## Instruments—找到被抛弃的内存

Allocations分析模板使用Allocations和VM Tracker测量你应用中的普通内存和虚拟内存使用。但是，跟踪被抛弃的内存，也就是那些被分配了却不再需要的内存，你需要严密关注Allocations检查器。该检查器测量了堆内存的使用，并跟踪所有的分配，包括类的特定对象分配。

由于被抛弃的内存仍然被你的应用所引用，检查器无法判断其重要性。要找到被抛弃的内存，你必须使用代分析以确保在重复执行相同的操作时内存不会持续增长。例如，开始和结束一个新的游戏，打开和关闭一个窗口，创建和删除一个连接，或者选中和取消一个偏好设置，所有这些操作在理论上应该都能返回到应用程序的前一个稳定的内存状态。经过多次大量的反复操作不应该导致意外的或无法控制的内存增长。Instruments可以帮助你把内存增长的阶段与特定的对象分配相关联，这样你就可以释放他们并减少你应用的内存占用。

### To look for abandoned memory in your app

1. 启动Instruments。
2. 当分析模板选择对话框出现时，点击Allocations。
3. 在目标设备和进程列表中选择你的设备和应用。
4. 点击Choose创建跟踪文档。
5. 点击时间轴窗格中的Allocations检查器。
6. 按下Command-2显示检视器窗格的显示设置区域。这样你就可以快速的访问到Mark Generation按钮，在你分析应用程序时，可以用该按钮标记操作的不同代。
7. 点击工具栏中的Record按钮()或按下Command-R开始记录。
8. 在应用中执行一小串可重复的操作。为了准确的生成趋势图，这一系列操作的开始和结束应该在应用中是相同的状态。
9. 点击检视器窗格中的Mark Generation按钮。在轨迹窗格中会出现一个标识，标识一个代。你标记的代的列表会显示在详情窗格里。每一个代都包含了从前一个代开始所发生的分配的列表。你也可以在记录完成后再标记代，只需要在轨迹窗格的时间轴里将检视的开头拖动到想要的地方再点击Mark Generation按钮即可。
10. 在监视详情窗格的时候重复步骤8和9若干次，直到你可以判断内存是否不受限制的增长。

IMPORTANT

注意：在最开始的几次迭代中，可能会由于缓存发生更多的分配。因此，为了建立基础标准创建一些初始代是非常重要的。然后再创建后续的代进行真正的分析。

1. 点击工具栏中的Stop按钮()或再次按下Command-R停止记录。
2. 在详情窗格扫过各个代，找到似乎表现出重复内存增长的地方。Growth列和# Persistent列会告诉你与前一个代相比增加了多少内存和发生了多少次分配。如果你的应用在某个操作之后返回到了其初始状态，那么这里就不会有一代一代的内存增长。
3. 点击每一代的展开三角形()，可以显示从前一代开始被分配的新的对象。
4. 找到那些持续存在的对象。如果你认定一个，就点击展开三角形()显示它的实例。
5. 选择一个对象的实例。
6. 按下Command-3在检视器的扩展详情区域显示选中实例的栈轨迹。栈轨迹可以提供分配该实例的完整的方法调用列表。
7. 在扩展详情区域点击Collapse按钮()可以隐藏栈轨迹中的系统调用。你的应用发起的调用会被标记成黑色并在前面加上一个用户代码图标().。
8. 双击栈轨迹中的方法可以在Instruments中显示其代码。
9. 点击详情板顶部的Xcode按钮()可以打开Xcode中的代码进行编辑。
10. 判断这次分配是否有用。如果没用，这就是被抛弃的内存，你应该解决它。