<https://blog.csdn.net/puppet_master/article/details/53548134>



所谓的RimLight边缘发光，也就是说对应我们当前视角方向，物体上位于边缘的地方额外加一个光的效果。那么，怎么判断一个点是否在物体的边缘呢？就是通过法线方向和视线方向的夹角来判断，当视线方向V与法线方向N垂直时，这个法线对应的面就与视线方向平行，说明当前这个点对于当前视角来说，就处于边缘；而当视线方向与法线方向一致时，这个法线对应的面就垂直于视线方向，说明当前是直视这个面，所有，我们就可以根据dot(N,V)来获得视线方向u法线方向的余弦值，通过这个值来区分给像素是否处在边缘，进而判断是否需要增加以及增加边缘光的强弱。

Shader "Unlit/XRay"

{

Properties

{

\_MainTex ("Texture", 2D) = "white" {}

\_RimColor ("Rim Color", Color) = (1.0, 1.0, 1.0, 1.0)

\_RimPower ("Rim Power", Range(0.01, 3.0)) = 0.1

}

SubShader

{

Tags { "Queue" = "Geometry" "RenderType" = "Opaque" }

Pass

{

ZTest LEqual

ZWrite On

CGPROGRAM

#pragma vertex vert

#pragma fragment frag

#include "UnityCG.cginc"

struct v2f

{

float4 pos : SV\_POSITION;

float2 uv : TEXCOORD0;

float4 worldPos : TEXCOORD1;

float3 worldViewDir : TEXCOORD2;

float3 worldNormalDir : TEXCOORD3;

};

sampler2D \_MainTex;

float4 \_MainTex\_ST;

fixed4 \_RimColor;

float \_RimPower;

v2f vert(appdata\_base v)

{

v2f o;

o.pos = UnityObjectToClipPos(v.vertex);

o.uv = TRANSFORM\_TEX(v.texcoord, \_MainTex);

o.worldPos = mul(unity\_ObjectToWorld, v.vertex);

o.worldViewDir = UnityWorldSpaceViewDir(o.worldPos);

o.worldNormalDir = UnityObjectToWorldNormal(v.normal);

return o;

}

fixed4 frag(v2f i) : SV\_Target

{

fixed3 worldViewDir = normalize(i.worldViewDir);

fixed3 worldNormalDir = normalize(i.worldNormalDir);

float cos = saturate(dot(worldViewDir, worldNormalDir));

float rim = pow(cos, \_RimPower);

fixed4 col = tex2D(\_MainTex, i.uv) \* rim + \_RimColor \* (1 - rim);

return col;

}

ENDCG

}

}

}

