**实现镜头景深效果**

Posted on 2013年06月25日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 12 次

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

|  |  |
| --- | --- |
| 001 |  |
| 002 | Shader "Hidden/Render DOF Factor" { |
| 003 | Properties { |
| 004 | \_MainTex ("Base", 2D) = "white" {} |
| 005 | \_Cutoff ("Cutoff", **float**) = 0.5 |
| 006 | } |
| 007 | *// Helper code used in all of the below subshaders* |
| 008 | CGINCLUDE |
| 009 | **struct** v2f { |
| 010 | float4 pos : POSITION; |
| 011 | **float** depth : TEXCOORD0; |
| 012 | }; |
| 013 | **struct** v2f\_uv { |
| 014 | float4 pos : POSITION; |
| 015 | float2 uv : TEXCOORD0; |
| 016 | **float** depth : TEXCOORD1; |
| 017 | }; |
| 018 | uniform float4 \_FocalParams; |
| 019 | half DOFFactor( **float** z ) { |
| 020 | **float** focalDist = \_FocalParams.x; |
| 021 | **float** invRange = \_FocalParams.w; |
| 022 | **float** fromFocal = z - focalDist; |
| 023 | **if**( fromFocal < 0.0 ) |
| 024 | fromFocal \*= 4.0; |
| 025 | **return** saturate( abs( fromFocal ) \* invRange ); |
| 026 | } |
| 027 | uniform sampler2D \_MainTex; |
| 028 | uniform **float** \_Cutoff; |
| 029 | half4 frag(v2f i) : COLOR { |
| 030 | **return** DOFFactor(i.depth); |
| 031 | } |
| 032 | half4 frag\_uv(v2f\_uv i) : COLOR { |
| 033 | half4 texcol = tex2D( \_MainTex, i.uv ); |
| 034 | clip( texcol.a - \_Cutoff ); |
| 035 | **return** DOFFactor(i.depth); |
| 036 | } |
| 037 | ENDCG |
| 038 | Category { |
| 039 | Fog { Mode Off } |
| 040 |  |
| 041 | *// regular opaque objects* |
| 042 | SubShader { |
| 043 | Tags { "RenderType"="Opaque" } |
| 044 | Pass { |
| 045 |  |
| 046 | CGPROGRAM |
| 047 | #pragma vertex vert |
| 048 | #pragma fragment frag |
| 049 | #include "UnityCG.cginc" |
| 050 | v2f vert( appdata\_base v ) { |
| 051 | v2f o; |
| 052 | o.pos = mul(glstate.matrix.mvp, v.vertex); |
| 053 | COMPUTE\_EYEDEPTH(o.depth); |
| 054 | **return** o; |
| 055 | } |
| 056 | ENDCG |
| 057 | } |
| 058 | } |
| 059 |  |
| 060 | *// transparent cutout objects* |
| 061 | SubShader { |
| 062 | Tags { "RenderType"="TransparentCutout" } |
| 063 | Pass { |
| 064 | Cull Off |
| 065 |  |
| 066 | CGPROGRAM |
| 067 | #pragma vertex vert |
| 068 | #pragma fragment frag\_uv |
| 069 | #include "UnityCG.cginc" |
| 070 | v2f\_uv vert( appdata\_base v ) { |
| 071 | v2f\_uv o; |
| 072 | o.pos = mul(glstate.matrix.mvp, v.vertex); |
| 073 | o.uv = v.texcoord; |
| 074 | COMPUTE\_EYEDEPTH(o.depth); |
| 075 | **return** o; |
| 076 | } |
| 077 | ENDCG |
| 078 | } |
| 079 | } |
| 080 |  |
| 081 | *// terrain tree bark* |
| 082 | SubShader { |
| 083 | Tags { "RenderType"="TreeOpaque" } |
| 084 | Pass { |
| 085 |  |
| 086 | CGPROGRAM |
| 087 | #pragma vertex vert |
| 088 | #pragma fragment frag |
| 089 | #include "UnityCG.cginc" |
| 090 | #include "TerrainEngine.cginc" |
| 091 | **struct** appdata { |
| 092 | float4 vertex : POSITION; |
| 093 | float4 color : COLOR; |
| 094 | }; |
| 095 | v2f vert( appdata v ) { |
| 096 | v2f o; |
| 097 | TerrainAnimateTree(v.vertex, v.color.w); |
| 098 | o.pos = mul( glstate.matrix.mvp, v.vertex ); |
| 099 | COMPUTE\_EYEDEPTH(o.depth); |
| 100 | **return** o; |
| 101 | } |
| 102 | ENDCG |
| 103 | } |
| 104 | } |
| 105 |  |
| 106 | *// terrain tree leaves* |
| 107 | SubShader { |
| 108 | Tags { "RenderType"="TreeTransparentCutout" } |
| 109 | Pass { |
| 110 | Cull Off |
| 111 |  |
| 112 | CGPROGRAM |
| 113 | #pragma vertex vert |
| 114 | #pragma fragment frag\_uv |
| 115 | #include "UnityCG.cginc" |
| 116 | #include "TerrainEngine.cginc" |
| 117 | **struct** appdata { |
| 118 | float4 vertex : POSITION; |
| 119 | float4 color : COLOR; |
| 120 | float4 texcoord : TEXCOORD0; |
| 121 | }; *//Unity3D教程手册：http://www.unitymanual.com* |
| 122 | v2f\_uv vert( appdata v ) { |
| 123 | v2f\_uv o; |
| 124 | TerrainAnimateTree(v.vertex, v.color.w); |
| 125 | o.pos = mul( glstate.matrix.mvp, v.vertex ); |
| 126 | o.uv = v.texcoord; |
| 127 | COMPUTE\_EYEDEPTH(o.depth); |
| 128 | **return** o; |
| 129 | } |
| 130 | ENDCG |
| 131 | } |
| 132 | } |
| 133 |  |
| 134 | *// terrain tree billboards* |
| 135 | SubShader { |
| 136 | Tags { "RenderType"="TreeBillboard" } |
| 137 | Pass { |
| 138 | Cull Off |
| 139 |  |
| 140 | CGPROGRAM |
| 141 | #pragma vertex vert |
| 142 | #pragma fragment frag\_tree |
| 143 | #include "UnityCG.cginc" |
| 144 | #include "TerrainEngine.cginc" |
| 145 | **struct** appdata { |
| 146 | float4 vertex : POSITION; |
| 147 | float4 color : COLOR; |
| 148 | float4 texcoord : TEXCOORD0; |
| 149 | }; |
| 150 | v2f\_uv vert( appdata\_tree\_billboard v ) { |
| 151 | v2f\_uv o; |
| 152 | TerrainBillboardTree(v.vertex, v.texcoord1.xy); |
| 153 | o.pos = mul( glstate.matrix.mvp, v.vertex ); |
| 154 | o.uv = v.texcoord; |
| 155 | COMPUTE\_EYEDEPTH(o.depth); |
| 156 | **return** o; |
| 157 | } |
| 158 | half4 frag\_tree(v2f\_uv i) : COLOR { |
| 159 | half4 texcol = tex2D( \_MainTex, i.uv ); |
| 160 | clip( texcol.a - 0.5 ); |
| 161 | **return** DOFFactor(i.depth); |
| 162 | } *//Unity3D教程手册：http://www.unitymanual.com* |
| 163 | ENDCG |
| 164 | } |
| 165 | } |
| 166 |  |
| 167 | *// terrain grass billboards* |
| 168 | SubShader { |
| 169 | Tags { "RenderType"="GrassBillboard" } |
| 170 | Pass { |
| 171 | Cull Off |
| 172 |  |
| 173 | CGPROGRAM |
| 174 | #pragma vertex vert |
| 175 | #pragma fragment frag\_uv |
| 176 | #pragma multi\_compile NO\_INTEL\_GMA\_X3100\_WORKAROUND INTEL\_GMA\_X3100\_WORKAROUND |
| 177 | #include "UnityCG.cginc" |
| 178 | #include "TerrainEngine.cginc" |
| 179 | v2f\_uv vert (appdata\_grass v) { |
| 180 | v2f\_uv o; |
| 181 | TerrainBillboardGrass(v.vertex, v.texcoord1.xy); |
| 182 | **float** waveAmount = v.texcoord1.y; |
| 183 | float4 dummyColor = 0; |
| 184 | TerrainWaveGrass (v.vertex, waveAmount, dummyColor, dummyColor); |
| 185 | o.pos = mul (glstate.matrix.mvp, v.vertex); |
| 186 | o.uv = v.texcoord; |
| 187 | COMPUTE\_EYEDEPTH(o.depth); |
| 188 | **return** o; |
| 189 | } |
| 190 | ENDCG |
| 191 | } |
| 192 | } |
| 193 |  |
| 194 | *// terrain grass non-billboards* |
| 195 | SubShader { |
| 196 | Tags { "RenderType"="Grass" } |
| 197 | Pass { |
| 198 | Cull Off |
| 199 |  |
| 200 | CGPROGRAM |
| 201 | #pragma vertex vert |
| 202 | #pragma fragment frag\_uv |
| 203 | #pragma multi\_compile NO\_INTEL\_GMA\_X3100\_WORKAROUND INTEL\_GMA\_X3100\_WORKAROUND |
| 204 | #include "UnityCG.cginc" |
| 205 | #include "TerrainEngine.cginc" |
| 206 | v2f\_uv vert (appdata\_grass v) { |
| 207 | v2f\_uv o; |
| 208 | **float** waveAmount = v.color.a \* \_WaveAndDistance.z; |
| 209 | float4 dummyColor = 0; |
| 210 | TerrainWaveGrass (v.vertex, waveAmount, dummyColor, dummyColor); |
| 211 | o.pos = mul (glstate.matrix.mvp, v.vertex); |
| 212 | o.uv = v.texcoord; |
| 213 | COMPUTE\_EYEDEPTH(o.depth); |
| 214 | **return** o; |
| 215 | } |
| 216 | ENDCG |
| 217 | } |
| 218 | } |
| 219 |  |
| 220 | } |
| 221 | } |
| 222 |  |
| 223 |  |