

Android手机自动化 测试交流



2013年10月

范围与目标

- + 交流对象：
 - 手机端软件测试人员
- + 交流目标：
 - 自动化测试背景
 - 自动化工具介绍
 - 自动化测试环境安装
 - Monkey runner 测试实例
- + 本次交流将占用100分钟

自动化测试背景

软件测试是软件产品开发过程中的重要环节。

测试对于发现软件中缺陷，保障软件的质量具有重要意义。

手工测试和自动化测试是软件测试的两种基本手段，二者相互补充，不可相互替代。

自动化测试能捕捉到资源冲突、多线程、死锁等手工测试难以发现的错误，可以高效的完成大量机械性、重复性测试，能模拟大量数据或大量并发用户等应用场景，从而有效的缩短软件测试周期，提高测试的可靠性及准确度。

自动化工具介绍

研究Android自动化测试方法，整理了一些工具，[Monkey](#)、[Monkeyrunner](#)等等。

目前看来, monkey这个程序其实并不完善, 不过, 作为折腾应用的使用还是足够的

接下来重点讲就是Monkey-->及进化的MonkeyRunner

自动化工具介绍

► Monkey

Monkey是**Android**中的一个命令行工具，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(如模拟用户触摸屏幕、滑动Trackball、按键等)，实现对正在开发的应用程序进行压力测试。**Monkey**测试是一种为了测试软件的稳定性、健壮性的快速有效的方法。

► MonkeyRunner

如果,把现阶段的**monkey**比做是幼儿园的小孩,那么**monkeyrunner**就是一个初中生了...它支持自己编写插件,控制事件,随时截图,简而言之,任何你在模拟器/设备中能做的事情,**MonkeyRunner**都能干,而且还可以记录和回放!!!

自动化测试环境安装

安装SDK (Software Development Kit, 即软件开发工具包)



下载解压后，勿忘添加环境变量：

Path = D:\android-sdk\tools

Monkey runner测试实例

Android自动化测试之Monkeyrunner的使用，是模拟人工的按键、触击屏幕、滑动屏幕、时延等操作来控制Android手机来达到自动化测试的效果，对比monkey功能要强大的多!它支持,自己编写插件,控制事件,随时截图,简而言之,任何你在模拟器/设备中能做的事情,MonkeyRunner都能干,而且还可以记录和回放!!!

1. 启动一个Android系统的模拟器，即为命令中device
2. 将cmd命令定位到android SDK的路径下的tools目录
3. 输入命令"monkeyrunner"，回车，进入Shell命令交互方式

导入monkeyrunner所要使用的模块：

A.直接在shell命令中输入（然后回车）：

```
from com.android.monkeyrunner import MonkeyRunner,  
MonkeyDevice,MonkeyImage
```

B.输入device = MonkeyRunner.waitForConnection()，然后回车

4. 到这儿，基本上算是完成了，至于参考文档中提到的传入apk则是为了对具体应用的控制；如果不传入则可以实时的控制模拟器，像您在玩自己的手机一样。
5. 确实，我们可以不必这样麻烦的一行行的输入命令，可以将这些指令都写到一个Python文件中

Monkey runner测试实例

```
C:\ 命令提示符 - monkeyrunner.bat
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\zhangwf>d:

D:\>cd D:\android-sdk\tools

D:\android-sdk\tools>monkeyrunner.bat
Jython 2.5.0 (Release_2_5_0:6476, Jun 16 2009, 13:33:26)
[Java HotSpot(TM) Client VM (Sun Microsystems Inc.)] on java1.6.0_07
>>>     from com.android.monkeyrunner import MonkeyRunner,
File "<stdin>", line 1
    from com.android.monkeyrunner import MonkeyRunner,
    ^
SyntaxError: no viable alternative at input '
>>> from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice,MonkeyImage
>>> device = MonkeyRunner.waitForConnection()
>>> device.press('KEYCODE_BACK',MonkeyDevice.DOWN_AND_UP)
>>> device.touch(201,740,MonkeyDevice.DOWN_AND_UP)
>>>
>>>
```


Monkey runner测试实例

```
0 10 20 30 40 50 60
1 #!D:\android-sdk\tools\monkeyrunner
2 from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice,I
3
4 # Connects to the current device, returning a MonkeyDevice object
5 device = MonkeyRunner.waitForConnection()
6
7 for i in range(0,3):
8     #device.touch(201,733,MonkeyDevice.DOWN_AND_UP)
9     #MonkeyRunner.sleep(20)
10    #device.press("KEYCODE_BACK",MonkeyDevice.DOWN_AND_UP)
11    #device.press('KEYCODE_BACK','DOWN_AND_UP')
12
13    device.touch(201,740,MonkeyDevice.DOWN_AND_UP)
14    MonkeyRunner.sleep(20)
15    device.press('KEYCODE_BACK',MonkeyDevice.DOWN_AND_UP)
16    MonkeyRunner.sleep(10)
17    #device.press('KEYCODE_BACK',MonkeyDevice.DOWN_AND_UP)
18    #device.press('KEYCODE_BACK','DOWN_AND_UP')
19
```

运行python
脚本实现自动
化测试

Monkey runner测试实例

```
0          10
1 #!D:\android-sdk\tools\monkeyrunner.bat
2 from com.android.monkeyrunner import MonkeyRunner, MonkeyDevice
3
4 # Connects to the first available device = MonkeyRunner.getDefault().
5 device = MonkeyRunner.getDefault().connectToDevice()
6
7 for i in range(0, 10):
8     #device.touch(200, 200, 0)
9     #MonkeyRunner.sleep(1)
10    #device.press(KeyEvent.KEYCODE_BACK)
11    #device.press(KeyEvent.KEYCODE_BACK)
12
13    device.touch(200, 200, 0)
14    MonkeyRunner.sleep(1)
15    device.press(KeyEvent.KEYCODE_BACK)
16    MonkeyRunner.sleep(1)
17    #device.press(KeyEvent.KEYCODE_BACK)
```

C:\ 命令提示符
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.
C:\Documents and Settings\zhangwf>d:
D:\>cd android-sdk\tools
D:\android-sdk\tools>monkeyrunner.bat demo.py
D:\android-sdk\tools>

Monkey runner测试实例

```
0 10 20 30 40 50 60
1 #!#!D:\android-sdk\tools\monkeyrunner
2 from com.android.monkeyrunner import MonkeyRunner, MonkeyDevice
3 from com.android.monkeyrunner.recorder import MonkeyRecorder
4
5 # Connects to the current device, returning a MonkeyDevice object
6 device = MonkeyRunner.waitForConnection()
7 MonkeyRecorder.start(device)
8 |
```

运行

recorder获取手机界面上按钮的坐标

Monkey runner测试实例

The image shows a screenshot of an Android emulator interface. At the top, there is a status bar with icons for signal, battery, and time (9:00). Below it is a search bar with the text "搜索". A dialog box titled "添加新商品" (Add New Product) is open, featuring a search bar with "搜索" and a list area. A virtual keyboard is visible at the bottom of the emulator.

Overlaid on the right side of the emulator is a Windows command prompt window titled "命令提示符 - monkeyrunner.bat recorder.py". The command prompt shows the following text:

```
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.
C:\Documents and Settings\zhangwf>d:
D:\>cd android-sdk\tools
D:\android-sdk\tools>monkeyrunner.bat demo.py
D:\android-sdk\tools>monkeyrunner.bat recorder.py
D:\android-sdk\tools>monkeyrunner.bat recorder.py
```

Below the command prompt, a code editor window shows the following code:

```
7 MonkeyRecorder.start(device)
8
```


Monkey runner测试实例

```
#!D:\android-sdk\tools\monkeyrunner
from com.android.monkeyrunner import

# Connects to the current device, re
device = MonkeyRunner.waitForConnect
device.touch(50, 1230, MonkeyDevice.
MonkeyRunner.sleep(2)
# Takes a screenshot
result = device.takeSnapshot()
result.writeToFile('d:\\snap1.png',
```



运行的结果截图保存D:\文件夹下

Monkey runner测试实例

L...	Time	PID	Application	Tag	Text
D	01-10 09:05:05.125	2108		dalvikvm	GC_EXPLICIT freed 11K, 49% fr
D	01-10 09:05:07.578	1778		FastDormancy	before ===== ENTER DORMANC
E	01-10 09:05:08.398	1635		AlarmManager...	android_server_AlarmManagerSe
V	01-10 09:05:08.398	1635			
V	01-10 09:05:08.421	1635			
I	01-10 09:05:08.445	1755			
E	01-10 09:05:08.492	1528			
I	01-10 09:05:09.453	1755			
E	01-10 09:05:10.351	1528			
V	01-10 09:05:10.523	1635			
I	01-10 09:05:11.453	1755			
I	01-10 09:05:12.460	1755			
D	01-10 09:05:13.531	2012			
D	01-10 09:05:15.585	1778			
D	01-10 09:05:15.585	1778		FastDormancy	[FD] ON default: true
D	01-10 09:05:15.585	1778		FastDormancy	Before mDormFlag: true in get
D	01-10 09:05:15.593	1778		FastDormancy	After mDormFlag: true in get
W	01-10 09:05:15.593	1778		FastDormancy	===== ENTER DORMANCY =====
D	01-10 09:05:15.640	1778		FastDormancy	===== ENTER DORMANCY =====
E	01-10 09:05:19.859	1635		AlarmManager...	andro
V	01-10 09:05:19.859	1635		AlarmManager	Adding
V	01-10 09:05:19.859	1635		AlarmManager	sending
D	01-10 09:05:19.875	2317		cpush	push on
D	01-10 09:05:19.882	2317		cpush	push onStart(): enter, messag
D	01-10 09:05:19.882	2317		cpush	onHandleIntent(): enter
D	01-10 09:05:19.882	2317		cpush	parseIntent(): enter
V	01-10 09:05:19.882	2317		cpush	action: null
E	01-10 09:05:19.882	2317		cpush	username = jlf_980c8248cecl
E	01-10 09:05:19.882	2317		cpush	password = jlf
I	01-10 09:05:19.882	2317		cpush	isconnected=true
E	01-10 09:05:19.882	2317		cpush	no connected again

```
01-10 09:05:08.398: E/AlarmManagerService(1635):  
android_server_AlarmManagerService_set to type=2, 6885.92000000
```

使用Logcat
查看手机Log

欢迎大家提问

Any Question ?

