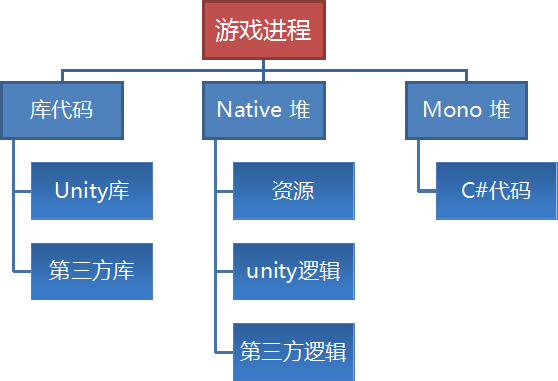
内存是手游的硬伤——Unity游戏Mono内存管理及泄漏

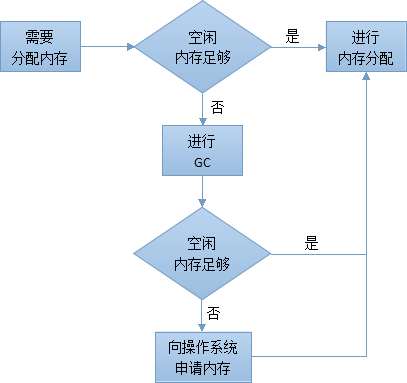
<http://wetest.qq.com/lab/view/135.html>

对于目前绝大多数基于Unity引擎开发的项目而言，其托管堆内存是由Mono分配和管理的。“托管” 的本意是Mono可以自动地改变堆的大小来适应你所需要的内存，并且适时地调用垃圾回收（Garbage Collection）操作来释放已经不需要的内存，从而降低开发人员在代码内存管理方面的门槛。Unity游戏在运行时的内存占用情况可以用下图表示：



Mono内存管理策略

Mono内存分为两部分，已用内存（used）和堆内存（heap），已用内存指的是mono实际需要使用的内存，堆内存指的是mono向操作系统申请的内存，两者的差值就是mono的空闲内存。



GC的主要作用在于从已用内存中找出那些不再需要使用的内存，并进行释放。Mono中的GC主要有以下几个步骤：

1.停止所有需要mono内存分配的线程。

2.遍历所有已用内存，找到那些不再需要使用的内存，并进行标记。

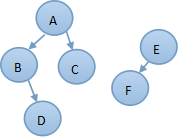
3.释放被标记的内存到空闲内存。

4.重新开始被停止的线程。

**另外，GC释放的内存只会留给mono使用，并不会交还给操作系统，因此mono堆内存是只增不减的。**

Mono内存泄漏分析

当GC的时候，以**全局数据区和当前寄存器中的对象**为根节点，按照引用关系进行遍历，对于遍历到的每一个对象，将其标记为活的（alive）。

****

对象已经不再需要使用却没有被GC回收的情况称为mono内存泄漏。在某对象超出其作用域时，我们 “忘记”清除对该无用对象的引用了。

解决办法

1. 如果某个静态资源引用一些new创建的动态资源，当这些资源不在需要时，程序并没有主动把静态资源的引用设置为null；objectA.a设置为null只是断绝了引用关系，真正对象的回收要等到GC的时候才会进行。
2. 而且GC也是用算法做的，难免会遇到一些考虑不全面的地方。

[**Cube**](http://mp.weixin.qq.com/s?__biz=MzA3NjA3NTI5Mg==&mid=2656328975&idx=1&sn=f998c1fc2214460b59aa547cc59824d5&scene=21#wechat_redirect)**工具现在提供免费使用**

**体验地址：**[http://wetest.qq.com/cube](http://wetest.qq.com/cube?from=content_columnist_cloud)

**关于测试报告的问题**：<http://wetest.qq.com/guide/view/?id=267>

**使用帮助**：<http://wetest.qq.com/guide/view/?id=266>

**常见问题**：<http://wetest.qq.com/guide/view/?id=268>