html

潜水的小懒猫博客 http://blog.sina.com.cn/s/articlelist_2312702844_6_1.html

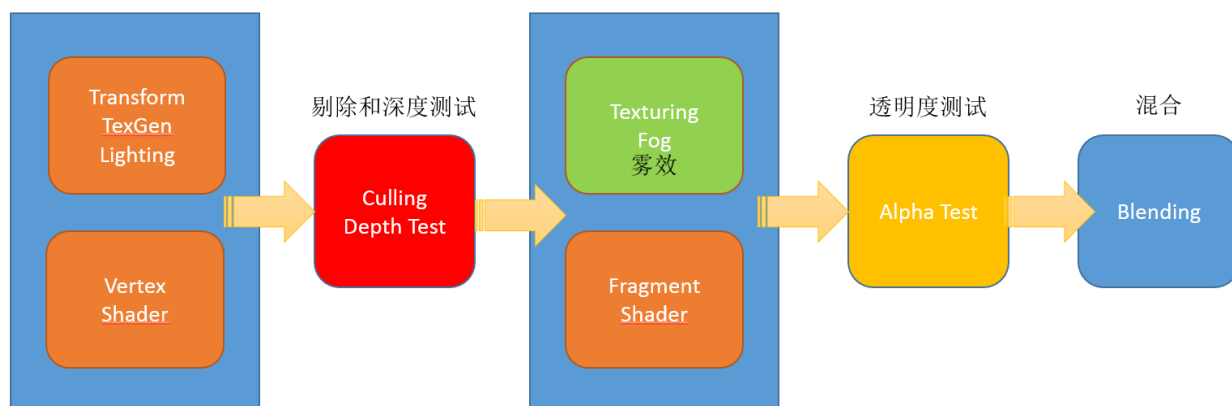
猫都能学会的Unity3D Shader <https://onevc.com/2013/07/shader-tutorial-1/>
<https://onevc.com/2013/08/shader-tutorial-2/>

蛮牛教育Shader编程教程 <http://edu.manew.com/course/96>

Unity官方; <http://docs.unity3d.com/Manual/SL-ShaderPrograms.html>

一、固定着色器——剔除和测试

以下是渲染管线中**Shader渲染**的大致流程



· **剔除（Culling）**：将被挡住或者视野以外的不可见的物体，根据优化需求，可以选择不绘制，以节省资源。

· **深度测试（Depth Testing）**：多个物体的遮挡，离相机的Z轴方向的距离大小

剔除和深度测试的一些关键词：

· Cull Back / Front / Off

- Cull Back——不绘制背离观察者的几何面
- Cull Front——不绘制面向观察者的几何面，用于由内自外的旋转对象
- Cull Off ——显示所有面。用于特殊效果。

· ZWrite On /Off

（是否和深度缓冲区内的深度值做比较）

用于控制是否将来之对象的像素写入深度缓冲（默认开启），

如果需要绘制纯色物体，便将此项打开，也就是写上ZWrite On。

如果要绘制半透明效果，关闭深度缓冲，则用ZWrite Off。

· ZTest Less/ Greater / LEqual / GEqual / Equal / NotEqual /Always

用于控制深度测试如何执行。

缺省值是LEqual （绘制和存在的对象一致或是在其中的对象；隐藏其背后的对象）

Greater	只渲染大
---------	------

	于 AlphaVal ue值的像 素
GEqual	只渲染大 于等于 AlphaVal ue值的像 素
Less	只渲染小 于 AlphaVal ue值的像 素
LEqual	只渲染小 于等于 AlphaVal ue值的像 素
Equal	只渲染等 于 AlphaVal ue值的像 素
NotEqual	只渲染不 等于 AlphaVal ue值的像 素
Always	渲染所有 像素，等 于关闭透 明度测 试。等于 用 AlphaTes t Off
Never	不渲染任 何像素

• **Offset Factor , Units**

此语句用两个参数（Facto和Units）来定义深度偏移。

- Factor参数表示 Z缩放的最大斜率的值。
- Units参数表示可分辨的最小深度缓冲区的值。

于是，我们就可以强制使位于同一位置上的两个集合体中的一个几何体绘制在另一个的上层。例如偏移量Offset 设为0, -1（即Factor=0, Units=-1）的值使得靠近摄像机的几何体忽略几何体的斜率，而偏移量为-1,-1（即Factor=-1, Units=-1）时，则会让几何体偏移一个微小的角度，让观察使看起来更近些。

· **透明度测试（Alpha Testing）**：在unity中是阻止像素被写到屏幕的最后机会；

在片元计算出最终需要渲染的颜色之后，可以通过将颜色的透明度和一个固定值进行比较，如果比较失败，则该像素不会显示输出。

Alpha测试在渲染凹形物体的透明部分时非常有用。显卡上有着每个像素写到屏幕上的深度记录。如果一个新的像素比原来的像素的深度深，那么新的像素就不会被写到屏幕中。

透明度测试的一些关键词：

· **AlphaTest Off**用于渲染所有像素（默认缺省）

· **AlphaTest comparison AlphaValue**

用于设定透明度测试只渲染在某一确定范围内的透明度值的像素。其中的comparison取值为下表之一：

而AlphaValue为一个范围在0到1之间的浮点值。也可以是一个指向浮点属性或是一个范围属性，在后一种情况下需要使用标准的方括号写法标注索引名字，如(**[变量名]**)。

Greater	Only render pixels whose alpha is greater than <i>AlphaValue</i> . 大于
GEqual	Only render pixels whose alpha is greater than or equal to <i>AlphaValue</i> . 大于等于
Less	Only render pixels

	whose alpha value is less than <i>AlphaValue</i> . 小于
LEqual	Only render pixels whose alpha value is less than or equal to from <i>AlphaValue</i> . 小于等于
Equal	Only render pixels whose alpha value equals <i>AlphaValue</i> . . 等于
NotEqual	Only render pixels whose alpha value differs from <i>AlphaValue</i> . 不等于
Always	Render all pixels. This is function

	ally equivale nt to <i>Alpha Test Off</i> . 渲染所有 像素，等 于关闭透 明度测试 AlphaTes t Off
Never	Don't render any pixels. 不 渲染任何 像素

· 基本雾效 (Fog)

雾效 (Fog) 参数用于控制雾相关的渲染

在计算机图形学中，雾化是通过混合已生成的像素的颜色和基于到镜头的距离来确定的最终的颜色来完成的。雾化不会改变已经混合的像素的透明度值，只是改变RGB值。

Unity中的缺省雾设定是基于Edit->RenderSettings中的设置的，且雾模式既可以是Exp2也可以是关闭；密度和颜色完全取自设定。

注意如果使用了片段着色器的话，着色器中的雾设定仍然是有效的。如果平台没有对固定管线的雾功能支持，Unity会实时补上着色器以支持需要的雾模式。

雾效的一些关键词：

· Fog { FogCommands }

用于设定雾命令的内容

FogCommands :

· Mode Off / Global / Linear / Exp / Exp2

用于定义雾模式。缺省是全局的，依据雾在渲染设定中是否打开确定可从无变化到平方值

· Color ColorValue

用于设定雾的颜色值

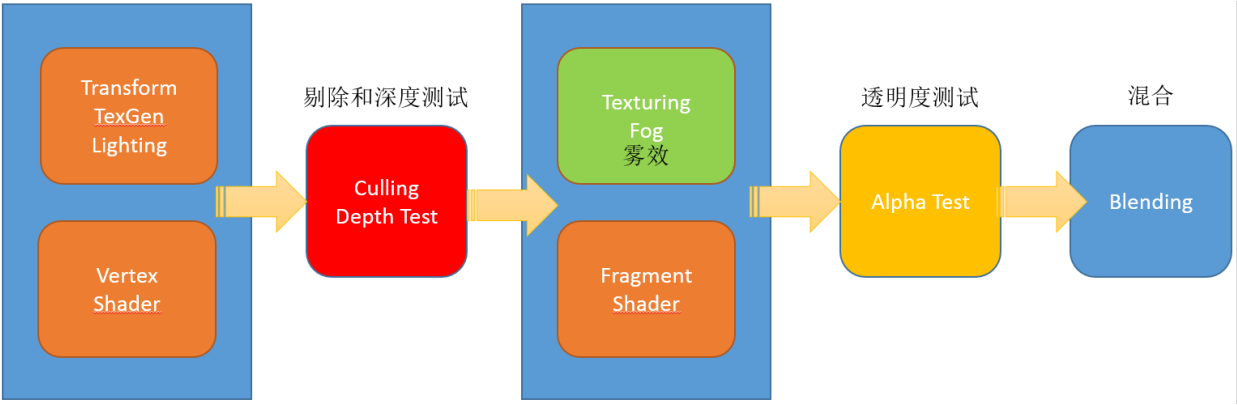
· Density FloatValue

以指数的方式设定雾的密度

· Range FloatValue,FloatValue

用于为linear类型的雾设定远近距离值

· 混合（Blend）：Shader渲染的最后一步
常用于制作透明物体、进行纹理混合



如上图所示，正被渲染的像素经过顶点光照、顶点着色器、剔除和深度测试，雾效、Alpha测试等一系列操作之后，最后一步便是混合操作。这个时候计算结果即将被输出到帧缓冲中。而混合操作，就是管理如何将这些像素输出到帧缓存中的这样一个过程——是直接替换原来的，是一加一的混合，还是有Alpha参与的不等比地混合等等。混合操作有两个对象：源和目标，因此也有两个对应的因子，即源因子和目标因子（对应于Blend SrcFactor DstFactor操作）。而如果我们把RGB颜色通道和Alpha通道分开来操作的话，混合就有了4个操作对象（对应于Blend SrcFactor DstFactor,SrcFactorA DstFactorA操作）。

混合操作的关键词：

Blend Off

Turn off blending 关闭混合

Blend SrcFactor DstFactor

基本的配置并启动混操作。

对产生的颜色乘以SrcFactor.对 已存在于屏幕的颜色乘以DstFactor，并且两者将被叠加在一起。

Blend SrcFactor DstFactor, SrcFactorA DstFactorA

同上，但是使用不同的要素来混合alpha通道，也就是有了4个操作对象

BlendOp Add |Min | Max | Sub | RevSub

此操作不和Blend操作一样 添加混合颜色在一起，而是 对它们做不同的操作。

而如下便是常用混合操作符（blend operations）的含义列举：

Add	将源像素和目标像素相加.
Sub	用源像素减去目标像素
RevSub	用目标像素减去源像素
Min	取目标像

	素和源像素颜色的较小者作为结果
Max	取目标像素和源像素颜色的较大者作为结果

混合因子（Blend factors）：

以下所有的属性都可作为**SrcFactor**或**DstFactor**。其中，Source指的是被计算过的颜色，Destination是已经在屏幕上的颜色。

One	值为1，使用此因子来让帧缓冲区源颜色或是目标颜色完全通过。
Zero	值为0，使用此因子来删除帧缓冲区源颜色或目标颜色的值。
SrcColor	使用此因子为将当前值乘以帧缓冲区源颜色的值
SrcAlpha	使用此因子为将当前值乘以帧缓冲区源颜色Alpha的值。
DstColor	使用此因子为将当前值乘以帧缓冲区源颜色的值。（修改为屏幕上的颜色）
DstAlpha	使用此因子为将当前值乘以帧缓冲区源颜色Alpha分量的值。（修改为屏幕上的颜色Alpha分量）
OneMinusSrcColor	使用此因子为将当前值乘以(1 -帧缓冲区源颜色值)
OneMinusSrcAlpha	使用此因子为将当前值乘以(1 -帧缓冲区源颜色Alpha分量的值)
OneMinusDstColor	使用此因子为将当前值乘以(1 –目标颜色值)
OneMinusDstAlpha	使用此因子为将当前值乘以(1 –目标Alpha分量的值)

在Pass{}中使用如下：

```
1. Blend SrcAlpha OneMinusSrcAlpha // Alpha混合
2. Blend One One // 相加
3. Blend One OneMinusDstColor // 比较柔和的相加 (SoftAdditive)
4. Blend DstColor Zero // 乘法
5. Blend DstColor SrcColor // 2倍乘法
```