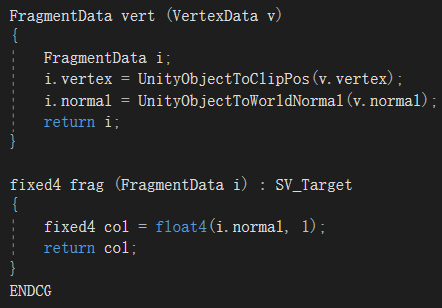
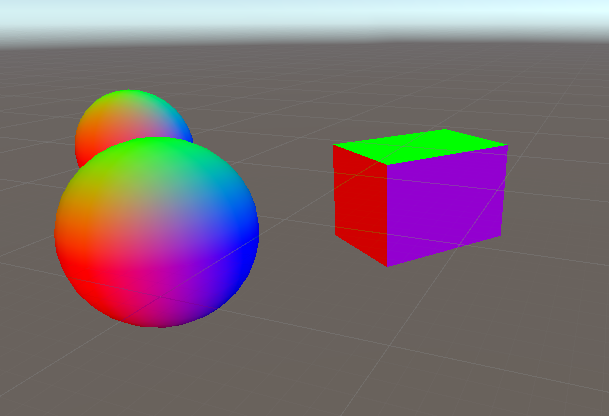
一、Shader 的实现过程、实现结果展示

1）法线贴图

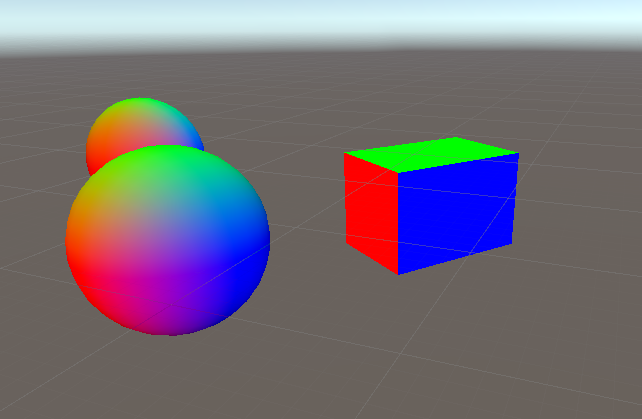
代码：



效果：



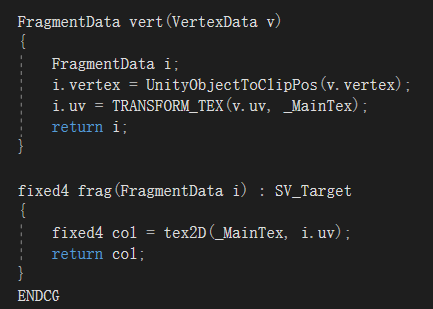
当FragmentData.normal不做变换时：



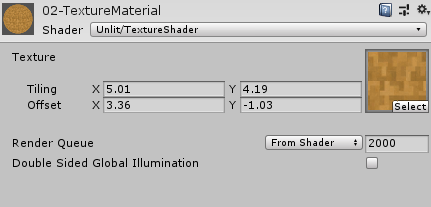
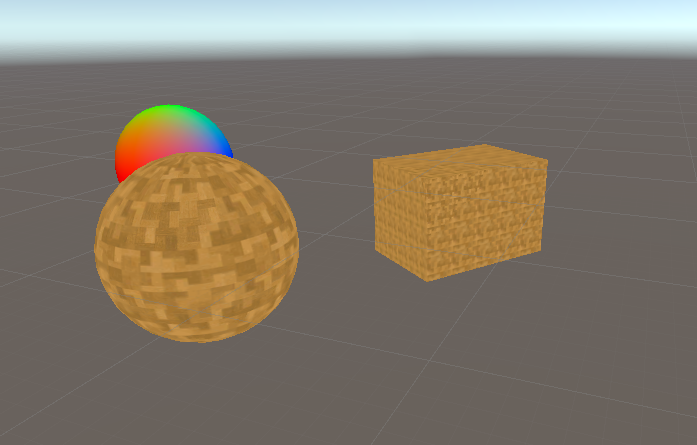
此时法线贴图随物体自身旋转，而不是在世界参照系中始终恒定。

1. 纹理贴图

代码：

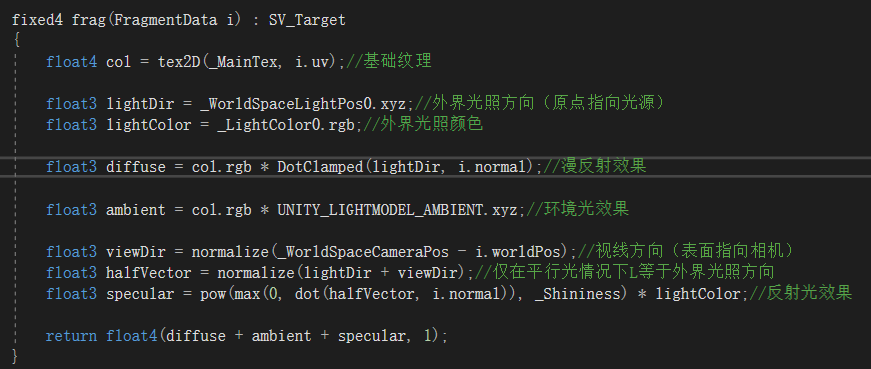


效果：

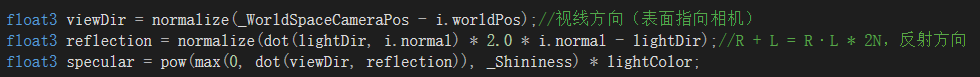


1. 光照效果

代码：

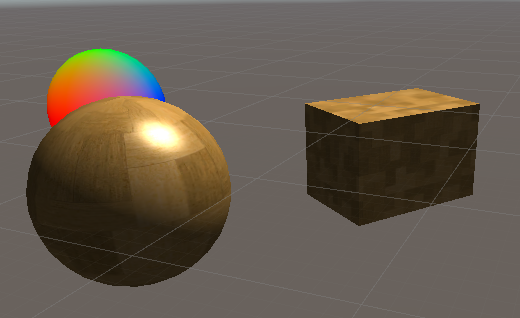
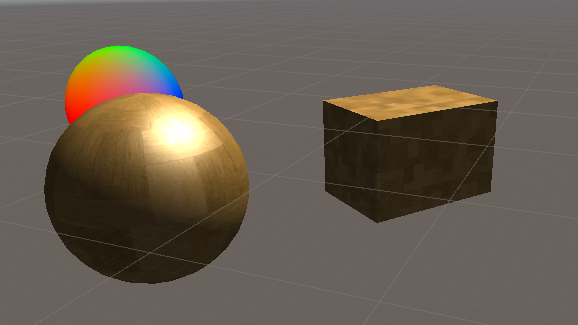
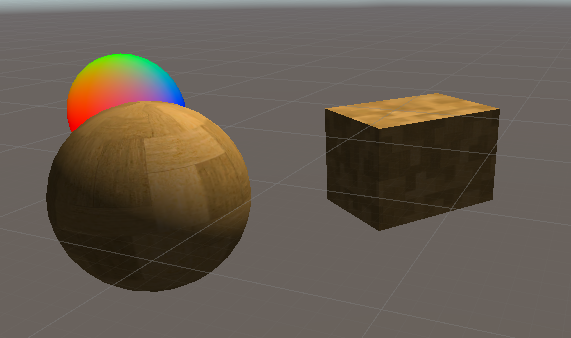
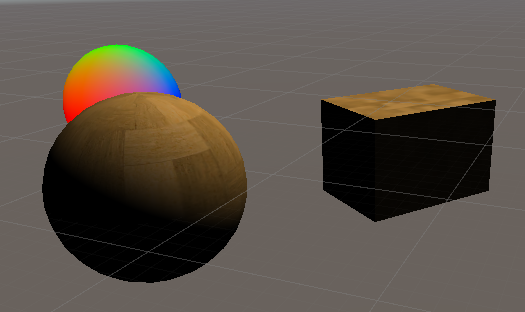


其中diffuse为漫反射效果的实现，ambient为环境光效果的实现，specular为Blinn Phong模型中镜面反射效果的实现



另外还实现了Phong光照模型的效果。

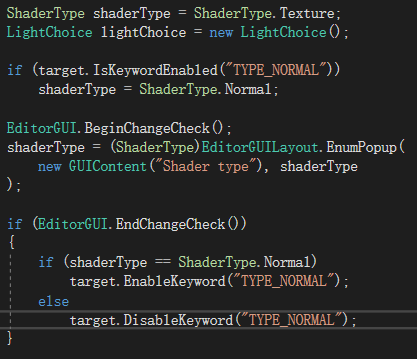
效果：



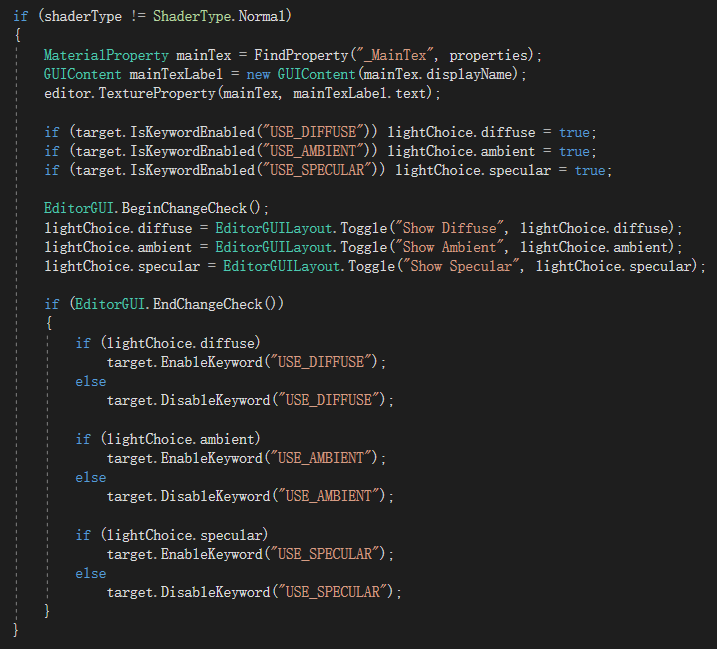
1. Shader GUI 的实现过程、实现结果展示

代码：

刚开始提供Normal和Texture两个选项



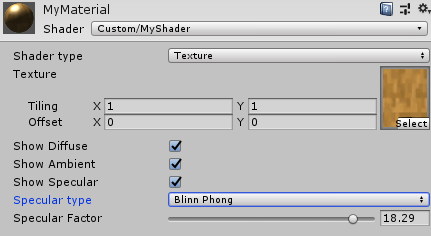
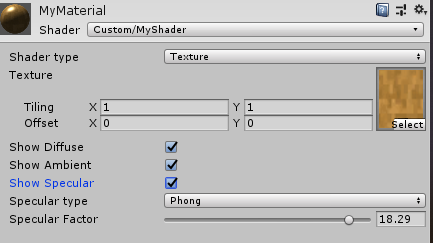
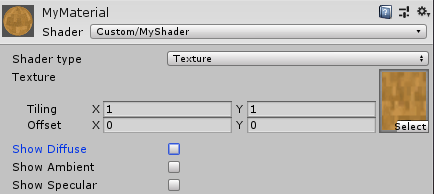
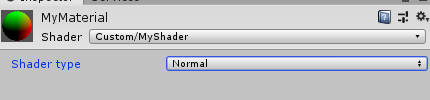
如果选择了Texture，则有Texture选择框和Diffuse，Ambient，Specular三个可勾选选项



如果勾选了Specular，可以选择使用Blinn Phong光照模型还是Phong光照模型，并且可以调节Shininess参数

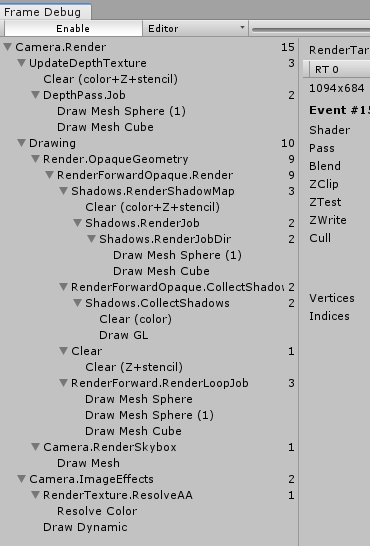


效果：

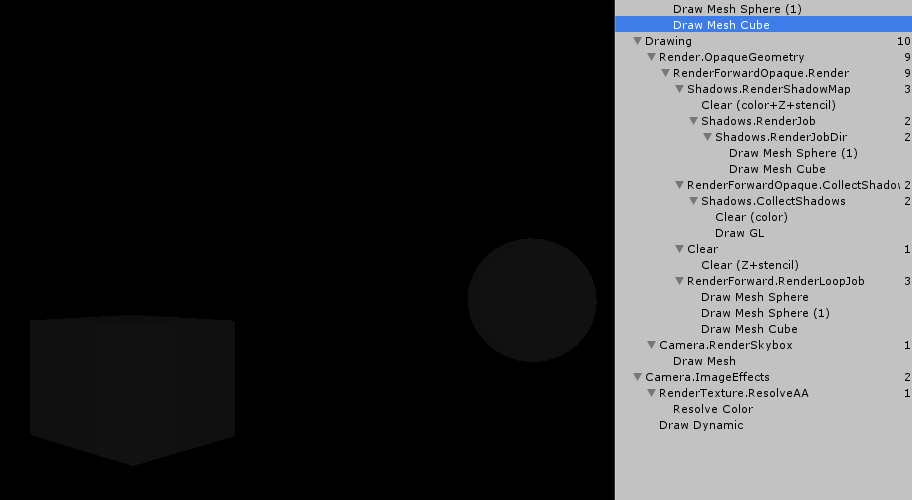
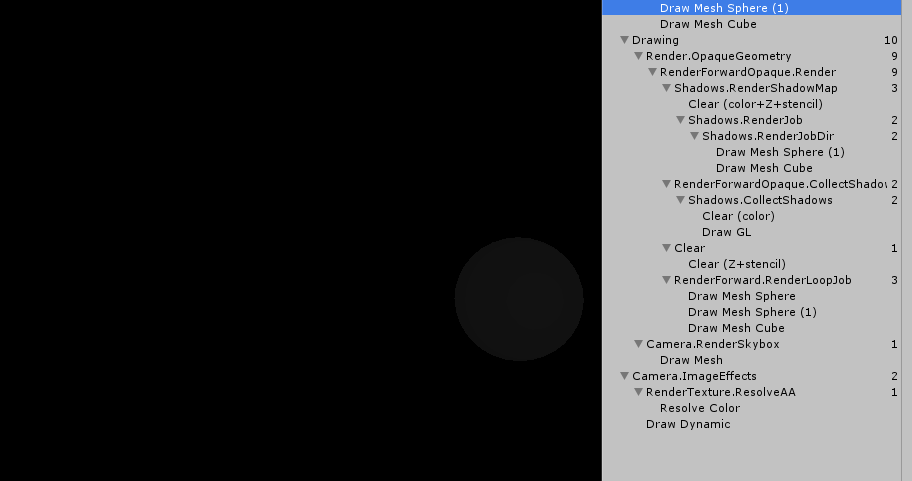


1. Debug 工具的使用过程

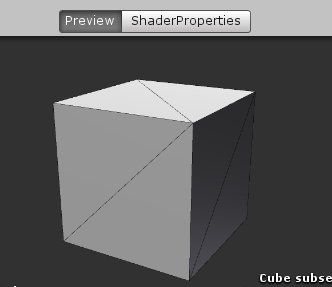
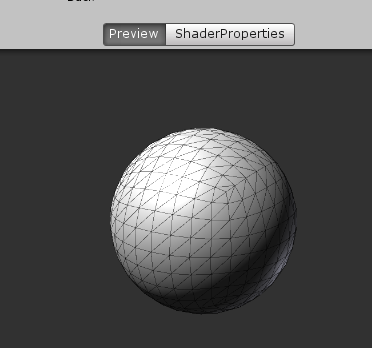
打开Frame Debug然后点击Enable



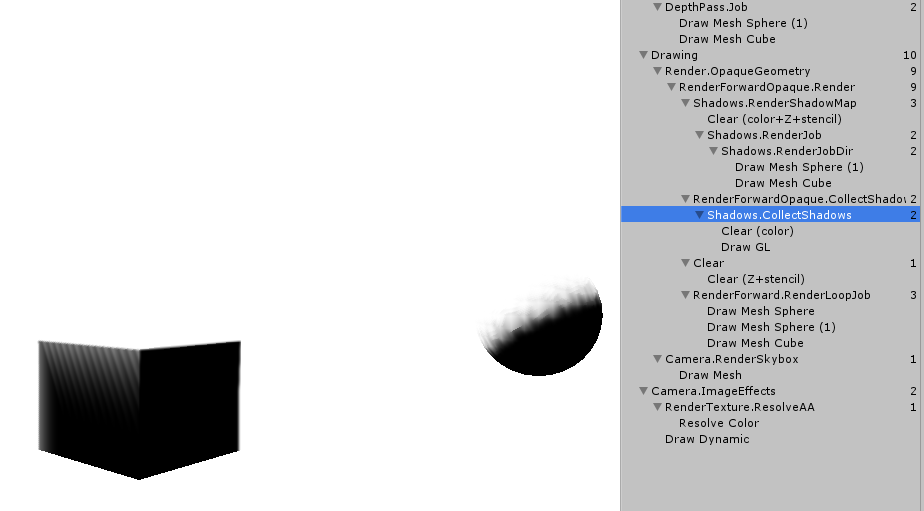
在Debugger中可以查看场景是如何被渲染的



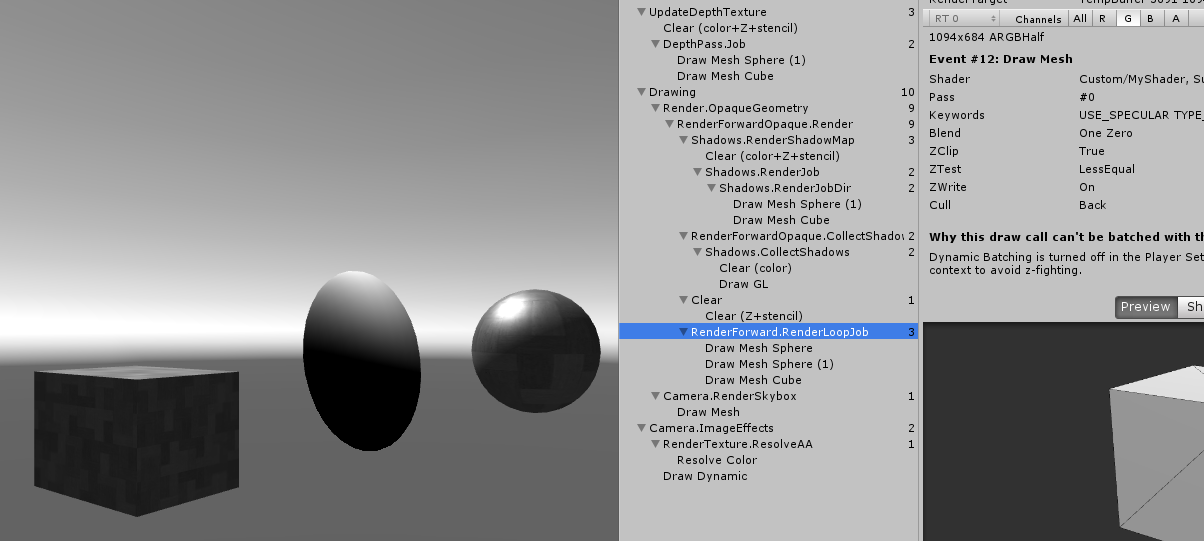
查看模型点面特征



查看部分渲染效果



RGB通道中的效果等等



1. 叠加波实现水面、海浪动态Shader

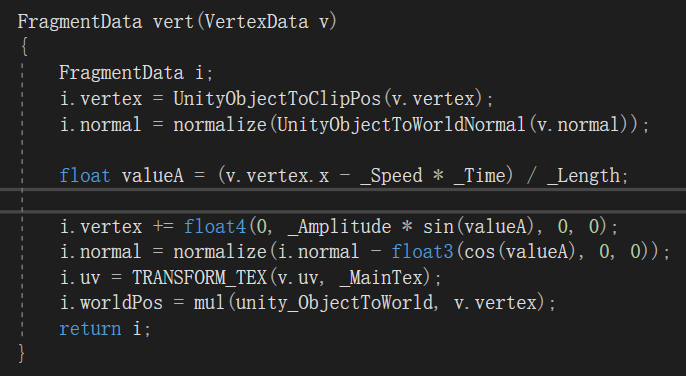
首先计算出波平面在x轴上的函数f(x) = H \* sin(2Π(x0 - vt)/L)

然后算出frag坐标相对vert坐标的变化是y的值增加了H \* sin(2Π(x0 - vt)/L)

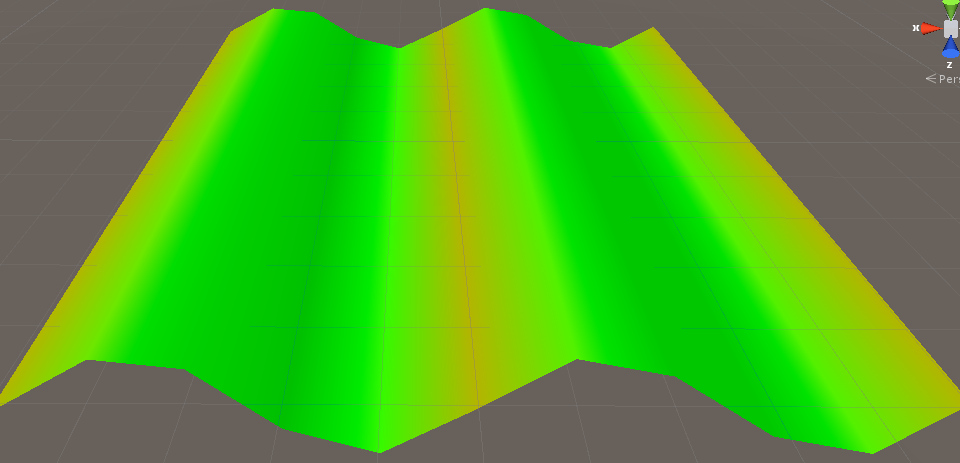
而法线向量的变化是x轴减少cos(2Π(x0 - vt)/L)

用法线贴图检测法线是否正确

代码：



效果：



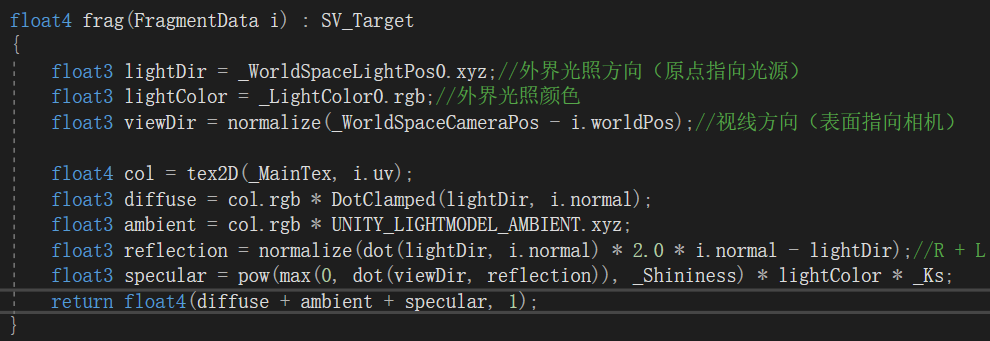
可以看出波浪的波峰和波谷法线向上，右侧的法线向右，符合预期。

然后将加载纹理贴图，并且根据法线计算出波浪的阴影和反光等

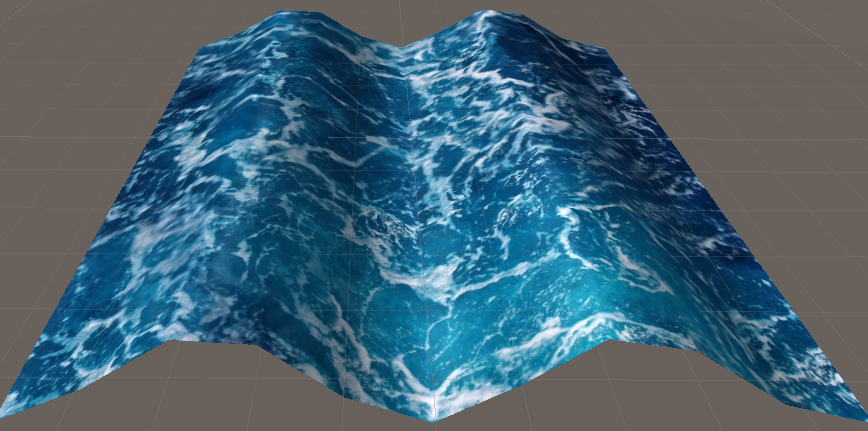
额外设置参数Ks以改变反光率

其他参数中Speed表示波的速率，Length表示波长，Amplitude表示波的振幅，Shininess表示海浪的光滑程度。

代码：

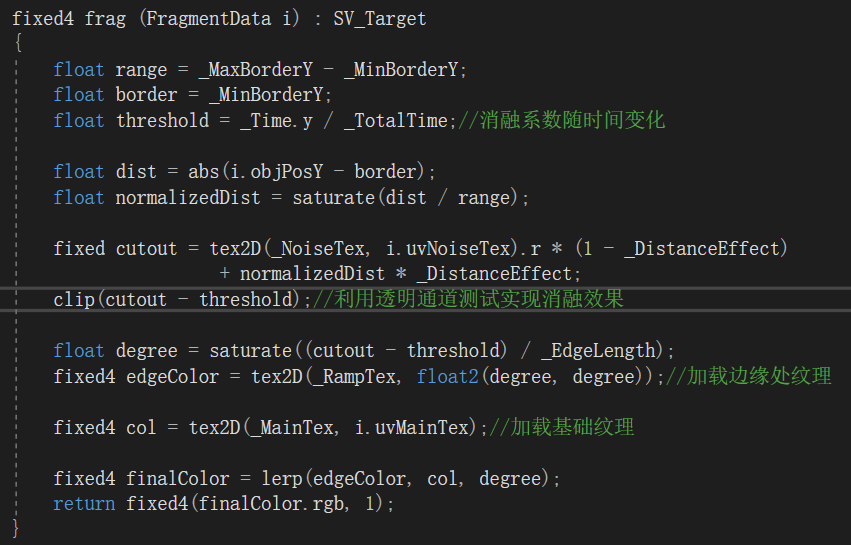


效果：

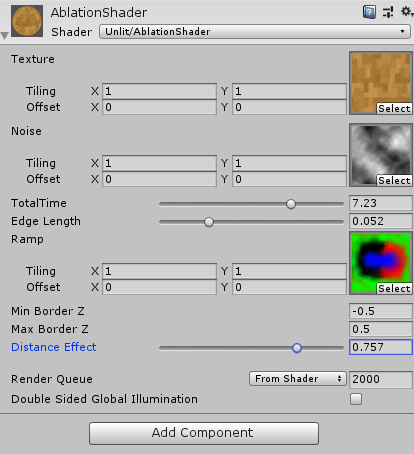


1. 使用噪声纹理实现动态消融效果

代码：



效果：



其中TotalTime表示消融的总耗时

Edge Length表示消融边缘的宽度

Distance Effect表示消融边缘的平整度

