**Unity3D脚本：Unity3D 车身与玻璃材质**

Posted on 2013年01月21日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 279 次

玻璃shader：

Shader "Reflective/Glass" {   
Properties {   
\_Color ("Main Color", Color) = (1,1,1,1)   
\_SpecColor ("Specular Color", Color) = (0.5, 0.5, 0.5, 1)   
\_Shininess ("Shininess", Range (0.01, 1)) = 0.078125   
\_ReflectColor ("Reflect Strength", Color) = (1,1,1,0.5)   
\_MainTex ("Base (RGB) Gloss (A)", 2D) = "white" {}   
\_Parallax ("Height", Range (0.005, 0.08)) = 0.02   
\_Cube ("Reflection Cubemap", Cube) = "\_Skybox" { TexGen CubeReflect }   
}   
SubShader   
{   
LOD 300   
Tags { "Queue"="Transparent" "IgnoreProjector"="True" "RenderType"="Transparent"}   
Blend one OneMinusDstColor   
ZWrite Off

// First pass does reflection cubemap   
Pass   
{   
Name "BASE"   
Tags {"LightMode" = "Always"}   
CGPROGRAM   
#pragma vertex vert   
#pragma fragment frag   
#pragma fragmentoption ARB\_fog\_exp2   
#pragma fragmentoption ARB\_precision\_hint\_fastest   
#include "UnityCG.cginc"

struct v2f {   
V2F\_POS\_FOG;   
float2 uv : TEXCOORD0;   
float3 I : TEXCOORD1;   
};

uniform float4 \_MainTex\_ST;

v2f vert(appdata\_tan v)   
{   
v2f o;   
PositionFog( v.vertex, o.pos, o.fog );   
o.uv = TRANSFORM\_TEX(v.texcoord,\_MainTex);

// calculate object space reflection vector   
float3 viewDir = ObjSpaceViewDir( v.vertex );   
float3 I = reflect( -viewDir, v.normal );

// transform to world space reflection vector   
o.I = mul( (float3x3)\_Object2World, I );

return o;   
}

uniform sampler2D \_MainTex;   
uniform samplerCUBE \_Cube;   
uniform float4 \_ReflectColor;

half4 frag (v2f i) : COLOR   
{   
half4 texcol = tex2D (\_MainTex, i.uv);   
half4 reflcol = texCUBE( \_Cube, i.I );   
reflcol \*= texcol.a;   
return reflcol \* \_ReflectColor;   
}   
ENDCG   
}

UsePass "Parallax Specular/PPL"

}   
FallBack "Reflective/VertexLit", 1   
}

车身shader用Reflective/Specular中的一种,然后加入以下脚本：

RenderToCubemap.js:  
// Attach this script to an object that uses a Reflective shader.  
// Realtime reflective cubemaps!  
@script ExecuteInEditMode

var cubemapSize = 128;  
var oneFacePerFrame = false;  
private var cam : Camera;  
private var rtex : RenderTexture;

function Start ()  
{  
// render all six faces at startup  
UpdateCubemap( 63 );  
}

function LateUpdate ()  
{  
if (oneFacePerFrame)  
{  
var faceToRender = Time.frameCount % 6;  
var faceMask = 1 << faceToRender;  
UpdateCubemap (faceMask);  
}  
else  
{  
UpdateCubemap (63); // all six faces  
}  
}

function UpdateCubemap (faceMask : int)  
{  
if (!cam)  
{  
var go = new GameObject ("CubemapCamera", Camera);  
go.hideFlags = HideFlags.HideAndDontSave;  
go.transform.position = transform.position;  
go.transform.rotation = Quaternion.identity;  
cam = go.camera;  
cam.farClipPlane = 100; // don't render very far into cubemap  
cam.enabled = false;  
}

if (!rtex)  
{  
rtex = new RenderTexture (cubemapSize, cubemapSize, 16);  
rtex.isPowerOfTwo = true;  
rtex.isCubemap = true;  
rtex.hideFlags = HideFlags.HideAndDontSave;  
renderer.sharedMaterial.SetTexture ("\_Cube", rtex);  
}

cam.transform.position = transform.position;  
cam.RenderToCubemap (rtex, faceMask);  
}

function OnDisable ()  
{  
DestroyImmediate (cam);  
DestroyImmediate (rtex);  
}