# 立方体纹理

立方体纹理 是 环境映射 的一种方法。

若希望 根据物体在场景中位置的不同，生成它们各自不同的立方体纹理。使用的是Unity的Camera.RenderToCubemap 函数，该函数可以把从任意位置观察到的场景图像存储到6张图中，从而创建出该微商对应的立方体纹理。

环境映射最常见的应用是反射和折射：

## 反射

天空盒中的物体，对天空盒进行反射。简单来说，就是通过反射方向，对立方体纹理进行采样。

1 反射方向是什么？

天空盒上的某一点，到物体表面，进入摄像机。我们知道 物体法线，知道摄像机位置，可以得到天空盒上的某一点。

v2f vert(a2v v)

{

v2f o;

o.pos = UnityObjectToClipPos(v.vertex);

o.worldPos = mul(unity\_ObjectToWorld, v.vertex).xyz;

o.worldNormal = UnityObjectToWorldNormal(v.normal);

o.worldViewDir = UnityWorldSpaceViewDir(o.worldPos);

o.worldRefl = reflect(-o.worldViewDir, o.worldNormal);//注意这里使用的是摄像机方向。

o.uv = TRANSFORM\_TEX(v.texcoord, \_MainTex);

TRANSFER\_SHADOW(o);

return o;

}

2 采样。

fixed4 frag(v2f i) : SV\_Target

{

float3 worldPos = normalize(i.worldPos);

float3 worldViewDir = normalize(i.worldViewDir);

float3 worldLightDir = UnityWorldSpaceLightDir(i.worldPos);

fixed3 uvColor = tex2D(\_MainTex, i.uv).rgb;

fixed3 albedo = \_Color.rgb \* uvColor;

fixed3 ambient = UNITY\_LIGHTMODEL\_AMBIENT.xyz \* albedo;

// 对Cube纹理采样，并乘以反射主颜色

fixed3 cubeColor = texCUBE(\_ReflectCube, i.worldRefl).rgb \* \_ReflectColor.rgb;

fixed3 diffuse = \_LightColor0.rgb \* albedo \* max(0, dot(i.worldNormal, worldLightDir));

UNITY\_LIGHT\_ATTENUATION(atten, i, i.worldPos);

// 设置最后的颜色，使用Lerp控制diffuse和 反射颜色的过度

fixed3 color = ambient + lerp(diffuse, cubeColor, \_ReflectAmount) \* atten;

return fixed4(color, 1.0);

}

## 折射

v2f vert(a2v v)

{

v2f o;

o.pos = UnityObjectToClipPos(v.vertex);

o.worldPos = mul(unity\_ObjectToWorld, v.vertex).xyz;

o.worldNormal = UnityObjectToWorldNormal(v.normal);

o.worldViewDir = UnityWorldSpaceViewDir(o.worldPos);

// 获取折射方向 refract(摄像机反方向，法线，入射率/出射率)

o.worldRefl = refract(-normalize(o.worldViewDir), normalize(o.worldNormal) , \_RefractRatio);

o.uv = TRANSFORM\_TEX(v.texcoord, \_MainTex);

TRANSFER\_SHADOW(o);

return o;

}

## 菲涅尔反射

反射的颜色 与 观察者看的角度有关.

例如 近处的水面可看见水底，远处的水面一片白茫茫

fixed3 cubeColor = texCUBE(\_RefractCube, i.worldRefl).rgb;

//菲涅尔的近似公式。

fixed fresnel = \_FresnelScale + (1 - \_FresnelScale)\*pow(1 - dot(worldViewDir, worldNormal), 5);

UNITY\_LIGHT\_ATTENUATION(atten, i, i.worldPos);

fixed3 color = ambient + lerp(diffuse, cubeColor, saturate(fresnel)) \* atten;