

# Unity3D中的协程

协程

# Unity3D中的协程



# 协程概念理解

- 与方法一样，协程也是一种程序组件。
- 相对一般方法而言，协程更为灵活，方法只有一个入口，而协程可以有多个入口和出口点，可以用协程来实现任何的方法。
- 协程更适合于用来实现如合作式多任务，迭代器，无限列表。
- 开始一个协程时，执行从被调用方法的起始处开始；然而，接下来的每次协程被调用时，从协程返回（`yield` `return`）的位置接着执行。
- 协程是单线程的。



# Unity中协程实现

- Unity协程实现主要组成
  - IEnumerator接口 提供迭代
  - yield return xxxx 指定下一次迭代入口
    - WaitForFixedUpdate 在FixedUpdate后执行
      - 适合于物理控制，
    - null、0、WaitForSeconds 在每帧Update后执行
      - 适合于分多帧完成一个任务，如A\*路径计算
    - WaitForEndOfFrame 在每帧结束后执行
      - 适合于每帧末尾执行的操作，如相机的控制
    - WWW 在请求结束后执行
      - 适合于网络下载数据。
  - StartCoroutine 开启协程



# 协程的停止

- StopCoroutine(string)
  - StopCoroutine( “协程方法名” )
  - 只能停止以StartCoroutine( “协程方法名” )开启的协程
- StopAllCoroutine()
  - 停止本对象中开启的所有协程



# 使用协程的注意



# 使用协程的注意

- 在程序中调用StopCoroutine()方法只能终止以字符串形式启动（开始）的协程；
- 多个协程可以同时运行，它们会根据各自的启动顺序来更新
- 协程可以嵌套任意多层
- 如果你想让多个脚本访问一个协程，那么你可以定义静态的协程；
- 协程不是多线程，它们运行在同一线程中，跟普通的脚本一样；
- 如果你的程序需要进行大量的计算，那么可以考虑在一个随时间进行的协程中处理它们；
- IEnumerator类型的方法不能带ref或者out型的参数
- 目前在Unity中没有简便的方法来检测作用于对象的协程数量以及具体是哪些协程作用在对象上。

