#### 自定义脚本的稳定性测试

常规Monkey测试执行的是随机的事件流，但如果只是想让Monkey测试某个特定场景这时候就需要用到自定义脚本了，Monkey支持执行用户自定义脚本的测试，用户只需要按照Monkey脚本的规范编写好脚本，存放到手机上，启动Monkey通过-f 参数调用脚本即可。

#### 测试案例

启动考研帮app3.1.0，然后跳过升级提示和引导页面，进入到登录界面并输入用户名密码进行登录。

#### 需求分析

* 从用户角度来思考步骤该怎样进行？
* 要操作的元素该如何定位？
* 怎样将操作步骤转化为测试脚本？

##### 获取元素坐标点位置

Monkey脚本只能通过**坐标**的方式来定位点击和移动事件的屏幕位置，这里就需要提前获取坐标信息。获取坐标信息的方法很多，最简单的方法就是打开手机中的开发人员选项，打开“显示指针位置”。随后，在屏幕上的每次操作，在导航栏上都会显示坐标信息。

##### Monkey脚本API简介

LaunchActivity(pkg\_name, cl\_name)：启动应用的Activity。参数：包名和启动的Activity。

Tap(x, y, tapDuration)： 模拟一次手指单击事件。参数：x,y为控件坐标，tapDuration为点击的持续时间，此参数可省略。

UserWait(sleepTime)： 休眠一段时间

DispatchPress(keyName)： 按键。参数： keycode。 RotateScreen(rotationDegree, persist)： 旋转屏幕。 参数：rotationDegree为旋转角度， e.g. 1代表90度；persist表示旋转之后是否固定，0表示旋转后恢复，非0则表示固定不变。

DispatchString(input)： 输入字符串。

DispatchFlip(true/false)： 打开或者关闭软键盘。

PressAndHold(x, y, pressDuration)： 模拟长按事件。

Drag(xStart, yStart, xEnd, yEnd, stepCount)： 用于模拟一个拖拽操作。

PinchZoom(x1Start, y1Start, x1End, y1End, x2Start, y2Start, x2End, y2End, stepCount)： 模拟缩放手势。

LongPress()： 长按2秒。

DeviceWakeUp()： 唤醒屏幕。

PowerLog(power\_log\_type, test\_case\_status)： 模拟电池电量信息。

WriteLog()： 将电池信息写入sd卡。

RunCmd(cmd)： 运行shell命令。

DispatchPointer(downtime,eventTime,action,x,yxpressure,size,metastate,xPrecision,yPrecision,device,edgeFlags)： 向指定位置，发送单个手势。

DispatchPointer(downtime,eventTime,action,x,yxpressure,size,metastate,xPrecision,yPrecision,device,edgeFilags)： 发送按键消息。

LaunchInstrumentation(test\_name,runner\_name)： 运行一个instrumentation测试用例。

DispatchTrackball： 模拟发送轨迹球事件。

ProfileWait： 等待5秒。

StartCaptureFramerate()： 获取帧率。

EndCaptureFramerate(input)： 结束获取帧率。

##### Monkey脚本格式

Monkey脚本主要包含两部分，一部分是头文件信息，一部分是具体的monkey命令。

type = raw events

count = 1

speed = 1.0

//下面为monkey命令

start data >>

具体的monkey脚本内容

##### 编写脚本

kyb.txt

#头文件信息

type = raw events

count = 1

speed = 1.0

#启动测试

start data >>

**LaunchActivity**(com.tal.kaoyan,com.tal.kaoyan.ui.activity.SplashActivity)

**UserWait**(2000)

**Tap**(624,900,1000) #点击取消升级

**UserWait**(2000)

**Tap**(806,64,1000) #点击跳过

**UserWait**(2000)

**Tap**(217,378,1000) #点击用户名输入框

**DispatchString**(zxw1234)

**UserWait**(2000)

**Tap**(197,461,1000) #点击密码输入框

**DispatchString**(zxw123456)

**UserWait**(2000)

**Tap**(343,637,1000) #点击登录按钮

##### 执行脚本

脚本编写完成后，传到手机设备上，然后执行。

adb push C:\Users\Shuqing\Desktop\kyb1.txt /sdcard

adb shell monkey -f /sdcard/kyb1.txt -v 1

执行结果

C:\Users\Shuqing>adb shell monkey -f /sdcard/kyb.txt -v 1

**:Monkey:** seed=1524592021303 count=1

**:IncludeCategory:** android.intent.category.LAUNCHER

**:IncludeCategory:** android.intent.category.MONKEY

Replaying 0 events with speed 1.0

**:Switch:** #Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10200000;component=com.tal.kaoyan/.ui.activity.SplashActivity;end

// Allowing start of Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.tal.kaoyan/.ui.activity.SplashActivity } in package com.tal.kaoyan

**:Sending Touch (ACTION\_DOWN):** 0:(267.0,1233.0)

// Allowing start of Intent { act=com.android.systemui.recent.action.TOGGLE\_RECENTS cmp=com.android.systemui/.recent.RecentsActivity } in package com.android.systemui

**:Sending Touch (ACTION\_UP):** 0:(267.0,1233.0)

Events injected: 5

:Sending rotation degree=0, persist=false

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

## Network stats: elapsed time=7201ms (0ms mobile, 0ms wifi, 7201ms not connected)

*// Monkey finished*

#### 注意事项

头文件代码书写注意“=”两边预留空格，否则会出现如下报错。

java.lang.NumberFormatException: Invalid int: ""

#### 参考资料

<https://www.cnblogs.com/mgzc-1508873480/p/7911373.html>