<http://wetest.qq.com/lab/view/150.html>

**内存泄漏的简单定义，就是申请了内存，却没有在该释放的时候释放**。

操作系统将会直接kill掉应用程序。**如果持续泄漏，将因内存占用过大而导致应用崩溃**。当然泄漏还有其他的危害，例如内存被无用对象占用，导致接下来的内存分配需要更高的时间成本，从而造成游戏的卡顿等等。

**代码中的泄漏 – Mono内存泄漏**

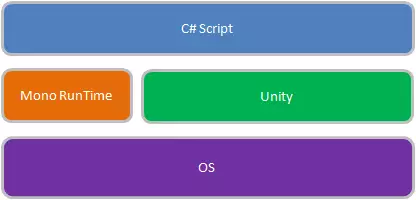
Mono内存泄漏的原因：

在Unity环境下，Mono堆内存的占用，是只会增加不会减少的。具体来说，可以将Mono堆，理解为一个内存池，每次Mono内存的申请，都会在池内进行分配；**释放的时候，也是归还给池，而不会归还给操作系统**。如果某次分配，发现池内内存不够了，则会对池进行扩建——向操作系统申请更多的内存扩大池以满足该次的内存分配。需要注意的是，每次对池的扩建，都是一次较大的内存分配，每次扩建，都会将池扩大6-10M左右（此处无官方数据，是观察所得）。

**资源中的泄漏 – Native内存泄漏**

资源加载之后占有了内存，但是在资源不用之后，没有将资源卸载导致内存的无谓占用。

上文中说的代码分配的内存，是通过Mono虚拟机，分配在Mono堆内存上的，其内存占用量一般较小，主要目的是程序猿在处理程序逻辑时使用；而Unity的资源，是通过Unity的C++层，分配在Native堆内存上的那部分内存。举个简单的例子，通过UnityEngine命名空间中的接口分配的内存，将会通过Unity分配在Native堆；通过System命名空间中的接口分配的内存，将会通过Mono Runtime分配在Mono堆。



还有一种资源上的泄漏，是因为Unity的一些接口在调用时会产生一份拷贝（例如Renderer.Material参考https://docs.unity3d.com/ScriptReference/Renderer-material.html），如果在使用上不注意的话，运行时会产生较多的资源拷贝，造成内存的无端浪费。