**JS脚本控制逐渐淡出的欢迎界面**

Posted on 2013年06月14日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 11 次

使用JS脚本控制开始界面的淡出并且最终显现出游戏界面。在游戏界面的设置中，开始界面的制作十分必要，是欢迎玩家并且简要介绍游戏背景的必备过程。

首先，先创建一个脚本文件，声明GUI必备的类：

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

|  |  |
| --- | --- |
| 001 | import System.Collections.Generic; |
| 002 |  |
| 003 |  |
| 004 | *// TextureGUI Class: create a basic class for creating and placing GUI elements* |
| 005 | *// texture = the texture to display* |
| 006 | *// offset = pixel offset from top left corner, can be modified for easy positioning* |
| 007 |  |
| 008 | **class** TextureGUI { |
| 009 | **var** texture:Texture; *//useful: texture.width, texture.height* |
| 010 | **var** offset:Vector2; *// .x and .y* |
| 011 | **private** **var** originalOffset:Vector2; *//store the original to correctly reset anchor point* |
| 012 | **enum** Point { TopLeft, TopRight, BottomLeft, BottomRight, Center} *//what part of texture to position around?* |
| 013 |  |
| 014 | **var** anchorPoint = Point.TopLeft; *// Unity default is from top left corner of texture* |
| 015 |  |
| 016 | function setAnchor() { *// meant to be run ONCE at Start.* |
| 017 | originalOffset = offset; |
| 018 | **if** (texture) { *// check for null texture* |
| 019 | **switch**(anchorPoint) { *//depending on where we want to center our offsets* |
| 020 | **case** anchorPoint.TopLeft: *// Unity default, do nothing* |
| 021 | **break**; |
| 022 | **case** anchorPoint.TopRight: *// Take the offset and go to the top right corner* |
| 023 | offset.x = originalOffset.x &minus; texture.width; |
| 024 | **break**; |
| 025 |  |
| 026 | **case** anchorPoint.BottomLeft: *// bottom left corner of texture* |
| 027 | offset.y = originalOffset.y &minus; texture.height; |
| 028 | **break**; |
| 029 |  |
| 030 | **case** anchorPoint.BottomRight: *//bottom right corner of texture* |
| 031 | offset.x = originalOffset.x &minus; texture.width; |
| 032 | offset.y = originalOffset.y &minus; texture.height; |
| 033 | **break**; |
| 034 |  |
| 035 | **case** anchorPoint.Center: *//and the center of the texture (useful for screen center textures)* |
| 036 | offset.x = originalOffset.x &minus; texture.width/2; |
| 037 | offset.y = originalOffset.y &minus; texture.height/2; |
| 038 | **break**; |
| 039 | } |
| 040 | } |
| 041 | } |
| 042 | } |
| 043 |  |
| 044 | *//Timer Class:* |
| 045 |  |
| 046 |  |
| 047 | **class** TimerGUI extends TextureGUI { *// Extend functionality from TextureGUI for a depreciating timer graphic* |
| 048 | **var** textureLEnd:Texture; *// left side of full texture (non stretching part)* |
| 049 | **var** offsetLEnd:Vector2; *// left side of full texture (non stretching part) start position* |
| 050 | **var** textureCenter:Texture; *// center of timer (will be stretched across width)* |
| 051 | **var** offsetCenter:Vector2; |
| 052 | **var** textureREnd:Texture; |
| 053 | **var** offsetREnd:Vector2; |
| 054 | **var** timerPerct:**float** = 1; *// percentage (0 to 1) this stretches the center* |
| 055 | **var** desiredWidth:**float** = 403; *// max width of the timer in pixels* |
| 056 |  |
| 057 | function setTime(newTime:**float**) { |
| 058 | timerPerct = newTime; *// sets the percent based on value<code lang="csharp">Unity3D教程手册* |
| 059 |  |
| 060 |  |
| 061 | } |
| 062 |  |
| 063 | } |
| 064 |  |
| 065 |  |
| 066 |  |
| 067 | *// SwitchGUI Class: Extends the TextureGUI to be able to load in multiple textures and switch between them* |
| 068 |  |
| 069 | **class** SwitchGUI extends TextureGUI { |
| 070 |  |
| 071 | **var** switchableTextures = new List.<Texture>(); |
| 072 |  |
| 073 | **var** currentTexture:**int** = 0; |
| 074 |  |
| 075 | function Start() { |
| 076 |  |
| 077 | **if** (switchableTextures.Count > 0) { |
| 078 |  |
| 079 | texture = switchableTextures[currentTexture]; |
| 080 |  |
| 081 | } |
| 082 |  |
| 083 | } |
| 084 |  |
| 085 | function changeTexture(switchTo:**int**) { |
| 086 |  |
| 087 | **if** (switchTo < switchableTextures.Count && switchTo >= 0) { |
| 088 |  |
| 089 | texture = switchableTextures[switchTo]; |
| 090 |  |
| 091 | currentTexture = switchTo; |
| 092 |  |
| 093 | } **else** { |
| 094 |  |
| 095 | *//Debug.Log( this + ": tried to call invalid part of switchTextures array!");* |
| 096 |  |
| 097 | } |
| 098 |  |
| 099 | } |
| 100 |  |
| 101 |  |
| 102 |  |
| 103 | function up() { |
| 104 |  |
| 105 | **if** ((currentTexture+1) < switchableTextures.Count) { |
| 106 |  |
| 107 | ++currentTexture; |
| 108 |  |
| 109 | texture = switchableTextures[currentTexture]; |
| 110 |  |
| 111 | } **else** { |
| 112 |  |
| 113 | *//Debug.Log( this + ": at the top!");* |
| 114 |  |
| 115 | } |
| 116 |  |
| 117 | } |
| 118 |  |
| 119 |  |
| 120 |  |
| 121 | function nextTexture() { |
| 122 |  |
| 123 | **if** ((currentTexture+1) < switchableTextures.Count) { *// if we are at the end of the array* |
| 124 |  |
| 125 | ++currentTexture; |
| 126 |  |
| 127 | texture = switchableTextures[currentTexture]; |
| 128 |  |
| 129 | } **else** {*// loop to the beginning* |
| 130 |  |
| 131 | currentTexture = 0; |
| 132 |  |
| 133 | texture = switchableTextures[currentTexture]; |
| 134 |  |
| 135 | } |
| 136 |  |
| 137 | } |
| 138 |  |
| 139 |  |
| 140 |  |
| 141 | function down() { |
| 142 |  |
| 143 | **if** ((currentTexture&minus;1) >= 0) { |
| 144 |  |
| 145 | &minus;&minus;currentTexture; |
| 146 |  |
| 147 | texture = switchableTextures[currentTexture]; |
| 148 |  |
| 149 | } **else** { |
| 150 |  |
| 151 | *//Debug.Log( this + ": at the bottom!");* |
| 152 |  |
| 153 | } |
| 154 |  |
| 155 | } |
| 156 |  |
| 157 |  |
| 158 |  |
| 159 | } |
| 160 |  |
| 161 |  |
| 162 |  |
| 163 | *// Location class:* |
| 164 |  |
| 165 |  |
| 166 |  |
| 167 |  |
| 168 |  |
| 169 | **class** Location { |
| 170 |  |
| 171 | **enum** Point { TopLeft, TopRight, BottomLeft, BottomRight, Center} |
| 172 |  |
| 173 |  |
| 174 |  |
| 175 | **var** pointLocation = Point.TopLeft; |
| 176 |  |
| 177 | **var** offset:Vector2; |
| 178 |  |
| 179 |  |
| 180 |  |
| 181 |  |
| 182 |  |
| 183 | function updateLocation() { |
| 184 |  |
| 185 | **switch**(pointLocation) { |
| 186 |  |
| 187 | **case** pointLocation.TopLeft: |
| 188 |  |
| 189 | offset = Vector2(0,0); |
| 190 |  |
| 191 | **break**; |
| 192 |  |
| 193 | **case** pointLocation.TopRight: |
| 194 |  |
| 195 | offset = Vector2(Screen.width,0); |
| 196 |  |
| 197 | **break**; |
| 198 |  |
| 199 |  |
| 200 |  |
| 201 | **case** pointLocation.BottomLeft: |
| 202 |  |
| 203 | offset = Vector2(0,Screen.height); |
| 204 |  |
| 205 | **break**; |
| 206 |  |
| 207 |  |
| 208 |  |
| 209 | **case** pointLocation.BottomRight: |
| 210 |  |
| 211 | offset = Vector2(Screen.width,Screen.height); |
| 212 |  |
| 213 | **break**; |
| 214 |  |
| 215 |  |
| 216 |  |
| 217 | **case** pointLocation.Center: |
| 218 |  |
| 219 | offset = Vector2(Screen.width/2,Screen.height/2); |
| 220 |  |
| 221 | **break**; |
| 222 |  |
| 223 | } |
| 224 |  |
| 225 | } |
| 226 |  |
| 227 | } |
| 228 |  |
| 229 |  |
| 230 |  |
| 231 | **class** TextureAnchor { |
| 232 |  |
| 233 | **enum** Point { TopLeft, TopRight, BottomLeft, BottomRight, Center} |
| 234 |  |
| 235 |  |
| 236 |  |
| 237 | **var** anchorPoint = Point.TopLeft; |
| 238 |  |
| 239 | **var** offset:Vector2; |
| 240 |  |
| 241 |  |
| 242 |  |
| 243 | function update() { |
| 244 |  |
| 245 | **switch**(anchorPoint) { |
| 246 |  |
| 247 | **case** anchorPoint.TopLeft: |
| 248 |  |
| 249 | offset = Vector2(0,0); |
| 250 |  |
| 251 | **break**; |
| 252 |  |
| 253 | **case** anchorPoint.TopRight: |
| 254 |  |
| 255 | offset = Vector2(Screen.width,0); |
| 256 |  |
| 257 | **break**; |
| 258 |  |
| 259 |  |
| 260 |  |
| 261 | **case** anchorPoint.BottomLeft: |
| 262 |  |
| 263 | offset = Vector2(0,Screen.height); |
| 264 |  |
| 265 | **break**; |
| 266 |  |
| 267 |  |
| 268 |  |
| 269 | **case** anchorPoint.BottomRight: |
| 270 |  |
| 271 | offset = Vector2(Screen.width,Screen.height); |
| 272 |  |
| 273 | **break**; |
| 274 |  |
| 275 |  |
| 276 |  |
| 277 | **case** anchorPoint.Center: |
| 278 |  |
| 279 | offset = Vector2(Screen.width/2,Screen.height/2); |
| 280 |  |
| 281 | **break**; |
| 282 |  |
| 283 | } |
| 284 |  |
| 285 | } |
| 286 |  |
| 287 | } |
| 288 |  |
| 289 |  |

然后创建一个用来实现欢迎界面全屏的脚本文件DisplayTextureFullScreen，脚本的源码如下：

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

|  |  |
| --- | --- |
| 01 | **var** graphic = TextureGUI(); *//(28,23);* |
| 02 |  |
| 03 | **var** GUIColor:Color; |
| 04 |  |
| 05 | function OnGUI() { |
| 06 |  |
| 07 | GUI.color = GUIColor; |
| 08 |  |
| 09 | **if** (graphic.texture) { |
| 10 |  |
| 11 | GUI.DrawTexture(Rect(graphic.offset.x,graphic.offset.y, |
| 12 |  |
| 13 | Screen.width,Screen.height), |
| 14 |  |
| 15 | graphic.texture,ScaleMode.StretchToFill,**true**); |
| 16 |  |
| 17 | } |
| 18 |  |
| 19 | } |
| 20 |  |
| 21 | function AlphaUp(change:**float**) { |
| 22 |  |
| 23 | GUIColor.a += change; |
| 24 |  |
| 25 | } |
| 26 |  |
| 27 | function setStartColor(color:Color) { |
| 28 |  |
| 29 | GUIColor = color; |
| 30 |  |
| 31 | } |
| 32 |  |
| 33 | function setDelay(delay:**float**) { |
| 34 |  |
| 35 | **if** (GUIColor.a > .5) { |
| 36 |  |
| 37 | GUIColor.a += delay; |
| 38 |  |
| 39 | } **else** { |
| 40 |  |
| 41 | GUIColor.a ?= delay; |
| 42 |  |
| 43 | } |
| 44 |  |
| 45 | }*// Unity3D教程手册* |
| 46 |  |
| 47 | function AlphaDown(change:**float**) { |
| 48 |  |
| 49 | GUIColor.a ?= change; |
| 50 |  |
| 51 | } |

可以看出其他的几个方法都是调节颜色和alpha值的函数，关键的函数是DrawTexture方法绘制纹理贴图。接下来再创建一个叫做GameState的JS脚本，源码如下：

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

|  |  |
| --- | --- |
| 01 | **var** renderOverlay : DisplayTextureFullScreen; |
| 02 |  |
| 03 |  |
| 04 | function Start() { |
| 05 |  |
| 06 | renderOverlay = GetComponent(DisplayTextureFullScreen); |
| 07 | renderOverlay.setStartColor(Color.white); |
| 08 | renderOverlay.setDelay(2.0); |
| 09 | } |
| 10 |  |
| 11 | function Update () { |
| 12 |  |
| 13 | **if** (renderOverlay.GUIColor.a > 0) { |
| 14 | renderOverlay.AlphaDown(Time.deltaTime); |
| 15 | } |
| 16 | } |

将这个和DisplayTextureFullScreen一起拖动到一个GameState的GameObject上面实现控制，并且在Inspector面板上拖动欢迎界面的贴图指向脚本中的Graphic。