**深度纹理**

**What:**

**Why:**

**How:**

**Unity是如何得到深度纹理的？**

使用延迟渲染路径时，深度纹理可以从G-buffer中访问到。

当无法直接获取深度缓存时，深度和法线纹理是通过一个单独的Pass渲染而得到的。Unity会使用着色器替换技术选择那些渲染类型为Opaque的物体，判断它们使用的渲染队列是否小于等于2500(内置的Background，Geometry和AlphaTest渲染队列均在此范围)，如果满足条件，就把它渲染到深度和法线纹理中。

**选择生成一张深度纹理时，Unity内部怎么处理？**

Unity会直接获取深度缓存或是使用着色器替换技术，选取需要的不透明物体，并使用它投射阴影时使用的Pass(即LightMode被设置为ShadowCaster的Pass)来得到深度纹理。如果Shader中不包含这样一个Pass，那么物体就不会出现在深度纹理中。

**如果选择生成一张深度+法线纹理，Unity会怎么处理？**

Unity会创建一张和屏幕分辨率相同，精度为32位的纹理，其中观察空间下的法线信息会被编码进纹理的R和G通道，而深度信息被编码进B和A通道。法线信息的获取在延迟渲染中非常容易，只需合并深度和法线缓存即可。而在前向渲染中，默认情况下是不会创建法线缓存的，因此Unity底层使用了一个单独的Pass把整个场景在此渲染一遍来完成。这个Pass被包含在Unity内置的一个UnityShader中。

builtin\_shaders-xxx/DefaultResources/Camera-DepthNormalTexture.shader。

**如何再Unity中获取深度和法线纹理？**

获取深度纹理：

camera.depthTextureMode = DepthTextureMode.Depth

在Shader中通过声明\_CameraDepthTexture变量来访问它。

获取深度+法线纹理：

camera.depthTextureMode = DepthTextureMode.DepthNormals

在Shader中通过声明\_CameraDepthNormalsTexture变量来访问它。

我们还可以组合这些模式，让一个摄像机同时产生一张深度和深度+法线纹理。

**Unity内置的宏**

深度纹理采样宏：

float d = SAMPLE\_DEPTH\_TEXTURE(\_CameraDepthTexture, i.uv)

SAMPLE\_DEPTH\_TEXTURE\_PROJ宏

SAMPLE\_DEPTH\_TEXTURE\_PROJ宏同样接受两个参数，深度纹理和一个float3或float4类型的纹理坐标，纹理坐标的前两个分量首先会除以最后一个分量，在进行纹理采样。如果提供了第四个分量，还会进行一次比较，通常用于阴影的实现。

SAMPLE\_DEPTH\_TEXTURE\_PROJ的第二个参数通常由顶点着色器输出插值而得的屏幕坐标：

float d = SAMPLE\_DEPTH\_TEXTURE\_PROJ(\_CameraDepthTexture,

UNITY\_PROJ\_COORD(i.scrPos))

其中，i.scrPos是在顶点着色器中通过调用ComputeScreenPos(o.pos)得到的屏幕坐标。

**如何获取深度值？**

这些深度值往往是非线性的，这种非线性来自于透视投影使用的裁剪矩阵。然而，在我们的计算过程中通常是需要线性的深度值，也就是说，我们需要把投影后的深度值变换到线性空间下，例如视角空间下的深度值。

Unity提供了两个辅助函数LinearEyeDepth和Linear01Depth。

LinearEyeDepth负责把深度纹理的采样结果转换到视角空间下的深度值，而Linear01Depth则会返回一个范围在[0,1]的线性深度值。

**如何获取深度+法线纹理？**

inline void DecodeDepthNormal(float4 enc, out float depth, out float3 normal){

depth = DecodeFloatRG(enc.zw);

normal = DecodeViewNormalStereo(enc);

}

获取的是范围为[0,1]的线性深度值和视角空间下的法线方向。