**不同版本及发布平台编写专用的程式码**

Posted on 2013年04月19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 205 次

当制作跨平台游戏时，虽然Unity已经相当方便了，但因为各平台间的效能及功能差异，我们常常需要附加专用的程式码来做一些处理，尽管Unity的API已经有Application.platform来让我们进行判断，但有些RuntimeClasses在不适用的平台Bulid时会回报错误，像是iPhoneSettings或是iPhoneKeyboard在Bulid成PC平台游戏时就会直接报错而无法Bulid。

另外，当Unity有新版本发行，在Script的语法上也可能做些变动，像是近期的3.4，官方的更新详细内容说明页面上就有提到对于Javascript的变动，也有提醒旧有程式可能在更新到新版Unity後可能会报错，此时，我们就必须自行调整程式内容，才能使游戏正常运作。

基于这些在编译时就会发生的错误，即使我们在程式中加入Application.platform或Application.unityVersion来判断执行平台或Unity版本，也没多大帮助，所以只好在不同平台编译时将暂时不能用的程式区段注解起来，以免发生错误，但是这种做法实在是不方便，特别是当程式越写越多时，出问题的机会就更大了；幸好，Unity在2.6及3.0之後分别为版本及平台提供了解决之道，所以我们只要为不同平台或Unity版本都撰写专用的程式码，然後利用#if、#else及#endif将这些程式码区隔开来，那么编译时就只会去编辑到需要的部份，可以直接排除这一类编译期发生的错误。

用来判断平台的常数有：

UNITY\_EDITOR

UNITY\_STANDALONE\_OSX

UNITY\_DASHBOARD\_WIDGET

UNITY\_STANDALONE\_WIN

UNITY\_WEBPLAYER

UNITY\_WII

UNITY\_IPHONE

UNITY\_ANDROID

UNITY\_PS3

UNITY\_XBOX360

在 Unity 编辑器中，这些常数就已是有效用的，而不用等到输出时才会生效，例如在 Build Settings 中是将平台切换到 Web Player ，此时虽然未将游戏输出，在程式码中判断为 UNITY\_WEBPLAYER 可执行的区域就会被编译，如此也大大增加了制作时期的便利性。

用来判断版本的常数则有：

UNITY\_2\_6

UNITY\_2\_6\_1

UNITY\_3\_0

UNITY\_3\_0\_0

UNITY\_3\_1

UNITY\_3\_2

UNITY\_3\_3

UNITY\_3\_4