

Monkey工具的使用

本章大纲

- Monkey介绍
- Monkey的基本使用
- Monkey的在项目中的应用

Monkey是什么？

- Monkey是Android系统自带的一个命令行工具，可以运行在Android模拟器里或真实设备中。它可以向被测应用发送伪随机(pseudo-random)的用户事件流(如点击、触摸、手势等)，实现对应用程序进行测试的目的。因此 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。

Monkey是什么？

- Monkey的特征
 - ✓ 测试的对象仅为应用程序包，有一定的局限性。
 - ✓ Monkey测试使用的事件流数据流是随机的，不能进行自定义。
 - ✓ 可对Monkey测试的对象，事件数量，类型，频率等进行设置。

官方文档: <https://developer.android.com/studio/test/Monkey.html>

源码: <https://github.com/aosp->

[mirror/platform_development/tree/ef7ff204e2ad4ba1094db8e9dc891a13c1f7ba31/cmds/monkey/src/com/android/commands/monkey](https://github.com/aosp-mirror/platform_development/tree/ef7ff204e2ad4ba1094db8e9dc891a13c1f7ba31/cmds/monkey/src/com/android/commands/monkey)

Monkey在哪？

- 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：/system/framework/Monkey.jar。
- Monkey.jar程序是由一个名为“Monkey”的Shell脚本来启动执行，shell脚本在Android文件系统中的存放路径 是：
/system/bin/Monkey；这样就可以通过在shell窗口中执行：adb shell Monkey {+命令参数} 来进行Monkey测试。

```
127!root@android:/system/bin # cat monkey
# Script to start "monkey" on the device, which has a very rudimentary
# shell.
#
base=/system
export CLASSPATH=$base/framework/monkey.jar
trap "" HUP
exec app_process $base/bin com.android.commands.monkey.Monkey $*
```

Monkey使用步骤

- 启动模拟器或者连接真机，adb devices验证
- 确定被测应用的包名（adb logcat|findstr
START）
- 使用Monkey发一条测试命令

本章大纲

- Monkey介绍
- Monkey的基本使用
- Monkey的在项目中的应用

如何确定被测应用的包名

- 方式一：问开发
- 方式二：uiautomatorviewer.bat

查看AppData\Local\Android\sdk\tools

- 方式三：aapt dump badging d:\group-612_0-meituan.apk

Monkey语法

- `adb shell monkey -p 包名 整数`

例: `adb shell monkey -p com.android.calendar 1000`

向日历app发送1000次随机事件

- `-p` 包名是指定测试的程序
- `-v` 参数打印信息的详细级别，可以跟多个（0~2级）

默认级别是0: `-v`只打印测试执行发送给被测Activity的事件

级别1: `-v -v` 增加了哪些Activity被选中，将在信息中被输出。

级别2: `-v -v -v` 打印哪些应用已经安装了但是却没有被选中的信息也被输出

Monkey结果分析

```
C:\WINDOWS\system32\cmd.exe
C:\Users\think>adb shell monkey -p com.xisue.zhoumo -v -v --throttle 1000 50
:Monkey: seed=1487767764756 count=50
:AllowPackage: com.xisue.zhoumo
:IncludeCategory: android.intent.category.LAUNCHER
:IncludeCategory: android.intent.category.MONKEY
// Selecting main activities from category android.intent.c
// + Using main activity com.xisue.zhoumo.ui.activity.Sta
age com.xisue.zhoumo>
// Selecting main activities from category android.intent.c
// Seeded: 1487767764756
Event percentages:
0: 15.0%
1: 10.0%
2: 2.0%
3: 15.0%
4: -0.0%
5: 25.0%
6: 15.0%
7: 2.0%
8: 2.0%
9: 1.0%
10: 13.0%
:Switch: #Intent;action=android.intent.action.MAIN;category
ory.LAUNCHER;launchFlags=0x10200000;component=com.xisue.zho
ActivityStart
```

```
public static final int FACTOR_TOUCH = 0;
public static final int FACTOR_MOTION = 1;
public static final int FACTOR_PINCHZOOM = 2;
public static final int FACTOR_TRACKBALL = 3;
public static final int FACTOR_ROTATION = 4;
public static final int FACTOR_PERMISSION = 5;
public static final int FACTOR_NAV = 6;
public static final int FACTOR_MAJORNAV = 7;
public static final int FACTOR_SYSOPS = 8;
public static final int FACTOR_APPSWITCH = 9;
public static final int FACTOR_FLIP = 10;
public static final int FACTOR_ANYTHING = 11;
public static final int FACTORZ_COUNT = 12; //

private static final int GESTURE_TAP = 0;
private static final int GESTURE_DRAG = 1;
private static final int GESTURE_PINCH_OR_ZOOM = 2;
```

Monkey结果分析

- 0: 触摸事件百分比，（一个在屏幕单一位置的按下抬起事件）即参数--pct-touch
- 1: 滑动事件百分比，（先在一个位置手指按下，滑动一段距离后再抬起手指的手势）即参数--pct-motion
- 2: 缩放事件百分比，即参数--pct-pinchzoom
- 3: 轨迹球事件百分比，（一系列随机移动单击事件，**智能手机上已经没有了**，就是类似手柄的方向键一样）即参数--pct-trackball
- 4: 屏幕旋转事件百分比，即参数--pct-rotation
- 5: 基本导航事件百分比，（老手机的上下左右键，**智能机上没有**）即参

Monkey结果分析

- 6: 主要导航事件百分比, (中间按键、回退按键、菜单按键) 即参数`--pct-majornav`
- 7: 系统事件百分比, (如Home, Back, Start Call, End Call, 音量控制) 即参数`--pct-syskeys`
- 8: Activity启动事件百分比, 即参数`--pct-appswitch`
- 9: 键盘翻转事件百分比, (点击输入框, 键盘弹起, 点击输入框以外区域, 键盘收回) 即参数`--pct-flip`
- 10: 其他事件百分比, 即参数`--pct-anyevent`

Monkey结果分析

- 找出Monkey出错的地方
- 查看Monkey里面出错前的一些事件动作，并手动执行该动作
- 若以上步骤不能找出，可以再次执行，注意seed值保持一致
- 注意第一个switch以及其他异常信息

Monkey测试的目的

- ANR问题：在日志中搜索“ANR”
 - Application Not Responding, 在Android上, 如果应用有一段时间不够灵敏, 系统会向用户显示一个对话框。“强制关闭或等待”。
- 崩溃问题：在日志中搜索“Exception”

Monkey的运行时机

- 首轮功能测试通过后
- 下班后夜间进行

本章大纲

- Monkey介绍
- Monkey的基本使用
- Monkey的在项目中的应用

延迟

- `adb shell Monkey -p com.android.calendar -v --throttle 200 1000`
- 向日历app发送1000次随机事件，每次间隔事件为200毫秒
- `--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生成和发送消息。

事件百分比

- `adb shell Monkey -p com.android.calendar -- throttle 200 -pct-touch 10 --pct-motion 20 -s 1234 -v 1000`
- 向日历app发送1000次随机事件，每次间隔事件为200毫秒，其中设定触屏的事件占比为10%，滑动事件占20%
- `--pct-touch` 设置在整个过程中点击操作所占的百分比
- `--pct-motion` 设置滑动在所有操作中的百分比
- `-s` 伪随机数生成器的种子值，前后两次执行将会生成相同的事件序列

事件百分比

- `--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。

项目实测命令

- `adb shell monkey -p 包名 --throttle 500 --ignore-crashes --ignore-timeouts --ignore-security-exceptions --ignore-native-crashes --monitor-native-crashes -v -v -v 1000000 > D:\demo\Monkey1.log`
- 向系统发送1000000 次随机事件，各个随机事件的时间间隔为500 毫秒，测试过程忽略安全、超时、崩溃等异

案例分析

```
:Sending Touch (ACTION_UP): 0:(621.4546, 578.2527)
Sleeping for 0 milliseconds
:Sending Key (ACTION_DOWN): 82      // KEYCODE_MENU
:Sending Key (ACTION_UP): 82      // KEYCODE_MENU
// CRASH: com.example.think.demol123 (pid 10341)
// Short Msg: java.lang.NumberFormatException
// Long Msg: java.lang.NumberFormatException: Invalid int: ""
// Build Label:
Android/samsung/samsung:4.4.2/KOT49H/3.7.3.1019:userdebug/test-keys
// Build Changelist: 3.7.3.1019
// Build Time: 1476849819000
// java.lang.NumberFormatException: Invalid int: ""
//   at java.lang.Integer.invalidInt(Integer.java:137)
//   at java.lang.Integer.parseInt(Integer.java:358)
//   at java.lang.Integer.parseInt(Integer.java:331)
//   at com.example.think.demol123.MainActivity$1.onClick
(MainActivity.java:26)
//   at android.view.View.performClick(View.java:4438)
//   at android.view.View$PerformClick.run(View.java:18422)
//   at android.os.Handler.handleCallback(Handler.java:733)
//   at android.os.Handler.dispatchMessage(Handler.java:95)
//   at android.os.Looper.loop(Looper.java:136)
//   at android.app.ActivityThread.main(ActivityThread.java:5045)
//   at java.lang.reflect.Method.invokeNative(Native Method)
//   at java.lang.reflect.Method.invoke(Method.java:515)
//   at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run
(ZygoteInit.java:779)
```

常见问题

- 连接真机，不稳定
- 一般是usb本身连接不够稳定，可以提升usb的供电量，不要使用一般的连接线。检查是否是接触不良。
- Monkey执行过程中有时候会触碰到通知栏等其他地方，如何避免？
- 官网下载Monkey源代码，进行二次开发。
- Monkey命令的使用需要组合多种参数方式来实现覆盖

MonkeyScript

- MonkeyScript是一组可以被Monkey识别的命令
- MonkeyScript可以完成重复固定的操作

二次开发

- DispatchPointer命令介绍
 - 命令说明：该命令用于向一个坐标点（即x坐标，y坐标）发送手势消息。
- LaunchActivity命令介绍
 - 命令说明：该命令用于启用任意引用的一个活动界面。
- UserWait命令介绍
 - 命令说明：该命令用于让脚本中断一段时间。
- Tap命令介绍
 - 命令说明：该命令用于模拟一次手指单击事件。

二次开发

```
adb push d:\demo\QQ.script
```

```
/data/tmp/qq.script
```

```
adb shell monkey -f /sdcard/qq.ms 1
```

目的:有针对性的进行稳定性测试

附录

- **monkeysript** - 定制化**monkey**流程
- <https://www.cnblogs.com/YatHo/p/7205162.html>