PowerPBS 参数手册

使用drp流程时,如果场景烘焙,主光源获取需要LightingProcess.cs配合.

Urp可以正常获取主光源.

目录

[PowerPBS 参数手册 1](#_Toc85119593)

[获取地址 2](#_Toc85119594)

[Settings 3](#_Toc85119595)

[DepthMode 3](#_Toc85119596)

[Shadow 3](#_Toc85119597)

[PbrMode 4](#_Toc85119598)

[Main 5](#_Toc85119599)

[MainProp 5](#_Toc85119600)

[PBRMask 5](#_Toc85119601)

[CustomSpecular 6](#_Toc85119602)

[ClearCoat 6](#_Toc85119603)

[Vertex 6](#_Toc85119604)

[Aniso 6](#_Toc85119605)

[预积分sss 8](#_Toc85119606)

[ssss 8](#_Toc85119607)

[Alpha 8](#_Toc85119608)

[AlphaTest 9](#_Toc85119609)

[AlphaPreMultiMode 9](#_Toc85119610)

[FresnelAlpha 9](#_Toc85119611)

[PresetBlendMode 9](#_Toc85119612)

[Detail Maps 10](#_Toc85119613)

[功能: 10](#_Toc85119614)

[detail的渲染顺序 10](#_Toc85119615)

[detail图集 10](#_Toc85119616)

[属性: 11](#_Toc85119617)

[Parallax\_SSS 11](#_Toc85119618)

[视差 11](#_Toc85119619)

[fastSSS 11](#_Toc85119620)

[Env(Lights) 12](#_Toc85119621)

[逐材质光照 12](#_Toc85119622)

[逐材质ibl 12](#_Toc85119623)

[Urp Additional Lights 12](#_Toc85119624)

[Specular options 13](#_Toc85119625)

[GI 13](#_Toc85119626)

[SH get lightDir 13](#_Toc85119627)

[Emission 13](#_Toc85119628)

[Alpha Blend Mode 13](#_Toc85119629)

[blend常用的设置: 14](#_Toc85119630)

[Strand 14](#_Toc85119631)

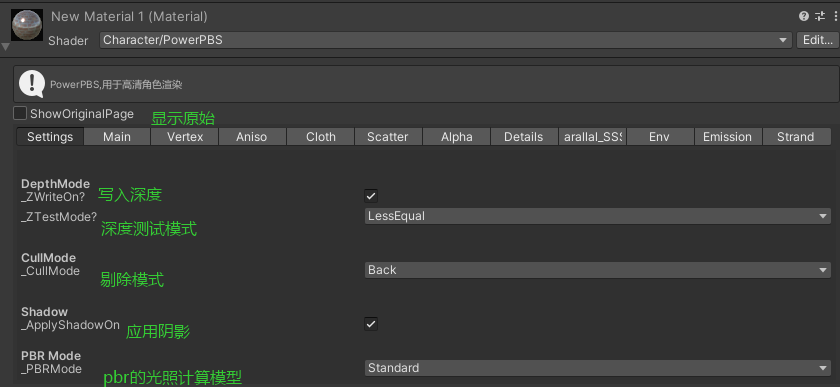
[RenderQueue 15](#_Toc85119632)

获取地址

中台svn : <http://svn.ifunplus.cn/svn/common/PowerPBS>

Github: <https://github.com/redcool/PowerPBS.git>

# Settings



DepthMode

**\_ZWriteOn**

是否写入深度缓冲,用于深度剔除

\_ZtestMode

确定深度剔除的方式

**\_**CullMode

正反面剔除模式

off : 正反面都渲染

front : 渲染背面

back : 渲染正面

Shadow

是否接收阴影

PbrMode

光照模型:



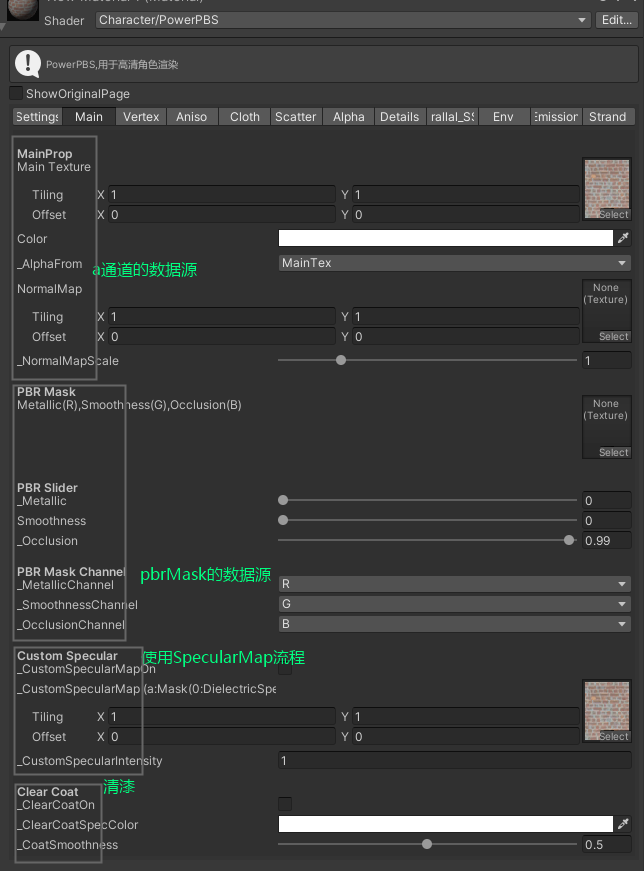
Standard : 通用的pbr渲染.公式为minimalist cooktorrance,大部分情况使用.

Aniso : 各向异性高光,公式为 ggx aniso,

Cloth : 通用的布料渲染,公式为AshihkminV+CharlieD

StrandSpec : 发丝各向异性高光.

****Main****



MainProp

MainTex : albedo(反照图,少一些暗部信息)

AlphaFrom : 选择alpha数据的来源,mainTex.a或pbrMask.a

PBRMask

Metallic(R),Smoothness(G),Occlusion(B) : (R通道:metallic,G:smoothness,B: occlusion)

最终的metallic = \_Metallic \* pbrMask.R.

**Metallic对diffColor,specColor的影响为:**

diffColor = lerp(0.96,0,metallic) \* albedo

specColor = lerp(0.04,albedo,metallic)

最终的Smoothness = \_Smoothness \* pbrMask.G.

决定高光的宽窄

最终Occlusion = pbrColor \* lerp(1,pbrMask.B,\_Occulsion).

决定物体表面的暗部.

Pbr渲染物体表面的暗部信息放到pbrMask.b通道.

CustomSpecular

指定specMap来使用specular流程.

Metallic流程高光色在lerp(0.04,albedo,metallic)之间

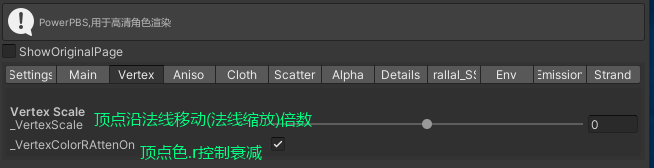
Specular流程,高光色来自specularMap

ClearCoat

清漆效果.

原理是,二重的pbr计算,coat pbr只有局部光照计算,无gi计算.

Vertex



让顶点沿着法线方向移动.

Aniso



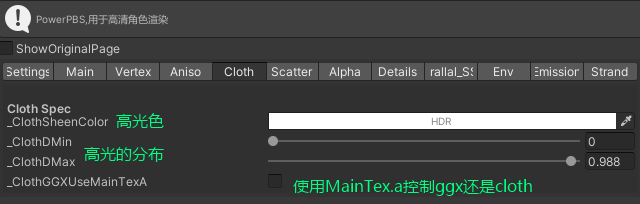
需要将Settings里pbrMode设置为 Aniso.

\_AnisoColor : 各异高光色

\_AnisoIntensity : 各异高光强度

\_AnisoRough : 各异高光分布(<0竖直分布,>0水平分布)

Cloth



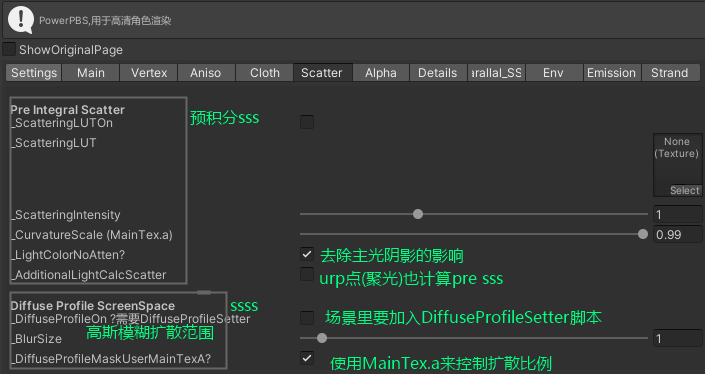
PbrMode需要选 Cloth

clothMin,ClothMax : 组合可以控制布料高光的分布的区域和扩散范围

ClothGGXUseMainTexA : 使用MainTex.a来控制像素使用pbr高光还是cloth高光.

可以用Metallic,smoothness来控制高光色,高光宽窄等

Scatter



Scatter可用于散射计算

预积分sss

原理: 使用(wnl,curvaScale)来采样ScatterLUT,再加到 pbr颜色上.

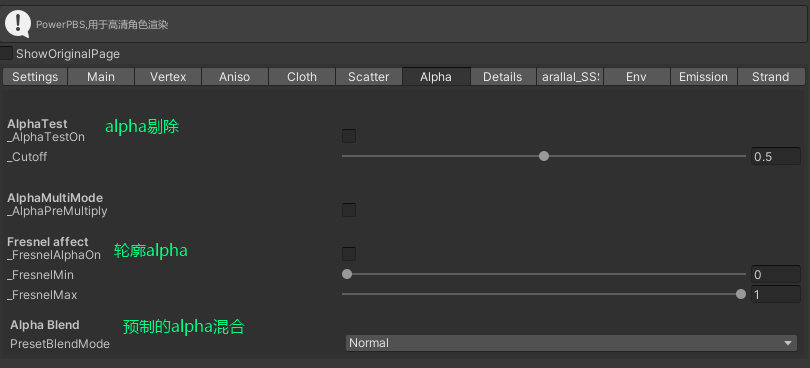
CurvatureScale : 控制v方向采样.

ssss

原理:使用高斯模糊来模拟光在物体内部的散射.

DiffuseProfileSetter计算高斯系数的颜色值与比例.

Alpha



AlphaTest

使用alpha控制像素剔除(会破坏ealyZ)

AlphaPreMultiMode

Alpha 预乘模式,可以渲染半透明物体(角膜,玻璃等)

会影响diffColor,alpha,最终的alpha受metallic影响.

公式:

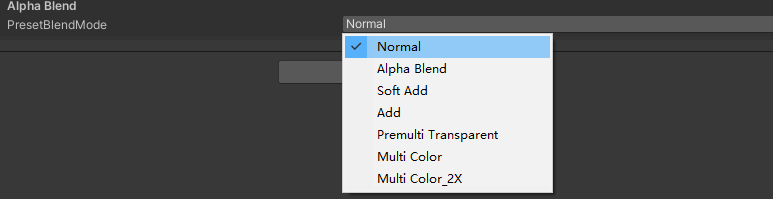
最终的diffColor = diffColor \* alpha

最终的alpha ~= lerp(alpha,1,metallic)

FresnelAlpha

最终的alpha受物体轮廓影响

PresetBlendMode



Alpha混合控制计算完的颜色值如何与帧缓冲已有的数值进行混合.

PresetBlendMode会同时控制材质的srcMode,dstMode

主要有:

Normal: 不混合,直接写入帧缓冲(blend one zero)

Alpha blend, 正常alpha 混合( Blend SrcAlpha OneMinusSrcAlpha)

SoftAdd ( Blend SrcAlpha One)

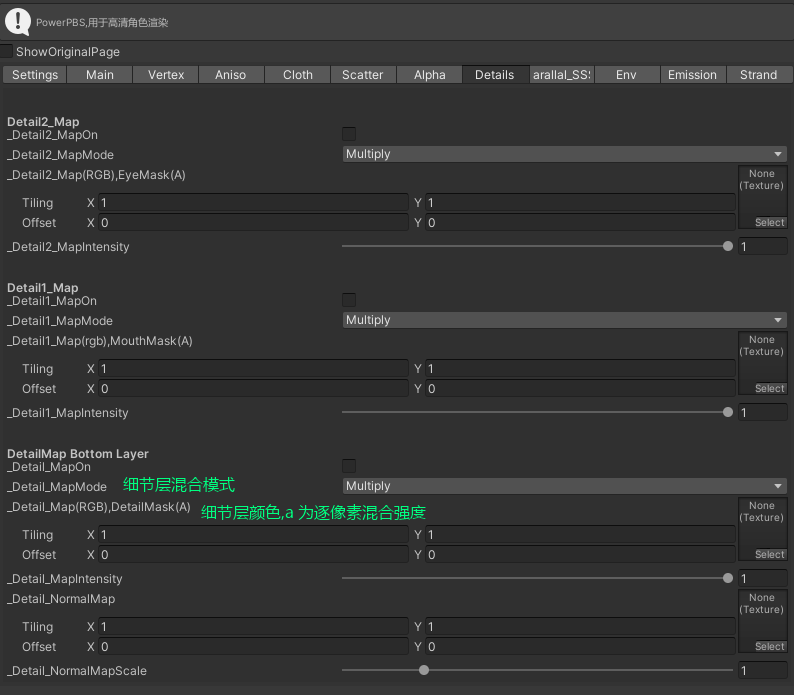
Add 叠加,(Blend One One)

PremultiTranparent, 屏幕色预乘(Blend One OneMinusSrcAlpha)

MultiColor (Blend DstColor Zero)

MultiColor 2x(Blend DstColor SrcColor)

****Detail Maps****



**功能:**

结合tiling,ofset可以大幅提高物体表面的细节丰富度.

主detail,用于通用的detail,如皮肤的细节,

detail1 ,可用于皮肤的 划痕, 疤痕, 斑点.

detail2,可用于嘴部细节,如唇彩,

**detail的渲染顺序**

高层的detail可以覆盖底层的detail.顺序从小到大.

**detail图集**

detail支持图集,设置tiling与offset,可以使用图集的不同区域.

可以将工程的所有细节部位整理到(1-5)个图集(2k), 每次的细节只使用图集的一部分,从而降低空间占用.

**属性:**

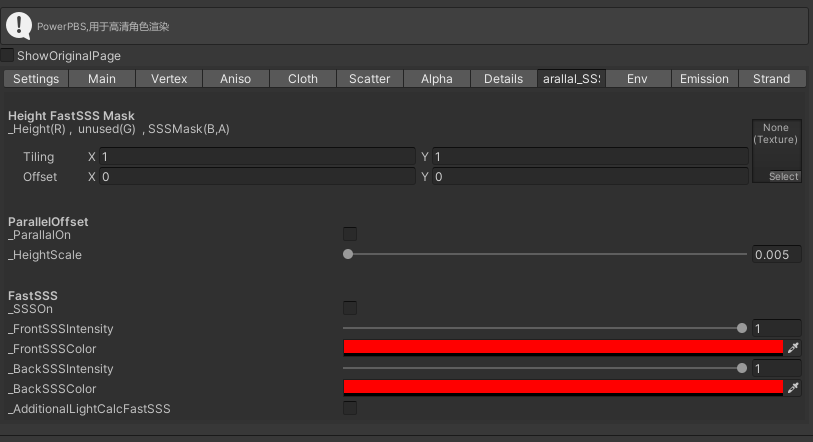
**\_**DetailMap(rgb提供颜色,a作为该detail的mask,控制detail的强度)

\_DetailMapIntensity : 细节纹理的强度

\_DetailNormalMap : 细节处的法线

\_DetailNormalMapScale : 细节法线倍数

Parallax\_SSS



视差

增强表面的深度感

原理,用切线空间的视线向量控制主纹理uv偏移.

\_Height(R),r通道用于确定像素的深度值

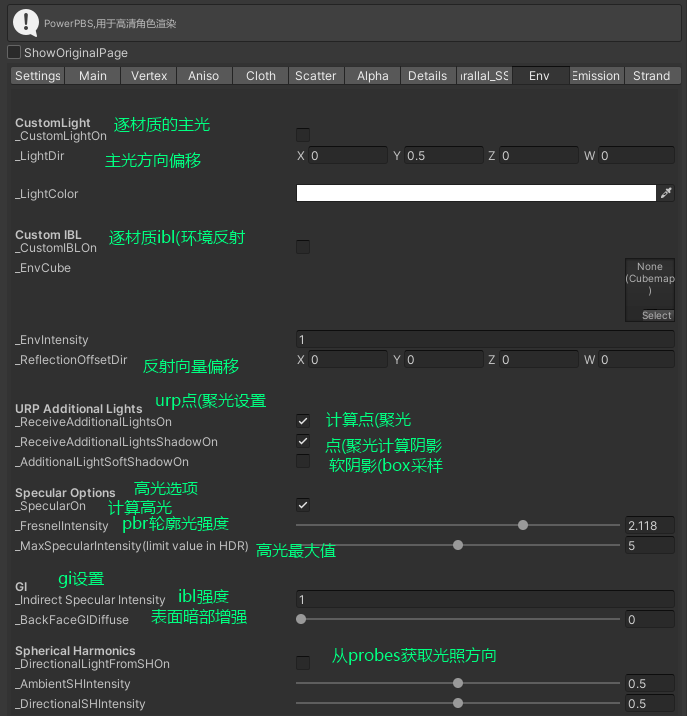
fastSSS

模拟光透.

原理: pbrColor叠加 dot(l,v)+dot(-l,v).

sssMask.ba,用于控制frontSSS,backSSS

Env(Lights)



逐材质光照

对mainLight进行修改

\_CustomLightOn : 开关

\_LightDir : 改变主光源的方向.

\_LightColor : 改变主光源的颜色.

逐材质ibl

替换ibl

默认情况使用Unity’s Lighting面板里的skybox(custom skybox)的hdr skybox.

Urp Additional Lights

控制如何计算点(聚光)灯.

Specular options

高光相关的控制开关

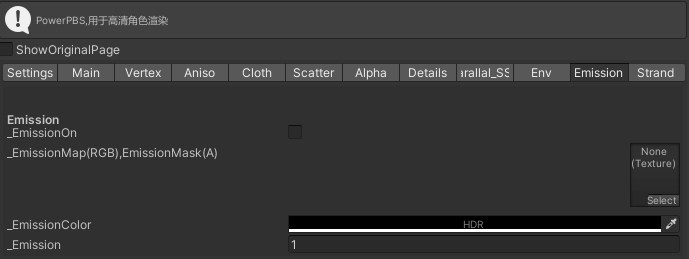
GI

对gi(diffuse,specular)进行控制

SH get lightDir

从probes中获取光照方向,再执行pbr光照计算.

****Emission****



用于自发光的控制.

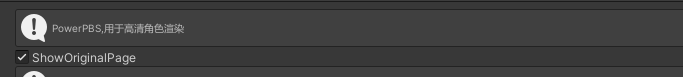
\_EmissionMap(RGB:自发光基础图,A: 自发光mask)

\_EmissionColor  : 自发光调色

\_Emission : 自发光强度

Alpha Blend Mode



勾上时可以找到 \_SrcMode,\_DstMode

**\_AlphaTestOn**: 开启 alphaTest,用于镂空

**AlphaBlendMode**: 控制新计算的像素颜色与已有的像素颜色的混合.

\_SrcMode : 新计算的像素颜色的模式

\_DstMode : 已有屏幕像素颜色的模式

blend常用的设置:

Blend one zero // 不混合,用于正常渲染

Blend SrcAlpha OneMinusSrcAlpha // 标准透明(会全透

Blend One OneMinusSrcAlpha // alpha预乘(玻璃效果,不会全透

Blend One One // 叠加,更亮

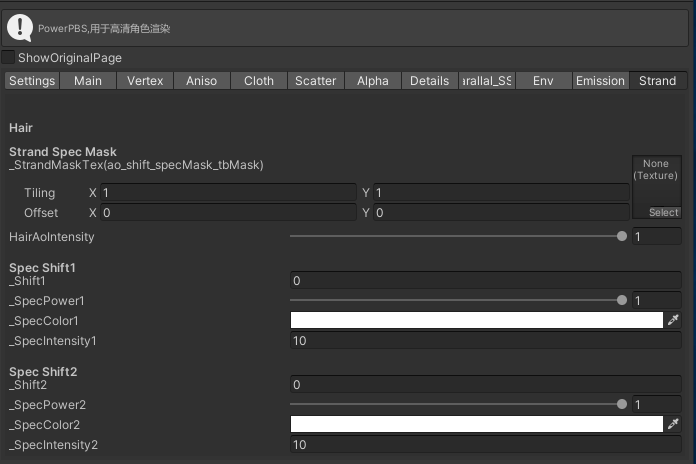
Blend OneMinusDstColor One // 柔和叠加(稍微加亮

DstColor Zero // 新旧颜色相乘

BlendDstColor SrcColor // 2倍相乘

详细文档: <https://docs.unity3d.com/Manual/SL-Blend.html>

****Strand****



这部分实现了strand specular(双重各向异性高光).

\_HairOn : 开启 strandSpecular,不用使用pbs高光(vd)

\_TBMaskMap (R):  用mask控制高光的走向,

mask里白色,高光横向分布(使用binormal数据),

mask里黑色,高光纵向分布(使用tangent数据)

\_ShiftTex(g:shift,b:mask) : 控制高光的基础偏移

\_Shift : 高光的偏移量

\_SpecPower : 高光的宽窄

\_SpecColor : 高光颜色

\_SpecIntensity : 高光的强度

**RenderQueue**

d72712804eb0491eee626abeb149e4bb

控制渲染此物体的顺序: 值越小越先渲染.会影响之后的混合

常见的数值如下:

Background : 1000,

Geometry: 2000

AlphaTest : 2450

Skybox : 2500

AlphaBlend : 3000