



唐老狮系列教程

消融效果 基本原理



唐老狮系列教程-消融效果 基本原理

主要讲解内容



唐老狮系列教程-消融效果 基本原理

主要讲解内容

1. 消融效果 是什么
2. 消融效果 基本原理



唐老狮系列教程-消融效果 基本原理

| 消融效果是什么



唐老狮系列教程-消融效果 基本原理

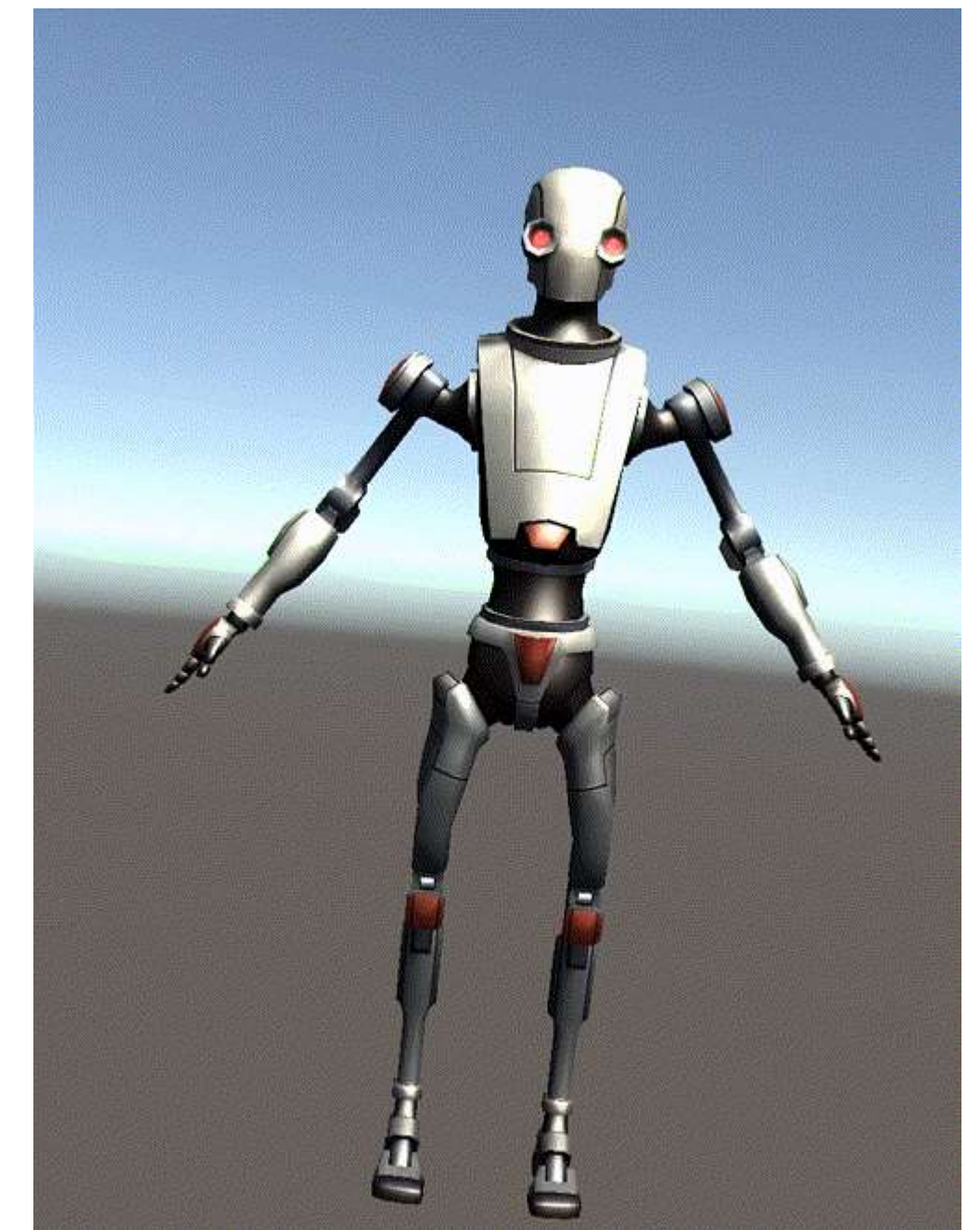
消融效果是什么

消融效果是模拟物体逐渐从屏幕上消失或溶解的过程

它通常利用**噪声纹理**实现，使物体按照某种规则逐渐透明或完全不可见。

这种效果常用于：

1. 角色死亡、传送场景
2. 魔法消失，比如燃烧、消失等





唐老狮系列教程-消融效果 基本原理

| 消融效果基本原理



唐老狮系列教程-消融效果 基本原理

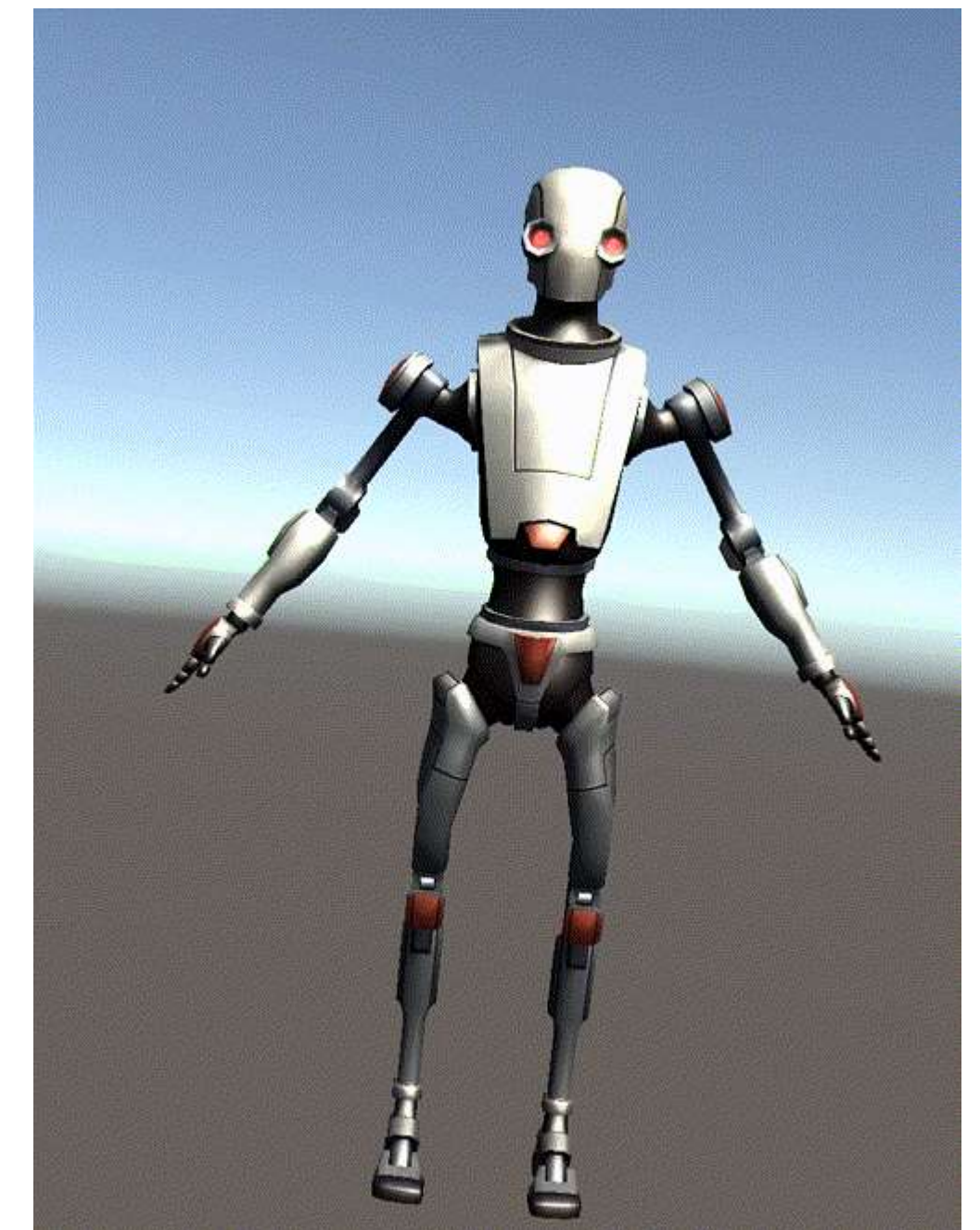
消融效果基本原理

一句话总结 消融效果 基本原理:

**通过对比噪声纹理值与消融进度参数，剔除低于阈值的像素，
同时在边缘添加渐变颜色实现动态溶解效果。**

关键点:

- 1. 如何剔除像素**
- 2. 如何处理边缘**





唐老狮系列教程-消融效果 基本原理

消融效果基本原理

1. 如何剔除像素

我们在片元着色器中**对噪声纹理进行采样**，由于噪声纹理是灰度图，只需**取出其中的RGB中的任意一通道的颜色来使用。**

再**自定义一个用于控制消融进度的参数** (0~1)

最后利用该片元的 噪声纹理值 - 进度参数

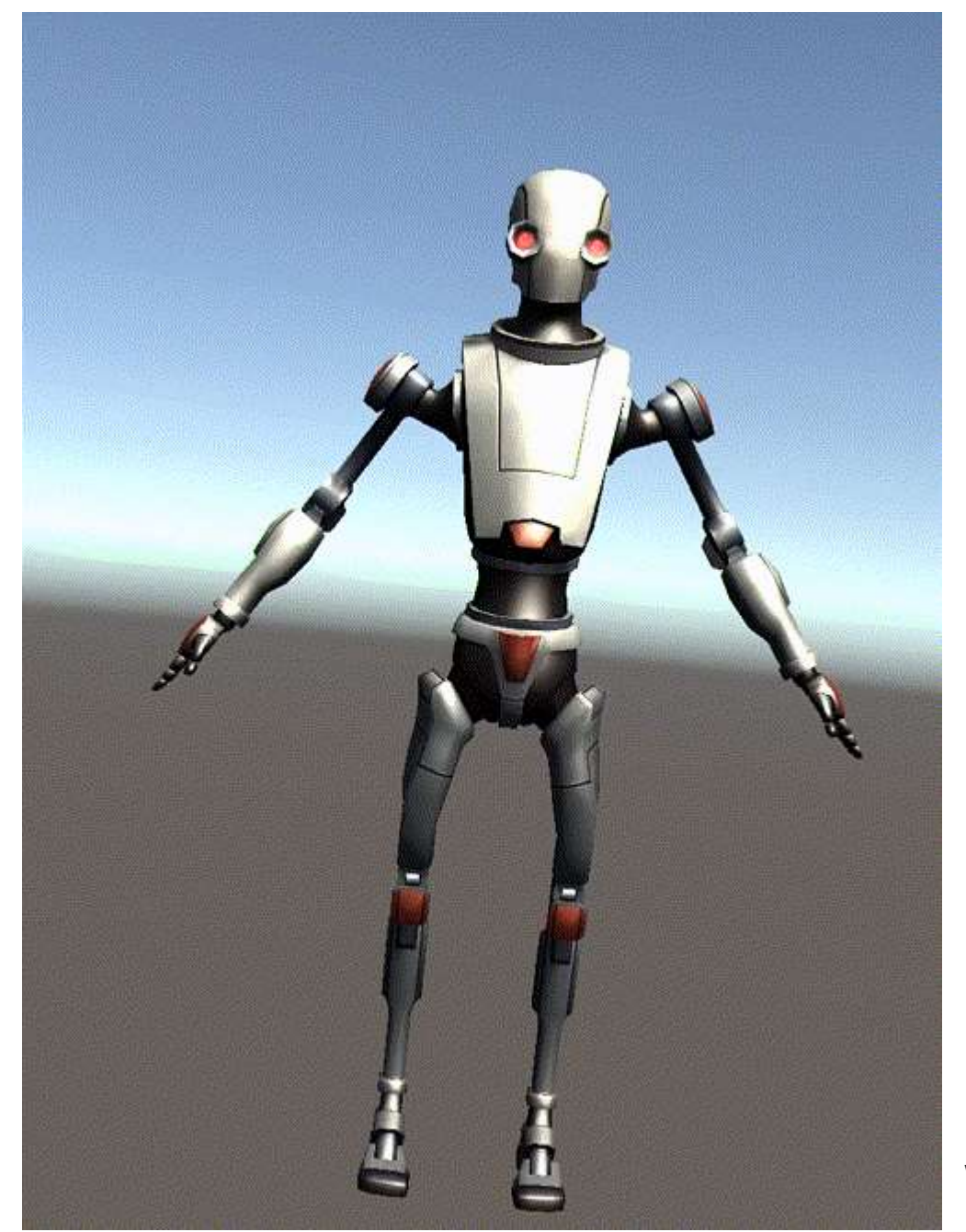
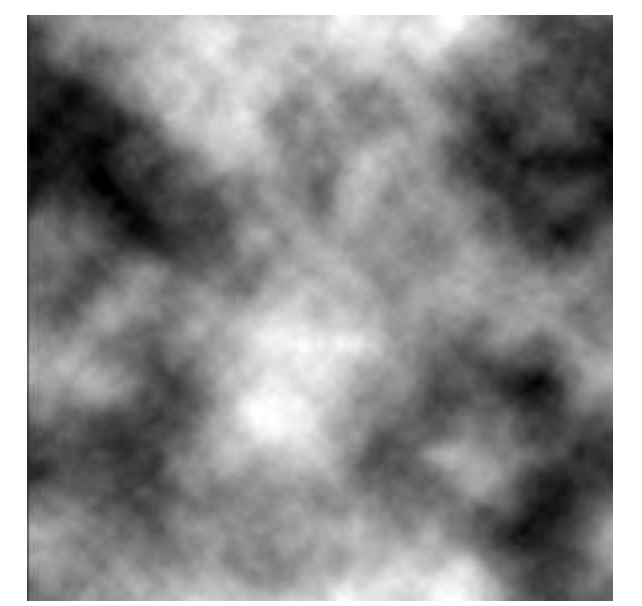
若小于0则不渲染该片元，通过控制进度参数

便可以控制消融程度了

```
fixed4 frag (v2f i) : SV_Target
{
    //噪声图上的颜色是线性随机的
    //我们通过一个消融阈值来控制
    //当颜色值(0~1)的值小于该阈值时，会抛弃对应像素不渲染
    //就会出现局部镂空效果
    fixed3 noiseColor = tex2D(_Noise, i.uv2.xy).rgb;
    clip(noiseColor.r - _Dissolve);
}
```

知识回顾:

Unity Shader中提供了一个内置函数 clip(x) 它的作用就是在片元着色器中调用时来丢弃片元的，我们之前在透明度测试时使用过。传入的值x小于0，则会丢弃当前片元，被丢弃的片元不会被进一步处理也就不会被渲染了





唐老狮系列教程-消融效果 基本原理

消融效果基本原理

2. 如何处理边缘

在处理边缘渐变颜色效果时，我们将使用Unity中内置的三个Shader函数：

1. smoothstep(a, b, x)

a起始值； b结束值； x输入值（用于在a和b之间平滑插值）

当 $x < a$ 时，返回0； 当 $x > b$ 时，返回1；

$a < x < b$ 时，返回0~1之间的值

2. lerp(a,b,t)

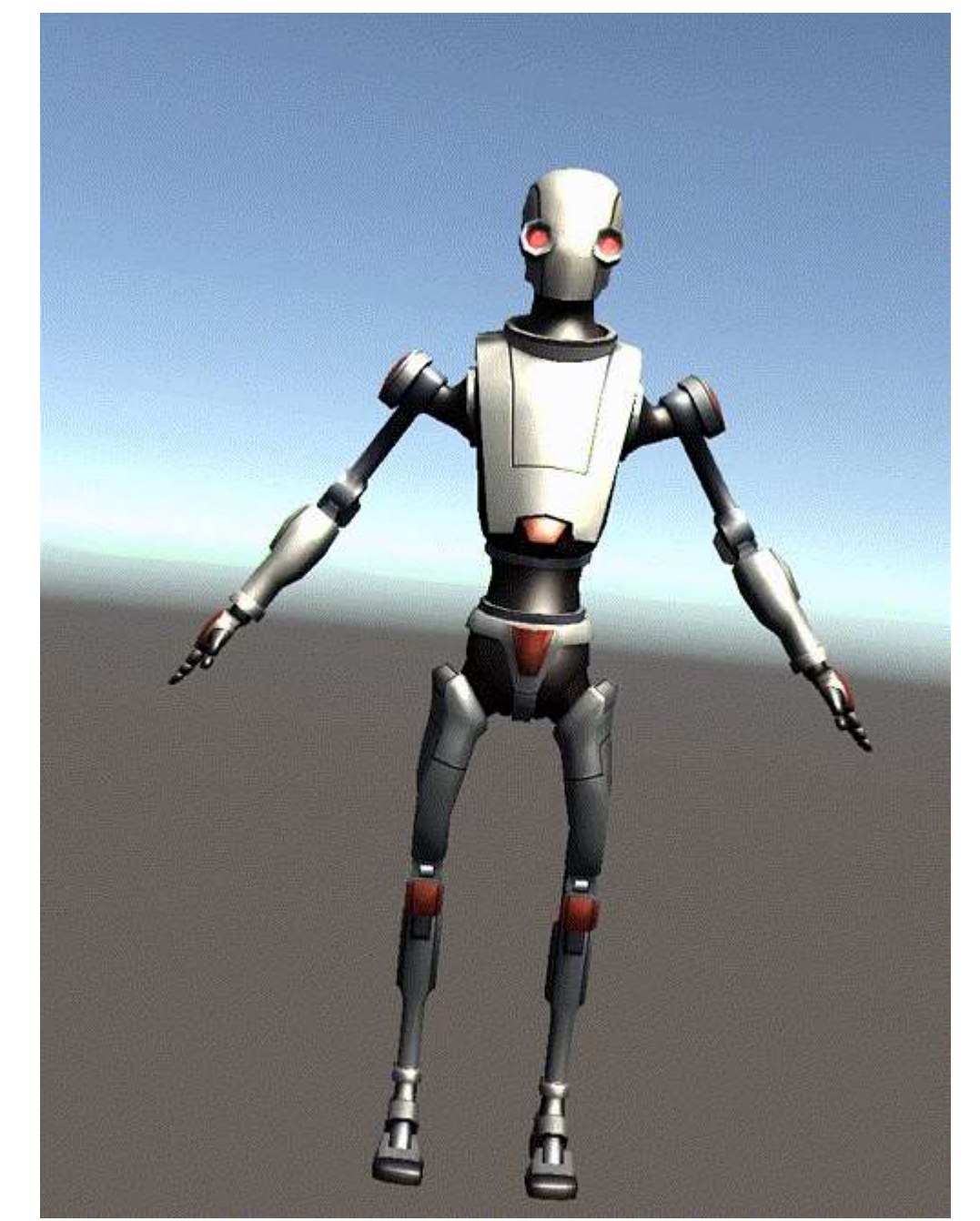
a起始值； b结束值； t插值因子

当 $t=0$ 时，返回a； 当 $t=1$ 时，返回b； 当 $0 < t < 1$ 时； 返回a和b之间的值

3.step(value, x)

value阈值， x输入值； 两值用于比较

$x < value$ ， 返回0； $x \geq value$ ， 返回1





唐老狮系列教程-消融效果 基本原理

消融效果基本原理

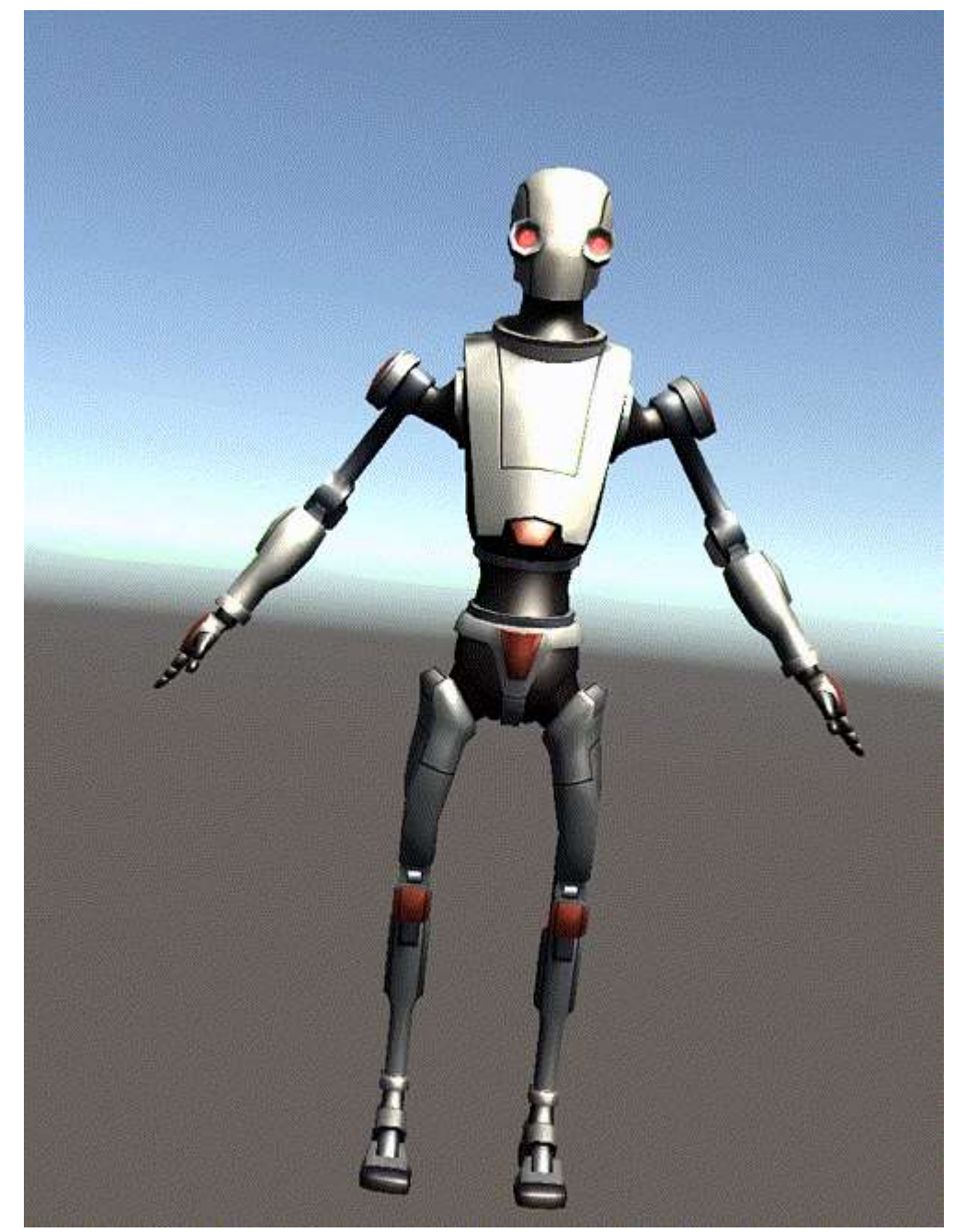
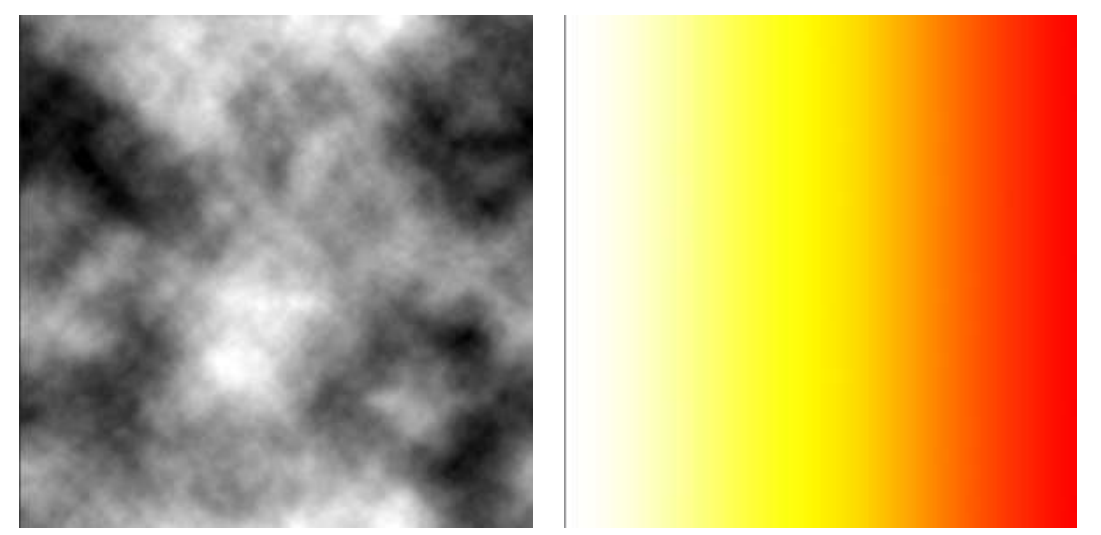
2. 如何处理边缘

首先我们利用smoothstep函数决定边缘颜色，我们利用噪声颜色值 - 消融进度值 得到一个剔除阈值value

然后自定义一个边缘范围值_Range，然后用smoothstep函数来得到一个值

```
fixed4 frag (v2f i) : SV_Target
{
    //噪声图上的颜色是线性随机的
    //我们通过一个消融阈值来控制
    //当颜色值(0~1)的值小于该阈值时，会抛弃对应像素不渲染
    //就会出现局部镂空效果
    fixed3 noiseColor = tex2D(_Noise, i.uv2.xy).rgb;
    clip(noiseColor.r - _Dissolve);
}
```

```
//边缘非常趋近于不渲染的像素点 noiseColor.r - _Dissolve 该值就越小
//该值越小 t越趋近于 1
fixed t = 1 - smoothstep(0.0, _Rang, noiseColor.r - _Dissolve);
//t越趋近于1 那么颜色就趋近于外火焰颜色
fixed3 gradientColor = tex2D(_Gradient, fixed2(t, 0.5)).rgb;
```





唐老狮系列教程-消融效果 基本原理

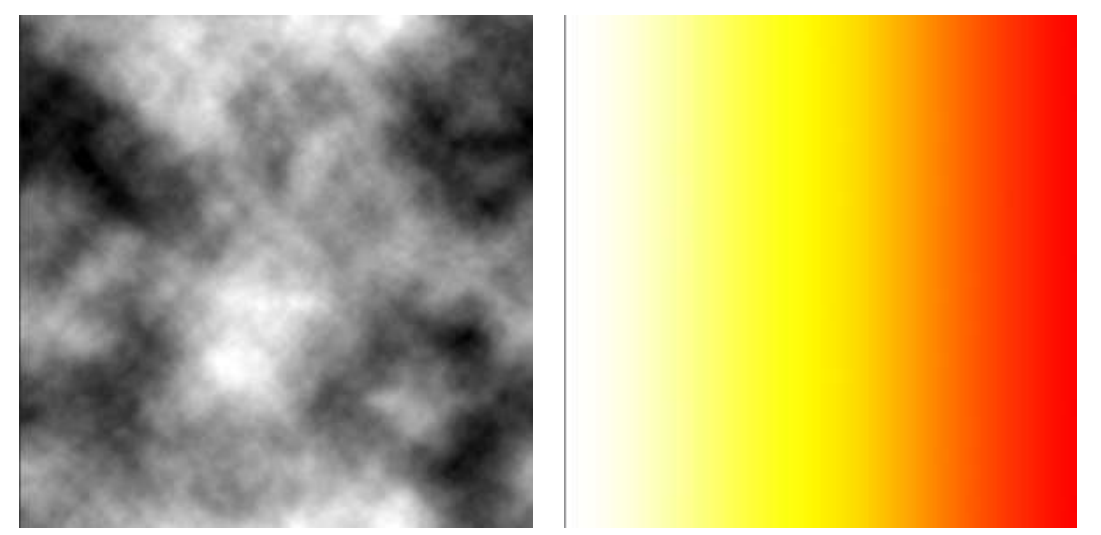
消融效果基本原理

2. 如何处理边缘

接着，我们在原本的颜色和渐变颜色之间进行插值，决定使用哪个颜色

```
//边缘非常趋近于不渲染的像素点 noiseColor.r - _Dissolve 该值就越小  
//该值越小 t越趋近于 1  
fixed t = 1 - smoothstep(0.0, _Rang, noiseColor.r - _Dissolve);  
//t越趋近于1 那么颜色就趋近于外火焰颜色  
fixed3 gradientColor = tex2D(_Gradient, fixed2(t, 0.5)).rgb;
```

```
//越区域边缘的像素点，越应该趋近于使用渐变纹理中的颜色  
//后面加入step(0.00001, _Dissolve)的目的是  
//当_Dissolve等于0时，t是有值的，同样会在两个颜色中插值  
//这样可以保证当_Dissolve为0时，始终不会使用渐变纹理中的颜色  
fixed3 finalColor = lerp(color, gradientColor, t * step(0.00001, _Dissolve));
```





唐老狮系列教程-消融效果 基本原理

消融效果基本原理

2. 如何处理边缘

利用smoothstep结合消融阈值来决定在渐变纹理中采用的渐变颜色

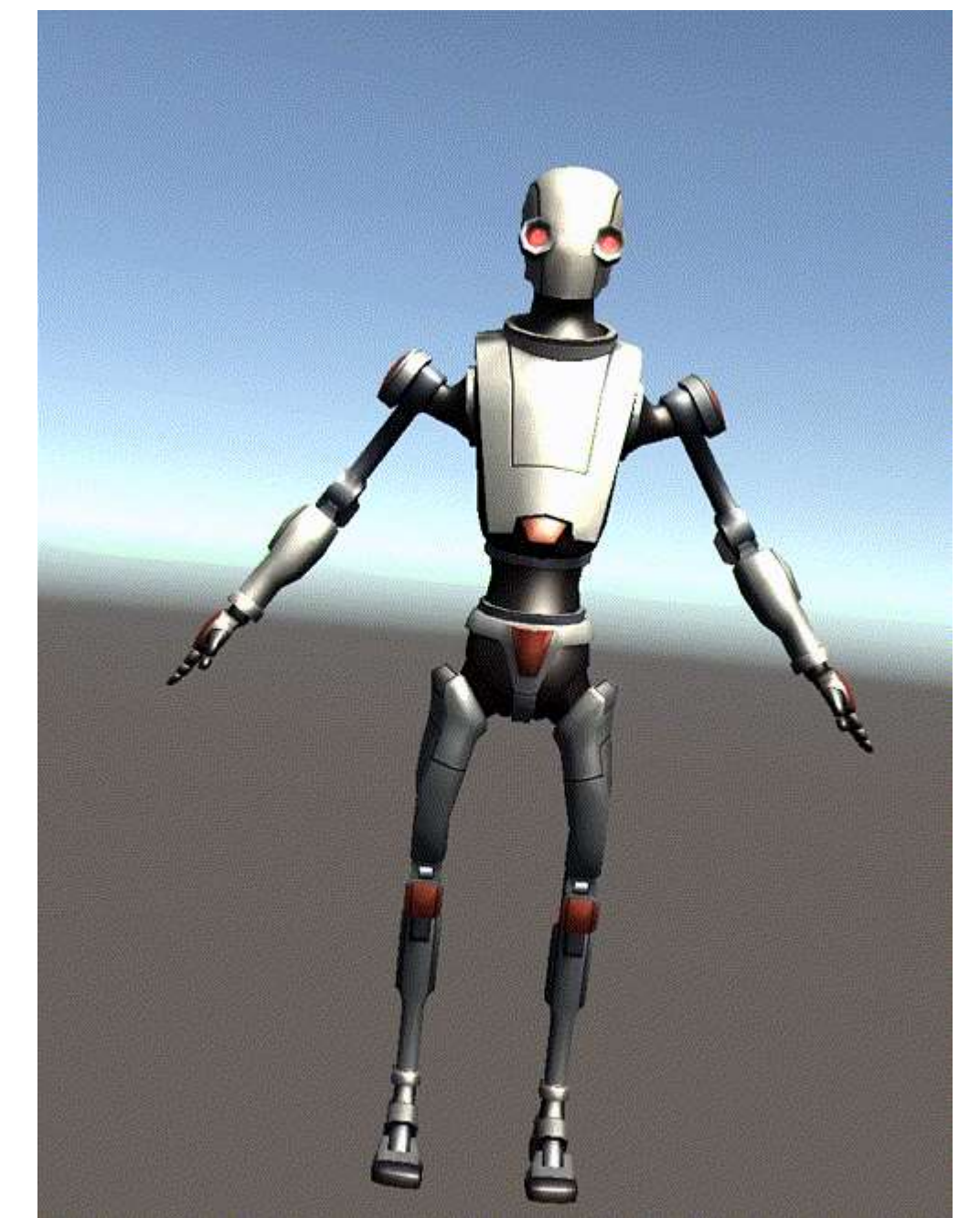
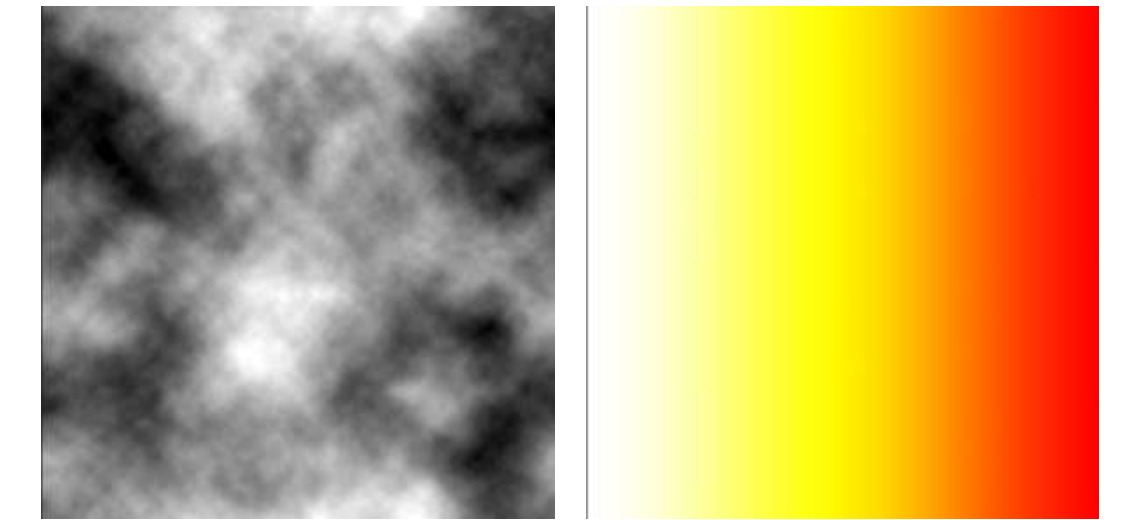
利用lerp来决定在原始颜色和边缘渐变颜色中使用哪个颜色

利用自定义参数来决定边缘范围

从而实现消融边缘渐变色

```
//边缘非常趋近于不渲染的像素点 noiseColor.r - _Dissolve 该值就越小  
//该值越小 t越趋近于 1  
fixed t = 1 - smoothstep(0.0, _Rang, noiseColor.r - _Dissolve);  
//t越趋近于1 那么颜色就趋近于外火焰颜色  
fixed3 gradientColor = tex2D(_Gradient, fixed2(t, 0.5)).rgb;
```

```
//越区域边缘的像素点, 越应该趋近于使用渐变纹理中的颜色  
//后面加入step(0.00001, _Dissolve)的目的是  
//当_Dissolve等于0时, t是有值的, 同样会在两个颜色中插值  
//这样可以保证当_Dissolve为0时, 始终不会使用渐变纹理中的颜色  
fixed3 finalColor = lerp(color, gradientColor, t * step(0.00001, _Dissolve));
```





唐老狮系列教程-消融效果 基本原理

| 总结



唐老狮系列教程-消融效果 基本原理

主要讲解内容

1. 消融效果 是什么

消融效果是利用噪声模拟物体逐渐从屏幕上消失或溶解的过程

2. 消融效果 基本原理

通过对比噪声纹理值与消融进度参数，剔除低于阈值的像素，同时在边缘添加渐变颜色实现动态溶解效果。

关键点：

1. 如何剔除像素

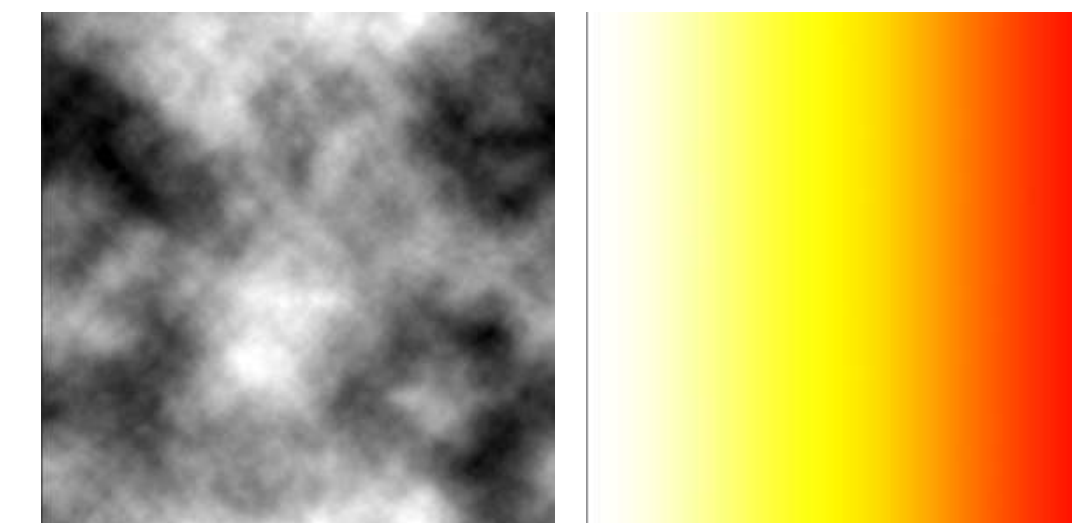
clip(噪声纹理值 - 消融进度参数)

2. 如何处理边缘

利用smoothstep结合消融阈值来决定在渐变纹理中采用的渐变颜色

利用lerp来决定在原始颜色和边缘渐变颜色中使用哪个颜色

利用自定义参数决定边缘





唐老狮系列教程

Thank

谢谢您的聆听

WELCOME
TO THE
UNITY
SPECIALTY COURSE
STUDY

版权所有：唐老狮 tpandme@163.com