

# Tree

## 简介

历史遗留产物,我只简单介绍一下。基于场景物体的光照计算,但有两点不同:

1. 顶点中提供了叶片摆动计算
2. 提供了Ambient-Occlusion的烘培

## 效果介绍



## 参数说明

1. Shake
  1. 开启抖动
  2. 抖动速度
  3. 抖动强度
  4. 风速
  5. 风力
  6. 底部限定
2. ambient occlusion (需要将模型放在“Ambient-Occlusion”目录下重导入)
  1. 开启内部暗部
  2. 内部阴暗度

## 技术说明

1. 抖动(这块的性能可能有些问题,因为树叶顶点挺多)

```
// 一个噪声t1影响周期性随风摆动,一个噪声t2控制自身随机的自由摆动
inline float3 Shake(float3 worldPos, half3 vColor, half3 verPos, half2 uv)
{
```

```

half nois = frac(sin(dot(uv, float2(12.9898, 78.233))) * 43758.5453);
half noisX = frac(sin(dot(worldPos.xz, float2(12.9898, 78.233))) * 43758.5453);
half noisY =
frac(sin(dot(worldPos.xy, float2(12.9898, 78.233))) * 43758.5453) * 0.25;
half noisZ =
frac(sin(dot(worldPos.zy, float2(12.9898, 78.233))) * 43758.5453) * 0.5;

half groupid = vColor.r + vColor.g + vColor.b * 0.5;
half wt = sin(_Time.y * _WindSpeed + worldPos.z);
// 随风向摇动
half t1 = sin((nois + wt * 10) * groupid * 3.1415 / 5) * _windRandom * 0.125;
// 自身随机摇动
half t2 = sin((_Time.y * _ShakeSpeed * 10 * noisX * noisZ) * groupid *
3.1415 / 8) * _ShakeStrength * noisY;
// _ShakeGradual取0存在精度问题，导致_ShakeGradual实际为极小的负值，最终产生一个极大的
y偏移
half y = pow(smoothstep(-1, 1, verPos.y), saturate(_ShakeGradual) + 0.01);
worldPos.xz += t1 * y;
worldPos.xyz += t2 * y;
return worldPos;
}

```

## 2. 环境遮蔽，Unity的实现机制

1. 需要将模型放在“Ambient-Occlusion”目录下重导入，来把ao烘焙到tangent.w上

```
o.darkInside = v.tangent.w * _DarkStrength;
```

2. [参考](#)