2024年12月8日星期日

圆角图片shader学习笔记

参考连接：https://blog.csdn.net/Vitens/article/details/107371190?spm=

圆角图像生成原理：

首先应明确一点，unity中的纹理坐标的原点是左下角，右上角为（1，1）

图表, 折线图

描述已自动生成

如上图所示，蓝色部分为unity默认的纹理坐标系。

如果要在unity默认的纹理坐标系中进行计算的话，会有很多不便。

为了方便后续的计算，定义一个坐标系，以unity纹理坐标系的(0.5,0.5)为原点。如图红色部分的坐标系。

图表, 形状

描述已自动生成

在该坐标系下，可以将所有的uv转换到第一象限进行计算。

absUV = abs(o.uv);

在我们的期望坐标系下，构造一个圆,圆的半径为\_R。取值范围为[0,0.5],0代表贴图的四个角为直角。

此时，可以得出一个阙值(圆心位置)，当absUV中有一个值在阙值之内，就表明这个点不需要进行透明处理，即1- (stepX \* stepY)中，当在阙值内时，则将其值设置为0。

threshold = 0.5 - \_R;

stepX = step(threshold , absUV.x);

stepY = step(threshold , absUV.y);

alpha = 1 – stepX \* stepY;

此时在经过片元着色器处理后就得到了四个角根据\_R的值动态变化的图形，但是，其四个角是正方形，并不是我们期望的圆角。

对于圆角图形，我们仅期望如图中最右上角的区域为透明。

于是，还需要再进行一步计算，即uv坐标与圆心之间的距离是否小于半径，如果小于半径则不需要对其进行透明处理。

首先计算出圆心坐标：

center = fixed2(threshold , threshold);

再求出一个向量

v = center – absUV;

对比半径的平方与uv坐标到圆心之间距离的平方，如果在圆内，则不需要进行透明处理

stepL = step(\_R\*\_R , dot(v,v));

alpha = 1 – stepX \* stepY \* stepL;

在上述alpha值的计算中，只要x在阙值内、y在阙值内、absUV落在圆内，三个条件满足一个则不需要对图像进行透明处理。

将以上算法添加在片元着色器中，可以得到如下的效果：

