Android Monkey Test

Monkey测试是Android自动化测试的一种手段，Monkey测试本身非常简单，就是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。

当Monkey程序在模拟器或设备运行的时候，如果用户出发了比如点击，触摸，手势或一些系统级别的事件的时候，它就会产生随机脉冲，所以可以用Monkey用随机重复的方法去负荷测试你开发的软件.

最简单的方法就是用用下面的命令来使用Monkey，这个命令将会启动你的软件并且触发500个事件.

$ adb shell monkey -v -p your.package.name 500

更多的关于命令Monkey的命令的信息，可以查看UI/Application

Exerciser Monkey documentation page.

com.android.stk

com.android.htmlviewer

com.android.globalsearch

com.android.gpstest

com.android.googlesearch

wnc.w88.engineermode

com.android.calculator2

com.android.inputmethod.latin

com.svox.pico

com.android.soundrecorder

com.android.packageinstaller

android.tts

com.android.spare\_parts

com.android.globaltime

com.android.sdksetup

com.quicinc.bluetooth

com.android.email

com.qualcomm.qx.neocore

com.android.development

com.android.term

com.android.server.vpn

com.android.contacts

com.android.camera

wnc.w88.factorymode

com.android.providers.applicati

com.android.providers.settings

com.android.voicedialer

com.android.phone

com.android.providers.contacts

com.android.launcher

com.android.mms

com.android.providers.telephony

com.android.providers.userdicti

com.android.providers.media

com.android.providers.calendar

com.android.calendar

com.android.providers.drm

com.android.providers.downloads

com.android.alarmclock

com.android.settings

com.android.qualcomm

com.android.browser

com.android.music

Monkey是一个命令行工具，可以运行在模拟器里或实际设备中。它向系统发送伪随机的用户事件流，实现对正在开发的应用程序进行压力测试。Monkey包括许多选项，它们大致分为四大类：

? 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。

? 调试选项。

在Monkey运行的时候，它生成事件，并把它们发给系统。同时，Monkey还对测试中的系统进行监测，对下列三种情况进行特殊处理：

? 如果限定了Monkey运行在一个或几个特定的包上，那么它会监测试图转到其它包的操作，并对其进行阻止。

? 如果应用程序崩溃或接收到任何失控异常，Monkey将停止并报错。

? 如果应用程序产生了应用程序不响应(application not responding)的错误，Monkey将会停止并报错。

按照选定的不同级别的反馈信息，在Monkey中还可以看到其执行过程报告和生成的事件。

Monkey基本用法 ：

可以通过开发机器上的命令行或脚本来启动Monkey。由于Monkey运行在模拟器/设备环境中，所以必须用其环境中的shell来进行启动。可以通过在每条命令前加上adb shell来达到目的，也可以进入Shell后直接输入Monkey命令。基本语法如下：

$ adb shell monkey [options]

如果不指定options，Monkey将以无反馈模式启动，并把事件任意发送到安装在目标环境中的全部包。下面是一个更为典型的命令行示例，它启动指定的应用程序，并向其发送500个伪随机事件：

$ adb shell monkey -p your.package.name -v 500

命令选项参考：

下表中列出了Monkey命令行可用的全部选项。

类别

选项

说明

常规

--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 。

事件

-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，事件将尽可能快地被产成。

--pct-touch <percent>

调整触摸事件的百分比(触摸事件是一个down-up事件，它发生在屏幕上的某单一位置)。

--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。

--pct-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。

约束限制

-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。

调试

--adb-no-events dos中输入“adb no events”

设置此选项，Monkey将执行初始启动，进入到一个测试Activity，然后不会再进一步生成事件。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接。

monkey [-p ALLOWED\_PACKAGE [-p ALLOWED\_PACKAGE] ...]

[-c MAIN\_CATEGORY [-c MAIN\_CATEGORY] ...]

[--ignore-crashes] [--ignore-timeouts]

[--ignore-security-exceptions] [--monitor-native-crashes]

[--kill-process-after-error] [--hprof]

[--pct-touch PERCENT] [--pct-motion PERCENT]

[--pct-trackball PERCENT] [--pct-syskeys PERCENT]

[--pct-nav PERCENT] [--pct-majornav PERCENT]

[--pct-appswitch PERCENT] [--pct-flip PERCENT]

[--pct-anyevent PERCENT]

[--wait-dbg] [--dbg-no-events] [-f scriptfile]

[--port port]

[-s SEED] [-v [-v] ...] [--throttle MILLISEC]

COUNT

讲了这么多，好像很枯燥，很难懂，即使看完了选项也不知道如何使用。

首先我们应该明白我们不可能使用monkey来制定做一样测试，重复做很多次，因为monkey是伪随机的，就是说虽然有一定规律可循，但是并不是我们就可以指定动作，但是我们可以通过添加命令选项来将操作限制在一定的范围内。

monkey -v -p com.android.camera --throttle 5000 --pct-anyevent 100 500

这条命令的解释是：

-v显示默认程度的信息；

-p com.android.camera是指定测试的程序，// Allowing start of Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.android.camera/.Camera } in package com.android.camera这是开始测试的camera的内容；

--throttle 5000 设定延时；

--pct-anyevent 100设定启动activity的百分比为100%。如果没有指定，我们可以看到在开始执行的时候8个事件（上面的选项中有介绍）的百分比如下：

# monkey -v -p com.android.camera --throttle 5000 500

Adb shell monkey -v -p com.android.camera --throttle 5000 500

DOS下显示情况：

:Monkey: seed=0 count=500

:AllowPackage: com.android.camera

:IncludeCategory: android.intent.category.LAUNCHER

:IncludeCategory: android.intent.category.MONKEY

// Event percentages:

// 0: 15.0%

// 1: 10.0%

// 2: 15.0%

// 3: 25.0%

// 4: 15.0%

// 5: 2.0%

// 6: 2.0%

// 7: 1.0%

// 8: 15.0%

指定事件之后

# monkey -v -p com.android.camera --throttle 5000 --pct-anyevent 100 500

monkey -v -p com.android.camera --throttle 5000 --pct-anyevent 100 500

:Monkey: seed=0 count=500

:AllowPackage: com.android.camera

:IncludeCategory: android.intent.category.LAUNCHER

:IncludeCategory: android.intent.category.MONKEY

// Event percentages:

// 0: 0.0%

// 1: 0.0%

// 2: 0.0%

// 3: 0.0%

// 4: 0.0%

// 5: 0.0%

// 6: 0.0%

// 7: 0.0%

// 8: 100.0%

最后count就是事件数设定为500

接下来看一个实例

Monkey的用法是$ adb shell monkey -p your.package.name -v 500 ，首先，我们要找到应用程序在Emulator中所对应的包名，我一开始是一个个目录找得，前7步完成了这项工作，最终发现应用程序包都在data/data下，你可以在shell中cd data/data，然后ls查看当前Emulator中的所有应用程序包。

假如我们想对SDK中的APIDemos做压力测试，

1.在Eclipse中新建工程，将: \android-sdk-windows\ platforms\android-2.0\samples\ ApiDemos添加到工程中点击运行，此时，他的APK应该加载到了Emulator上

2.在命令行输入adb shell

3.输入ls查看当前文件夹下的目录，执行结果如下

C:\Documents and Settings\Administrator>adb shell

# ls

ls

sqlite\_stmt\_journals

config

cache

sdcard

d

etc

system

sys

sbin

proc

init.rc

init.goldfish.rc

init

default.prop

data

root

dev

4.应用程序包都在data下，我们输入cd data进入data文件夹，如下

# cd data

cd data

5.输入ls查看文件夹下的内容，如下所示：

# ls

ls

misc

local

data

app-private

app

property

anr

backup

dontpanic

dalvik-cache

system

lost+found

6.还有个data，所有的应用程序就在这个data下了，进入这个data，然后输入ls如下：

# cd data

cd data

# ls

ls

com.android.mms

com.android.providers.applications

com.android.globalsearch

com.android.calculator2

com.android.spare\_parts

com.android.gesture.builder

com.android.music

com.android.sdksetup

com.android.packageinstaller

com.android.fallback

com.android.providers.settings

com.android.providers.drm

com.android.development

com.android.providers.telephony

com.android.inputmethod.pinyin

com.android.htmlviewer

com.android.settings

com.android.netspeed

com.android.providers.userdictionary

com.android.browser

com.android.contacts

com.android.alarmclock

com.android.camera

com.android.providers.contacts

jp.co.omronsoft.openwnn

com.android.launcher

com.android.phone

com.android.soundrecorder

com.google.android.providers.enhancedgooglesearch

com.svox.pico

com.android.customlocale

com.android.term

com.android.providers.downloads

com.android.providers.media

com.android.email

android.tts

com.android.inputmethod.latin

com.android.server.vpn

com.example.android.apis

7.最后一个就是我们刚才加载的APIDemo的应用程序包

8.可以直接输入monkey -p com.example.android.apis -v 50，结果如下，也可以退出shell，在命令行输入，adb shell monkey -p com.example.android.apis -v 500 ，运行过程中，Emulator中的应用程序在不断地切换画面，可以看一下，呵呵：）

# monkey -p com.example.android.apis -v 50

monkey -p com.example.android.apis -v 50

:Monkey: seed=0 count=50

:AllowPackage: com.example.android.apis

:IncludeCategory: android.intent.category.LAUNCHER

:IncludeCategory: android.intent.category.MONKEY

// Event percentages:

// 0: 15.0%

// 1: 10.0%

// 2: 15.0%

// 3: 25.0%

// 4: 15.0%

// 5: 2.0%

// 6: 2.0%

// 7: 1.0%

// 8: 15.0%

:Switch: #Intent;action=android.intent.action.MAIN;category=android.intent.categ

ory.LAUNCHER;launchFlags=0x10000000;component=com.example.android.apis/.ApiDemos

;end

// Allowing start of Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.example.android.apis/.ApiDemos } in package com.

example.android.apis

:Sending Pointer ACTION\_MOVE x=-4.0 y=2.0

:Sending Pointer ACTION\_UP x=0.0 y=0.0

// Allowing start of Intent { cmp=com.example.android.apis/.ApiDemos } in pa

ckage com.example.android.apis

:Sending Pointer ACTION\_DOWN x=207.0 y=282.0

:Sending Pointer ACTION\_UP x=189.0 y=289.0

// Allowing start of Intent { cmp=com.example.android.apis/.app.Intents } in

package com.example.android.apis

:Sending Pointer ACTION\_DOWN x=95.0 y=259.0

:Sending Pointer ACTION\_UP x=95.0 y=259.0

:Sending Pointer ACTION\_DOWN x=295.0 y=223.0

:Sending Pointer ACTION\_UP x=290.0 y=213.0

:Sending Pointer ACTION\_MOVE x=-5.0 y=3.0

:Dropped: keys=0 pointers=0 trackballs=0 flips=0

## Network stats: elapsed time=3799ms (3799ms mobile, 0ms wifi, 0ms not connecte

d)