**协程**

大多数脚本代码只显示在性能跟踪内的位于特定Unity回调调用下某一个位置。但是，协程的 CPU 代码总是出现在跟踪内的哪里？

协程中的所有初始代码（从协程方法的开始一直到第一次暂停）将出现在跟踪过程中任何启动协程的位置。通常出现在调用什么方法的位置？

协程代码的所有其余部分（从第一次恢复一直到完成执行）将显示在 Unity 主循环内出现的什么行中？

启动协程引起的内存压力等于固定开销成本加上什么？

启动协程的代码将构造并调用此对象，然后 Unity 的什么在每当满足协程的暂停条件时再次调用此对象？

虽然嵌套的协程非常有利于确保代码的条理性和进行维护，但协程跟踪对象本身会导致产生什么？

如果一个协程几乎每帧都运行并且在长时间运行操作中不会暂停，那么用什么回调来替换该协程通常更合理一些？

禁用对象时，协程不会停止，只有怎么样时才会停止？

在处理什么操作（例如等待 HTTP 传输、资源加载或文件 I/O 完成）时，最适合使用协程？