Unity-Chan Toon Shader 2 简介

Unity-Chan Toon Shader (UTS) 是一种用于图像和视频的卡通着色器，旨在满足从事 cel-shaded 3DCG 动画的创作者的需求。

此 Toon 着色器配置为轻松生成角色设计所必需的所有阴影，例如突出角色模型各个部分形式的阴影，而无需担心光源的位置和强度。

该着色器特别强大的功能之一是能够使用着色器本身内的滑块调整阴影，从而无需多个光源。

除了Base Color、1st Shade Color和2nd Shade Color的 3 个基本层之外，颜色和纹理还可以接受多种自定义选项，例如High Color、Rim Light、MatCap（球体贴图）和Emissive（光发射）

该着色器还有 2 个选项用于创建设计所需的固定阴影：位置贴图，为每个阴影分配一个设置的投射点，以及着色等级贴图，它可以根据照明调整阴影强度。

还有其他一些方便的工具，例如通过刘海可以看到眼睛和眉毛的程度，可以进一步强调动漫风格的外观。

添加后期效果允许 UTS2 使用非真实感渲染 (NPR) 来创建可以使用支持基于物理渲染 (PBR) 的标准着色器制作的任何类似图片的元素

UTS2 着色器分为两大类。

\* DoubleShadeWithFeather：UTS2 的标准着色器。允许 2 种阴影颜色（双阴影颜色）和颜色之间的渐变（羽化）。

\* ShadingGradeMap：更高级的 UTS2 着色器。除了 DoubleShadeWithFeather 功能之外，此着色器还可以保存称为 ShadingGradeMap 的特殊贴图。

选择使用哪种着色器取决于个人喜好，

但通常DoubleShadeWithFeather更适合需要清晰、定义明确的颜色的cel样式，

而ShadingGradeMap可能更适合颜色更模糊的插图样式。

此外，将Toon放在着色器名称的开头意味着着色器可以使用对象反转公式创建轮廓。

UTS2中的轮廓有很多自定义选项：使用专用纹理生成的轮廓的强度、与基础颜色的混合级别、相机基础偏移等。

着色器名称在末尾也有名称块，如剪裁。这些表示以下几种功能。

   Clipping：着色器包含一个裁剪蒙版，这是一种能够裁剪、溶解等功能的“纹理删除器”。

   TransClipping：也表示剪切蒙版，但在省略纹理时会考虑蒙版的α透明度。这样可以更好地省略，但会产生比剪裁更大的负载。

   StencilMask：使用模具缓冲区指定某些部分通过其他部分的可见度。例如，对于动漫风格的角色，可能需要始终保持眉毛可见，而不是被角色的刘海遮住。该着色器必须始终与StencilOut类型着色器一起使用。

   StencilOut：与StencilMask类型着色器一起使用。在上面的示例中，此着色器将设置在“刘海”部分，以使其透明，以便“眉毛”部分可见。

The Shaders in the UnityChanToonShader/NoOutline Folder

NoOutline文件夹中的着色器名称前面有ToonColor，但此处表示着色器没有轮廓功能。

没有轮廓功能的零件将少一个绘制过程，因此这些着色器非常适合不需要轮廓或使用高精度卡通线着色器(如PSOFT铅笔4 Line for Unity)的设计。

某些NoOutline着色器的名称末尾有 Transparent名称块。

这些是特殊的透明度着色器。它们适用于具有“脸红”外观的部件，或者用于玻璃和类似玻璃的物体。

The Shaders in the UnityChanToonShader/AngelRing Folder

AngelRing文件夹包含具有“天使环”功能的着色器。

“天使戒指”是一种高光效果，如下图所示。从相机的角度看，它们有一个固定的位置。

只有UTS2的高规格ShadingGradeMap着色器及其变体ShadingGradeMap\_TransClipping着色器才具有此“天使环”功能。

这些着色器主要用于“头发”部分，这使它们属于与模具省略的StencilOut着色器相似的类别。

The Shaders in the UnityChanToonShader/Mobile Folder

轻量级版本的着色器适用于Mobile和VR内容，通常不会改变对象的外观，可以在Mobile文件夹中找到。

为了使这些着色器在移动平台上更亮，以下功能受到限制。

只能使用一个实时平行光(也不支持多个光源和实时点光源)。

通过使用烘焙的点光源和光探头来支持点光源。这可能需要对Gi\_Intenity进行某些调整。

移动着色器特性与普通的Toon\_DoubleShadeWithFeather着色器和Toon\_ShadingGradeMap着色器兼容，因此如果上述功能可接受，则可以用移动着色器替换其标准版本，以提高渲染性能。

Mobile/AngelRing文件夹包含支持“Angel Ring”功能的移动着色器。

每个着色器的基本功能与其标准版本的对应功能相同。

The Shaders in the UnityChanToonShader/Tessellation Folder

支持DirectX 11 Phong细分的UTS2着色器可以在细分文件夹中找到。

Phong细分会更正细分曲面的位置，以便它们与网格的法线向量更紧密地对齐。这是一种对低多边形网格进行平滑的有效方法。

在Windows环境中，UTS2只能与DirectX 11一起使用Phong细分。

细分/灯光文件夹包含轻量级版本的细分着色器，具有与移动着色器相同的限制。

其余文件夹还包含前面描述的着色器版本，并添加了对Phong细分的支持。

添加Phong细分可显著提高轮廓和其他小细节(如嘴唇)的质量。

这些着色器主要用于预渲染内容，但是，它们也适用于高端VR角色内容，要求角色离用户非常近。

The shaders in the UnityChanToonShader/Helper Folder

Helper文件夹包含仅用于显示轮廓对象的着色器。

轮廓对象可以作为多材质应用于角色。

通过转到网格的蒙皮网格渲染器&gt;材质并将大小增加1来添加轮廓，然后注册所需的轮廓材质。

警告：大纲将被添加到其他材料之上，这将增加PC的负担。使用这些着色器时请记住这一点。

Sample Scenes

通过打开项目并打开\Assets\Sample Scenes文件夹，可以找到以下示例场景。



Box Projection：使用长方体投影为黑暗的房间照明。





ToonShader：插图样式着色器的设置。





ToonShader\_CelLook：大提琴样式着色器的设置。





ToonShader\_Emitve：具有发射器的着色器的设置。





ToonShader\_Firefy：多个实时点光源。





Baked Normal\Cube\_HardEdge：烘焙法线引用。





Sample\Sample：基本UTS2着色器简介。





ShaderBall\ShaderBall：示例着色球上的UTS2设置。





PointLightTest\PointLightTest：带点光源的大提琴样式内容示例。





SSAO Test\SSAO中的SSAO测试。





NormalMap\NormalMap：将法线贴图与UTS2一起使用的技巧。





LightAndShadows\LightAndShadows：标准着色器和UTS2之间的比较。





AngelRing\AngelRing：天使环和ShadingGradeMap示例。





MatCapMask\MatCapMask：MatcapMask示例。





EmissiveAnimation\EmisssiveAnimation：EmissiveAnimation示例。





Mirror\MirrorTest.Unity：镜像对象的示例场景检查



每个场景都用作相关着色器和照明设置的参考。

在创建您自己的场景时，它们应该会派上用场。

https://developer.unity.cn/projects/04-cong-ling-kai-shi-de-qia-tong-xuan-ran-pbrpian

二分色卡通渲染的局限性，以及将卡通材质和PBR材质进行结合的一些方法

基于二分色的卡通渲染都可以表达的材质



布料和皮肤材质





金属材质





清漆材质





表面有半透明层的多层材质。比如瓷器表面的半透明釉质，漆面上的半透明涂层。





光滑的皮革



表达场景吃力

场景所包含的材质非常丰富，只靠二分色难以表现足够多的材质。 卡通渲染一方面减少了色阶，营造出卡通的风格。但另一方面也因为色阶的减少，难以表达多样的材质

二分人物和PBR场景结合

目前出现了一些将卡通渲染的人物和PBR材质场景进行结合的游戏，如《蓝色协议》、《二之国》等

PBR（Physically based rendering），采用了物理真实的光照模型。和卡通渲染所属的NPR领域看起来是2个路线。不过两者也是进行结合的。

两种风格融合方法



柔化明暗边界



但是可以通过柔化明暗交界，让风格往厚涂的方向过渡，这会让人物和场景看起来更统一

《原神》中的角色对明暗边界进行了一定的柔化，使得角色和场景的融合感更强



人物和场景受相同光源的影响



为角色添加多光源的支持。在场景中添加多个光源，同时作用于场景和角色。如果场景和角色有同样的受光，就能提高两者的融合感



风格化PBR渲染



主机游戏 《重力眩晕》。它的渲染可以说是风格化PBR的一个实现。使用PBR的光照计算，再通过减少色阶来增加卡通感，做出了很有自己风格的画面。不过因为减少了色阶，导致材质的表达上有些不足。如衣服上的金属材质，并不太有金属的感觉。这也是通过减少色阶来实现风格化PBR的一个问题。如果既要体现出卡通感，又要保留PBR丰富的材质表达力，做到兼得鱼和熊掌，还需要想其他的方法。

例子

相对近期的游戏《嗜血代码》

人物的皮肤是偏向二分色的卡通材质，服装方面则采用了PBR材质。这也是卡通渲染和PBR结合的一个方法，用卡通渲染来表现它比较擅长的皮肤、布料等材质。然后使用PBR材质来表现其他对材质感要求高的部分

在移动平台上，我看到《战双帕弥什》也采用了类似的方案。角色的皮肤、布料等材质使用了二分色的光照计算，而金属和皮革的材质则使用了PBR的光照计算，通过蒙版对两种光照计算进行区分。即保持了人物的卡通感，又通过PBR材质表现了卡通材质难以表现的材质感，达到了不错的材质效果。

总结

在本节中，介绍了二分色的卡通渲染的局限性，使卡通材质和PBR材质能更好的融合在画面中的一些方法，以及市面上的一些游戏是如何通过引入PBR材质来弥补卡通渲染在材质表达方面的不足的。

RimLight Settings 边缘光设置

LightDirection Mask 灯光方向遮罩（灯光反方向 无边缘光）

Antipodean(Ap)\_RimLight 反方向的等用其他颜色衬托

1. 采样边缘光遮罩 o1
2. 计算边缘光材质混合光照 a1
3. 计算遮蔽强度（控制a1结果的范围）b1
4. a1与b1计算混合后得到结果 c1
5. 算上反向边缘光 d1
6. 用d1与c1混合后，再混合o1，得到结果e1
7. 最后将e1和高光混合

边缘光原理：



RimLight边缘发光，对应当前视角方向，物体上位于边缘的地方额外加一个光的效果。

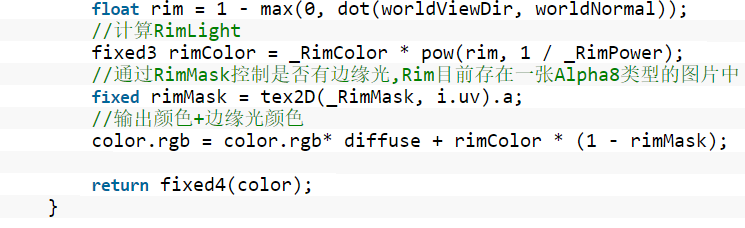
那么，怎么判断一个点是否在物体的边缘呢？

通过法线方向和视线方向的夹角来判断。当视线方向V与法线方向N垂直时，这个法线对应的面就与视线方向平行，说明当前这个点对于当前视角来说，就处在边缘；而视线方向与法线方向一致时，这个法线对应的面就垂直于视线方向，说明当前是直视这个面。所以，我们就可以根据dot（N,V）来获得视线方向与法线方向的余弦值，通过这个值来区分该像素是否处在边缘，进而判断是否需要增加以及增加边缘光的强弱。



通过Mask图控制特定部位有边缘光

解决全身边缘光问题，只希望盔甲部分有光效果，而其他部分没有自发光，我们就需要用Mask图来控制，用一张Alpha8的灰度图来控制是否开启边缘光效果，将上面的shader简单修改一下，增加Mask图的通道：

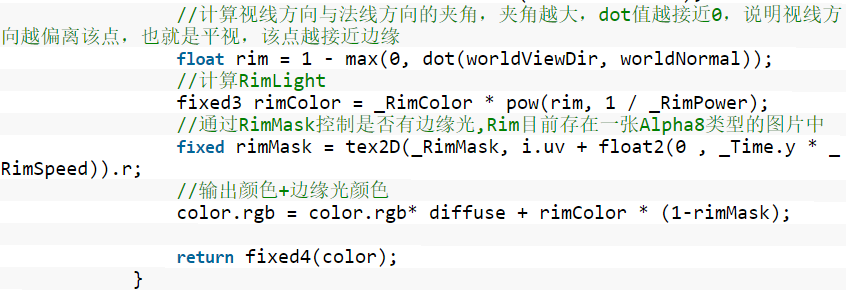


动态RimLight效果

边缘光使用一张动态图，达到流动效果。思路跟Mask图一样，只不过这次用一张扫面线类型的纹理，然后通过这张纹理的采样滚动，达到动态效果。用到\_Time向量，这个向量中几个值都是跟时间相关的，这里用\_Time.y就可以获得时间了。

用一张Mask纹理，白色代表有边缘光，黑色无边缘光，通过采样这张Mask纹理，控制模型上显示边缘光的部分：







博文：<https://blog.csdn.net/linuxheik/article/details/78912050>

内置变量

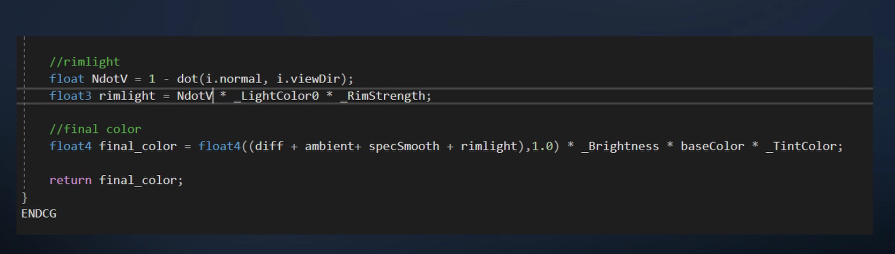


Unity Shader-后处理：屏幕水波效果

<https://blog.csdn.net/puppet_master/article/details/52975666>

Unity Shader-后处理：简单的颜色调整（亮度，饱和度，对比度）

<https://blog.csdn.net/puppet_master/article/details/52423905>



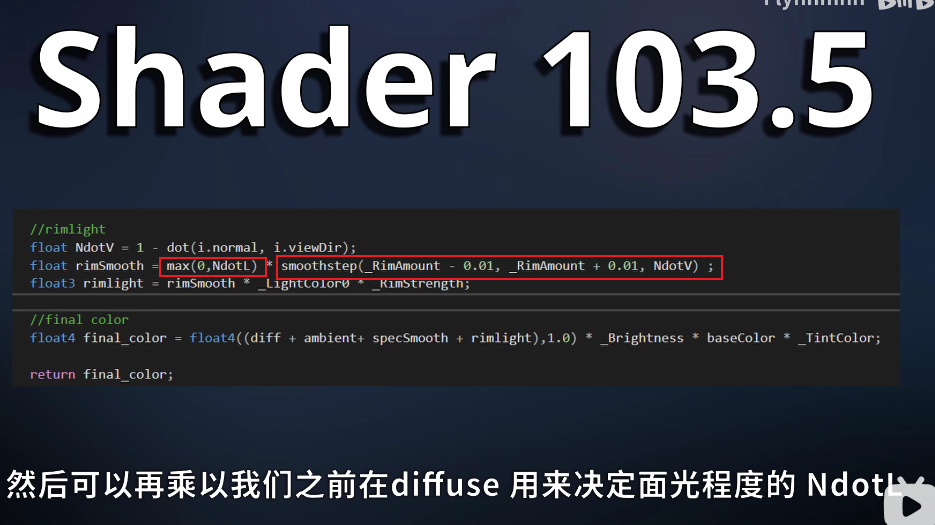


整个模型轮廓都有边缘光

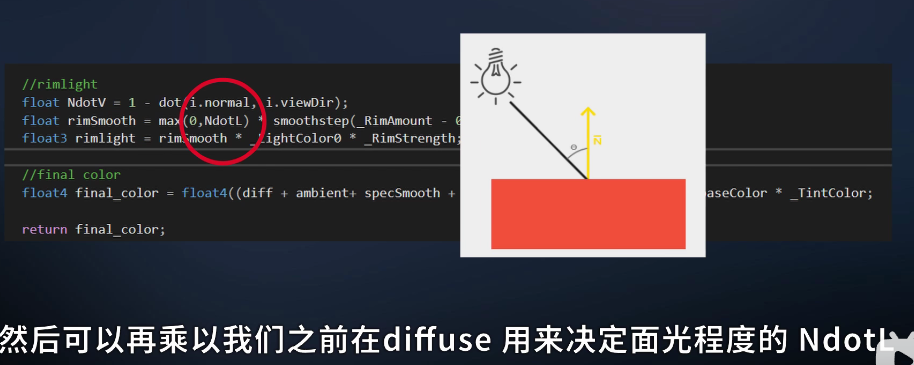
但是在Toon shader中不适用

通过smoothstep使得在边界更加清晰

为了让边缘光不产生锯齿和毛糙，会用smoothstep函数来平滑过渡



Max(0,NdotL)



实现只在朝向有光源时有边缘光

加判断范围 实现边缘分界效果





网址：<https://www.bilibili.com/video/BV1ia4y177GJ>