**Unity3D中如何终止协同程序Coroutine**

Posted on 2013年03月05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 199 次

在[**Unity3D**](http://www.unitymanual.com)中，使用MonoBehaviour.StartCoroutine方法即可开启一个协同程序，也就是说该方法必须在MonoBehaviour或继承于MonoBehaviour的类中调用。

在[**Unity3D**](http://www.unitymanual.com/category/manual)中，使用StartCoroutine(string methodName)和StartCoroutine(IEnumerator routine)都可以开启一个线程。区别在于使用字符串作为参数可以开启线程并在线程结束前终止线程，相反使用IEnumerator 作为参数只能等待线程的结束而不能随时终止(除非使用StopAllCoroutines()方法)；另外使用字符串作为参数时，开启线程时最多只能传递一个参数，并且性能消耗会更大一点，而使用IEnumerator 作为参数则没有这个限制。

在[**Unity3D**](http://www.unitymanual.com/category/script)中，使用StopCoroutine(string methodName)来终止一个协同程序，使用StopAllCoroutines()来终止所有可以终止的协同程序，但这两个方法都只能终止该MonoBehaviour中的协同程序。

还有一种方法可以终止协同程序，即将协同程序所在gameobject的active属性设置为false，当再次设置active为ture时，协同程序并不会再开启；如是将协同程序所在脚本的enabled设置为false则不会生效。这是因为协同程序被开启后作为一个线程在运行，而MonoBehaviour也是一个线程，他们成为互不干扰的模块，除非代码中用调用，他们共同作用于同一个对象，只有当对象不可见才能同时终止这两个线程。然而，为了管理我们额外开启的线程，Unity3D将协同程序的调用放在了MonoBehaviour中，这样我们在编程时就可以方便的调用指定脚本中的协同程序，而不是无法去管理，特别是对于只根据方法名来判断线程的方式在多人开发中很容易出错，这样的设计保证了对象、脚本的条理化管理，并防止了重名。