<https://blog.csdn.net/u011047171/article/details/48807105>

碎屏特效主要原理就是通过一张法线贴图存储的值对屏幕UV坐标进行扭曲即可

Shader文件：

Shader "Unlit/BrokenScreen"

{

Properties

{

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

\_BumpMap ("Bump Map", 2D) = "bump" {}

\_BumpScale ("Bump Scale", Float) = 0.5

}

SubShader

{

Tags { "RenderType"="Opaque" }

Pass

{

Tags { "LightMode" = "ForwardBase" }

Cull Off

ZWrite Off

ZTest Always

CGPROGRAM

#include "UnityCG.cginc"

#pragma vertex vert

#pragma fragment frag

struct v2f

{

float4 pos : SV\_POSITION;

fixed4 uv : TEXCOORD0;

};

sampler2D \_MainTex;

float4 \_MainTex\_ST;

sampler2D \_BumpMap;

float4 \_BumpMap\_ST;

fixed4 \_BumpMap\_TexelSize;

float \_BumpScale;

float \_Saturation;

v2f vert(appdata\_img v)

{

v2f o;

UNITY\_INITIALIZE\_OUTPUT(v2f, o);

o.pos = UnityObjectToClipPos(v.vertex);

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

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

return o;

}

fixed4 frag(v2f i) : SV\_TARGET

{

fixed2 bump = UnpackNormal(tex2D(\_BumpMap, i.uv.zw)).rg;

// 这里主要是大片的正常区域解析得到的bump是(1,1,255)，为了不影响正常采样，需要偏移一个纹素

i.uv.xy = (bump - \_BumpMap\_TexelSize.xy) \* \_BumpScale + i.uv.xy;

fixed4 col = tex2D(\_MainTex, i.uv.xy);

fixed4 lum = Luminance(col);

col = lerp(col, lum, \_Saturation);

return fixed4(col);

}

ENDCG

}

}

}

C#文件：

[ExecuteInEditMode]

public class BrokenScreen : MonoBehaviour

{

public Shader brokenScreenShader;

private Material brokenScreenMaterial = null;

public Material material

{

get

{

if (brokenScreenMaterial) return brokenScreenMaterial;

brokenScreenMaterial = new Material(brokenScreenShader);

return brokenScreenMaterial;

}

}

[Range(0.0f, 2.0f)]

public float bumpScale = 0.5f;

[Range(0.0f, 1.0f)]

public float saturation = 0.5f;

void OnRenderImage(RenderTexture src, RenderTexture dest)

{

if (material)

{

material.SetFloat("\_BumpScale", bumpScale);

material.SetFloat("\_Saturation", saturation);

Graphics.Blit(src, dest, material);

}

else

{

Graphics.Blit(src, dest);

}

}

}



