**Unity3D教程：掌握多线程！**

Posted on 2013年05月27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 46 次

**什么叫线程？**

在程序中，这些独立运行的程序片断叫作“线程”（Thread），利用线程，用户可按下一个按钮，然后程序会立即作出响应，而不是让用户等待程序完成了当前任务以后才开始响应。

**如何掌握多线程？**

1.每个窗体都有自己的都在不同的线程上运行，如果需要在窗体之间交互，就需要在线程之间交互。

2.当线程Sleep时，系统就退出执行队列一段时间，当睡眠结束时，系统会产生一个时钟中断，从而使线程回到执行队列中，从而恢复线程的执行。

3.如果父线程先于子线程结束，那么子线程将在父线程结束的同时被迫结束。Thread.Join()方法使父线程等待，直到子线程结束。Abort()方法带来的后果是不可恢复的终止线程。Unity3D教程手册

4.起始线程可以称之为主线程，如果所有的前台线程都停止了，那么主线程可以终止，而所有的后台线程都将无条件终止。后台线程跟前台线程只有一个区别，那就是后台线程不妨碍程序的终止。一旦一个进程所有的前台线程都终止后，CLR将通过调用任意一个存活中的后台进程的Abort()方法来彻底终止进程。

6.挂起，睡眠（都可称为－－阻塞，暂停）与Thread.Sleep 不同，Thread.Suspend 不会使线程立即停止执行。直到线程到达安全点之后它才可以将该线程挂起。如果线程尚未启动或已经停止，则它将不能挂起。调用 Thread.Resume 将使另一个线程跳出挂起状态并使该线程继续执行。一个线程不能对另一个线程调用Sleep ，但是一个线程可以对另一个线程调用Suspend。还可以使用许多其它的方式来阻塞线程。例如，可以通过调用 Thread.Join 使一个线程等待另一个线程（子线程）停止。使用Monitor.Wait使一个线程等待访问一个同步对象。

5.关键字lock可以把一段代码定义为互斥段（critical section），互斥段在一个时刻内只允许一个线程进入执行，而其他线程必须等待。多线程公用一个对象时，就不应该使用lock关键字了，这里Monitor，Monitor提供了使线程共享资源的方案。Monitor类可以锁定一个对象，一个线程只有得到这把锁才可以对该对象进行操作。

如：

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

|  |  |
| --- | --- |
| 1 | Monitor.Enter(obj); |
| 2 |  |
| 3 | *//现在oQueue对象只能被当前线程操纵了* |
| 4 |  |
| 5 | Monitor.Exit(obj); |

6.一个进程开始时至少会有一个主线程 ( 即主执行实例 ) ，这是在系统加载你的程序的时候所创建的主执行流程。而消息队列则是与线程 ( Thread ) 相关的，在似win2k上一个线程有一个且只有一个消息队列 ( queue ) 与之相对应。消息队列是在什么时候生成的呢？ 在似win2k系统上，从一开始创建线程就已经有了。一个线程可以创建多个窗体。统发送给这些窗口的消息都统一发送到同一个 消息队列 中，幸亏消息结构中有msg.hwnd指出该条消息与哪一个窗口相关， DispatchMessage() 函数就是依照这个保证消息分派处理自动化而且不会出错！Unity3D教程手册

7.每个窗体都属于创建它的线程，在一线程中直接访或间接问其它线程中的窗体的方法将导致运行时错误（VS2005）。

解决方法：使用窗体从Control继承而来的Control.Invoke（Delegate）方法。该方法将在创建窗体的线程上执行委托指向的方法。

注意：在VS2003下，可以在一个线程中直接或间接调用另一个线程中的窗体的方法，而不会导致运行时错误。