**Unity3D教程：Unity与Xperia Game pad**

Posted on 2013年05月14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 65 次

如果有Xperia Play且常玩游戏的朋友因该都有发现，目前支援Xperia Game pad的游戏其实也满多的，其中Unity的作品也不少，开发商Angry Mob Games在早期的一款作品“Guerrilla Bob”，早在2011年初就支援了Xperia Play的Game pad。

[](http://www.unitymanual.com/wp-content/uploads/2013/05/QQ截图20130514101420.png)

Unity3D教程：Unity与Xperia Game pad

最近发现，原来Unity的Script API本身并没有完整支援Xperia Play的Game Pad，开发者需要透过Unity提共的JAVA函数库来启动这项功能。其实，你只需要先透过Unity提共的AndroridJavaObject去实体化JAVA函数库的类别，之后再透过function去check手机目前的状态，来判断操作的模式，如下所示：

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

|  |  |
| --- | --- |
| 01 | **protected** **void** Awake () { |
| 02 |  |
| 03 | InitAndroidConfigLink(); |
| 04 |  |
| 05 | } |
| 06 |  |
| 07 | **private** AndroidJavaObject \_m\_currentConfig = **null**; |
| 08 |  |
| 09 | **private** **void** InitAndroidConfigLink() |
| 10 |  |
| 11 | { |
| 12 |  |
| 13 | **using**( AndroidJavaClass player = new AndroidJavaClass(“com.unity3d.player.UnityPlayer”) ) { |
| 14 |  |
| 15 | AndroidJavaObject activity = player.GetStatic(“currentActivity”); |
| 16 |  |
| 17 | \_m\_currentConfig = activity.Call(“getResources”).Call(“getConfiguration”); |
| 18 |  |
| 19 | } |
| 20 |  |
| 21 | } |

一切的重点就在你取得\_m\_currentConfig 的设定，有了这个设定後你就可以自行Check手机的Game Pad是否已经开启。这部分你可以每个Update检查或隔几秒检察，使用者是否开启或关闭Game Pad，不过更好的做法我想因该是Coroutine，自动动又方便。来看以下程式码：

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

|  |  |
| --- | --- |
| 01 | **protected** **void** Start () { |
| 02 |  |
| 03 | StartCoroutine( CheckForVirtualGamePadStatus() ); |
| 04 |  |
| 05 | } |
| 06 |  |
| 07 | **private** IEnumerator CheckForVirtualGamePadStatus() |
| 08 |  |
| 09 | { |
| 10 |  |
| 11 | **const** **int** NAVIGATIONHIDDEN\_UNDEFINED = 0; |
| 12 |  |
| 13 | *//const int NAVIGATIONHIDDEN\_NO = 1;* |
| 14 |  |
| 15 | **const** **int** NAVIGATIONHIDDEN\_YES = 2; |
| 16 |  |
| 17 | **int** nav; |
| 18 |  |
| 19 | **while**(**true**) { |
| 20 |  |
| 21 | nav = \_m\_currentConfig.**Get**(“navigationHidden”); |
| 22 |  |
| 23 | **if**( nav == NAVIGATIONHIDDEN\_YES || |
| 24 |  |
| 25 | nav == NAVIGATIONHIDDEN\_UNDEFINED ) { |
| 26 |  |
| 27 | **if**( \_m\_VirtualGamePad == **true** ) { |
| 28 |  |
| 29 | \_m\_VirtualGamePad = **false**; |
| 30 |  |
| 31 | } |
| 32 |  |
| 33 | } **else** { |
| 34 |  |
| 35 | **if**( \_m\_VirtualGamePad == **false** ) { |
| 36 |  |
| 37 | \_m\_VirtualGamePad = **true**; |
| 38 |  |
| 39 | } |
| 40 |  |
| 41 | } |
| 42 |  |
| 43 | **yield** **return** new WaitForSeconds(2.0f); |
| 44 |  |
| 45 | } |
| 46 |  |
| 47 | } |
| 48 |  |
| 49 | **protected** **void** Update () { |
| 50 |  |
| 51 | **if**( \_m\_VirtualGamePad ) { *//Do something for Xperia Gamepad... }* |
| 52 |  |
| 53 | **else** { *//Do others... }* |
| 54 |  |
| 55 | } |

接着剩下的就是Key Codes的对应！如下表：

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

|  |  |
| --- | --- |
| 01 | Input.GetKey (KeyCode.LeftShift) = Left shoulder |
| 02 |  |
| 03 | Input.GetKey (KeyCode.RightShift) = Right shoulder |
| 04 |  |
| 05 | Input.GetKey (KeyCode.UpArrow) = Up |
| 06 |  |
| 07 | Input.GetKey (KeyCode.LeftArrow) = Left |
| 08 |  |
| 09 | Input.GetKey (KeyCode.RightArrow) = Right |
| 10 |  |
| 11 | Input.GetKey (KeyCode.DownArrow) = Down |
| 12 |  |
| 13 | Input.GetKey (“joystick button 2”) = / |
| 14 |  |
| 15 | Input.GetKey (“joystick button 1”) = [] |
| 16 |  |
| 17 | Input.GetKey (“joystick button 3”) = O |
| 18 |  |
| 19 | Input.GetKey (“joystick button 0”) = X |
| 20 |  |
| 21 | Input.GetKey (KeyCode.Pause) = **SELECT** |
| 22 |  |
| 23 | Input.GetKey (KeyCode.**Return**) = START |
| 24 |  |
| 25 | Input.GetKey(KeyCode.Menu) = Android Menu |
| 26 |  |
| 27 | Input.GetKey(KeyCode.Escape) = Android Back |
| 28 |  |
| 29 | AndroidInput = TouchPad |