**有关延时执行的yield\coroutine相关用法**

Posted on 2013年07月21日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script) /被围观 31 次

1.yield： 声明一种特殊的return，可以传值如Waitforsecond(5.0)，实现延时或某一事件发生后触发。如果有需要有依赖关系顺序执行的函数，如：

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

|  |  |
| --- | --- |
| 01 | **private** **var** state = 0; |
| 02 |  |
| 03 | function Update() { |
| 04 |  |
| 05 | **if** (state == 0) { |
| 06 |  |
| 07 | *// do step 0* |
| 08 |  |
| 09 | state = 1; |
| 10 |  |
| 11 | **return**; |
| 12 |  |
| 13 | } |
| 14 |  |
| 15 | **if** (state == 1) { |
| 16 |  |
| 17 | *// do step 1* |
| 18 |  |
| 19 | state = 2; |
| 20 |  |
| 21 | **return**; |
| 22 |  |
| 23 | } |
| 24 |  |
| 25 | *// ...* |
| 26 |  |
| 27 | } |
| 28 |  |
| 29 | 可这样使用 **yield** |
| 30 |  |
| 31 | **while**(**true**) { |
| 32 |  |
| 33 | *// do step 0* |
| 34 |  |
| 35 | **yield**; *// wait for one frame* |
| 36 |  |
| 37 | *// do step 1* |
| 38 |  |
| 39 | **yield**; *// wait for one frame* |
| 40 |  |
| 41 | *// ...* |
| 42 |  |
| 43 | } |
| 44 |  |
| 45 | &nbsp; |

可以把 yield 理解为一种挂起 ,当函数执行到它时不在返回而是执行下一个函数的语句，一旦收到yield后定义的函数为true的消息，则执行yield返回并进行下面的代码。

2.coroutine:作为function StartCoroutine (routine : IEnumerator) 的返回值。一个函数中包含yield声明，就可以构成一个coroutine，这个函数的调用时必由StartCoroutine来进行调用，如下：

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

|  |  |
| --- | --- |
| 01 | function Awake(){ |
| 02 |  |
| 03 | StartCoroutine(MyDo()); |
| 04 |  |
| 05 | } |
| 06 |  |
| 07 | function MyDo(){ |
| 08 |  |
| 09 | **while** (**true**) { |
| 10 |  |
| 11 | print(“first”); |
| 12 |  |
| 13 | **yield** WaitForSeconds(3.0f); |
| 14 |  |
| 15 | print(“second”); |
| 16 |  |
| 17 | **yield** ; |
| 18 |  |
| 19 | **break**; |
| 20 |  |
| 21 | } |
| 22 |  |
| 23 | } |

也可以在如start等公用函数中直接使用yield，不用StartCoroutine.当消息返回后，系统会自动重新执行yield后的代码，如下：

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

|  |  |
| --- | --- |
| 01 | function Start () { |
| 02 |  |
| 03 | *// Start a download of the given URL* |
| 04 |  |
| 05 | **var** www : WWW = new WWW (url); |
| 06 |  |
| 07 | *// Wait for download to complete* |
| 08 |  |
| 09 | **yield** www; |
| 10 |  |
| 11 | *// assign texture* |
| 12 |  |
| 13 | renderer.material.mainTexture = www.texture; |
| 14 |  |
| 15 | } |