



唐老狮系列教程

Bloom效果基本原理



唐老狮系列教程-Bloom效果基本原理

| 主要讲解内容



唐老狮系列教程-Bloom效果基本原理

主要讲解内容

1. Bloom效果是什么
2. Bloom效果的基本原理



唐老狮系列教程-Bloom效果基本原理

| Bloom效果是什么



唐老狮系列教程-Bloom效果基本原理

Bloom效果是什么

Bloom效果（中文也可以叫做高光溢出效果），是一种使**画面中亮度较高的区域产生一种光晕或发光效果**的图像处理技术

Bloom效果的主要目的是模拟现实世界中强光源在相机镜头或人眼中造成的散射和反射现象
使得画面中较亮的区域“扩散”到周围的区域，造成一种朦胧的效果





唐老狮系列教程-Bloom效果基本原理

Bloom效果的基本原理



唐老狮系列教程-Bloom效果基本原理

Bloom效果的基本原理

三步骤概括Unity Shader中实现Bloom效果的基本原理：

- 1.提取：** 提取原图像中的亮度区域存储到一张新纹理中
- 2.模糊：** 将提取出来的纹理进行模糊处理（一般采用高斯模糊）
- 3.合成：** 将模糊处理后的亮度纹理和源纹理进行颜色叠加

关键知识点：

多个Pass的运用，

如何提取，

如何模糊，

如何合成



唐老狮系列教程-Bloom效果基本原理

Bloom效果的基本原理 —— 多个Pass的运用

通过高斯模糊效果的学习，

我们知道可以在屏幕后处理时，单独调用某个Pass对渲染纹理进行处理

只需要利用Unity中的三个函数即可

RenderTarget.GetTemporary(纹理宽, 纹理高, 0)

Graphics.Blit(源纹理, 目标纹理, 材质, passID)

RenderTarget.ReleaseTemporary(渲染纹理对象)



唐老狮系列教程-Bloom效果基本原理

Bloom效果的基本原理 —— 多个Pass的运用

处理高斯模糊效果时，我们使用了2个Pass，分别用来计算水平卷积核竖直卷积
而我们在处理Bloom效果时将使用**4个Pass**，他们分别是：

1. 用于 提取亮度区域 存储到新纹理中的 1 个 Pass
2. 用于 模糊处理提取出来的纹理 的高斯模糊的 2 个 Pass（可以复用高斯模糊Shader中写好的Pass）
3. 用于 与原图像进行合成 的 1 个 Pass

通过使用这4个Pass，我们就能够完成

提取、模糊、合成 三个步骤



唐老狮系列教程-Bloom效果基本原理

Bloom效果的基本原理 —— 如何提取

我们需要在Shader中声明一个**亮度阈值变量**，亮度低于该值的区域不会被提取

主要用于 **“提取”** Pass的片元着色器函数当中，

用当前像素的灰度**值** $L = 0.2125 * R + 0.7154 * G + 0.0721 * B$ 与 **亮度阈值变量** 进行计算

```
fixed luminance(fixed4 color) {  
    return 0.2125 * color.r + 0.7154 * color.g + 0.0721 * color.b;  
}  
  
fixed4 fragExtractBright(v2f i) : SV_Target {  
    fixed4 c = tex2D(_MainTex, i.uv);  
    fixed val = clamp(luminance(c) - _LuminanceThreshold, 0.0, 1.0);  
  
    return c * val;  
}
```




唐老狮系列教程-Bloom效果基本原理

Bloom效果的基本原理 —— 如何提取

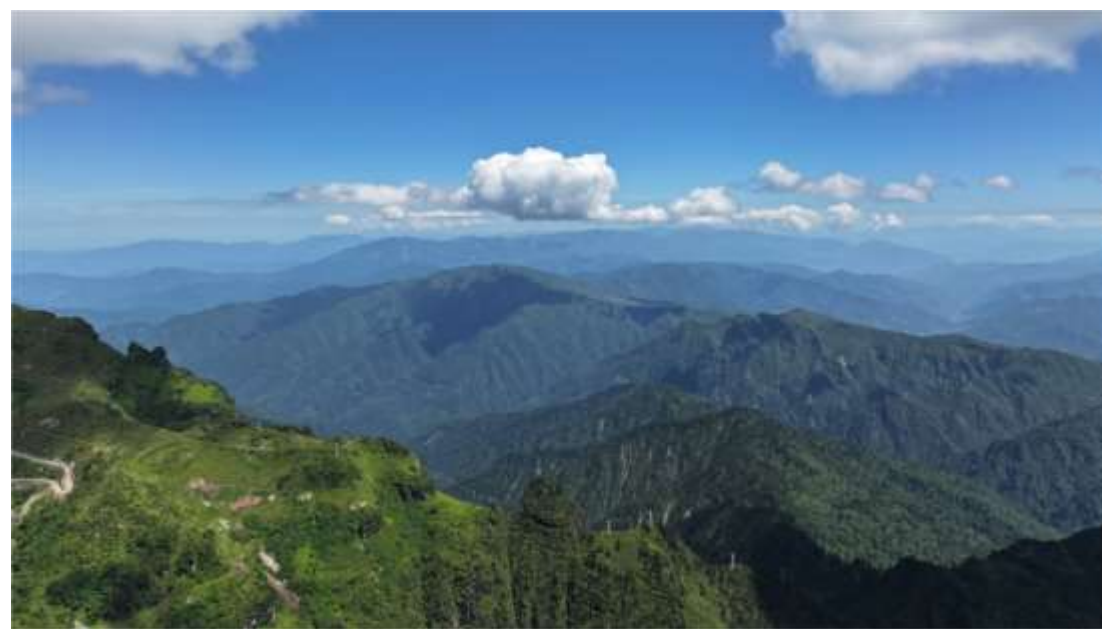
```
fixed luminance(fixed4 color) {  
    return 0.2125 * color.r + 0.7154 * color.g + 0.0721 * color.b;  
}  
  
fixed4 fragExtractBright(v2f i) : SV_Target {  
    fixed4 c = tex2D(_MainTex, i.uv);  
    fixed val = clamp(luminance(c) - _LuminanceThreshold, 0.0, 1.0);  
  
    return c * val;  
}
```

灰度值 - 亮度阈值变量：是为了仅保留超过阈值的部分，可以提取出图像中亮度较高的地方。

Clamp函数：如果结果小于0则为0，大于1则为1。得到的val表示像素的亮度贡献

颜色*亮度贡献：基于亮度阈值调节颜色的亮度，若val为0则为黑色，越接近1越接近原始颜色

这样做的目的是保留高亮区域的颜色信息，同时衰减低亮区域的颜色。



原图



阈值为0



阈值为0.5



唐老狮系列教程-Bloom效果基本原理

Bloom效果的基本原理 —— 如何提取

如果 **“提取”** Pass是Shader中的第一个Pass，那么我们完全可以利用
RenderTexture.GetTemporary 和 Graphics.Blit 函数将源纹理提取亮度信息后存储到缓存区中

```
RenderTexture buffer0 = RenderTexture.GetTemporary(rtW, rtH, 0);  
buffer0.filterMode = FilterMode.Bilinear;  
  
Graphics.Blit(src, buffer0, material, 0);
```

有了提取出来的纹理信息，我们就可以进行第二步 **“模糊”**



唐老狮系列教程-Bloom效果基本原理

Bloom效果的基本原理 —— 如何模糊

由于我们目前已经在高斯模糊的课程中写好了两个用于进行水平和竖直卷积运算的Pass

因此我们可以使用以前学习SubShader语句时学习的**UsePass**指令

只需要在之前的高斯模糊Shader中为两个Pass取名，然后在**Bloom的Shader中复用即可**

```
#region 知识点二 Pass的名字
//主要作用:
//我们对Pass命名的主要目的
//可以利用UsePass命令在其他Shader当中复用该Pass的代码,
//只需要在其他Shader当中使用
//UsePass "Shader路径/Pass名"
//注意:
//Unity内部会把Pass名称转换为大写字母
//因此在使用UsePass命令时必须使用大写形式的名字

//Pass{
//  Name MyPass
//}

//在其他Shader中复用该Pass代码时
//UsePass "TeachShader/Lesson4/MYPASS"
#endregion
```

复用了两个Pass后，我们只需要在Bloom效果对应的C#代码中对 **“提取”** 出来的纹理
进行高斯模糊Pass的调用即可（逻辑和高斯模糊一致）



唐老狮系列教程-Bloom效果基本原理

Bloom效果的基本原理 —— 如何模糊

接着，我们需要在Bloom效果的Shader中声明一个**纹理属性 _Bloom**

用于存储模糊处理完毕后的纹理

在C#代码中完成高斯模糊处理后，只需要将缓存区内容，写入材质球中的纹理属性即可

```
material.SetTexture("_Bloom", buffer0);
```

完成这一步后，在Shader中就能够得到**模糊处理后的亮度纹理信息**了

我们便可以进行第三步 **“合成”**



唐老狮系列教程-Bloom效果基本原理

Bloom效果的基本原理 —— 如何合成

在“**合成**”的Pass 中

我们只需要用

主纹理_MainTex（其中使用的纹理是屏幕原图像）和

纹理属性_Bloom（其中使用的纹理是模糊处理后的亮度纹理信息）

进行颜色叠加即可，我们对两张纹理进行采样，将获取到的**颜色信息**进行**加法**运算

因为颜色相加带来的效果就是增加亮度，使得原本高亮的部分变得更加显眼

从而达到Bloom效果（高光溢出效果）



唐老狮系列教程-Bloom效果基本原理

| 总结



唐老狮系列教程-Bloom效果基本原理

总结

1. Bloom效果是什么

高光溢出效果

使画面中亮度较高的区域产生一种光晕或发光效果的图像处理技术

2. Bloom效果的基本原理

基本原理就是利用4个Pass进行3个处理步骤

提取(1个Pass)：提取原图像中的亮度区域存储到一张新纹理中

模糊(2个Pass)：将提取出来的纹理进行模糊处理（一般采用高斯模糊）

合成(1个Pass)：将模糊处理后的亮度纹理和源纹理进行颜色叠加



唐老狮系列教程

Thank

谢谢您的聆听