

Unity Shader_5-----CG内置函数

CG函数：

UnpackNormal()函数

UnpackNormal接受一个fixed4的输入，并将其转换为所对应的法线值（fixed3），并将其赋给输出的Normal，就可以参与到光线运算中完成接下来的渲染工作了。

一个调用示例：

```
o.Normal = UnpackNormal (tex2D (_BumpMap,IN.uv_BumpMap));
```

saturate()函数

saturate的字面解释是浸湿，浸透。其作用其实也就是将取值转化为[0,1]之内的一个值。

其可选的原型如下：

1. float saturate(float x);
2. float1 saturate(float1 x);
3. float2 saturate(float2 x);
4. float3 saturate(float3 x);
5. float4 saturate(float4 x);
6. half saturate(half x);
7. half1 saturate(half1 x);
8. half2 saturate(half2 x);
9. half3 saturate(half3 x);
10. half4 saturate(half4 x);
11. fixed saturate(fixed x);
12. fixed1 saturate(fixed1 x);
13. fixed2 saturate(fixed2 x);
14. fixed3 saturate(fixed3 x);
15. fixed4 saturate(fixed4 x);

其唯一的一个参数x表示矢量或者标量的饱和值（Vector or scalar to saturate.），也就是将这个x转化为[0,1]之内的值。

其返回值：

- 如果x取值小于0，则返回值为0.
- 如果x取值大于1，则返回值为1.
- 若x在0到1之间，则直接返回x的值。

其代码实现大致如下：

```
float saturate(float x)
```

```
{  
    return max(0,min(1, x));  
}
```

一个调用示例：

```
half rim = 1.0 - saturate(dot(normalize(IN.viewDir), o.Normal));
```

dot()函数

dot函数顾名思义，是高等数学中的点积操作，用于返回两个向量的标量积。

可选原型如下：

1. **float** dot(**float** a, **float** b);
2. **float** dot(float1 a, float1 b);
3. **float** dot(float2 a, float2 b);
4. **float** dot(float3 a, float3 b);
5. **float** dot(float4 a, float4 b);
6. half dot(half a, half b);
7. half dot(half1 a, half1 b);
8. half dot(half2 a, half2 b);
9. half dot(half3 a, half3 b);
10. half dot(half4 a, half4 b);
11. fixed dot(fixed a, fixed b);
12. fixed dot(fixed1 a, fixed1 b);
13. fixed dot(fixed2 a, fixed2 b);
14. fixed dot(fixed3 a, fixed3 b);
15. fixed dot(fixed4 a, fixed4 b);

其代码实现大致是这样的：

```
float dot(float4 a, float4 b)  
{  
    return a.x*b.x + a.y*b.y + a.z*b.z + a.w*b.w;  
}
```

一个调用示例：

```
float answer= dot (normalize(IN.viewDir),o.Normal);
```

tex2D()函数

让我们看一看CG中用得比较多的用于2D纹理采样的tex2D函数的用法。其备选的原型也是非常之多：

1. float4 tex2D(sampler2D samp, float2 s)
2. float4 tex2D(sampler2D samp, float2 s, int texelOff)
3. float4 tex2D(sampler2D samp, float3 s)
4. float4 tex2D(sampler2D samp, float3 s, int texelOff)
5. float4 tex2D(sampler2D samp, float2 s, float2 dx, float2 dy)
6. float4 tex2D(sampler2D samp, float2 s, float2 dx, float2 dy, int texelOff)
7. float4 tex2D(sampler2D samp, float3 s, float2 dx, float2 dy)
8. float4 tex2D(sampler2D samp, float3 s, float2 dx, float2 dy, int texelOff)
9. int4 tex2D(isampler2D samp, float2 s)
10. int4 tex2D(isampler2D samp, float2 s, int texelOff)
11. int4 tex2D(isampler2D samp, float2 s, float2 dx, float2 dy)
12. int4 tex2D(isampler2D samp, float2 s, float2 dx, float2 dy, int texelOff)
13. unsigned int4 tex2D(usampler2D samp, float2s)
14. unsigned int4 tex2D(usampler2D samp, float2s, int texelOff)
15. unsigned int4 tex2D(usampler2D samp, float2s, float2 dx, float2 dy)
16. unsigned int4 tex2D(usampler2D samp, float2s, float2 dx, float2 dy, int texelOff)

参数简介：

samp - 需要查找的采样对象，也就是填个纹理对象在这里。

s - 需进行查找的纹理坐标。

dx - 预计算的沿X轴方向的导数。

dy - 预计算的沿Y轴方向的导数。

texelOff - 添加给最终纹理的偏移量

而其返回值，自然是查找到的纹理。

最后，看一个综合了本次讲解的四个函数（UnpackNormal、saturate、tex2D、dot）的Surface Shader中surf函数的示例：

```
// 【2】 表面着色函数的编写
```

```
void surf (Input IN, inout SurfaceOutput o)
```

```
{  
    //从主纹理获取rgb颜色值  
    o.Albedo= tex2D (_MainTex, IN.uv_MainTex).rgb;  
    //从凹凸纹理获取法线值  
    o.Normal= UnpackNormal (tex2D (_BumpMap, IN.uv_BumpMap));  
    //从_RimColor参数获取自发光颜色  
    half rim = 1.0 - saturate(dot (normalize(IN.viewDir), o.Normal));  
    o.Emission= _RimColor.rgb * pow (rim, _RimPower);  
}
```

