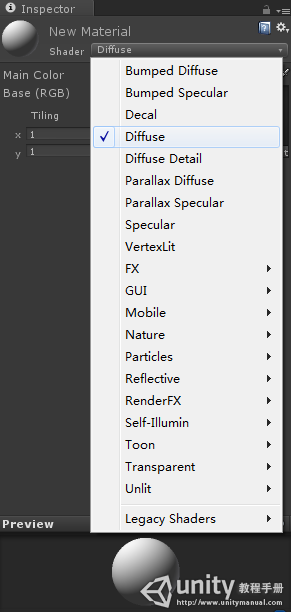
**Unity3D教程：着色器**

Posted on 2013年01月11日 by U3d / [Unity3D 基础教程](http://www.unitymanual.com/category/manual/unity3d-%e5%9f%ba%e7%a1%80%e6%95%99%e7%a8%8b)/被围观 526 次

如下图所示，在Project面板中新建一个New Material.这样子就准备好了一个空白的着色器。它能支持的类型也就在Shader中全部显示出来。不得不说，很好很强大。



Unity3D教程：着色器

所有这些Pixel[像素]着色器，默认的Diffuse算是资源开销较小的渲染选择。随后那些带有凹凸贴图的着色器，其次就是Bumped Specular[凹凸高光]，Parallax Specular[视差高光]类等开销大。所有基于顶点光照的着色器开销都比Pixel着色器的开销小。但是相对来说效果可能不是那么强大，毕竟打造好的视觉效果增大开销是在所难免。

着色器共有的主要属性：

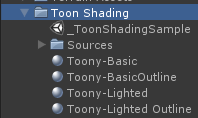
Diffuse[散射]：定义对象的基本颜色，可以使用贴图或者简单的颜色选择器来控制散射，所有的着色器都带有某种类型的散射属性。

Bump and Parallax Bump[凹凸和视差凹凸]：名称中带有Bump或者Parallax的着色器都带有的一个属性，使得他们可以模拟对象的高度和深度。比如我们需要对对象的每一个伤痕或者细节进行建模，这样很耗费资源，此时聚划算的方法就是使用凹凸贴图来模拟。

Specular[高光]：带有Specular属性的着色器都可以使对象在光源照射下具有漂亮的光泽，注意这和反射是不同的。

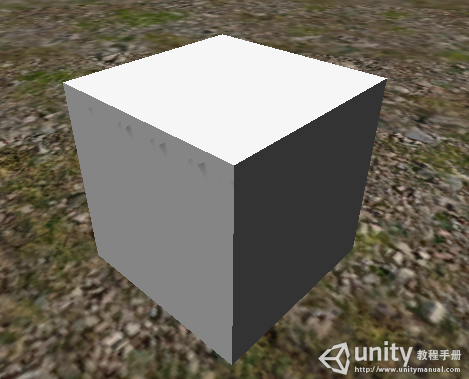
下面，创建一个吧：

1：先把ToonShading



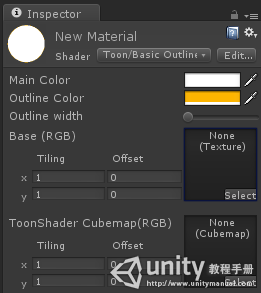
Unity3D教程：着色器

导入Project，在Hierarchy面板中创建一个Cube，此时看到的只是个纯色的方体。



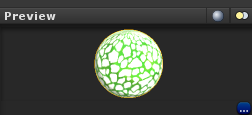
Unity3D教程：着色器

2：在Project面板中创建一个Materials，将着色器的Shader选为：Toon—〉BasicOutLine。

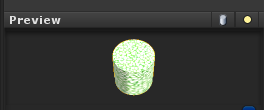


Unity3D教程：着色器

3：点击上图中的Base(RGB)右侧的方框的Select按钮，选一个你喜欢的图片吧。下边的那一个ToonShader Cubemap的Select选toony lighing。这样子我们的着色器就可以使用了。预览一下：点击Preview那一栏的右边的原型按钮，可以变换模型哦。你可以对模型进行各个方向的拖拉查看。

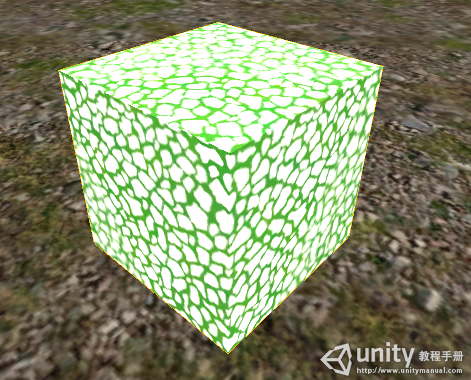


Unity3D教程：着色器



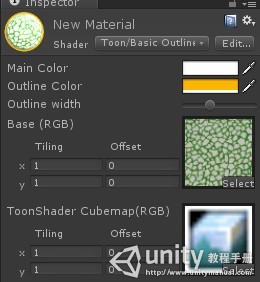
Unity3D教程：着色器

4：把我们Project面板的着色器直接拖给场景的Cube物体。

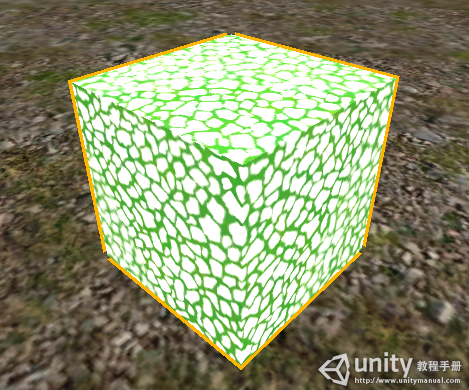


Unity3D教程：着色器

我把outlineColor和outlinewidth都改一下。



Unity3D教程：着色器



Unity3D教程：着色器

关于天空盒[SkyBox]的着色器，就是Shader下的RenderFx/SkyBox。选择他以后你会看到下边列出了Front—Back—Left—Right—Up—Down的每个填充项。你用对应的天空贴图填充，就可以得到一个天空盒着色器了。

动态表面效果着色器脚本：



Unity3D教程：着色器

Shader "Effects/Plasma" {  
Properties {  
\_MainTex ("Base (RGB)", RECT) = "white" {}  
\_ColorBand ("Base (RGB)", 2D) = "white" {}  
}  
SubShader {  
ZTest Always Cull Off ZWrite Off  
Fog { Mode off }  
Pass {  
CGPROGRAM  
#pragma vertex vert\_img  
#pragma fragment fragSh  
#pragma fragmentoption ARB\_precision\_hint\_fastest  
#include "UnityCG.cginc"  
uniform samplerRECT \_MainTex;  
uniform sampler2D \_ColorBand;  
float4 fragSh (v2f\_img i) : COLOR  
{  
float2 ca = float2(0.2, 0.2);  
float2 cb = float2(0.7, 0.9);  
float da = distance(i.uv, ca);  
float db = distance(i.uv, cb);  
float c1 = sin(da \* \_CosTime.y \* 16 + \_Time.x);  
float c2 = cos(i.uv.x \* 8 + \_Time.z);  
float c3 = cos(db \* 14) + \_CosTime.x;  
float p = (c1 + c2 + c3) / 3;  
return tex2D(\_ColorBand, float2(p, 0.5));  
}  
ENDCG  
}  
}  
Fallback off  
}