**使用Unity3D制作2D游戏的重点做法**

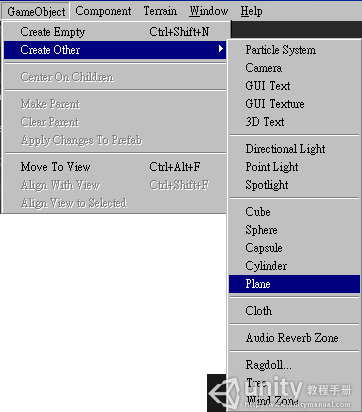
Posted on 2013年04月22日 by U3d / [Unity3D 基础教程](http://www.unitymanual.com/category/manual/unity3d-%e5%9f%ba%e7%a1%80%e6%95%99%e7%a8%8b)/被围观 338 次

官网上有提供一个[2D游戏的教学范例](http://unity3d.com/support/resources/tutorials/2d-gameplay-tutorial)，这个游戏只不过把镜头摆放在横向位置跟随角色移动，游戏内物件运动时固定一个轴不动，使他像横轴式的2D游戏来玩，本质上其实还是个3D游戏，所以如果没有3D建模的话不就没办法做2D游戏了？其实这些并没有定论，就看制作者如何运用Unity来完成自己的游戏。

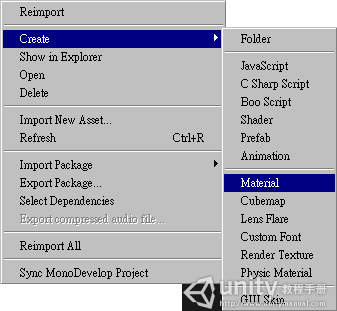
首先我们先来说为何用Unity来做2D游戏，除了3D与2D游戏各有其乐趣之外，最主要的还是Unity方便使用的渲染及物理引擎，如果是像官方范例那样使用3D建模来做，这些物件只需要做过一次，将来要转成3D游戏的玩法，也不需要重新建模，重复利用性高，对将来节约开发成本很有帮助，如果不使用3D建模的方式而改以平面贴图的物件来做2D游戏，对於各种物理现象如碰撞、重力、加速度等，也可以直接使用Unity的Component来完成，而且游戏空间本身就是个3D空间，所以要制作出拥有多层次场景的2D游戏也会变得较为方便，试想自己写程式来做2D游戏，光是碰撞、重力这种物理动作就够伤脑筋了，再加上光影效果等变化，势必要多加很多更细部的贴图，而不如Unity加个灯光或什麽的，由3D渲染及物理引擎自行去运作即可。

使用3D建模的方式制作2D游戏，直接看官网范例即可，其实和做3D游戏没啥差别，如果使用平面贴图物件来做的话，只需要注意到几个重点就可以很容易完成。

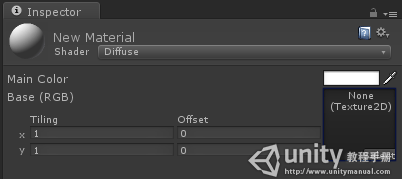
首先，就是如何贴图呢？也许直觉会想到使用GUI来贴图，但别忘了GUI是处在画面的表面空间，是纯2D的空间，一方面会受到画面宽高解析度的影响，一方面是无法套用3D物理引擎的东西，而且GUIClass只能使用於OnGUI()中，根据我的使用经验，OnGUI()在每frame会执行数次，如果贴图的图档设定不够优化的话，在较好的硬体可能不受影响，但在iPhone3G或iPodTouch等较旧机型硬体上，如此大量GUI贴图及运算将影响到执行效能，降低整个游戏的顺畅度，所以应该避免利用GUI贴图来制作2D游戏内容，当然也可能利用一些程式设计技巧来达到使用GUI制作2D游戏的目的，但这并不是这边想要说明的方式，而且我并不建议那样做，GUI还是用在静态的讯息表现上就好了；既然不建议利用GUI制作，那麽如何贴图呢？大家应该都知道使用Unity选单建立平面物件GameObject>CreateObject>Plane，这个平面有200个面，对我们来讲并不需要那麽多，所以可以自行使用3D建模软体自己拉个只有2面的平面来汇入Unity使用，建好平面物件後，在Project视窗中建立材质球并拉给已建立的平面物件即完成贴图了，当然在这个材质球未有任何Texture的情况下是看不到任何贴图的，此时只需要将欲贴图的图片档拉给材质球就行了，材质球预设的Shader是Diffuse，会使贴图贴满整个四方形的平面，这样贴图的运用就不灵活了，通常我们会使用许多去背图，此时只需要将Shader改为Transparent系列的Shader就可以达到去背的效果(通常没特殊需求，我是使用Transparent>Diffuse)，如此贴图的部份就完成了。

[](http://www.unitymanual.com/wp-content/uploads/2013/04/001.png)

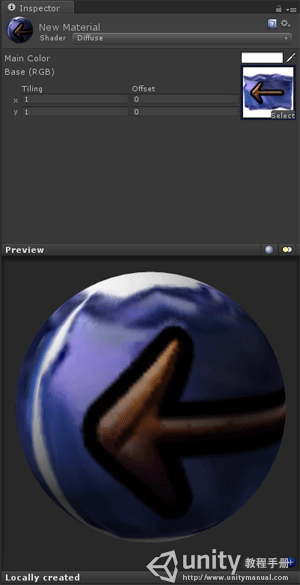
使用Unity选单建立平面物件

[](http://www.unitymanual.com/wp-content/uploads/2013/04/002.png)

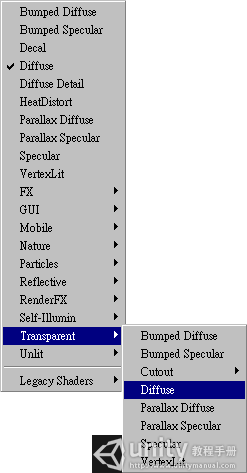
建立材质球

[](http://www.unitymanual.com/wp-content/uploads/2013/04/003.png)

材质球未有任何Texture的情况

[](http://www.unitymanual.com/wp-content/uploads/2013/04/004.png)

材质球预设的Shader是Diffuse

[](http://www.unitymanual.com/wp-content/uploads/2013/04/005.png)

Shader改为Transparent系列

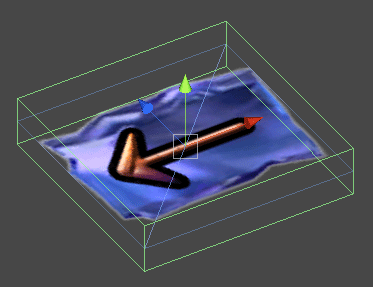
[](http://www.unitymanual.com/wp-content/uploads/2013/04/006.png)

Shader 是 Diffuse

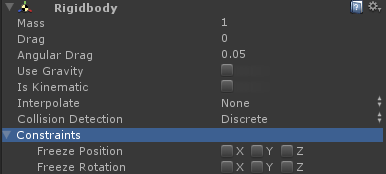
[](http://www.unitymanual.com/wp-content/uploads/2013/04/007.png)

Shader改为Transparent>Diffuse

此时在镜头前，此平面物件看起来就像是一般2D画面的贴图而已，但Scene视窗中可以很明显的看出它是3D空间中的物件，你可以任意的为其加入其他Component或自己写的Script来做控制，此时就可以为其加入刚体及碰撞体，那麽许多物理动作就能简单又快速的完成了；此时会发生一些问题是在碰撞时可能会使平面转动或是碰撞结果产生错误方向的位移，毕竟2D游戏只有X轴和Y轴，碰撞後多了Z轴的位移不就把画面破坏掉了，以往我们可能需要利用Update()或FixedUpdate()不断的将平面的角度及位置调整回来，如今的Unity版本(3.3)则可以直接在刚体的Component(Rigidbody)上直接冻结某个轴的旋转及位移，使其在碰撞後不受影响；当然，这个冻结项目只在物理引擎运算下有作用，我们还是可以透过程式码去操作已冻结的轴。

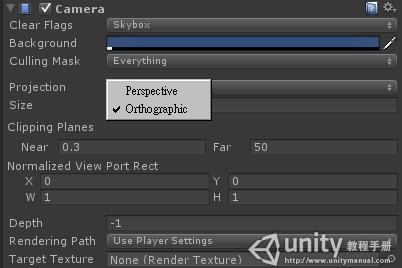
[](http://www.unitymanual.com/wp-content/uploads/2013/04/008.png)

加入刚体及碰撞体

[](http://www.unitymanual.com/wp-content/uploads/2013/04/009.png)

刚体可限制并冻结指定轴

接下来是镜头的设定，通常预设投射(Projection)方式是透视(Perspective)，这使画面有景深的立体感，有些2D游戏的背景或物件表现需要让远一点的东西较小，近一点的东西看起来较大，或是背景映出前景的影子等，那麽直接使用预设的Perspective即可，但如果你希望制作的2D游戏是整个平面化的，希望场景内配置的物件不会因为其中一个较远或较近而影响到大小或呈现角度，此时就必须将Projection改为Orthographic，那麽画面内的镜头可视范围将不会有任何远近感，达到完全平面的效果。

[](http://www.unitymanual.com/wp-content/uploads/2013/04/010.png)

改变镜头的投射方式

基本上，有了以上三个重点，就能轻易的利用Unity3D制作出2D游戏，其他的就看设计者如何利用Unity做更多灵活的变化罗！