**卡通渲染**

**What:**

**Why:**

**How:**

**如何实现卡通渲染？**

要实现卡通渲染有很多方法，其中之一就是基于色调的着色技术。在实现中，我们往往会使用漫反射系数对一张一维纹理进行采样，以控制漫反射的色调。卡通风格的高光往往是一块块分界明显的纯色区域。

除了光照模型不同之外，卡通风格通常还需要在物体边缘部分绘制轮廓。本节，我们使用基于模型的描边方法。这种方法简单，很多情况下也有不错的效果。

**常见的5种绘制模型轮廓的方法。**

1. **基于观察视角和表面法线的轮廓线渲染。**使用视角方向和表面法线的点乘结果来得到轮廓线的信息。这种方法简单快速，一个Pass中就得到渲染效果，但局限性很大，很多模型渲染出来的描边效果不尽如人意；
2. **过程式几何轮廓线渲染。**这种方法的核心是使用两个Pass渲染。第一个Pass渲染背面的面片，并使用某些技术让它的轮廓可见；第二个Pass再正常渲染正面的面片。这种方法快速有效，并且适用于绝大多数表面平滑的模型，但它的缺点是不适合类似于立方体这样的平整的模型；
3. **基于图像处理的轮廓线渲染。**在12，13章介绍的边缘检测方法就属于这个类别。这种方法的优点在于，可以适用于任何种类的模型。但是一些深度和法线变化很小的轮廓无法被检测出来，例如桌子上的纸张；
4. **基于轮廓边检测的轮廓线渲染。**上面提到的各种方法，一个最大的问题是，无法控制轮廓线的渲染风格。对于一些情况，我们希望可以渲染出独特风格的轮廓线，例如水墨风格等。为此，我们希望可以检测出精确的轮廓边，然后直接渲染它们。检测一条边是否是轮廓边的公式很简单，我们只需要检查和这条边相邻的两个三角形面片是否满足：(n0·v > 0) ≠ (n1·v > 0)。上述公式本质上就在于检查两个相邻的三角面片是否一个朝正面，一个朝背面。缺点是动画连贯性会有问题。
5. **混合上述几种渲染方法。**

**本节我们使用的渲染轮廓线方案是什么？**

使用过程式几何轮廓线渲染的方法来对模型进行轮廓描边。我们将使用两个Pass渲染模型：在第一个Pass中，我们会使用轮廓线颜色渲染整个背面的面片，并在视角空间下把模型顶点沿着法线方向向外扩张一段距离，以此来让背部轮廓线可见。

代码如下：viewPos = viewPos + viewNormal \* \_Outline;

为了防止内凹的模型发生背面面片遮挡正面面片的情况。在扩张背面顶点前，我们首先对顶点法线的z分量进行处理，扩展后的背面更加扁平化：

viewNormal.z = -0.5;

viewNormal = normalize(viewNormal);

viewPos = viewPos + viewNormal \* \_Outline;

**添加高光**

对于卡通渲染需要的高光反射光照模型，我们计算normal和halfDir的点乘结果，我们把该值和一个阈值进行比较，如果小于该阈值，则高光反射系数为0，否则返回1。