**Unity3D教程：使用C#编写脚本**

Posted on 2013年06月03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 78 次

在Unity中可以使用JavaScript、C#和Boo来编写脚本。在使用C#编写脚本时，需注意以下几个特点：

1、所有脚本都继承至MonoBehaviour

所有的行为脚本都必须直接地或间接地继承至MonoBehaviour。在使用JavaScript时，这种继承关系将自动生成，但在使用C#编写脚本时，必须显式地进行定义。通过模板创建的C#脚本时（Asset->Create->CSharpScript），自动生成的代码已经包含了这样的定义：

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

|  |  |
| --- | --- |
| 01 | *//C#* |
| 02 |  |
| 03 | publicclassNewBehaviourScript:MonoBehaviour |
| 04 |  |
| 05 | { |
| 06 |  |
| 07 | ... |
| 08 |  |
| 09 | } |
| 10 |  |

2、使用Awake或Start方法来进行初始化

在使用JavaScript时，所有在函数外的代码，在使用C#时，都必须把它们放置在Awake或Start方法中。Awake和Start方法的不同点在于：前者在场景进行加载时被调用；后者在调用Update或FixedUpdate方法之前被调用。因此，Awake方法在Start方法之前被调用。Unity3D教程手册

3、类名必须与文件名相同

在JavaScript中，类名被隐式地设置为文件名。但在C#中，必须手动地进行设置。

4、在C#中，协同程序的使用语法与JavaScript不同。

协同程序必须返回一个IEnumerator类型，并且使用yieldreturn...来代替JavaScript中的yield...。

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

|  |  |
| --- | --- |
| 01 | usingSystem.Collections; |
| 02 |  |
| 03 | usingUnityEngine; |
| 04 |  |
| 05 | publicclassNewBehaviourScript:MonoBehaviour |
| 06 |  |
| 07 | { |
| 08 |  |
| 09 | *//C#coroutine* |
| 10 |  |
| 11 | IEnumeratorSomeCoroutine() |
| 12 |  |
| 13 | { |
| 14 |  |
| 15 | *//Waitforoneframe* |
| 16 |  |
| 17 | yieldreturn0; |
| 18 |  |
| 19 | } |
| 20 |  |
| 21 | *//Waitfortwoseconds* |
| 22 |  |
| 23 | yieldreturnnewWaitForSeconds(2); |
| 24 |  |
| 25 | } |

5、不能使用命名空间(Namespace)

当前的Unity不支持把脚本放到一个命名空间中。这可能在将来的版本中所有改变。

6、只有成员变量才能被序列化并显示在检查器中

私有和保护成员变量仅仅在专家模式（ExpertMode）下才显示在检查器中。属性不能被序列化并显示在检查器中。

7、避免使用构造器

永远不要使用构造器来初始化值，而应该在Awake或Start中进行，到时Unity会自动触发相应的构造器，即使在编辑模式（EditMode）修改成员变量的值，也会自动触发。这通常在脚本编译完成后直接发生，以便为它们设置默认值。这样设计的主要原因有两个：构造器的调用不可预期；构造器的调用可能为了预置的或未活动的游戏对象。Unity3D教程手册

例如：单例模式（SingletonPattern）对象使用构造器进行初始化时，可能会导致值为空的严重后果。因此，应该在Awake方法中来初始化单例模式对象。请记住：任何继承至MonoBehaviour的类中，都不能包含使用构造器的代码。