

水材质模块功能记录

材质属性

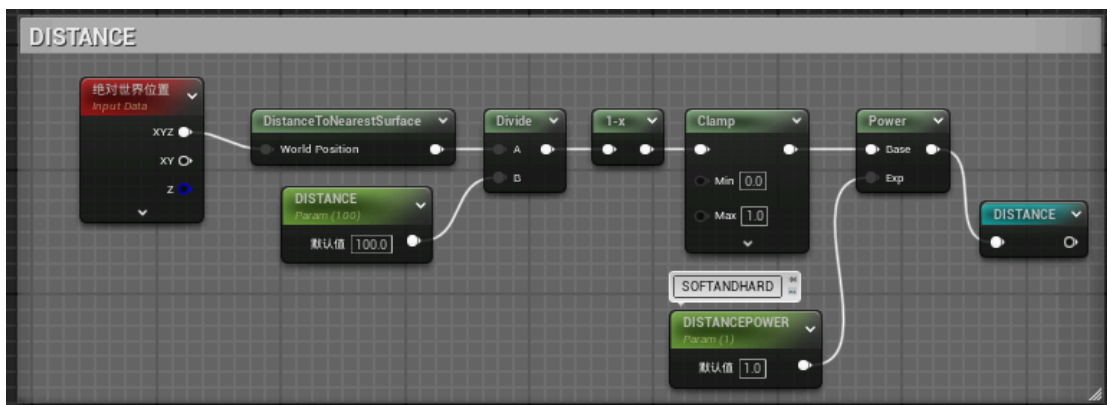
半透明默认光照 (光照模式半透明体积, 半透明通道: 景深前)

折射方法: 像素正常偏移

随 Nanite 使用, 开启距离场计算

基础计算部分

距离场

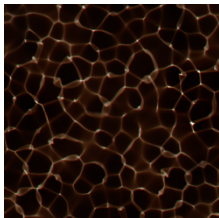


Offset

通过世界位置的 xy 轴数值输入进 panner 节点对线性颜色类型的贴图 UV 做动画, 然后通过此图的 xy 值还原成法线的 xyz 轴数值, 将还原后的 normal 值 xy 分别和远距离法线强度, 近距离法线强度相乘, 再和原本的 z 轴组合成两个 vector3。

相机位置减去世界位置计算相机到点的距离长度然后 $\sim 1/2$, 归一化, 然后作为远和近的法线的过渡值。

基础色和发光色部分



用焦散图作为 maske 图渲染水体颜色

加入泡沫边缘

焦散计算方法

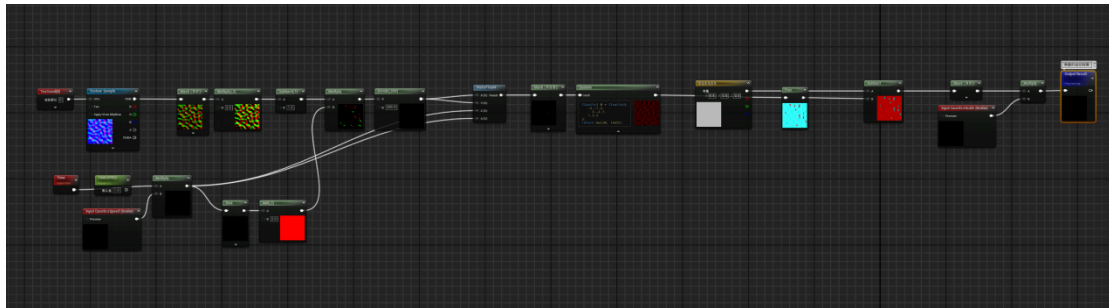
1 首先计算水深:

$$\text{depth} = \frac{\text{场景深度} \times (\text{相机高度} - \text{世界位置})}{\text{像素深度}} - (\text{相机高度} - \text{世界位置})$$

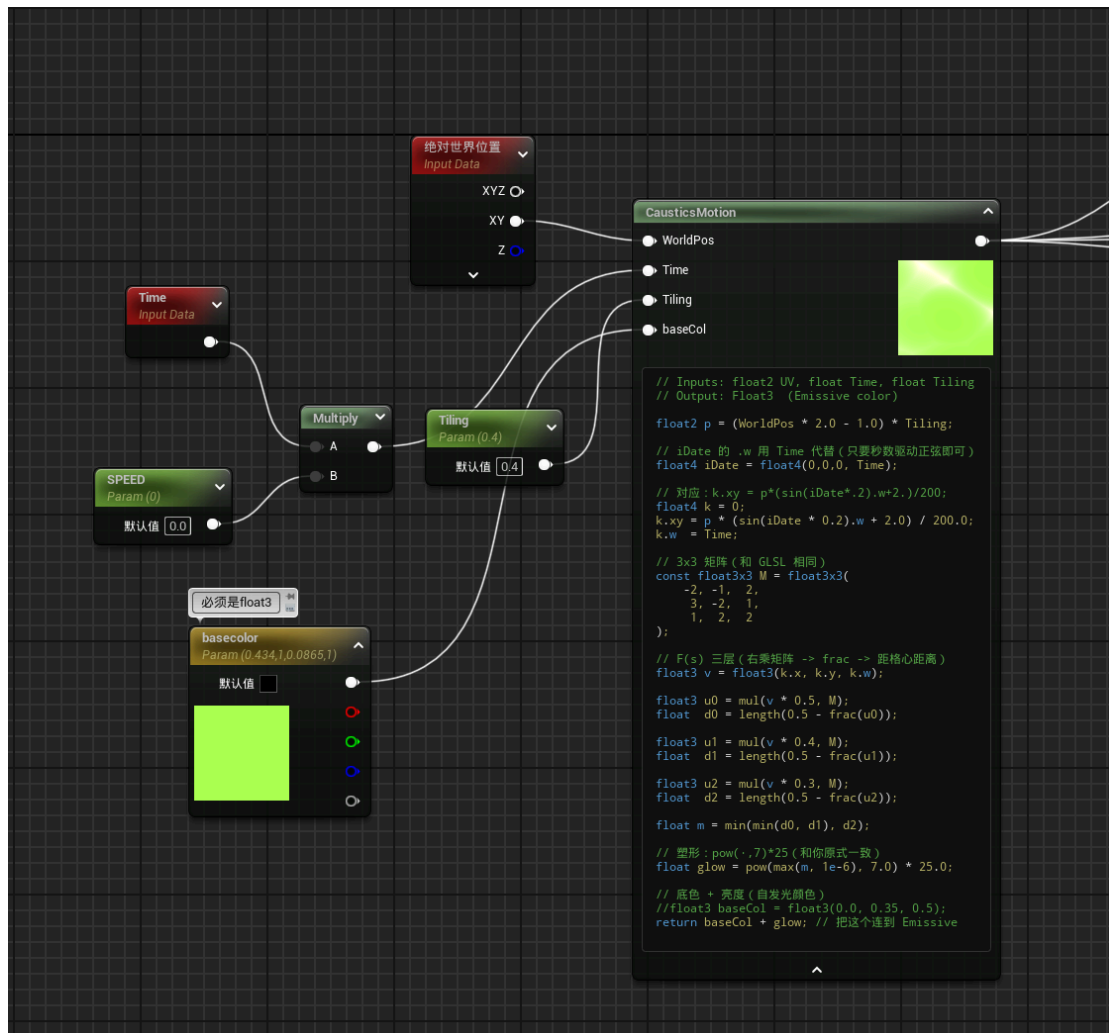
将场景与像素深度的比例映射到水的 z 轴高度再减去原本 z 轴高度得到水深

2 焦散的运动

获取 UV 通过法线进行扰动，计算后封装成焦散的材质函数



第一版



The image displays a Unity Shader Graph for a custom 'CAUSTICS' material. The graph is set against a dark grid background. Key nodes and their connections include:

- TexCoord0** (Texture Coordinate) and **DEPTH XY** (Depth) are inputs to a **Multiply** node.
- UV Tiling** (UV Tiling) is an input to a **Texture Sample** node.
- Texture Sample** (Texture Sample) outputs to a **Multiply** node.
- Mask (R,G)** (Mask (R,G)) and **Caustic Inten** (Caustic Inten) are inputs to a **Multiply** node.
- Multiply (0.001)** (Multiply (0.001)) is a constant multiplier.
- Color** (Color) is a color input.
- Time** (Time) and **SPEED** (Speed) are inputs to a **Multiply** node.
- CAUSTICS COLOR** (CAUSTICS COLOR) is the final output node.

The **Custom** node contains the following C# code:

```
// compile-time constants
static const float MAX = 0.2318520718;
static const int MAX_ITER = 5;
float inten = CausticInten;

// some math ...
float2 p = frac(CV * TAU, TAU) * 250.0;
float time = Time * 0.5 + 23.1;

float2 ii = p;
float c = 1.0;

[unroll]
for (int n = 0; n < MAX_ITER; n++)
{
    float t = time * (1.0 - (3.5 / (float)(n + 1)));
    if (p.x < float(t))
        cosf(-11.2) * sinf(-11.2);
        sinf(-11.2) * cosf(-11.2);
    };

    float2 demon = float2(
        p.x / (sin(11.2) + 1) / inten,
        p.y / (cos(11.2) + 1) / inten);
    };

    c += 1.0 / length(demon);
}

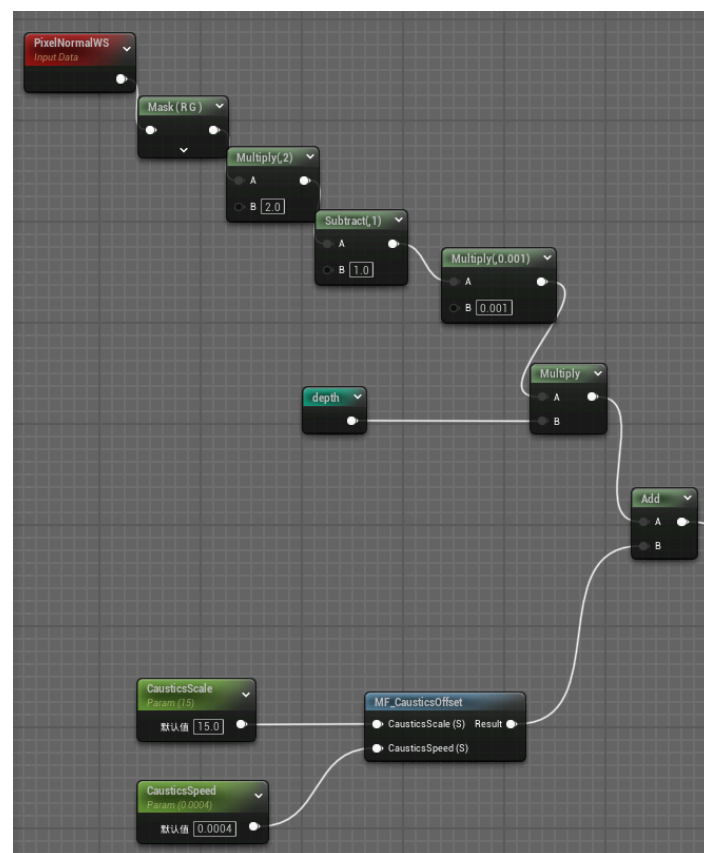
c /= (float)MAX_ITER;
c = 1.17 - pow(c, 1.4);

float3 color = pow(abs(c), 8.0).xxx;
color = clamp(color) * CausticsColor * 0.0, 1.0);
color = lerp(color, float3(1.0, 1.0, 1.0), 0.3);

// Custom node must 'return'
return color;
```

3 将像素法线映射到[-1,1]得到水面切线位置，将这个值和水深相乘，再和焦散 UV 值相加，就得到了焦散的正确位置和颜色

4 通过焦散贴图的 R 和 B 作为 Mask 遮罩最后输出基础色



泡沫部分

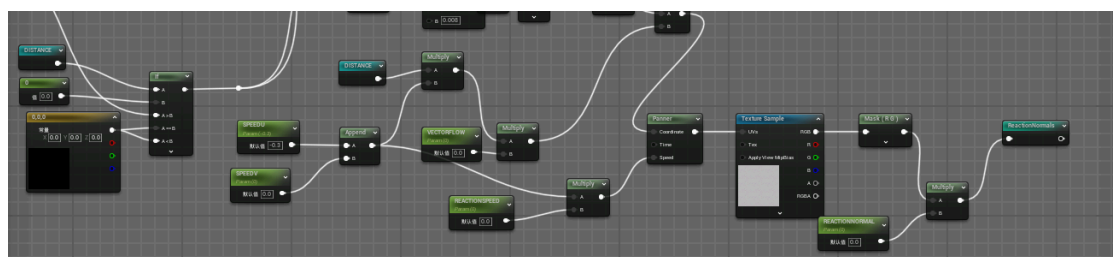
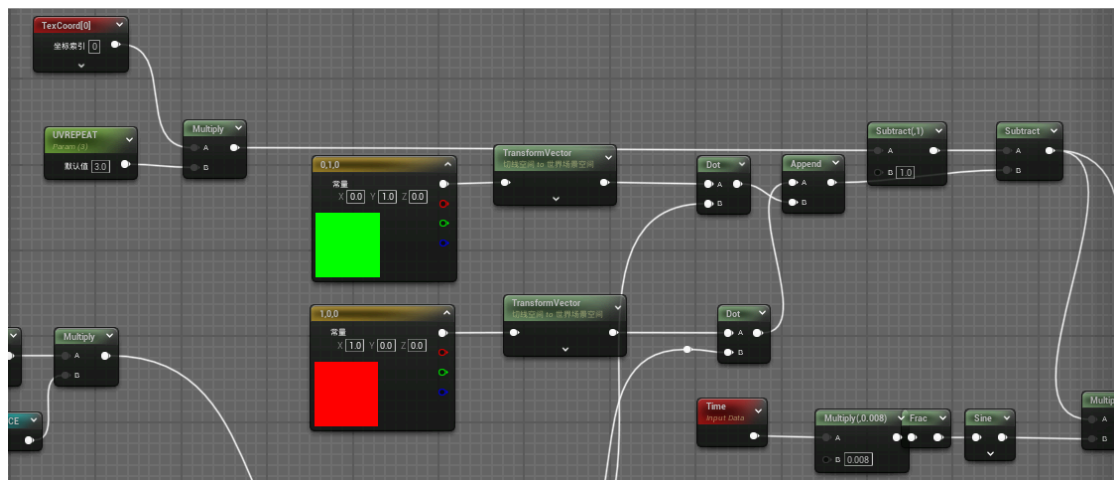
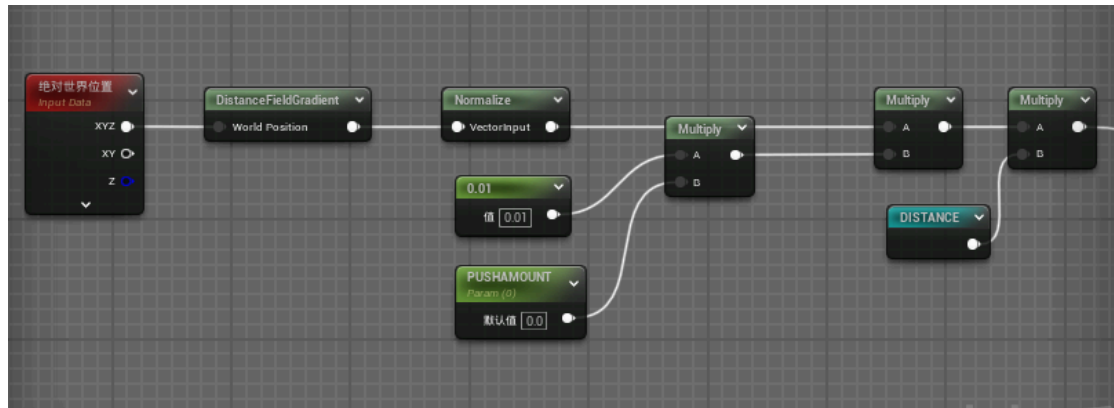
使用距离场作为 UV 的 U 的 mask，对泡沫贴图进行 offset 扰动，和基础色混合，创建 staticbool

控制是否开启泡沫显现

加入菲涅尔做水面边缘颜色

法线部分

距离场控制交互法线



和使用了 offset 的 UV 的法线贴图相加，使用 staticbool 控制

加入边缘计算

