**使用GUI制作时钟倒数效果**

Posted on 2013年04月15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 200 次

[**Unity3D教程**](http://www.unitymanual.com/category/manual)：如何使用GUI制作时钟倒数效果？在游戏中以圆形时钟指针那样缩减倒数的方式尤其常见，例如技能的冷却时间倒数；但是当我们要使用 GUI 制作这种效果时，会发现不容易做到，因为 GUI 不能指定区块将图片在该区块范围内的图素遮挡为透明的，此时，我们必须使用其他技巧来完成这个目的。

首先我们必须将底图切割为半圆的图，而倒数效果的图则是与底图相同大小的白色半透明图片。我们以图层的概念为这四张图重新排列成为五个图层，前后顺序排列它们的深度，然后以圆心座标为半透明图的旋转枢纽，使 GUI 依序做以下几个动作就可完成效果：

刚开始时先将最前面层的那个半圆图及左右半透明图隐藏起来（关闭 script 的执行或使该 GUI.color.a 为 0）。

当开始倒数时先显示左右半透明图。

右半透明图开始旋转，旋转过程如果从正面看可以看到部份被左底图挡住。

当右半透明图旋转 180 度后，将完全被左底图档住，此时显示最前面层的右半圆图。

左半透明图开始旋转，旋转过程如果从正面看可以看到部份被最前面层的右半圆图挡住。

当左半透明图旋转 180 度后，将完全被最前面层的右半圆图挡住，此时即完成倒数。

最后再将最前面层的那个半圆图及左右半透明图隐藏起来，并将左右半透明图的旋转角度恢复为 0 度。

撰写 GUI 专用的程式码，如下：

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

|  |  |
| --- | --- |
| 01 | **public** **class** MyDrawTexture : MonoBehaviour { |
| 02 |  |
| 03 | *//貼圖* |
| 04 | **public** Texture2D texture; |
| 05 | *//大小位置* |
| 06 | **public** Rect position; |
| 07 | *//深度* |
| 08 | **public** **int** depth; |
| 09 | *//角度* |
| 10 | **public** **float** angle; |
| 11 | *//旋轉樞紐* |
| 12 | **public** Vector2 pivot; |
| 13 |  |
| 14 | **void** OnGUI(){ |
| 15 |  |
| 16 | **if**(angle != 0) GUIUtility.RotateAroundPivot(angle, pivot); |
| 17 |  |
| 18 | GUI.depth = depth; |
| 19 | GUI.DrawTexture(position , texture , ScaleMode.StretchToFill); |
| 20 | } |
| 21 | } |

建立空物件（GameObject / Create Empty），将此程式拉给此物件附加上成为物件的 Component。

在 Hierarchy 中复制出另外四个物件并更改名称，并建一个空物件做为父物件，把这五个 GUI 物件都拉进去做为子物件。

设置各物件的大小位置（假设图的大小为 22\*44）、深度（在前面的数值较小）、旋转枢纽座标。

开启 Animation View （Window / Animation） 并在 Hierarchy 中选择这些 GUI 物件的父物件，在 Animation View 新增动画剪辑，新增完成後，在 Inspector 可以看到多出 Animation Component，在 Animation View 中会发现可对此物件及其子物件编辑动画曲线。

在 Hierarchy 点击名为 front 的物件，并在 Animation View 中将它的 Enable 栏位设定 0 秒的值为 0 及 0.5 秒（0:30） 的值为 1 的线段。

在 Hierarchy 点击名为 modalLeft 的物件，并在 Animation View 中将它的 Angle 栏位设定 0.5 秒（0:30）的值为 0 及 1 秒（1:00） 的值为 180 的线段。

在 Hierarchy 点击名为 modalRight 的物件，并在 Animation View 中将它的 Angle 栏位设定 0 秒的值为 0 及 0.5 秒（0:30） 的值为 180 的线段。

接下来执行游戏就能看到像时钟转动的效果。

到这边算是完成效果的实现了，但是离实际运用还有一些事情要考虑… 每个 GUI 物件都是个别设定位置，但这个效果至少要五个物件在一起，如果要移动它的位置不就要反覆做五次修改座标的动作，这样的话，好像也不是很方便；所以我们可以使用 GUI.BeginGroup（） 及 GUI.EndGroup（） 将它们包起来统一设置，另外就是旋转枢纽也有类似的情形，所以我们也可将它调整到一个地方设置就好，做法如下：

将上面程式码中的 public int depth; 、 public Vector2 pivot; 及 OnGUI（） 整个删除掉，因为用不到了。

撰写 GUI 的 Group 程式码如下：

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

|  |  |
| --- | --- |
| 01 | **public** **class** MyDrawTextureGroup : MonoBehaviour { |
| 02 |  |
| 03 | **public** Rect position; |
| 04 |  |
| 05 | **public** Vector2 pivot; |
| 06 |  |
| 07 | **public** MyDrawTexture[] guis; |
| 08 |  |
| 09 | **void** OnGUI(){ |
| 10 |  |
| 11 | GUI.BeginGroup(position); |
| 12 |  |
| 13 | **foreach**(MyDrawTexture \_gui **in** guis){ |
| 14 |  |
| 15 | **if**(\_gui == **null** || !\_gui.enabled) **continue**; |
| 16 |  |
| 17 | Matrix4x4 \_matrix = GUI.matrix; |
| 18 |  |
| 19 | **if**(\_gui.angle != 0) GUIUtility.RotateAroundPivot(\_gui.angle, pivot); |
| 20 |  |
| 21 | GUI.DrawTexture(\_gui.position , \_gui.texture , ScaleMode.StretchToFill); |
| 22 |  |
| 23 | GUI.matrix = \_matrix; |
| 24 | } |
| 25 |  |
| 26 | GUI.EndGroup(); |
| 27 | } |
| 28 | } |

把这支程式拉给这五个 GUI 物件的父物件，设置宽高及旋转枢纽座标後，依前後关系将五个 GUI 物件拉到 Inspector 中的 Guis 栏位的阵列元素中。

此时再次执行游戏，获得的效果是一样的，所不同的是，当改变位置时只需要在父物件调整即可。

最后，我们可能想要调整效果动作的时间，希望速度要快一点或慢一点，那么只需要点击一下父物件，在 Animation View 把有设置动画线段的物件展开来 （ front、 modalLeft、modalRight ）就能同时看到它们的线段及 key，此时只要在上方的横条拖曳滑鼠框选该时间范围内各线段的 key，然后拉动横条上的 key 就能任意调整多条线段的关键时间点，这样就能很方便的调整速度。