

目录

1. Monkey 的含义.....	2
2. Monkey 的运用.....	2
3. Mongkey 语句分类及 API.....	2
4. Monkey 产生 LOG 的解析.....	5
5. Monkey 的使用场景.....	5
6. Monkey 使用注意点.....	7
7. Monkey 结束方法.....	7
8. MonkeyScript.....	7
9. MonkeyScript 的 API.....	7
10. MonkeyScript 的简单使用.....	8
11. 程序异常出错术语介绍.....	9

1. Monkey 的含义

monkey 测试的原理就是利用 socket 通讯的方式来模拟用户的按键输入，触摸屏输入，手势输入等，看设备多长时间会出异常。

Monkey 程序由 Android 系统自带，使用 Java 语言写成，在 Android 文件系统中的存放路径是：
/sdk/sdk/tools/lib/monkey.jar。

备注：monkey 环境搭建详情请查看：<http://jingyan.baidu.com/article/14bd256e2b190bbb6d261228.html>

2. Monkey 的运用

主要运用于压力测试，分为整机压力测试及单个 APP 测试。

3. Monkey 语句分类及 API

adb -s +设备 ID shell monkey -p +包名 -s 种子数+事件百分比+异常处理+log 等级+延迟事件+事件数 >导出 log 的路径

如：adb -s PJQDU15731003937 shell monkey -p com.m4399.gamecenter -s 168 --pct-touch 80 --pct-motion 20 --ignore-crashes -v -v --throttle 500 10000 >d:\monkey.log

注：运行 adb 命令之前先运行 adb devices，确认下设备是否有连接，如果没有连接，有问题可以看下是不是没有把类似手机助手的软件给关掉。详情遇到的问题可以查看：

<http://www.51testing.com/html/17/521817-850713.html>

Monkey 命令可简单分为如下四类：

基本配置选项，如设置尝试的事件数量

运行约束选项，如设置只对单独的一个包进行测试

事件类型和频率，如设置触摸屏事件的发生频率

调试选项，如设置忽略应用程序异常，继续向系统发送事件，直到计数完成

类别	选项	说明
常规选项	-help	列出简单的用法
	-v	命令行的每一个“-v”将增加反馈信息的级别。 Level 0(缺省值):除启动提示、测试完成和最终结果之外, 提供较少信息。 Level 1:提供较为详细的测试信息, 如逐个发送到 Activity 的事件。 Level 2:提供更加详细的设置信息, 如测试中被选中的或未被选中的 Activity
约束选项	-p <allowed-package-name>	如果用此参数指定了一个或几个包, Monkey 将只允许系统启动这些包里的 Activity。如果你的应用程序还需要访问其它包里的 Activity(如选择取一个联系人), 那些包也需要在此同时指定。如果不指定任何包, Monkey 将允许系统启动全部包里的 Activity。要指定多个包, 需要使用多个 -p 选项, 每个-p 选项只能用于一个包。
	-c <main-category>	如果用此参数指定了一个或几个类别, Monkey 将只允许系统启动被这些类别中的某个类别列出的 Activity。如果不指定任何类别, Monkey 将选择下列类别中列出的 Activity: Intent.CATEGORY_LAUNCHER 或 Intent.CATEGORY_MONKEY。要指定多个类别, 需要使用多个-c 选项, 每个-c 选项只能用于一个类别。
事件选项	-s <seed>	伪随机数生成器的 seed 值。如果用相同的 seed 值再次运行 Monkey, 它将生成相同的事件序列。
	--throttle <milliseconds>	在事件之间插入固定延迟。通过这个选项可以减缓 Monkey 的执行速度。如果不指定该选项, Monkey 将不会被延迟。
	--pct-motion <percent>	调整动作事件的百分比(动作事件由屏幕上某处的一个 down 事件、一系列的伪随机事件和一个 up 事件组成)。
	--pct-trackball <percent>	调整轨迹事件的百分比(轨迹事件由一个或几个随机的移动组成, 有时还伴随有点击)。
	--pct-nav <percent>	调整“基本”导航事件的百分比(导航事件由来自

事件选项		方向输入设备的 up/down/left/right 组成)。
	--pct-majornav <percent>	调整“主要”导航事件的百分比(这些导航事件通常引发图形界面中的动作, 如: 5-way 键盘的中间按键、回退按键、菜单按键)
	--pct-syskeys <percent>	调整“系统”按键事件的百分比(这些按键通常被保留, 由系统使用, 如 Home、Back、Start Call、End Call 及音量控制键)。
	--pct-appswitch <percent>	调整启动 Activity 的百分比。在随机间隔里, Monkey 将执行一个 startActivity()调用, 作为最大程度覆盖包中全部 Activity 的一种方法。
	--percent-anyevent <percent>	调整其它类型事件的百分比。它包罗了所有其它类型的事件, 如: 按键、其它不常用的设备按钮、等等。

调试选项	--dgb-no-events	设置此选项, Monkey 将执行初始启动, 进入到一个测试 Activity, 然后不会再生成事件。为了得到最佳结果, 把它与 -v、一个或几个包约束、以及一个保持 Monkey 运行 30 秒或更长时间的非零值联合起来, 从而提供一个环境, 可以监视应用程序所调用的包之间的转换。
	--hprof	设置此选项, 将在 Monkey 事件序列之前和之后立即生成 profiling 报告。这将会在 data/misc 中生成大文件(~5Mb), 所以要小心使用它
	--ignore-crashes	通常, 当应用程序崩溃或发生任何失控异常时, Monkey 将停止运行。如果设置此选项, Monkey 将继续向系统发送事件, 直到计数完成。
	--ignore-timeouts	通常, 当应用程序发生任何超时错误(如“Application Not Responding”对话框)时, Monkey 将停止运行。如果设置此选项, Monkey 将继续向系统发送事件, 直到计数完成。
	--ignore-security-exceptions	通常, 当应用程序发生许可错误(如启动一个需要某些许可的 Activity)时, Monkey 将停止运行。如果设置了此选项, Monkey 将继续向系统发送事件, 直到计数完成。
	--kill-process-after-error	通常, 当 Monkey 由于一个错误而停止时, 出错的应用程序将继续处于运行状态。当设置了此选项时, 将会通知系统停止发生错误的进程。注意, 正常的(成功的)结束, 并没有停止启动的进程, 设备只是在结束事件之后, 简单地保持在最后的状态。
	--monitor-native-crashes	监视并报告 Android 系统中本地代码的崩溃事件。如果设置了 --kill-process-after-error, 系统将停止运行。
	--wait-dbg	停止执行中的 Monkey, 直到有调试器和它相连

4. Monkey 产生 LOG 的解析

案例讲解

注意点:

ID	Event percentages	事件	
0	15%	--pct-touch	触摸事件
1	10%	--pct-motion	滑动事件
2	2%	--pct-pinchzoom	手势事件
3	15%	--pct-trackball	滚动事件
4	0%	--pct-rotation	旋转事件
5	25%	--pct-nav	导航事件（方向）
6	15%	--pct-majornav	主要导航事件
7	2%	--pct-syskeys	系统案件事件
8	2%	--pct-appswitch	activity 切换事件
9	1%	--pct-flip	调整点击事件
10	13%	--pct-anyevent	调整其他事件

Log 关键词搜索:

ANR Fatal Crash Exception (查看包名)

5. Monkey 的使用场景

5.1 策略

◆ 异常策略

跑完: 全部异常忽略

专门测试某个异常: 不忽略某个异常, 出现某个异常即停止测试

人在时候: 上班过程中跑不忽略异常, 出现异常马上停止, 可以马上处理

验收策略: 去除全部异常, 出现错误则停止, 则验收不通过

◆ 延时策略

低延时

高延时

随机延时

用户操作延时

◆ 事件数量

常规测试: 10W

压力型测试: 30W

稳定性测试: 50W

长时间执行: 100W

5.2 命令组合

应用选取策略+随机种子策略+事件策略+异常策略+延时策略+事件数量

◆ 应用选择策略

单应用

多应用组合

黑白名单组合

整机测试

◆ 随机种子策略

固定种子，从小到大

随机种子

◆ 事件策略

用户故事策略：依据常见的用户场景划分各事件百分比

应用特性策略：依据应用策略对 Monkey 各事件进行划分百分比

专项测试策略：对某个事件提高到很高的百分比，对应用进行专项测试

5.3 monkey 使用策略

✓ 需求开发

轻量 延迟高些 针对专项

✓ 迭代完成

高压 全面测试 各种组合命令 专项 延时低 事件多 关注性能

✓ 稳定版

虐待 全面测试 加大压力 事件大 专项 性能

✓ 验收

全部测试 不忽略异常

5.4 专项测试

专项名称	压力事件	理论依据
旋转压力测试	--pct-rotation	旋转处理是否得当，数据刷新是否合理，数据传递是否正确，经典旋转内存泄漏等
应用开启测试	--pct-appswitch	开关的数据文件数据，网络处理，初始化等
手势压力测试	--pct-pinchzoom	对手势支持者，压力代码处理是否合理
滑动压力测试	--pct-motion	页面左右与上下滑动，页面快速切换，列表刷新等
关键按钮压力测试	--pct-nav	常用的案件对页面应用，对常用按钮的处理
导航按钮压力测试	--pct-major-nav	焦点的切换问题
其他按钮压力测试	--pct-anyevent	是否会对特殊按钮响应出错

6. Monkey 使用注意点

- 事件的总百分比不能超过 100%
- 事件的百分比要写出，不能空白，空白当事件没有 100%的时候，会自动分配
- 记得加种子，如果没有加种子的话，不方便问题的回归
- 注意编写格式（会提供一个脚本工具）

7. Monkey 结束方法

方法一、

前提：手机需要有 ROOT 权限

Adb shell ps |findstr "monkey"

Adb shell kill +PID

方法二、

拔电池关机

方法三、

Adb reboot

8. MonkeyScript

优点：简单，快捷，不需要借助任何工具，只要一个记事本文件

缺点：实现坐标、案件等基本操作的相应步骤，是没有逻辑判断的

9. MonkeyScript 的 API

API	说明
<u>LaunchActivity(Pkg_name,cl_name)</u>	启动应用的Activity
<u>Tap(x, y, tapDuration)</u>	模拟一次手指单击事件
<u>DispatchPress(keyName)</u>	按键
<u>RotateScreen(rotationDegree,peresist)</u>	选择屏幕
<u>DispatchFlip(true/false)</u>	打开或者关闭软键盘
<u>LongPress()</u>	长按两秒
<u>PressAndHold(x, y, pressDuration)</u>	模拟长按事件
<u>DispatchString(input)</u>	输入字符串
<u>Drag(xStart,yStart,xEnd,yEnd,stepCount)</u>	用于模拟一个拖拽操作
<u>PinchZoom(pt1xStart,pt1yStart,pt1xEnd,pt1yEnd,pt2xStart,pt2yStart,pt2xEnd,pt2yEnd,stepCount)</u>	模拟缩放手势
<u>UserWait(sleepTime)</u>	让脚步中断一段时间
<u>DeviceWakeUp()</u>	唤醒屏幕

10. MonkeyScript 的简单使用

涉及工具: adb, uiautomatorviewer.bat, hierarchyviewer.bat

注意: uiautomatorviewer 只能到 SDK 的 tools 去启动, 而不要直接 DOS 输入启动, 不然会出现端口被占用, 手机连接失败

案例讲解 (游戏盒 1 分钟只能发表 2 条动态验证)

```
#Start Script
type = raw events
count = 49
speed = 1.0
#以下为monkey命令
start data >>
LaunchActivity(com.m4399.gamecenter,com.m4399.gamecenter.ApplicationActivity)
UserWait(5000)
#点击动态
Tap(324,1693)
#点击动态编辑按钮
Tap(987,139)
UserWait(2000)
#输入任意一个动态
DispatchString(4399)
UserWait(2000)
#点击发布
Tap(987,139)
UserWait(5000)
#点击动态编辑按钮
Tap(987,139)
UserWait(2000)
#输入任意一个动态
DispatchString(youxibox)
UserWait(2000)
#点击发布
Tap(987,139)
UserWait(5000)
#退出游戏盒
DispatchPress(3)
DispatchPress(3)
```

程序写好以后, 然后保存在 D 盘命令为 dongtaifabiao, 然后推送到手机目录, 命令为 adb push d:\dongtaifabiao.txt /data/local/tmp/, 推送成功后, 可以检查下, 检查命令为


```

C:\Users\Administrator>adb shell
shell@hwRIO-AL00:/ $ cd data/local/tmp
cd data/local/tmp
shell@hwRIO-AL00:/data/local/tmp $ ls
ls
123.hprof
124.hprof
1458550925
1460086182
2016-01-28-11com.baidu.appsearch{6FCE2CC3-9DE8-4EE2-9662-82E4FC4140C8}.apk
2016-04-05-16com.baidu.appsearch{A5B59010-69F7-4F6F-A570-8A3EF5A2A14F}.apk
4.7.txt
aapt2
aapt_android
box2
busybox
busybox1
busybox2
cn.m4399.coresdk
cn.m4399.gamedemo
com.example.jimmie.listeninstalldemo
com.m4399.forums
com.m4399.gamecenter
com.m4399.gamecenter-build-id.txt
dalvik-cache
dongtaifabiao.txt
ipc_property
mtools2
n1.hprof
pccd.pid
pcchannel
test1.jar
uidump.xml
shell@hwRIO-AL00:/data/local/tmp $ _

```

检查完毕后，在运行 `adb shell monkey -f /data/local/tmp/dongtaifabiao.txt --throttle 500`
`2>D:\test.log`

然后就开始了自动发动态的测试了。

注意：逗号跟符号都要用英文的

11. 程序异常出错术语介绍

App Crash :

1. 空指针异常：错误日志定位 `java.lang.NullPointerException`
2. 安全异常：错误日志定位 `java.lang.SecurityException`
3. 无法找到类：错误日志定位 `java.lang.ClassNotFoundException`
4. 动态库加载异常：错误日志定位 `java.lang.UnsatisfiedLinkError`
5. 无效参数错误：错误日志定位 `java.lang.IllegalArgumentException`
6. 调用抽象方法异常：错误日志定位 `java.lang.AbstractMethodError`
7. 违法访问错误：错误日志定位 `java.lang.IllegalAccessError`
8. 越界错误：错误日志定位 `java.lang.IndexOutOfBoundsException`

App ANR :

1. 空指针异常：错误日志定位 `java.lang.NullPointerException`,
2. 按键或触摸事件特定时间内无响应：异常日志定位 `keyDispatchingTimedOut`
3. 广播无响应：异常日志定位 `Broadcastof Intent`

QQ: 519256650