

Monkey 调研报告

李俞乐 19335111

一、Monkey测试的技术方案

- 工作原理

- Monkey 测试原理就是利用 socket 通讯的方式来模拟用户的按键输入、触摸屏输入、手势输入等事件，并查看设备多长时间会出现异常，当 Monkey 程序在设备上运行时，可以模拟发送随机重复事件去负荷测试被测软件，实现对正在开发的应用程序进行压力测试。
- Monkey 测试会对测试中的系统进行监测，默认状态下会对下列三种情况进行特殊处理：
 - (1) 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；
 - (2) 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；
 - (3) 如果应用程序产生了应用程序不响应 ANR(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。

- ADB 命令参数

安卓调试桥 (Android DeBug Bridge, ADB),是一种可以用来操作手机设备或模拟器的命令行工具，实验中将在adb shell 环境下来完成 Monkey 测试

- 基础参数

- 最简单的 Monkey 命令：`Monkey 1000`，表示向测试系统发送1000个伪随机事件

- -v 参数

用于指定反馈信息级别，即测试日志的详细程度，共分为3级，默认为 -v

-v: 只提供启动提示、测试完成提示和最终结果等少量信息

-v -v: 较为详细的日志，包括每个发送到 Activity 的时间信息

-v -v -v: 最详细的日志，包括测试中选中/未选中的 Activity 信息

- -p 参数

由于Monkey测试是针对某个特定的应用进行操作的，因此需要为Monkey指定一个安装包的名字，而参数-p的作用就是由于指定安装包，使用方式为 `-p packagename`。若没有指定安装包，则Monkey会在测试设备上随机启动应用进行测试；若需要同时测试多个应用，则可以 `-p packagename1 -p packagename2` 的方式来指定应用

- -s 参数

-s 参数表示输入的种子值 (seed)，由于Monkey向测试设备中发送的是伪随机的事件序列，而 seed 则是用于产生一个特定的伪随机事件序列，相同的 seed 值将会产生相同的事件序列，因此测试时需要记录每一次的 seed 值，方便后续的重复验证。

- --throttle 参数

用于设置执行操作的延迟时间（毫秒），即两个事件的发送时间间隔。如果没有指定延迟时间，则Monkey将会以最快的速度生成和发送时间

○ 调试选项

■ --ignore-crashes

用于指定当应用程序崩溃时，Monkey是否停止运行。若使用此参数，Monkey在发送事件的过程中将会忽略应用程序崩溃，直至事件计数完成

■ ---ignore-timeouts

用于指定当应用程序发生程序未响应（ANR）错误时，Monkey是否停止运行。若使用此参数，Monkey在发送事件的过程中将会忽略应用程序发生的 ANR 错误，直至事件计数完成

■ --ignore-security-exceptions

用于指定当应用程序发生许可错误时（如证书许可、网络许可等），Monkey是否停止运行。若使用此参数，Monkey在发送事件的过程中将会忽略应用程序发生的许可错误，直至事件计数完成

■ --kill-process-after-error

用于指定当应用程序发生错误时，是否停止其运行。若使用此参数，则当应用程序发生错误时，应用程序停止运行并保持在发生错误时的状态（系统并不会结束该应用程序的进程）

■ --monitor-native-crashes

用于指定是否监视并报告应用程序发生崩溃的本地代码

■ --hprof

该选项设置后，将会在 Monkey 事件序列前后立刻生成report

○ 事件类型

Monkey 发送的伪随机事件序列中主要可以分为11类事件，默认情况下各类事件地比例是随机分配的，也可以为各类事件指定百分比

- --pct-touch 触摸事件
- --pct-motion 滑动事件
- --pct-pinchzoom 缩放事件
- --pct-trackball 轨迹球事件
- --pct-rotation 屏幕旋转事件
- --pct-nav 基本导航事件
- --pct-majornav 主要导航事件
- --pct-flip 键盘事件
- --pct-appswitch 切换 Activity 事件
- --pct-syskeys 系统按键事件
- --pct-anyevent 其他类型事件（除以上事件外的其他事件）

• Monkey 命令示例

```
1 adb shell monkey -p com.nox.game.mla --throttle 100 --pct-touch 30 --pct-motion 30 --ignore-crashes --ignore-timeouts --ignore-security-exceptions --ignore-native-crashes --monitor-native-crashes -v -v -v -s 1234 100 >d:\log.txt
```

二、实现Monke测试的具体过程

- 实验环境

实验中使用 Android SDK 工具，需要自行下载安装并完成系统环境变量的配置，设备平台为夜神模拟器上创建的 Android 7 (64bit) 系统，目标应用为游戏《超级玛丽》

- 模拟器配置

- 将模拟器的分辨率和窗口大小设置为手机版
- 在模拟器的设置 -> 关于平板电脑中多次点击版本号开启开发者模式，并在开发者选项中打开 USB 调试选项



- 在模拟器下载安装好游戏《超级玛丽》



- Monkey 测试具体过程

- 在 windows 的命令中使用 `adb connect 127.0.0.1:62001` 命令连接模拟器（62001是夜神模拟器的默认端口），连接完成后可以使用 `adb devices` 命令查看已连接的设备

```
PS C:\Users\Lenovo> adb connect 127.0.0.1:62001
connected to 127.0.0.1:62001
PS C:\Users\Lenovo> adb devices
List of devices attached
127.0.0.1:62001 device
```

- 使用 `adb shell pm list package` 命令查看设备上的所有安装包，也可以通过 `adb shell dumpsys activity activities | findstr mResumedActivity` 命令来获取正在运行的应用程序的安装包名字

- 查看设备上所有安装包的名字

```
PS C:\Users\Lenovo> adb shell pm list package
package:com.android.cts.priv.ctsshim
package:com.android.providers.telephony
package:com.android.providers.calendar
package:com.android.providers.media
package:com.android.wallpapercropper
package:com.android.documentsui
package:com.android.externalstorage
package:com.android.htmlviewer
package:com.android.mms.service
package:com.android.providers.downloads
package:com.android.browser
package:com.android.inputmethod.pinyin
package:com.android.defcontainer
package:com.android.certinstaller
package:com.android.carrierconfig
package:android
package:com.android.contacts
package:com.android.camera2
package:com.nox.game.mla
package:com.android.egg
package:com.android.mtp
package:com.android.launcher3
package:com.android.backupconfirm
package:com.android.provision
package:com.android.statementservice
package:com.android.Calendar
package:com.android.providers.settings
package:com.android.sharedstoragebackup
package:com.android.printspooler
package:com.android.dreams.basic
package:com.android.inputdevices
package:com.google.android.webview
package:android.ext.shared
package:com.android.onetimeinitializer
package:com.android.server.telecom
package:com.android.keychain
package:com.android.gallery3d
package:android.ext.services
```

- 查看设备上正在运行的应用程序

```
PS C:\Users\Lenovo> adb shell dumpsys activity activities | findstr mResumedActivity
mResumedActivity: ActivityRecord{e3ec09b u0 com.nox.game.mla/com.nox.game.fc.FcEmulatorActivity t7}
PS C:\Users\Lenovo> |
```

- 分别使用以下命令执行 Monkey 测试，测试信息将会被写入到测试日志中

- 未指定事件百分比

```
PS C:\Users\Lenovo> adb shell monkey -p com.nox.game.mla --throttle 100 --ignore-crashes --ignore-timeouts --ignore-security-exceptions --ignore-native-crashes --monitor-native-crashes -v -v -s 1234 1000 >d:\log.txt
```

- 指定事件百分比

```
PS C:\Users\Lenovo> adb shell monkey -p com.nox.game.mla --throttle 100 --pct-touch 30 --pct-motion 30 --ignore-crashes --ignore-timeouts --ignore-security-exceptions --ignore-native-crashes --monitor-native-crashes -v -v -s 1234 1000 >d:\log2.txt
```

○ 测试日志分析

测试日志记录了 seed 值、测试事件数、各类事件的百分比、延迟时间以及Monkey发送的伪随机事件序列，若在测试过程中程序出现错误，日志中也会有具体的记录，可以在日志搜索 ANR、CRASH 或者 EXCEPTION 等关键词来快速定位程序发生错误的位置。

■ 未指定事件百分比的测试日志

```
log.txt - 记事本
文件(E)  编辑(E)  格式(O)  查看(V)  帮助(H)
|Monkey: seed=1234 count=1000
:AllowPackage: com.nox.game.mla
:IncludeCategory: android.intent.category.LAUNCHER
:IncludeCategory: android.intent.category.MONKEY
// Selecting main activities from category android.intent.category.LAUNCHER
// - NOT USING main activity com.android.browser.BrowserActivity (from package com.android.browser)
// - NOT USING main activity com.android.camera.CameraLauncher (from package com.android.camera2)
// - NOT USING main activity com.android.gallery3d.app.GalleryActivity (from package com.android.gallery3d)
// - NOT USING main activity com.android.settings.Settings (from package com.android.settings)
// - NOT USING main activity com.amaze.filemanager.activities.MainActivity (from package com.amaze.filemanager)
// - NOT USING main activity com.android.calendar.ui.activity.MainActivity (from package com.android.calendar)
// - NOT USING main activity com.android.calculator2.MainActivity (from package com.android.calculator2)
// - NOT USING main activity com.android.deskclock.MainActivity (from package com.android.deskclock)
// - NOT USING main activity com.android.documentsui.LauncherActivity (from package com.android.documentsui)
// + Using main activity com.qihoo.util.StartActivity (from package com.nox.game.mla)
// Selecting main activities from category android.intent.category.MONKEY
// - NOT USING main activity com.android.launcher3.launcher3.Launcher (from package com.android.launcher3)
// - NOT USING main activity com.android.settings.Settings$RunningServicesActivity (from package com.android.settings)
// - NOT USING main activity com.android.settings.Settings$StorageUseActivity (from package com.android.settings)
// Seeded: 1234
// Event percentages:
// 0: 15.0%
// 1: 10.0%
// 2: 2.0%
// 3: 15.0%
// 4: -0.0%
// 5: -0.0%
// 6: 25.0%
// 7: 15.0%
// 8: 2.0%
// 9: 2.0%
// 10: 1.0%
// 11: 13.0%
:Switch: #Intent:action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10200000;component=com.nox.game.mla/com.qihoo.util.StartActivity;end
// Allowing start of Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.nox.game.mla/com.qihoo.util.StartActivity } in package com.nox.game.mla
Sleeping for 100 milliseconds
:Sending Touch (ACTION_DOWN): 0:(1133.0,720.0)
:Sending Touch (ACTION_POINTER_DOWN 1): 0:(1136.6451,719.86176) 1:(529.0,49.0)
:Sending Touch (ACTION_MOVE): 0:(1144.796,714.21783) 1:(528.1555,29.714542)
:Sending Touch (ACTION_MOVE): 0:(1151.8645,714.07245) 1:(526.16943,21.434278)
:Sending Touch (ACTION_MOVE): 0:(1160.4515,713.4345) 1:(519.3396,20.424614)
:Sending Touch (ACTION_MOVE): 0:(1161.0051,711.2358) 1:(519.1815,4.967703)
:Sending Touch (ACTION_MOVE): 0:(1166.4049,708.8342) 1:(512.75354,0.0)
:Sending Touch (ACTION_MOVE): 0:(1182.1748,705.2427) 1:(508.97098,0.0)
```

■ 指定事件百分比的测试日志

```
log2.txt - 记事本
文件(E)  编辑(E)  格式(O)  查看(V)  帮助(H)
|Monkey: seed=1234 count=1000
:AllowPackage: com.nox.game.mla
:IncludeCategory: android.intent.category.LAUNCHER
:IncludeCategory: android.intent.category.MONKEY
// Selecting main activities from category android.intent.category.LAUNCHER
// - NOT USING main activity com.android.browser.BrowserActivity (from package com.android.browser)
// - NOT USING main activity com.android.camera.CameraLauncher (from package com.android.camera2)
// - NOT USING main activity com.android.gallery3d.app.GalleryActivity (from package com.android.gallery3d)
// - NOT USING main activity com.android.settings.Settings (from package com.android.settings)
// - NOT USING main activity com.amaze.filemanager.activities.MainActivity (from package com.amaze.filemanager)
// - NOT USING main activity com.android.calendar.ui.activity.MainActivity (from package com.android.calendar)
// - NOT USING main activity com.android.calculator2.MainActivity (from package com.android.calculator2)
// - NOT USING main activity com.android.deskclock.MainActivity (from package com.android.deskclock)
// - NOT USING main activity com.android.documentsui.LauncherActivity (from package com.android.documentsui)
// + Using main activity com.qihoo.util.StartActivity (from package com.nox.game.mla)
// Selecting main activities from category android.intent.category.MONKEY
// - NOT USING main activity com.android.launcher3.launcher3.Launcher (from package com.android.launcher3)
// - NOT USING main activity com.android.settings.Settings$RunningServicesActivity (from package com.android.settings)
// - NOT USING main activity com.android.settings.Settings$StorageUseActivity (from package com.android.settings)
// Seeded: 1234
// Event percentages:
// 0: 30.0%
// 1: 30.0%
// 2: 1.0666667%
// 3: 8.0%
// 4: -0.0%
// 5: -0.0%
// 6: 13.333334%
// 7: 8.0%
// 8: 1.0666667%
// 9: 1.0666667%
// 10: 0.53333336%
// 11: 6.933334%
:Switch: #Intent:action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10200000;component=com.nox.game.mla/com.
```

三、常规Monkey测试的缺陷

- 只能进行有限的基本测试，并且常规 Monkey 测试使用的事件数据流是随机的，不能进行自定义
- 常规 Monkey 的测试对象是针对整个 Android 系统或者某个应用，但是无法针对应用里的某个模块或者界面。这样在产品迭代过程中存在的一个局限是：每个迭代时做的稳定性测试都重新做一遍，而无法针对该迭代新增的功能模块做单独的稳定性测试
- 传统 Monkey工具的操作是随机的，到达深层界面的概率极低，这样可能导致一些用户路径操作不到，即会出现遍历界面有限的问题
- 由于Monkey测试具有随机性且由 Monkey 生成的测试序列不容易以文档的形式描述用例，因此 Monkey 测试过程中发现的 Bug 难以重现，即无法准确得知Bug的复现步骤
- Monkey测试的随机性会导致测试过程中出现大量的无效操作，同时也使得传统的 Monkey 测试只能作为稳定性测试工具，难以在其上进行功能扩展。
- 在 Monkey 测试中，由于用户事件的随机性，可能会导致 Monkey 卡在某些简单的页面，也有可能会出现路径回环，导致。另外，在一些可操作内容较少或者界面控件较少且控件位置较偏的页面上，Monkey测试效果不佳
- Monkey 测试可能需要持续很长时间才能发现 Bug，并且Monkey 测试有可能会发现业务范围之外的 Bug，分析起来也很耗时

四、智能Monkey测试

4.1、思路和方法

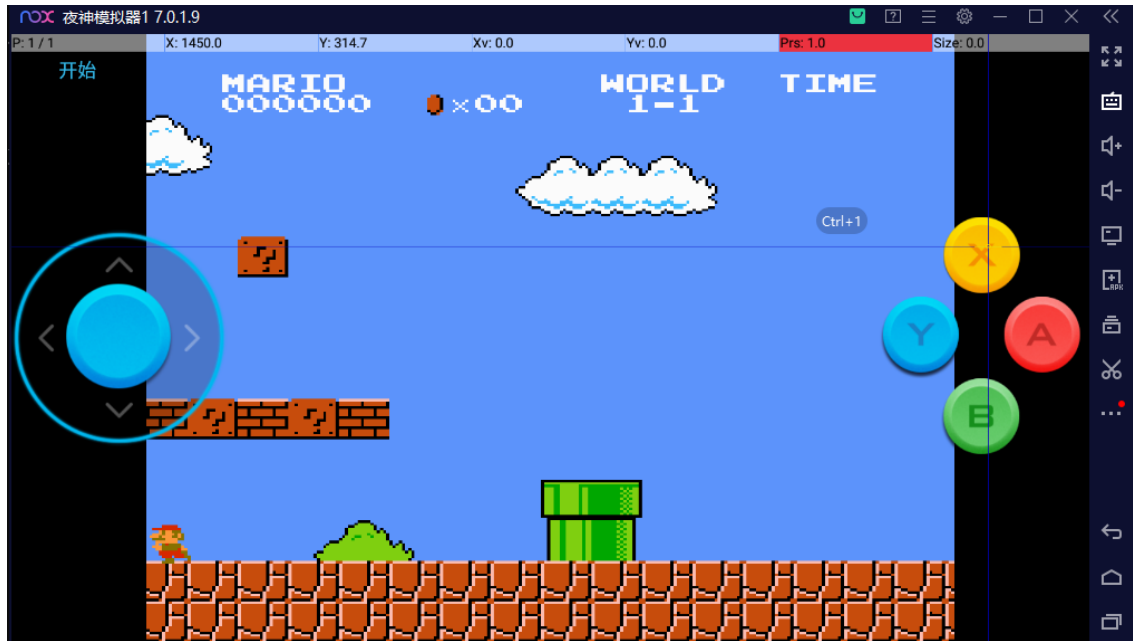
常规 Monkey 测试中会生成一个伪随机的事件序列，但是其中大部分事件对于目标应用来说都是无效操作并且伪随机事件序列达到深层的概率极低，另外，常规 Monkey 测试的操作对象是某个应用，无法针对应用中的特定界面进行测试。因此，我们可以使用 MonkeyScript 来编写自定义 Monkey 测试脚本，使得 Monkey 可以针对特定界面发送指定操作事件进行稳定性测试。

4.2、实现智能Monkey测试

- 脚本是针对应用的特定界面进行测试，因此需要确定应用的主界面名，主界面名的获取方法如下
 - 命令行依次输入命令 `adb logcat -c` 和 `adb logcat ActivityManager:I *:s`
 - 在模拟器上手动打开测试应用程序
 - 在命令行的输出中可以找到应用的主界面名

```
PS C:\Users\Lenovo> adb logcat -c
PS C:\Users\Lenovo> adb logcat ActivityManager:I *:s
----- beginning of system
12-19 13:40:46.145 1644 1833 I ActivityManager: START u0 {act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x10200000 cmp=com.nox
.game.mla/com.qihoo.util.StartActivity bnds=[18,652][162,912] (has extras)} from uid 1000 on display 0
----- beginning of main
12-19 13:40:46.183 1644 3220 I ActivityManager: Config changes=480 {1.0 460mcc7mnc [zh_CN] ldltr sm450dp w800dp h426dp 320dpi nrm1 long land finger -keyb/
v/h -nav/h s.11}
12-19 13:40:46.221 1644 3303 I ActivityManager: START u0 {act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x10000000 cmp=com.nox
.game.mla/com.nox.game.fc.FcEmulatorActivity bnds=[18,652][162,912] (has extras)} from uid 1000 on display 0
12-19 13:40:46.785 1644 1665 I ActivityManager: Displayed com.nox.game.mla/com.nox.game.fc.FcEmulatorActivity: +435ms (total +549ms)
```


- 脚本中的点击事件、拖拽事件、点击并长按事件等都需要指定具体的坐标值，则在编写脚本时需要确定界面中有效控件的坐标。夜神模拟器中可以在设置 -> 开发者选项 -> 打开指针位置，设置后在模拟器中任意位置点击都会显示点击位置的具体坐标值，因此，我们可以通过依次点击当前界面中的各个控件来得到其坐标



• MonkeyScript 脚本

```

1  type=user
2  count=10
3  speed=1.0
4  start data >>
5  #启动界面
6  LaunchActivity(com.nox.game.mla, com.nox.game.fc.FcEmulatorActivity)
7  UserWait(3000)
8  DispatchPointer(0,0,0,1440,324,0,0,0,0,0,0,0) #屏幕点击事件
9  DispatchPointer(0,0,1,1440,324,0,0,0,0,0,0,0)
10 UserWait(500) #等待500毫秒
11 DispatchPointer(0,0,0,1523,447,0,0,0,0,0,0,0)
12 DispatchPointer(0,0,1,1523,447,0,0,0,0,0,0,0)
13 UserWait(500)
14 DispatchPointer(0,0,0,1344,438,0,0,0,0,0,0,0)
15 DispatchPointer(0,0,1,1344,438,0,0,0,0,0,0,0)
16 UserWait(500)
17 DispatchPointer(0,0,0,1439,563,0,0,0,0,0,0,0)
18 DispatchPointer(0,0,1,1439,563,0,0,0,0,0,0,0)
19 UserWait(500)
20 DispatchPointer(0,0,0,100,50,0,0,0,0,0,0,0)
21 DispatchPointer(0,0,1,100,50,0,0,0,0,0,0,0)
22 UserWait(500)
23 RotateScreen(2,1) #屏幕旋转180度
24 UserWait(500)
25 LongPress() #长按
26 UserWait(500)
27 PressAndHold(1440, 324, 10) #点击并长按

```



```

28 PressAndHold(1523, 447, 10)
29 PressAndHold(1344, 438, 10)
30 PressAndHold(1439, 563, 10)
31 UserWait(500)
32 Drag(158,450,150,311,500)    #模拟按住并拖拽
33 UserWait(500)
34 Drag(158,450,10,450,500)
35 UserWait(500)
36 Drag(158,450,160,600,500)
37 UserWait(500)
38 Drag(158,450,303,448,500)
39 UserWait(500)
40 DispatchPress(4)    #连续点击两次返回键
41 DispatchPress(4)

```

- 脚本运行方法

```

1  #在测试系统中创建脚本文件夹
2  adb shell mkdir /sdcard/script
3  #将脚本文件push到文件夹中
4  adb push test_search /sdcard/script
5  #运行文件中的脚本文件，可以指定脚本运行次数
6  adb shell Monkey -f /sdcard/script/test_search 100 >d:\script.log

```

```

PS C:\Users\Lenovo> adb shell mkdir /sdcard/script
PS C:\Users\Lenovo> adb push D:\script /sdcard/script
D:\script: 1 file pushed, 0 skipped. 0.7 MB/s (1014 bytes in 0.001s)
PS C:\Users\Lenovo> adb shell monkey -f /sdcard/script/script -v -v 100 >d:\script.log

```

- 分析测试日志

测试日志记录的事件序列与脚本中自定义的事件序列相同，脚本执行结束后日志会有 `Monkey finished` 的记录

```
:Sending Touch (ACTION_MOVE): 0:(293.42883,448.13303)
:Sending Touch (ACTION_MOVE): 0:(293.71884,448.12903)
:Sending Touch (ACTION_MOVE): 0:(294.00885,448.12503)
:Sending Touch (ACTION_MOVE): 0:(294.29886,448.12103)
:Sending Touch (ACTION_MOVE): 0:(294.58887,448.11703)
:Sending Touch (ACTION_MOVE): 0:(294.87888,448.11304)
:Sending Touch (ACTION_MOVE): 0:(295.16888,448.10904)
:Sending Touch (ACTION_MOVE): 0:(295.45889,448.10504)
:Sending Touch (ACTION_MOVE): 0:(295.7489,448.10104)
:Sending Touch (ACTION_MOVE): 0:(296.0389,448.09705)
:Sending Touch (ACTION_MOVE): 0:(296.32892,448.09305)
:Sending Touch (ACTION_MOVE): 0:(296.61893,448.08905)
:Sending Touch (ACTION_MOVE): 0:(296.90894,448.08505)
:Sending Touch (ACTION_MOVE): 0:(297.19894,448.08105)
:Sending Touch (ACTION_MOVE): 0:(297.48895,448.07706)
:Sending Touch (ACTION_MOVE): 0:(297.77896,448.07306)
:Sending Touch (ACTION_MOVE): 0:(298.06897,448.06906)
:Sending Touch (ACTION_MOVE): 0:(298.35898,448.06506)
:Sending Touch (ACTION_MOVE): 0:(298.649,448.06107)
:Sending Touch (ACTION_MOVE): 0:(298.939,448.05707)
:Sending Touch (ACTION_MOVE): 0:(299.229,448.05307)
:Sending Touch (ACTION_MOVE): 0:(299.519,448.04907)
:Sending Touch (ACTION_MOVE): 0:(299.80902,448.04507)
:Sending Touch (ACTION_MOVE): 0:(300.09903,448.04108)
:Sending Touch (ACTION_MOVE): 0:(300.38904,448.03708)
:Sending Touch (ACTION_MOVE): 0:(300.67905,448.03308)
:Sending Touch (ACTION_MOVE): 0:(300.96906,448.02908)
:Sending Touch (ACTION_MOVE): 0:(301.25906,448.0251)
:Sending Touch (ACTION_MOVE): 0:(301.54907,448.0211)
:Sending Touch (ACTION_MOVE): 0:(301.83908,448.0171)
:Sending Touch (ACTION_MOVE): 0:(302.1291,448.0131)
:Sending Touch (ACTION_MOVE): 0:(302.4191,448.0091)
:Sending Touch (ACTION_MOVE): 0:(302.7091,448.0051)
:Sending Touch (ACTION_MOVE): 0:(302.9991,448.0011)
:Sending Touch (ACTION_UP): 0:(302.9991,448.0011)
  // [calendar_time:2021-12-19 13:22:41.532 system_uptime:1271910]
  // Sending event #204600
Sleeping for 0 milliseconds
  // [calendar_time:2021-12-19 13:22:41.532 system_uptime:1271910]
  // Sending event #204600
Events injected: 204600
:Sending rotation degree=0, persist=false
:Dropped: keys=0 pointers=0 trackballs=0 flips=0 rotations=0
## Network stats: elapsed time=1094945ms (0ms mobile, 0ms wifi, 1094945ms not connected)
// Monkey finished
```

工具包链接：

- **Android SDK:** developer.android.com
- **夜神模拟器:** <https://www.yeshen.com/>

参考资料

[Android Monkey压力测试环境搭建及使用 - 每一天，为明天。 - 博客园 \(cnblogs.com\)](#)

[Monkey测试：Monkey的简单使用 - 飘着的石头 - 博客园 \(cnblogs.com\)](#)

[Monkey测试详解 - 榴莲Alice - 博客园 \(cnblogs.com\)](#)

[智能monkey之monkeyscript（一） - Young哥哥 - 博客园 \(cnblogs.com\)](#)

[\(58条消息\) MonkeyScript的简单使用_QXY93的博客-CSDN博客](#)

[Monkeyscript---获取包名主界面名和位置坐标的几种方法 - bling123 - 博客园 \(cnblogs.com\)](#)

[monkey使用脚本编写 - 钟鼎山林 - 博客园 \(cnblogs.com\)](#)