**GUI的旋转**

Posted on 2013年04月22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 132 次

首先，这里所要说的GUI并不是指GUIText和GUITexture，而是指只能在OnGUI()中运行的GUI方法，这些GUI可用的功能相当的多，通常多用于表单功能或是一些讯息的呈现，但也有很多时候我们希望UI能都以图形方式来呈现，所以我们会像制作网页或2D游戏那样在画面上贴了很多的图片，使画面更活泼。

当我们的UI画面多以图形表现时，一定会希望多点动态，例如位移和缩放，这些只要调整Rect()的数值就能简单办到，但是当需要旋转时又该怎么做呢？

这个时候，我们需要用到GUIUtility.RotateAroundPivot()来为GUI做旋转的动作，它可以直接定义一个中心点，并以这个点为圆心来旋转，详细说明可查看官网说明；当使用GUIUtility.RotateAroundPivot()来旋转GUI时，将会发现这一句程式码之后的全部GUI都被旋转了，可是我们只想要单一个GUI元素被旋转，此时不就要再加一句GUIUtility.RotateAroundPivot()并给予相反的同值角度来将后面的GUI转回来，如此看来，每次需要旋转时，GUIUtility.RotateAroundPivot()必须呼叫两次，如果有多个GUI需要旋转，而且转速及角度都可能不同时，那么不是每个GUI转完后马上再呼叫一次调正回来，就是要每个需要旋转的GUI计算相对角度，真是好麻烦啊！

我们看一下官网的说明，会发现GUIUtility.RotateAroundPivot()是利用修改GUI.matrix来使全部的GUI元素旋转，也就是说，我们只要在GUIUtility.RotateAroundPivot()之前将GUI.matrix暂存到一个变数里，之后在不需要旋转的GUI元素之前将GUI.matrix恢复原来的值，那么后续的元素就不会跟着旋转了，如此就不用重复呼叫GUIUtility.RotateAroundPivot()，而且每个需要旋转的GUI元素只需要顾好本身此时应该转到第几度，而不受其他旋转的GUI元素影响，这样程式逻辑运算上就简单很多了。

以下为参考范例：

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

|  |  |
| --- | --- |
| 01 | **public** **var** image : Texture2D; |
| 02 | **public** **var** rotSpeed : **float** = 100.0; |
| 03 | **private** **var** rotAngle : **float**; |
| 04 |  |
| 05 | function Update(){ |
| 06 | rotAngle += Time.deltaTime \* rotSpeed; |
| 07 | } |
| 08 |  |
| 09 | function OnGUI(){ |
| 10 |  |
| 11 | **var** \_matrix : Matrix4x4 = GUI.matrix; |
| 12 |  |
| 13 | GUIUtility.RotateAroundPivot (rotAngle, Vector2(100,100)); |
| 14 | GUI.DrawTexture (Rect(50 , 50 , 100 , 100),image,ScaleMode.StretchToFill); |
| 15 | GUI.matrix = \_matrix; |
| 16 |  |
| 17 | GUI.Label (Rect (10, 10, 100, 20), "Hello World!"); |
| 18 | } |