**二、Monkey测试策略**

测试策略就是指我们将如何开展我们的测试活动，即“测什么”和“如何测”。

具体来说，就是答好和产品测试相关的六大问题：

 测试的对象和范围是什么？

 测试的目标是什么？

 测试的重点和难点是什么？

 测试的深度和广度？

 如何安排各种测试活动（先测试什么，再测试什么）？

 如何评价测试的效果？

**1. 测试分类**

    Monkey测试针对不同的对象和不同的目的采用不同的测试策略，首先测试的对象、目的及类型如下：

**1）测试的对象和范围：**

单个目标APP：公司或项目组需要测试的一个APP。

同类型APP：比如，教育类的APP，同时测试比较结果。

**2）测试的目标：**

不忽略异常的测试：白天测试，及时反馈和解决问题。

忽略异常的测试：晚上过夜测试，继续执行后续的测试。

特殊场景测试：假如我们想验证横竖屏切换功能是否正常，那就需要选择--pct-rotation参数，调大横竖屏切换事件的比例；假如我们想验证重复某种特定操作时，应用是否会存在异常，那可以选择-f参数，自定义Monkey脚本进行验证。

**3）测试的类型：**

稳定性测试：测试APP是否存在ANR和Crash等问题。

性能测试：测试APP启动慢，卡顿，CPU不足/内存泄露，应用耗电快等问题。

压力测试：测试APP在快速的事件响应的过程中，程序是否能正常运行。

**2. Monkey测试策略制定**

如何让测试策略真正起到指导测试的作用？我们还需要一套方法来指导我们制定测试策略的整个过程，“四步测试策略制定法”。如图4所示：



图4 四步测试策略制定法

**1）针对单个目标APP（稳定性和性能测试）**

针对单个目标APP的目的**是充分挖掘自身的问题，提高APP的稳定性和性能**。当然，性能参数获取要借助自动化的手段实现，也是我们封装autoMonkey的目的之一。

**(1) 不忽略异常**

白天，针对单个APP的稳定性测试时，则选择目标APP且不忽略异常的命令执行。**如果出现异常，及时分析原因和开发沟通。**

**例如：**

monkey -p $package\_name --throttle 1000 --randomize-throttle -s 10000 --pct-touch 40 --pct-motion 25 --pct-appswitch 10 --pct-rotation 10  --pct-majornav 10 -v -v -v 10000 > $path\monkey\_test.txt

**说明：**

参数--throttle设置固定延时，其值为1000毫秒。

参数--randomize-throttle为随机延时，即在0 -- 1000毫秒之内随机延时。希望能更真实地模拟用户的使用场景。

参数-s为伪随机事件流，用于复现问题。在自动化封装是，可以生产一个随机整数；否则，所有随机事件都是相同的。

**(2) 忽略异常**

晚上，在进行单个APP的过夜测试时，则选择目标APP且忽略异常的命令执行。让Monkey在遇到崩溃或没有响应的时候，能在日志中记录相关信息，并继续执行后续的测试。这样可以在测试过程中，尽可能多地发现APP中存在的问题而不影响Monkey的执行。

**例如：**

monkey -p $package\_name --throttle 1000 -s 10000 --pct-touch 40 --pct-motion 25 --pct-appswitch 10 --pct-rotation 10  --pct-majornav 10 --ignore-crashes --ignore-timeouts --ignore-security-exceptions --ignore-native-carshes --monitor-native-crashes -v -v -v 15000 > $path\monkey\_test.txt

**(3) 特殊场景**

根据项目不同的需求和出现的问题，针对特定的场景进行测试。如，验证横竖屏切换功能是否正常，那就需要选择--pct-rotation参数，调大横竖屏切换事件的比例。

**例如：**

monkey -p $package\_name --throttle 1000 -s 1000 --pct-rotation 80 -v -v -v 15000 > $path\monkey\_test.txt

**2）针对同类型APP（性能测试）**

针对同类型APP的目的是与同类型的APP比较，分析各自的优缺点。找到自己的不足，努力提高APP的性能。同样，性能参数获取要借助自动化的手段实现。

**(1) 不忽略异常**

把同类型的APP导入白名单，通过白名单进行测试；不忽略异常原因与单个APP稳定性测试的第一个场景类似。

**例如：**

monkey --pkg-whitelist-file /data/whitelist.txt -hprof --throttle 500 -s 10000 --pct-touch 40 --pct-motion 25 --pct-appswitch 10 --pct-rotation 10  --pct-majornav 10 -v -v -v 15000 >  $path\monkey\_test.txt

说明： 参数-hprof在出现内存问题时，存储一份内存“快照”。

**(2) 忽略异常**

把同类型的APP导入白名单，通过白名单进行测试；忽略异常原因与单个APP稳定性测试的第二个场景类似。

**例如：**

monkey --pkg-whitelist-file /data/whitelist.txt--throttle 1000 -hprof --throttle 500 -s 100 --pct-touch 40 --pct-motion 25 --pct-appswitch 10 --pct-rotation 10  --pct-majornav 10 --ignore-crashes --ignore-timeouts--ignore-security-exceptions --ignore-native-carshes --monitor-native-crashes-v -v -v 15000 >  $path\monkey\_test.txt

**3）针对单个目标APP（压力测试）**

根据项目需求，缩短monkey测试中事件与事件之间的延迟时间，验证在快速的事件响应的过程中，APP是否能正常运行。

**例如：**

monkey -p $package\_name -hprof --throttle 200 -s 10000 --pct-touch 40 --pct-motion 25 --pct-appswitch 10 --pct-rotation 10  --pct-majornav 10 --ignore-crashes --ignore-timeouts --ignore-security-exceptions --ignore-native-carshes --monitor-native-crashes -v -v -v 15000 > $path\monkey\_test.txt

**说明：**

参数--throttle设置固定延时，其值为200毫秒。可以根据手机性能的高低端调整。

小  结

上面介绍了几种常见的Monkey测试方法，**但在实际项目中，选择哪种Monkey测试策略，则需要根据实际项目的情况来做判断。主要是看测试目的及被测应用自身的特点。**

假如我们想验证横竖屏切换功能是否正常，那就需要选择--pct-rotation参数，调大横竖屏切换事件的占比进行Monkey测试；

假如我们想验证重复某种特定操作时，应用是否会存在异常，那可以选择-f参数，自定义Monkey脚本进行验证；

假如我们想验证长时间操作时应用是否会存在内存泄漏，那就需要结合-hprof参数和dumpsys meminfo $package\_name or $pid进行Monkey性能测试。