

LightBridge

美术需要一个透明、发光、流动的光桥，能够沿UV方向从隐藏到出现。

效果展示



参数说明

1. 主贴图
2. 主贴图颜色
3. 主贴图流动速度
4. LayerX: 第X层发光贴图
5. LayerX颜色: 第X层颜色
6. 噪声流动速度: 第X层流动速度
7. 遮罩: rgb三通道对应Layer1~3
8. 噪声: 扰动噪声
9. 噪声强度
10. 噪声流动速度
11. 是否应用噪声遮罩: 将扰动噪声同时作为颜色遮罩, 应用
12. 渐变进度: 沿UV的V方向延展的百分比
13. 渐变宽度
14. 渐变头部颜色: 控制渐变头部的颜色(乘法)

技术介绍

1. 发光: 使用四层发光纹理和一个遮罩图(rgb分别控制后三层发光纹理), 再用一个噪声贴图来扭曲发光纹理

```
// 避免时间过大
float t = fmod(_Time.y, 100.0);
float2 q = 0;
// 多层噪声嵌套得到流动的噪声(噪声图最好小点, 有点费)
q.x = tex2D(_NoiseTex, i.uv0.zw).r;
q.y = tex2D(_NoiseTex, i.uv0.zw + float2(0.37, 0.59)).r;
float p = tex2D(_NoiseTex, i.uv0.zw + q + float2(1.7, 9.2)) + t
* float2(_NoiseFlow.xy)).r;
// 遮罩
half3 mask = tex2D(_MaskTex, i.uv0.xy).rgb;
float3 mainCol = tex2D(_MainTex, i.uv2.zw).rgb *
_MainTexColor.rgb;
// 后三层受噪声扭曲
float3 layer1 = tex2D(_Layer1Tex, i.uv1.xy +
_NoiseIntensity*p).rgb * _Layer1Color.rgb;
float3 layer2 = tex2D(_Layer2Tex, i.uv1.zw +
_NoiseIntensity*p).rgb * _Layer2Color.rgb;
float3 layer3 = tex2D(_Layer3Tex, i.uv2.xy +
_NoiseIntensity*p).rgb * _Layer3Color.rgb;
// .....
// 后三层受遮罩控制
half4 col = half4((mainCol + layer1 * mask.r + layer2 *
mask.g + layer3 * mask.b) * fade, 1);
```

2. 沿UV的V方向控制光桥延展方向

```
float fade =  
smoothstep(_GradientPercent+_Gradientwidth,_GradientPercent,i.uv0.y);  
//获取中间虚化的部分  
float fade2 =  
smoothstep(_GradientPercent,_GradientPercent+_Gradientwidth,i.uv0.y);  
float GradienFade = fade * fade2;
```