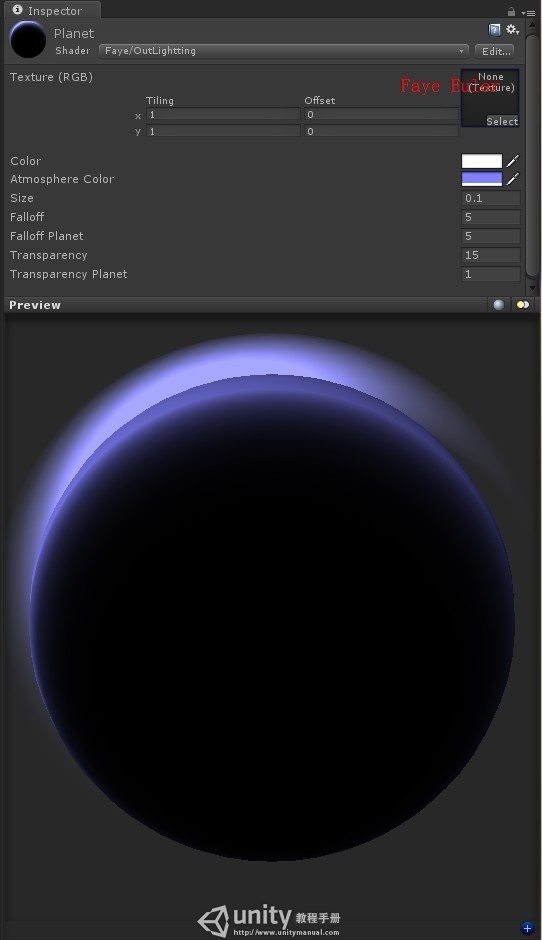
**外发光材质Shader**

Posted on 2013年06月28日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 166 次

游戏中我们经常会遇到需要制作外发光效果的地方，比如武器，比如坐骑要带一个牛掰的光晕。今天我找到了个好效果：

[](http://www.unitymanual.com/wp-content/uploads/2013/06/6ad33d35gd387b66ab80d690.jpeg)

外发光材质Shader

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 001 | Shader "Faye/OutLightting" |
| 002 | { |
| 003 | Properties |
| 004 | { |
| 005 | \_MainTex("Texture (RGB)", 2D) = "black" {} |
| 006 | \_Color("Color", Color) = (0, 0, 0, 1) |
| 007 | \_AtmoColor("Atmosphere Color", Color) = (0.5, 0.5, 1.0, 1) |
| 008 | \_Size("Size", **Float**) = 0.1 |
| 009 | \_Falloff("Falloff", **Float**) = 5 |
| 010 | \_FalloffPlanet("Falloff Planet", **Float**) = 5 |
| 011 | \_Transparency("Transparency", **Float**) = 15 |
| 012 | \_TransparencyPlanet("Transparency Planet", **Float**) = 1 |
| 013 | } |
| 014 |  |
| 015 | SubShader |
| 016 | { |
| 017 | Pass |
| 018 | { |
| 019 | Name "PlanetBase" |
| 020 | Tags {"LightMode" = "Always"} |
| 021 | Cull Back |
| 022 |  |
| 023 | CGPROGRAM |
| 024 | #pragma vertex vert |
| 025 | #pragma fragment frag |
| 026 |  |
| 027 | #pragma fragmentoption ARB\_fog\_exp2 |
| 028 | #pragma fragmentoption ARB\_precision\_hint\_fastest |
| 029 |  |
| 030 | #include "UnityCG.cginc" |
| 031 |  |
| 032 | uniform sampler2D \_MainTex; |
| 033 | uniform float4 \_MainTex\_ST; |
| 034 | uniform float4 \_Color; |
| 035 | uniform float4 \_AtmoColor; |
| 036 | uniform **float** \_FalloffPlanet; |
| 037 | uniform **float** \_TransparencyPlanet; |
| 038 |  |
| 039 | **struct** v2f |
| 040 | { |
| 041 | float4 pos : SV\_POSITION; |
| 042 | float3 normal : TEXCOORD0; |
| 043 | float3 worldvertpos : TEXCOORD1; |
| 044 | float2 texcoord : TEXCOORD2; |
| 045 | }; |
| 046 |  |
| 047 | v2f vert(appdata\_base v) |
| 048 | { |
| 049 | v2f o; |
| 050 |  |
| 051 | o.pos = mul (UNITY\_MATRIX\_MVP, v.vertex); |
| 052 | o.normal = v.normal; |
| 053 | o.worldvertpos = mul(\_Object2World, v.vertex).xyz; |
| 054 | o.texcoord = TRANSFORM\_TEX(v.texcoord, \_MainTex); |
| 055 |  |
| 056 | **return** o; |
| 057 | } |
| 058 |  |
| 059 | float4 frag(v2f i) : COLOR |
| 060 | { |
| 061 | i.normal = normalize(i.normal); |
| 062 | float3 viewdir = normalize(\_WorldSpaceCameraPos-i.worldvertpos); |
| 063 |  |
| 064 | float4 atmo = \_AtmoColor; |
| 065 | atmo.a = pow(1.0-saturate(dot(viewdir, i.normal)), \_FalloffPlanet); |
| 066 | atmo.a \*= \_TransparencyPlanet\*\_Color; |
| 067 | *//Unity3D教程手册：http://www.unitymanual.com* |
| 068 | float4 color = tex2D(\_MainTex, i.texcoord)\*\_Color; |
| 069 | color.rgb = lerp(color.rgb, atmo.rgb, atmo.a); |
| 070 |  |
| 071 | **return** color\*dot(normalize(i.worldvertpos-\_WorldSpaceLightPos0), i.normal); |
| 072 | } |
| 073 | ENDCG |
| 074 | } |
| 075 |  |
| 076 | Pass |
| 077 | { |
| 078 | Name "AtmosphereBase" |
| 079 | Tags {"LightMode" = "Always"} |
| 080 | Cull Front |
| 081 | Blend SrcAlpha One |
| 082 |  |
| 083 | CGPROGRAM |
| 084 | #pragma vertex vert |
| 085 | #pragma fragment frag |
| 086 |  |
| 087 | #pragma fragmentoption ARB\_fog\_exp2 |
| 088 | #pragma fragmentoption ARB\_precision\_hint\_fastest |
| 089 |  |
| 090 | #include "UnityCG.cginc" |
| 091 |  |
| 092 | uniform float4 \_Color; |
| 093 | uniform float4 \_AtmoColor; |
| 094 | uniform **float** \_Size; |
| 095 | uniform **float** \_Falloff; |
| 096 | uniform **float** \_Transparency; |
| 097 |  |
| 098 | **struct** v2f |
| 099 | { |
| 100 | float4 pos : SV\_POSITION; |
| 101 | float3 normal : TEXCOORD0; |
| 102 | float3 worldvertpos : TEXCOORD1; |
| 103 | }; |
| 104 | *//Unity3D教程手册：http://www.unitymanual.com* |
| 105 | v2f vert(appdata\_base v) |
| 106 | { |
| 107 | v2f o; |
| 108 |  |
| 109 | v.vertex.xyz += v.normal\*\_Size; |
| 110 | o.pos = mul (UNITY\_MATRIX\_MVP, v.vertex); |
| 111 | o.normal = v.normal; |
| 112 | o.worldvertpos = mul(\_Object2World, v.vertex); |
| 113 |  |
| 114 | **return** o; |
| 115 | } |
| 116 |  |
| 117 | float4 frag(v2f i) : COLOR |
| 118 | { |
| 119 | i.normal = normalize(i.normal); |
| 120 | float3 viewdir = normalize(i.worldvertpos-\_WorldSpaceCameraPos); |
| 121 |  |
| 122 | float4 color = \_AtmoColor; |
| 123 | color.a = pow(saturate(dot(viewdir, i.normal)), \_Falloff); |
| 124 | color.a \*= \_Transparency\*\_Color\*dot(normalize(i.worldvertpos-\_WorldSpaceLightPos0), i.normal); |
| 125 | **return** color; |
| 126 | } |
| 127 | ENDCG |
| 128 | } |
| 129 | } |
| 130 |  |
| 131 | FallBack "Diffuse" |
| 132 | } |