**Unity3D手游开发：如何实现最高分的存储与显示（一）**

Posted on 2013年07月01日 by U3d / [Unity3D 基础教程](http://www.unitymanual.com/category/manual/unity3d-%e5%9f%ba%e7%a1%80%e6%95%99%e7%a8%8b)/被围观 92 次

（一）初始化显示的高分榜

游戏中的数据存储是很常见的问题，比如想在高分榜里显示如下的数据：下面就是HighScore的制作流程。

首先要搭建好显示的框架，包括背景图片和橙色的边框两个部分，创建一个空的GameObject用来存放这些GUI，并将其命名为HighscoresOBJ，上面拖拽三个脚本文件。在这三个脚本文件之前，先创建\_GUIClasses.js这个基本的脚本提供相关的类和方法：

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

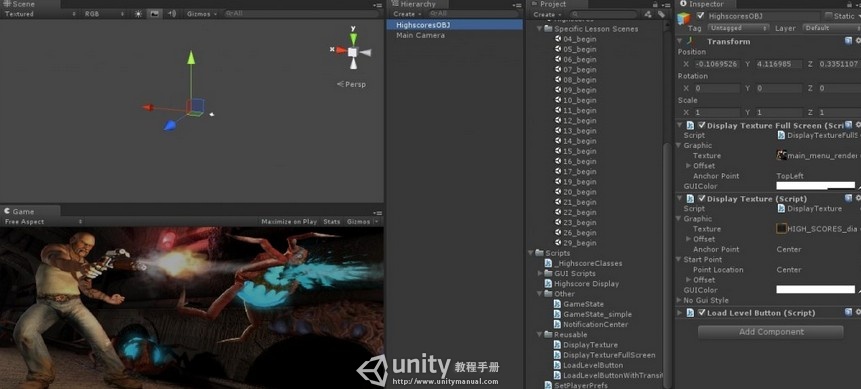
|  |  |
| --- | --- |
| 001 | import System.Collections.Generic; |
| 002 |  |
| 003 |  |
| 004 | *// TextureGUI Class: create a basic class for creating and placing GUI elements* |
| 005 | *// texture = the texture to display* |
| 006 | *// offset = pixel offset from top left corner, can be modified 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 - texture.width; |
| 024 | **break**; |
| 025 |  |
| 026 | **case** anchorPoint.BottomLeft: *// bottom left corner of texture* |
| 027 | offset.y = originalOffset.y - texture.height; |
| 028 | **break**; |
| 029 |  |
| 030 | **case** anchorPoint.BottomRight: *//bottom right corner of texture* |
| 031 | offset.x = originalOffset.x - texture.width; |
| 032 | offset.y = originalOffset.y - 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 - texture.width/2; |
| 037 | offset.y = originalOffset.y - 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* |
| 059 | } |
| 060 | } |
| 061 |  |
| 062 | *// SwitchGUI Class: Extends the TextureGUI to be able to load in multiple textures and switch between them* |
| 063 | **class** SwitchGUI extends TextureGUI { |
| 064 | **var** switchableTextures = new List.<Texture>(); |
| 065 | **var** currentTexture:**int** = 0; |
| 066 |  |
| 067 | function changeTexture(switchTo:**int**) { |
| 068 | **if** (switchTo < switchableTextures.Count && switchTo >= 0) { |
| 069 | texture = switchableTextures[switchTo]; |
| 070 | currentTexture = switchTo; |
| 071 | } **else** { |
| 072 | *//Debug.Log( this + ": tried to call invalid part of switchTextures array!");* |
| 073 | } *//Unity3D教程手册：www.unitymanual.com* |
| 074 | } |
| 075 |  |
| 076 | function up() { |
| 077 | **if** ((currentTexture+1) < switchableTextures.Count) { |
| 078 | ++currentTexture; |
| 079 | texture = switchableTextures[currentTexture]; |
| 080 | } **else** { |
| 081 | *//Debug.Log( this + ": at the top!");* |
| 082 | } |
| 083 | } |
| 084 |  |
| 085 | function nextTexture() { |
| 086 | **if** ((currentTexture+1) < switchableTextures.Count) { *// if we are at the end of the array* |
| 087 | ++currentTexture; |
| 088 | texture = switchableTextures[currentTexture]; |
| 089 | } **else** {*// loop to the beginning* |
| 090 | currentTexture = 0; |
| 091 | texture = switchableTextures[currentTexture]; |
| 092 | } |
| 093 | } |
| 094 |  |
| 095 | function down() { |
| 096 | **if** ((currentTexture-1) >= 0) { |
| 097 | --currentTexture; |
| 098 | texture = switchableTextures[currentTexture]; |
| 099 | } **else** { |
| 100 | *//Debug.Log( this + ": at the bottom!");* |
| 101 | } *//Unity3D教程手册：www.unitymanual.com* |
| 102 | } |
| 103 |  |
| 104 | } |
| 105 |  |
| 106 | *// Location class:* |
| 107 |  |
| 108 | **class** Location { |
| 109 | **enum** Point { TopLeft, TopRight, BottomLeft, BottomRight, Center} |
| 110 |  |
| 111 | **var** pointLocation = Point.TopLeft; |
| 112 | **var** offset:Vector2; |
| 113 |  |
| 114 |  |
| 115 | function updateLocation() { |
| 116 | **switch**(pointLocation) { |
| 117 | **case** pointLocation.TopLeft: |
| 118 | offset = Vector2(0,0); |
| 119 | **break**; |
| 120 | **case** pointLocation.TopRight: |
| 121 | offset = Vector2(Screen.width,0); |
| 122 | **break**; |
| 123 |  |
| 124 | **case** pointLocation.BottomLeft: |
| 125 | offset = Vector2(0,Screen.height); |
| 126 | **break**; |
| 127 |  |
| 128 | **case** pointLocation.BottomRight: |
| 129 | offset = Vector2(Screen.width,Screen.height); |
| 130 | **break**; |
| 131 |  |
| 132 | **case** pointLocation.Center: |
| 133 | offset = Vector2(Screen.width/2,Screen.height/2); |
| 134 | **break**; |
| 135 | } |
| 136 | } |
| 137 | } |

第一个脚本DisplayTextureFullScreen.js用来实现图片的全屏显示：

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

|  |  |
| --- | --- |
| 01 | @script ExecuteInEditMode() |
| 02 | **var** graphic = TextureGUI(); *//(28,23);* |
| 03 | **var** GUIColor:Color; |
| 04 |  |
| 05 | function OnGUI() { |
| 06 | GUI.color = GUIColor; |
| 07 | **if** (graphic.texture) { |
| 08 | GUI.DrawTexture(Rect(graphic.offset.x,graphic.offset.y, |
| 09 | Screen.width,Screen.height), |
| 10 | graphic.texture,ScaleMode.StretchToFill,**true**); |
| 11 | } |
| 12 | } |
| 13 |  |
| 14 | function AlphaUp(change:**float**) { |
| 15 | GUIColor.a += change; |
| 16 | } |
| 17 |  |
| 18 | function setStartColor(color:Color) { |
| 19 | GUIColor = color; |
| 20 | } |
| 21 | *//Unity3D教程手册：www.unitymanual.com* |
| 22 |  |
| 23 | function setDelay(delay:**float**) { |
| 24 | **if** (GUIColor.a > .5) { |
| 25 | GUIColor.a += delay; |
| 26 | } **else** { |
| 27 | GUIColor.a -= delay; |
| 28 | } |
| 29 | } |
| 30 |  |
| 31 | function AlphaDown(change:**float**) { |
| 32 | GUIColor.a -= change; |
| 33 | } |

其中，@script ExecuteInEditMode()这句话的意思是在编辑模式里运行。

[](http://www.unitymanual.com/wp-content/uploads/2013/07/1.jpg)

Unity3D手游开发：如何实现最高分的存储与显示（一）

脚本默认在play点击是才运行，这样添加之后便可以在Game界面直接看到运行效果。

具体解释：Unity Script Reference传送门

第二个脚本DisplayTexture.js是用来显示高分榜的橙色外框的：

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

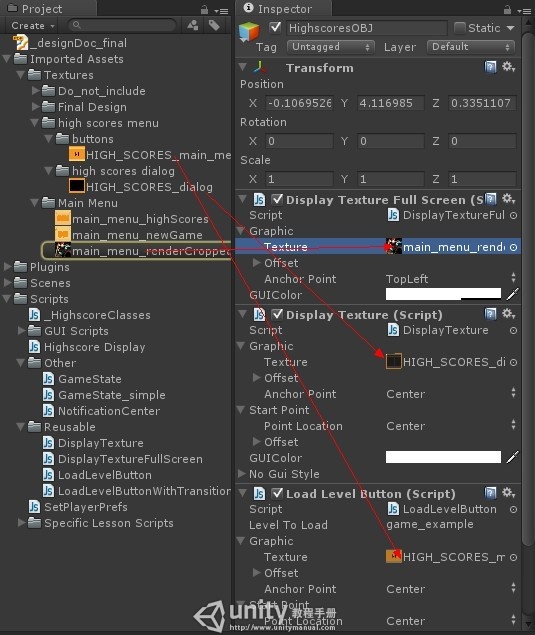
|  |  |
| --- | --- |
| 01 | **var** graphic = TextureGUI(); |
| 02 | **var** startPoint = Location(); |
| 03 | **var** GUIColor:Color = Color.white; |
| 04 | **var** noGuiStyle : GUIStyle; |
| 05 |  |
| 06 | function Start() { |
| 07 | graphic.setAnchor(); |
| 08 | } |
| 09 |  |
| 10 | function Update() { |
| 11 | **if** (graphic.texture){ |
| 12 | startPoint.updateLocation(); |
| 13 | } |
| 14 | } |
| 15 | *//Unity3D教程手册：www.unitymanual.com* |
| 16 | function OnGUI() { |
| 17 | **if** (graphic.texture){ |
| 18 | GUI.color = GUIColor; |
| 19 | GUI.Box(Rect(graphic.offset.x+startPoint.offset.x, |
| 20 | graphic.offset.y+startPoint.offset.y, |
| 21 | graphic.texture.width,graphic.texture.height), |
| 22 | graphic.texture,noGuiStyle); |
| 23 | } |
| 24 | } |

第三个脚本LoadLevelButton.js是用来显示那个MainMenu按钮的：

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

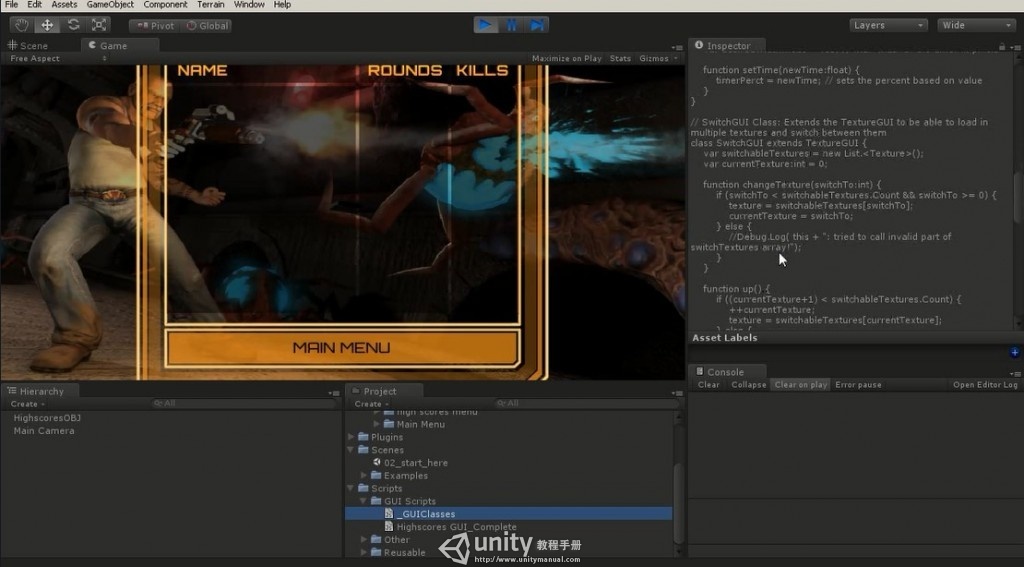
|  |  |
| --- | --- |
| 01 | **var** levelToLoad:**String**; |
| 02 | **var** graphic = TextureGUI(); *//(28,23);* |
| 03 | **var** startPoint = Location(); |
| 04 | **var** GUIColor:Color = Color.white; |
| 05 | **var** noGuiStyle : GUIStyle; |
| 06 |  |
| 07 | function Start() { |
| 08 | graphic.setAnchor(); |
| 09 | } |
| 10 |  |
| 11 | function Update() { |
| 12 | **if** (graphic.texture){ |
| 13 | startPoint.updateLocation(); |
| 14 | } |
| 15 | } |
| 16 |  |
| 17 | function OnGUI() { |
| 18 | GUI.color = GUIColor; |
| 19 | **if** (GUI.Button(Rect(graphic.offset.x+startPoint.offset.x,graphic.offset.y+startPoint.offset.y,graphic.texture.width,graphic.texture.height),graphic.texture,noGuiStyle)) { |
| 20 | PlayerPrefs.Save(); |
| 21 | Application.LoadLevel(levelToLoad); |
| 22 | } |
| 23 | } |

并且给他们挨个分配好相应的贴图：

[](http://www.unitymanual.com/wp-content/uploads/2013/07/2.jpg)

Unity3D手游开发：如何实现最高分的存储与显示（一）

这样再运行项目，就可以看到一个简单框架已经搭好了：

[](http://www.unitymanual.com/wp-content/uploads/2013/07/3.jpg)

Unity3D手游开发：如何实现最高分的存储与显示（一）

本文转自：汪海的实验室

**Unity3D手游开发：如何实现最高分的存储与显示（二）**

Posted on 2013年07月04日 by U3d / [Unity3D 基础教程](http://www.unitymanual.com/category/manual/unity3d-%e5%9f%ba%e7%a1%80%e6%95%99%e7%a8%8b)/被围观 101 次

（二）使用GUI.Box显示文字

创建脚本文件TestGUI.js：

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

|  |  |
| --- | --- |
| 1 | function OnGUI() { |
| 2 |  |
| 3 | GUI.Box(Rect(0,0,120,30),"This is a title"); |
| 4 | } |

将其拖拽到HighscoresOBJ上面，运行游戏便会发现窗口中多了一行标签：

[](http://www.unitymanual.com/wp-content/uploads/2013/07/16.jpg)

Unity3D手游开发：如何实现最高分的存储与显示（二）

这显然不是想要的结果。

Unity中的原点和很多其他软件一样默认在左上角，Rect参数列表中的第1、2表示坐标，3、4表示宽度和高度。接下来便是调整坐标使其在一个合适的位置。将脚本做如下修改：

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

|  |  |
| --- | --- |
| 1 | **var** boxStartLocation:Vector2; |
| 2 |  |
| 3 | *// draw a text string to the screen* |
| 4 |  |
| 5 | function OnGUI() { |
| 6 |  |
| 7 | GUI.Box(Rect(boxStartLocation.x,boxStartLocation.y,120,30),"This is a title"); |
| 8 |  |
| 9 | } |

这样我们就能在Unity中显式的调整坐标了：

[](http://www.unitymanual.com/wp-content/uploads/2013/07/24.jpg)

Unity3D手游开发：如何实现最高分的存储与显示（二）

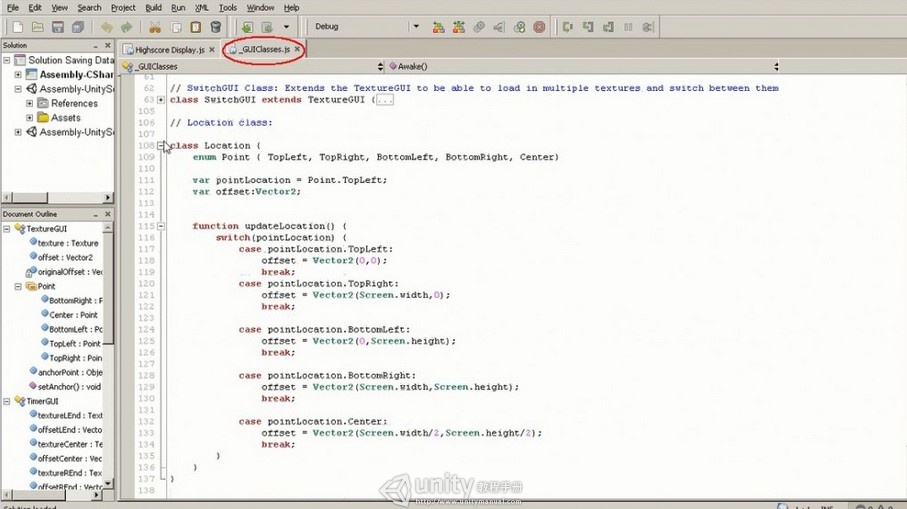
这是一种非常直观的方法，但是你会发现这个绝对坐标存在一个巨大的隐患：不知道设备的分辨率，所以会导致不同的设备显示的位置可能会不一样。

比如调整窗口的大小，这个BOX的坐标不会变化但是其他的位置都变了：

[](http://www.unitymanual.com/wp-content/uploads/2013/07/35.jpg)

Unity3D手游开发：如何实现最高分的存储与显示（二）

这个时候我们就需要自定义一个Location类来实现定位。

[](http://www.unitymanual.com/wp-content/uploads/2013/07/44.jpg)

Unity3D手游开发：如何实现最高分的存储与显示（二）

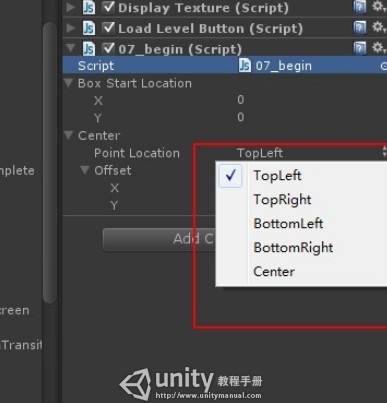
从代码中可以看到已经封装了各种可能遇到的需求，比如居中布局就是将offset设置为屏幕Screen的一半大小。

在原来的脚本上也稍作改动：

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

|  |  |
| --- | --- |
| 01 | **var** boxStartLocation:Vector2; |
| 02 | **var** center = Location(); |
| 03 |  |
| 04 | function Update() { |
| 05 | center.updateLocation(); |
| 06 | } |
| 07 |  |
| 08 | *// draw a text string to the screen* |
| 09 |  |
| 10 | function OnGUI() { |
| 11 |  |
| 12 | GUI.Box(Rect(center.offset.x + boxStartLocation.x, center.offset.y + boxStartLocation.y,120,30),"This is a title"); |
| 13 |  |
| 14 | } |

这是便会发现Inpector面板中多了几行设置：

[](http://www.unitymanual.com/wp-content/uploads/2013/07/52.jpg)

Unity3D手游开发：如何实现最高分的存储与显示（二）

这里的offset是自动运算的，无法手动修改它。但是我们可以调整BoxStartLocation中的X和Y来进行局部调整。

本文转自：汪海的实验室