<https://blog.csdn.net/puppet_master/article/details/73478905>

**X光效果**

原理：人物通过两个Pass渲染，第一个Pass渲染X光的Pass，这个Pass在渲染的时候用ZTest Greater，在渲染的时候，有遮挡的部分会被显示出来，而未被遮挡的部分由于深度为1测试不会通过，，第二个Pass正常渲染，设置ZTest LEqual，被遮挡的部分不渲染，值渲染被遮挡的部分。我们先用一个shader+边缘光的效果测试一下效果

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+1" "RenderType" = "Opaque" }

// 被遮挡部分

Pass

{

// 需要混合，这里的目标颜色不要用1-SrcAlpha，因为它仅仅只是展示人物，并非要两者半透明效果

Blend SrcAlpha One

ZTest Greater

ZWrite Off // 此时不能写入，如果大于的时候写入，在第二个pass就会将整个人物全部渲染出来

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 = \_RimColor \* (1 - rim);

return col;

}

ENDCG

}

// 未被遮挡部分

Pass

{

ZTest LEqual

ZWrite On

CGPROGRAM

#pragma vertex vert

#pragma fragment frag

#include "UnityCG.cginc"

struct v2f

{

float2 uv : TEXCOORD0;

float4 vertex : SV\_POSITION;

};

sampler2D \_MainTex;

float4 \_MainTex\_ST;

v2f vert (appdata\_base v)

{

v2f o;

o.vertex = UnityObjectToClipPos(v.vertex);

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

return o;

}

fixed4 frag (v2f i) : SV\_Target

{

return tex2D(\_MainTex, i.uv);

}

ENDCG

}

}

}

