卡通风格的渲染

1. 渲染轮廓线

与屏幕后处理对模型描边不同，这是基于模型的描边方法。

绘制轮廓渲染线的方法：

1.基于观察角度和表面法线的轮廓线描边，这种方法简单快速，但是效果不尽如人意。

2.过程式几何轮廓线渲染。这种方法效果还可以，但是不太适用于带有尖角的物体。

3.基于图像处理的轮廓线描边法，使用古早的图像处理算法，比如边缘检测方法来实现。这种方法适用于任何模型，但是如果深度和法线变化很小，边缘检测就很难识别到边缘。

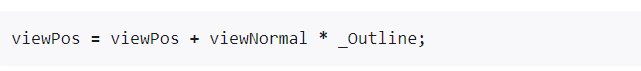
4.基于轮廓边检测的轮廓线渲染，上面提到的几种方法，它们都有一个最大的问题，即它们没有办法控制轮廓线的风格，对于一些特殊情况，我们希望可以渲染出轮廓线风格，例如水墨风格等，这种方法也有很大的缺点，除了实现起来相对复杂之外，它还会出现动画的连贯性问题，也就是说，由于是单帧提取轮廓，所以在帧与帧之间会出现跳跃性。

5.最后一种方法就是混合了上述几种渲染方式，在不同的部分用不同的方式来完成轮廓线渲染。

本例中使用过程式几何渲染方法。

绘制两个pass，第一个pass在视角空间下将顶点沿着法线向外扩张一部分的距离，绘制成纯黑色，第二个pass正常的绘制原来的物体，这样，就可以产生描边效果。

第一个pass实现方法



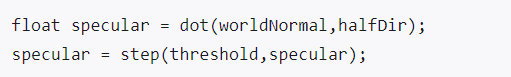
但是如果直接使用顶点进行法线方向的扩张，对于一些内凹的模型，就可能发生正面面片被背面面片遮挡的情况，为了尽可能的防止这种情况的出现，在扩张背面的顶点之前，我们必须先对顶点法线的z分量进行处理，使它们等于一个定值，然后再对法线归一化后，进行扩张。如下：

文本

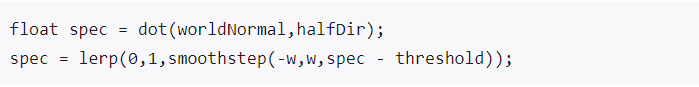
低可信度描述已自动生成

高光：

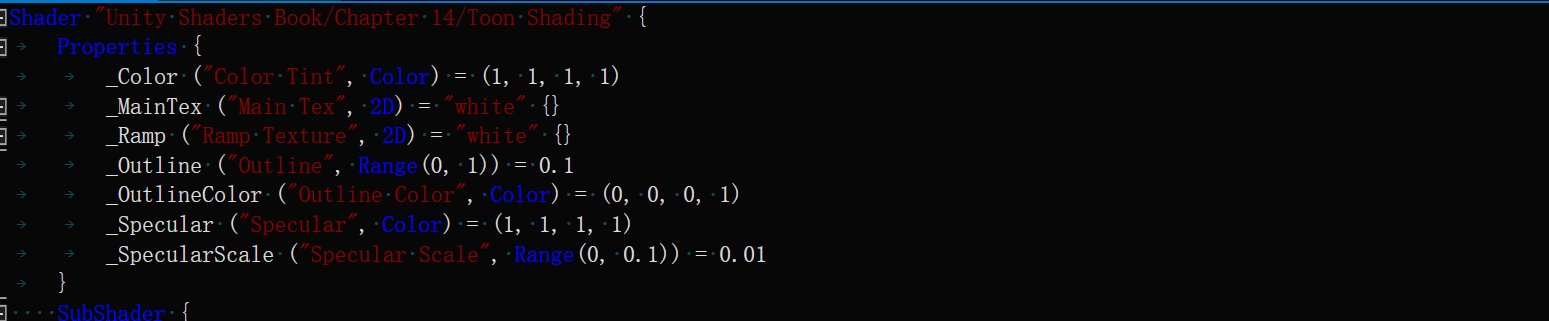
卡通风格的高光往往是模型上一块块分界明显的纯色区域，在布林-冯模型的基础上添加一个阈值做判断，把半程向量和法线的点乘结果与之比较，若大于这个阈值则高光反射系数为1，否则为0



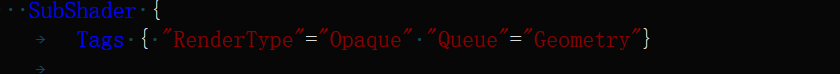
采用的是step函数，它接受一个阈值，然后将传入的参数划归到0和1的两个输出。不过这种从0到1的突变会让高光边界产生非常明显的锯齿。为了防止这种情况，在边缘处做一个过渡。



使用了smoothstep函数来处理，它接受两个阈值a和b（a < b），如果目标参数小于a，那么为0，如果目标参数大于b，则为1，否则，返回原值，然后我们把原值再丢入一个插值函数中，用该插值函数在0到1之间过度。这样就可以在[a,b]之间，也就是高光的边界区域，得到一个从0到1的变化，尽管可以把w设定为一个很小的值，但是这里还是选用了fwidth函数，它可以计算出领域像素之间的导数值，用于作为阈值。



\_Ramp是控制漫反射色调的渐变纹理，\_Outline控制轮廓线宽度，控制颜色的是\_OutlineColor，\_SpecularScale控制高光反射阈值



第一个Pass渲染背面：

形状

中度可信度描述已自动生成

形状

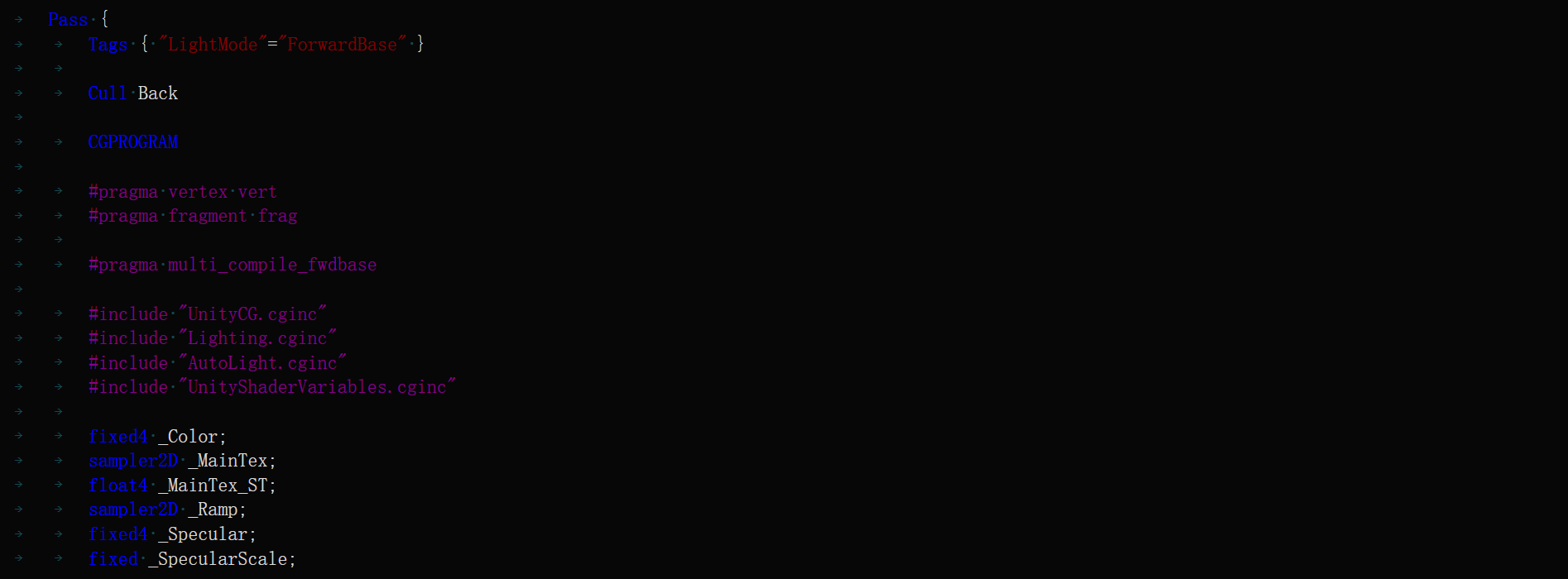
中度可信度描述已自动生成

电脑萤幕画面

描述已自动生成

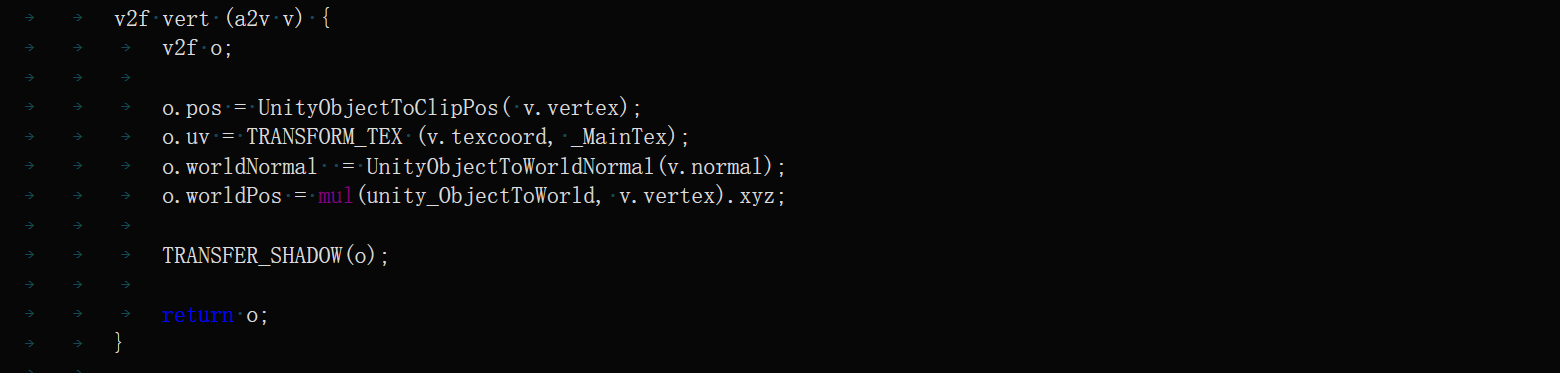
顶点着色器中把法线变换到视角空间中进行描边，原因是描边的宽度现在是相对世界空间不变的，当摄像机拉近后，显示就会变粗。我们期望无论摄像机拉近拉远，描边的粗细都能不变。要解决这个问题应通过将法线外扩的大小调整为使用NDC空间的距离进行外扩。

第二个Pass渲染正面，定义光照：



形状

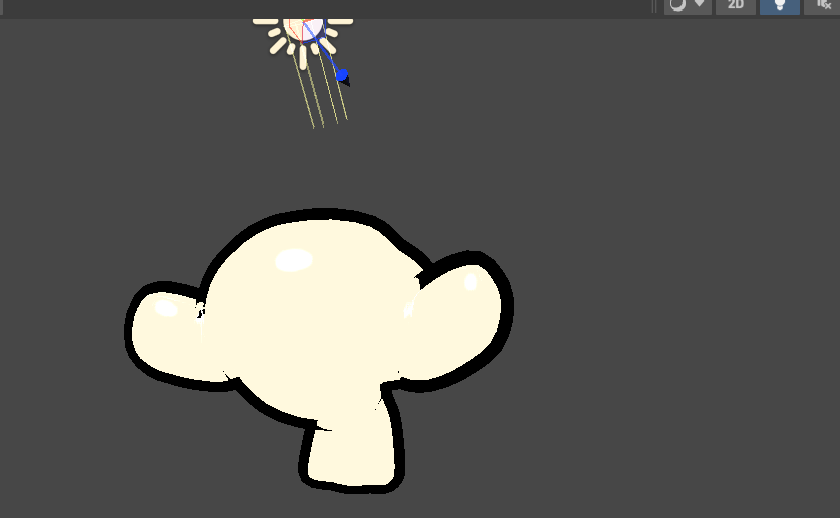
中度可信度描述已自动生成



文本

描述已自动生成

片元着色器中首先计算了布林-冯模型光照所需的各项参数，然后对纹理采样求出材料的反射率，UNITY\_KIGHTMODEL\_AMBIENT求出环境光，然后求出阴影值，之后求半兰伯特漫反射系数，和阴影值相乘得到最终的漫反射系数，使用它对阴影纹理采样和反射率光照颜色相乘得到最终的漫反射光。镜面反射中使用fwidth对高光区域的边界进行抗锯齿处理并将计算得到的镜面反射系数和颜色相乘得到镜面反射部分。最后使用step函数，在阈值为0时可以完全消除高光反射，最后相加求出光照。



素描风格的渲染

是另外一种不同类型的非真实感渲染，它有点类似于风格转换，比如把一张图改成浮雕风格，或者素描风格，或者水墨风格。这就是非常有名的色调艺术映射（Tonal Art Map，TAM）

一些文字和图案

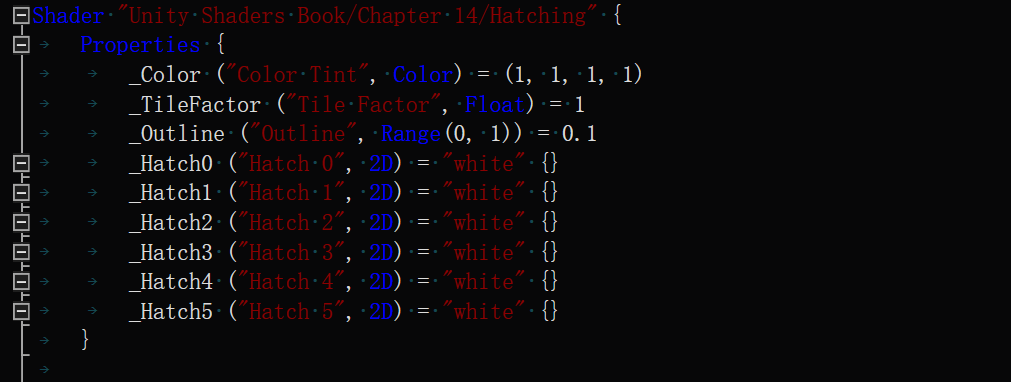
描述已自动生成

我们需要准备目标风格下的纹理图，并且用它们来表达出阴影，和渐变纹理仍然是一个概念。就是把原本的漫反射的值（一个具有大小的值）映射到这些纹理上。值比较大的对应着颜色比较浅的纹理，值比较小的对应颜色比较深的纹理。

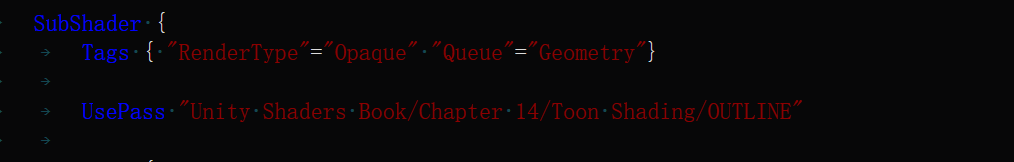
当我们计算出diffuse的值是多少的时候



需要用这个值来在6张纹理之间选择一个合适的纹理来表达出当前的漫反射亮度。那么我们可以把亮度除以6，因为亮度原本是一个0到1的值，比如亮度在[0,1/6]之间，我们就选择最暗的图。亮度在[1/6,2/6]之间，我们就选择倒数第二张图。



\_Color控制模型颜色，\_TileFactor是纹理的平铺系数，它越大模型上的线条越密



使用上例中的描边Pass

文本

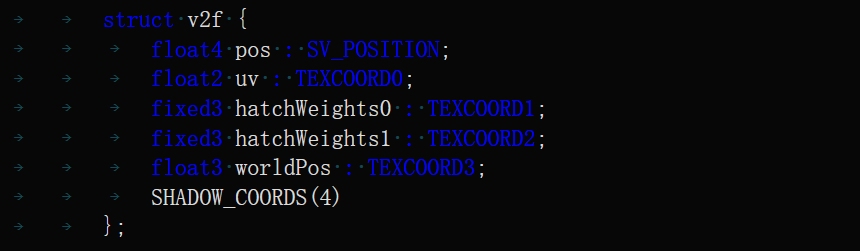
描述已自动生成

图形用户界面, 文本

描述已自动生成

文本

描述已自动生成



因为一共赋值了六张纹理，所以需要六个混合权重，把它们存储在两个fixed3类型变量中



文本

描述已自动生成

顶点着色器中进行顶点的基本变换，使用\_TileFactor得到纹理采样坐标，在计算六张纹理的混合权重前进行逐顶点光照计算得到漫反射系数，把混合权重全部初始化为0，把[0,7]区间均匀划分为7个子区间，声明一个hatchfactor用于计算对应的纹理混合权重

文本

中度可信度描述已自动生成

片元着色器中对每张纹理采样并和对应的权重值相乘得到每张纹理的采样颜色，并且计算一个纯白色的贡献度，因为我们希望在光照最亮的部分是白色的，最后混合各个颜色求出渲染结果

