

Monkey 测试的调研与实践

姓名：郝裕玮

班级：计科 1 班

学号：18329015

目录

1 Monkey 测试基础知识	3
1.1 Monkey 简介	3
1.2 Monkey 参数	4
2 实验环境搭建	7
2.1 硬件环境	7
2.2 软件配置	7
3 Monkey 测试过程	14
3.1 测试步骤	14
3.2 结果分析	16
4 Monkey 测试优缺点	20
5 智能 Monkey 测试	21
5.1 Maxim & UICrawler	21
5.2 Maxim 环境配置	23
5.3 命令行运行 Maxim	23
5.4 分析日志	24
5.5 UICrawler 环境配置	29
5.6 UICrawler 运行结果	31
6 实验感想	32

1 Monkey 测试基础知识

1.1 Monkey 简介

Monkey 测试是 Android 自动化测试 APP 的手段之一。monkey 译为猴子，所以从原理上来说，它的自动化测试就是像一只什么都不懂的猴子一样在软件上乱敲按键。

Monkey 测试是一个命令行工具，它可以运行在模拟器或实际设备中。它通过向系统发送伪随机的用户事件流（如按键输入、触摸屏输入、滑动 Trackball、手势输入等操作），实现对全系统或某个应用程序进行压力测试。

Monkey 工具存在 Android 系统中，使用 Java 语言写成。其 jar 包在 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。

monkey 需要通过 adb 工具来唤醒，即在 cmd 窗口中执行：adb shell monkey {+命令参数} 来进行 Monkey 测试。

1.2 Monkey 参数

(1) -p:

用于指定的一个或多个应用包，当指定包后，monkey 将只允许系统启动指定的 app，如果没有指定，monkey 将允许系统启动设备中的所有 app。

```
#指定 1 个应用
#com.htc.Weather 为包名，100 是事件计数（即让 Monkey 程序模拟 100 次随机用户事件）。
adb shell monkey -p com.htc.Weather 100

#指定多个应用
adb shell monkey -p com.htc.Weather -p com.htc.pdfreader -p com.htc.photo.widgets 100

#不指定应用
#Monkey 随机启动 APP 并发送 100 个随机事件。
adb shell monkey 100
```

(2) -v:

用于指定反馈信息级别，即日志的详细程度，总共有 3 个级别，一般测试的时候选择最高级别获取最详细的日志。

```
#Level0: 仅提供启动提示、测试完成和最终结果等少量信息
adb shell monkey -p com.htc.Weather -v 100

#Level1: 提供较为详细的日志，包括每个发送到 Activity 的事件信息
adb shell monkey -p com.htc.Weather -v -v 100

#Level2: 最详细的日志，包括了测试中选中/未选中的 Activity 信息
adb shell monkey -p com.htc.Weather -v -v -v 100
```

(3) -s:

用于指定 monkey 生成器的 seed 值，一个 seed 值对应一个事件序列，当你使用进行 monkey 测试出现异常后，在日志中提取 seed 值并进行指定，可生成相同的事件序列，有利于 bug 的重现。

```
#测试 1
adb shell monkey -p com.htc.Weather -s 10 100
#测试 2
adb shell monkey -p com.htc.Weather -s 10 100

#两次测试的效果是相同的。
#因为模拟的用户操作序列（每次操作按照一定的先后顺序所组成的一系列操作，即一个序列）是一样的。
#操作序列虽然是随机生成的，但是只要我们指定了相同的 seed 值，就可以保证两次测试产生的随机操作序列是完全相同的。
#所以这个操作序列是伪随机的。
```

(4) --throttle:

用于指定用户操作事件的延迟时间，单位为毫秒，若不指定该参数，则 Monkey 会尽可能快地发送事件。参数缺省值为 300ms。

```
adb shell monkey -p com.htc.Weather --throttle 3000 100
```

(5) --ignore-crashes:

用于指定当应用程序崩溃时（Force & Close 错误），Monkey 是否停止运行。如果使用此参数，即使应用程序崩溃，Monkey 依然会发送事件，直到事件计数完成。

```
#测试过程中即使 Weather 程序崩溃，Monkey 依然会继续发送事件直到事件数目达到 1000 为止
adb shell monkey -p com.htc.Weather --ignore-crashes 1000

#测试过程中，如果 Weather 程序崩溃，Monkey 将会停止运行
adb shell monkey -p com.htc.Weather 1000
```

(6) --ignore-timeouts

用于指定当应用程序发生 ANR (Application No Responding) 错误时, Monkey 是否停止运行。如果使用此参数, 即使应用程序发生 ANR 错误, Monkey 依然会发送事件, 直到事件计数完成。

(7) --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。如果使用此参数, 即使应用程序发生许可错误, Monkey 依然会发送事件, 直到事件计数完成。

(8) --pct- {+事件类别} {+事件类别百分比}

用于指定每种类别事件的数目百分比 (在 Monkey 事件序列中, 该类事件数目占总事件数目的百分比)

① --pct-touch {+百分比}

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)

```
adb shell monkey -p com.htc.Weather --pct-touch 10 1000
```

② --pct-motion {+百分比}

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)

```
adb shell monkey -p com.htc.Weather --pct-motion 20 1000
```

③ --pct-trackball {+百分比}

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)

```
adb shell monkey -p com.htc.Weather --pct-trackball 30 1000
```

④ --pct-nav {+百分比}

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)

以上仅举例 4 条, 其余指令在后续报告中会补充说明。

2 实验环境搭建

2.1 硬件环境

Windows 系统 + 安卓手机

2.2 软件配置

adb 介绍:

Android Debug Bridge (安卓调试桥) tools。它就是一个命令行窗口, 用于通过电脑端与模拟器或者是设备之间的交互。

adb 是一个 C/S 架构的应用程序, 由三部分组成:

① 运行在 PC 端的 adb client:

命令行程序 adb 用于从 shell 或脚本中运行 adb 命令。首先, adb 程序尝试定位主机上的 adb 服务器, 如果找不到 adb 服务器, adb 程

序自动启动一个 adb 服务器。接下来，当设备的 adbd 和 PC 端的 adb server 建立连接后，adb client 就可以向 adb server 发送服务请求。

② 运行在 PC 端的 adb server:

adb server 是运行在主机上的一个后台进程。它的作用在于检测 USB 端口感知设备的连接和拔除，以及模拟器实例的启动或停止，adb server 还需要将 adb client 的请求通过 USB 或者 TCP 的方式发送到对应的 adbd 上；

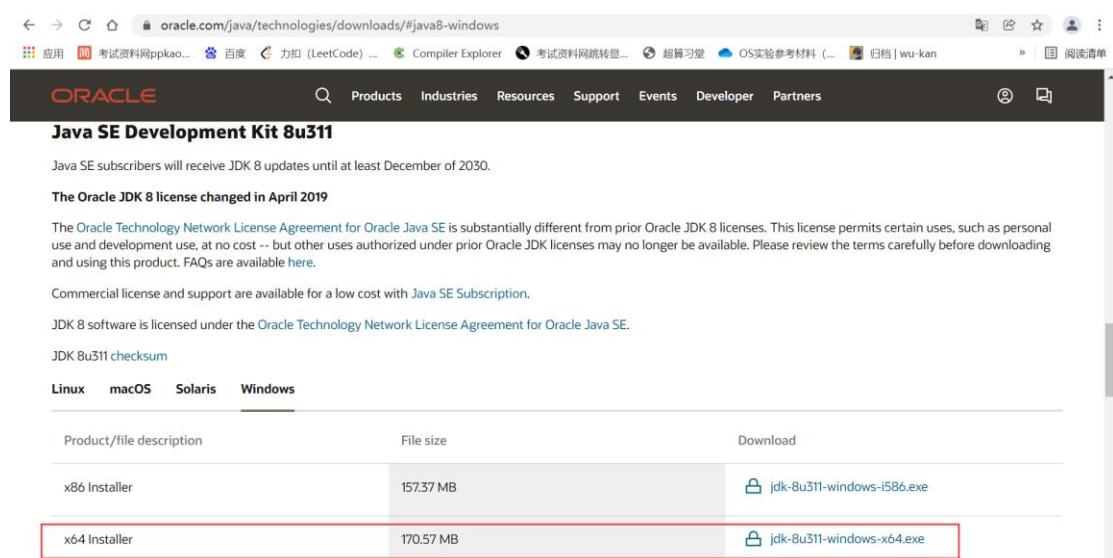
③ 运行在设备端的常驻进程 adb demon (adbd):

程序 adbd 作为一个后台进程在 Android 设备或模拟器系统中运行。它的作用是连接 adb 服务器，并且为运行在主机上的客户端提供一些服务。

配置步骤如下：

(1) 配置 Java 环境，安装 JDK

访问 <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> 安装 Windows 安装包



The screenshot shows the Oracle Java SE Development Kit 8u311 download page. The page includes the Oracle logo, navigation links (Products, Industries, Resources, Support, Events, Developer, Partners), and a table of download links for Windows. The table has three columns: Product/file description, File size, and Download. The x64 Installer row is highlighted with a red box, showing a file size of 170.57 MB and a download link to jdk-8u311-windows-x64.exe.

Product/file description	File size	Download
x86 Installer	157.37 MB	jdk-8u311-windows-i586.exe
x64 Installer	170.57 MB	jdk-8u311-windows-x64.exe

下载完执行安装.exe 文件默认安装即可。

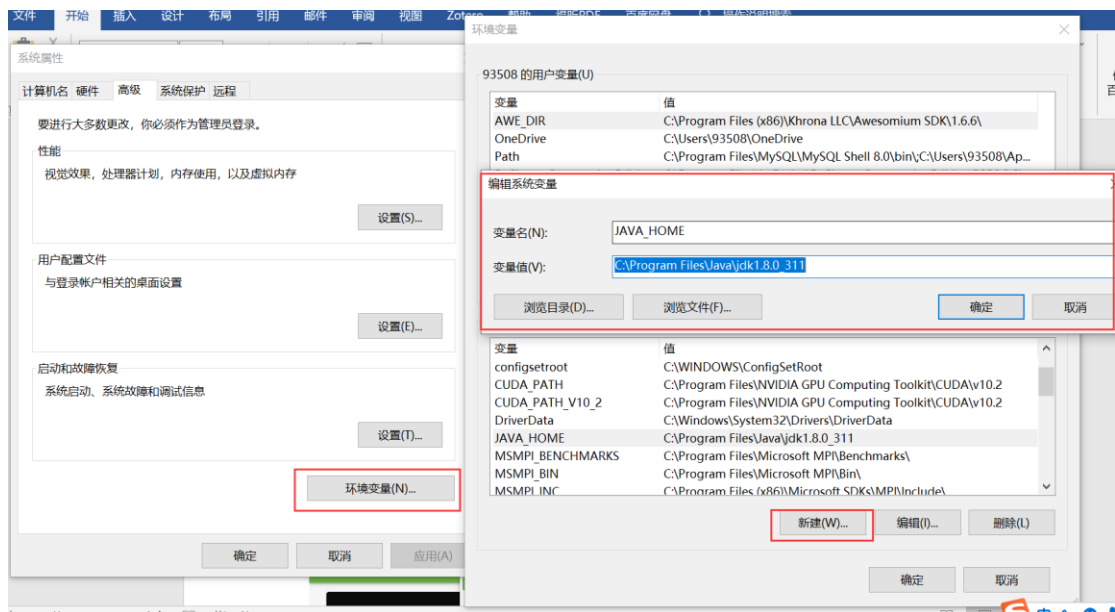
(2) JDK 环境变量配置

安装完成后，需要配置环境变量。

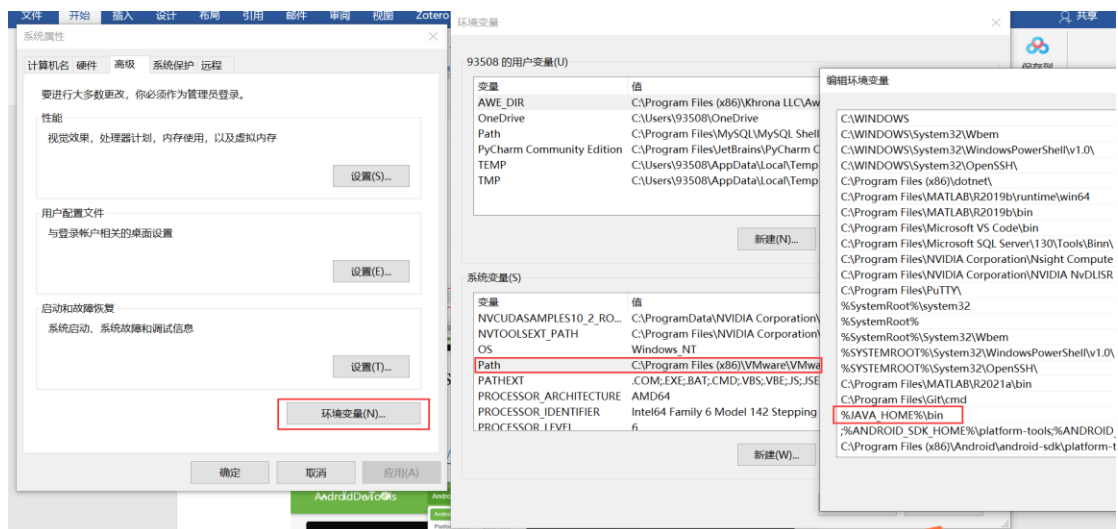
查看高级系统设置—环境变量—系统变量—新建

① 添加 JAVA_HOME 环境变量

路径为：C:\Program Files\Java\jdk1.8.0_311

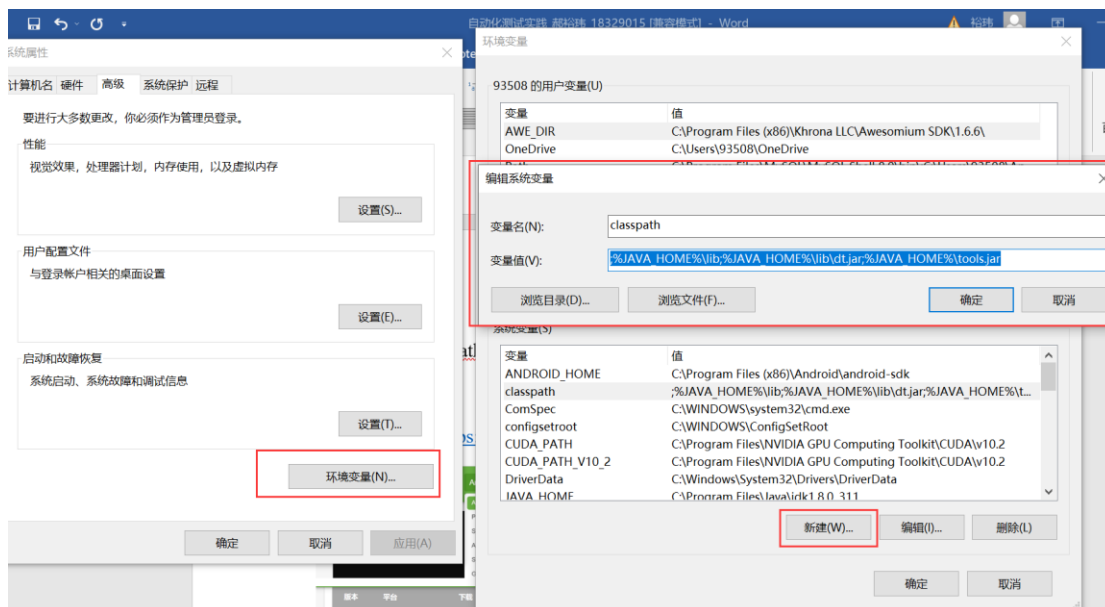


② 修改 Path 环境变量，在 Path 中添加%JAVA_HOME%\bin



③ 添加 classpath 环境变量：

路径为：;%JAVA_HOME%\lib;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\tools.jar

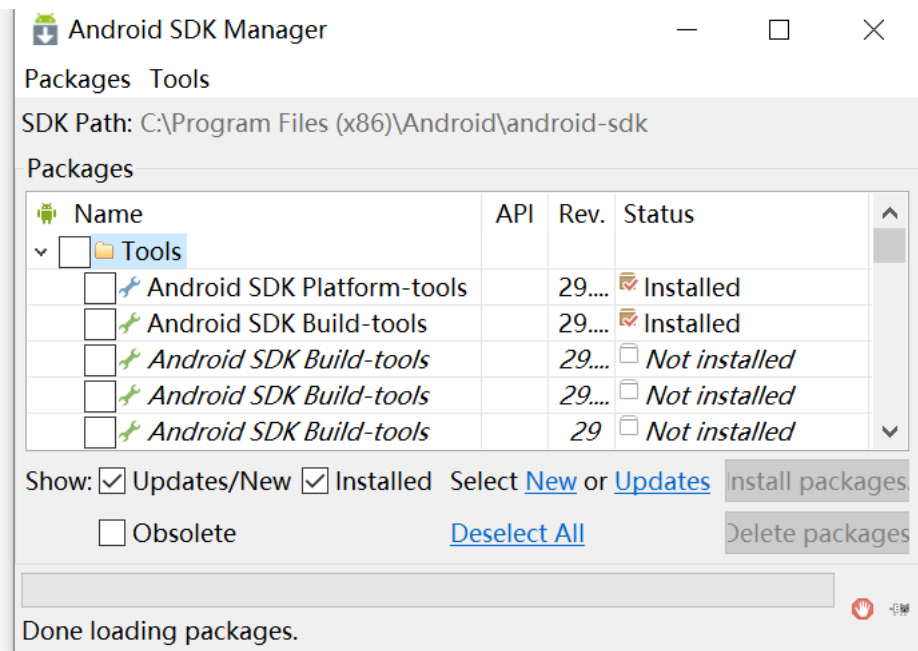


(3) 进入 <https://www.androiddevtools.cn/> 下载 SDK 安装包

The screenshot shows the Android DevTools website. The 'SDK Tools' link is highlighted in the navigation menu. Below, a table lists the download links for the SDK tools and the installer.

版本	平台	下载	大小	SHA-1校验码	官方SHA-1校验码截图
3859397	Windows	sdk-tools-windows-3859397.zip	132 MB	7f6037d3a7d6789b4fdc06ee7af041e071e9860c5f166f7a4eb5913df9871fd2	查看
	Mac OS X	sdk-tools-darwin-3859397.zip	82 MB	4a81754a760fce88cba74d69c364b05b31c53d57b26f9f82355c61d5fe4b9d9	
	Linux	sdk-tools-linux-3859397.zip	130 MB	444e22ce8ca0f6735bda4b85175ed3731cae3ffa695ca18119cbacef1c1bea0	
24.4.1	Windows	installer_r24.4.1-windows.exe	144 MB	f9b59d72413649d31e633207e31f456443e7ea0b	查看
		android-sdk_r24.4.1-windows.zip	190 MB	66b6a6433053c152b22bf8cab19c0f3fef4eba49	
	Mac OS X	android-sdk_r24.4.1-macosx.zip	98 MB	85a9cccb0b1f9e6f1f616335c5f07107553840cd	
	Linux	androiddk_r24.4.1-linux.tgz	311 MB	725bb360f07d04eacff5a2d57abdd49061326d	

解压后，双击 SDK Manager.exe 打开 SDK 管理器，并按照默认配置安装相关安装包即可。

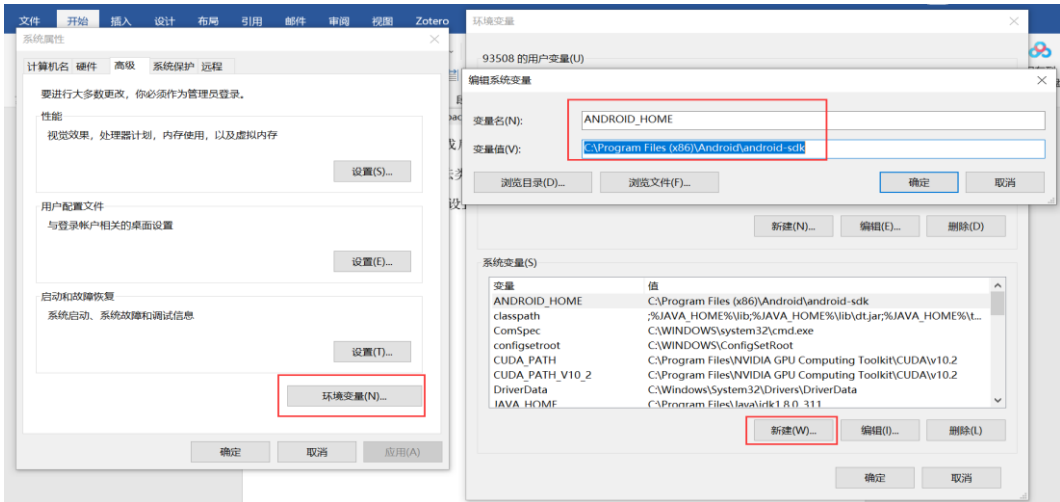


(4) 安装完成后，需要配置环境变量。方法和设置 JAVA JDK 的环境变量的方法类似。

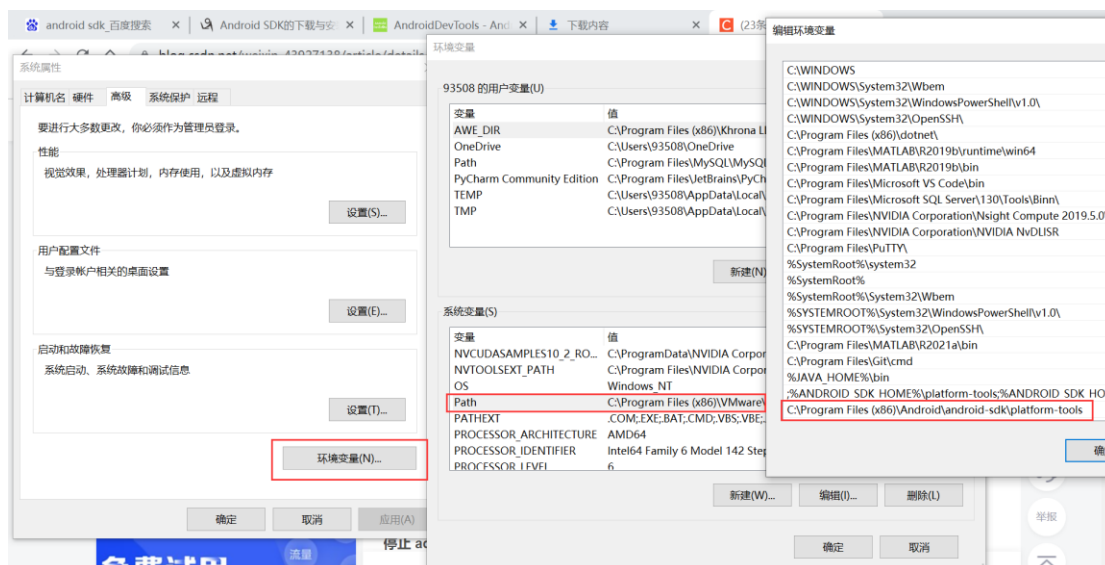
查看高级系统设置—环境变量—系统变量—新建

① 添加 ANDROID_HOME 环境变量：

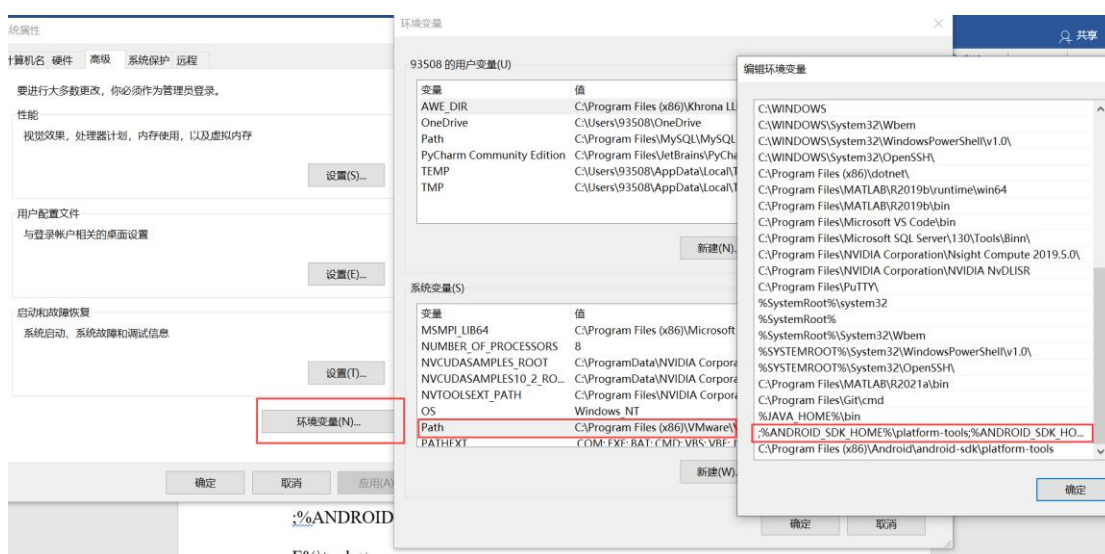
路径为：C:\Program Files (x86)\Android\android-sdk



② 将 adb.exe 在 SDK 里的路径添加到环境变量 Path 中



③ 将 SDK 的 platform-tools 和 tool 的路径添加到环境变量 Path 中



在 Path 变量中添加如下路径：

```
;%ANDROID_SDK_HOME%\platform-tools;%ANDROID_SDK_HOME%\tools
```

(5) 检验 JDK 和 adb 安装是否成功

① 打开 cmd，输入 java 和 javac

C:\WINDOWS\system32\cmd.exe

Microsoft Windows [版本 10.0.19043.1348]
(c) Microsoft Corporation。保留所有权利。

C:\Users\93508>java

用法: java [options] <主类> [args...]

(执行类)

或 java [options] -jar <jar 文件> [args...]

(执行 jar 文件)

或 java [options] -m <模块>[/<主类>] [args...]

java [options] --module <模块>[/<主类>] [args...]

(执行模块中的主类)

或 java [options] <源文件> [args]

(执行单个源文件程序)

将主类、源文件、-jar <jar 文件>、-m 或
--module <模块>/<主类> 后的参数作为参数
传递到主类。

其中，选项包括：

-cp <目录和 zip/jar 文件的类搜索路径>
-classpath <目录和 zip/jar 文件的类搜索路径>
--class-path <目录和 zip/jar 文件的类搜索路径>
使用 ; 分隔的，用于搜索类文件的目录，JAR 档案
和 ZIP 档案列表。
-p <模块路径>
--module-path <模块路径>...
用 ; 分隔的目录列表，每个目录
都是一个包含模块的目录。
--upgrade-module-path <模块路径>...

C:\WINDOWS\system32\cmd.exe

C:\Users\93508>javac

用法: javac <options> <source files>

其中，可能的选项包括：

@<filename>	从文件读取选项和文件名
-Akey[=value]	传递给注释处理程序的选项
--add-modules <模块>(<模块>)*	除了初始模块之外要解析的根模块；如果 <module> 为 ALL-MODULE-PATH，则为模块路径中的所有模块。
--boot-class-path <path>, -bootclasspath <path>	覆盖引导类文件的位置
--class-path <path>, -classpath <path>, -cp <path>	指定查找用户类文件和注释处理程序的位置
-d <directory>	指定放置生成的类文件的位置
-deprecation	输出使用已过时的 API 的源位置
--enable-preview	启用预览语言功能。要与 -source 或 --release 一起使用。
-encoding <encoding>	指定源文件使用的字符编码
-endorseddirs <dirs>	覆盖签名的标准路径的位置
-extdirs <dirs>	覆盖所安装扩展的位置
-g	生成所有调试信息
-g: {lines, vars, source}	只生成某些调试信息
-g:none	不生成任何调试信息
-h <directory>	指定放置生成的本机标头文件的位置
--help, -help, -?	输出此帮助消息
--help-extra, -X	输出额外选项的帮助
-implicit: {none, class}	指定是否为隐式引用文件生成类文件
-J<flag>	直接将 <标记> 传递给运行时系统
--limit-modules <模块>(<模块>)*	限制可观察模块的领域
--module <模块>(<模块>)*, -m <模块>(<模块>)*	

② 打开 cmd，输入 adb version



```
C:\WINDOWS\system32\cmd.exe
C:\Users\93508>adb version
Android Debug Bridge version 1.0.41
Version 29.0.6-6198805
Installed as C:\Program Files (x86)\Android\android-sdk\platform-tools\adb.exe
C:\Users\93508>
```

至此，所有配置均已完成。

3 Monkey 测试过程

3.1 测试步骤

在手机中：设置—关于手机—连续点击“版本号”7次，即可进入到开发者模式。然后可以到：设置—系统和更新—开发人员选项—调试里打开 USB 调试并授予 adb 相关权限。

USB 线连接安卓机和电脑，打开 cmd，输入 adb devices 查看设备连接情况。



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.19043.1348]
(c) Microsoft Corporation。保留所有权利。

C:\Users\93508>adb devices
List of devices attached
FEC0220710015466      device

C:\Users\93508>
```

由上图可知，设备连接成功。

输入 adb shell pm list packages -3 查看手机上所有的第三方安装包

```
C:\WINDOWS\system32\cmd.exe

C:\Users\93508>adb shell pm list packages -3
package:com.jwi3n4.wu0eji6
package:com.hisense.liuanbus
package:tuoyan.com.xinghuo_daying
package:com.baidu.student
package:com.tencent.tmgp.sgame
package:com.kuto.vpn
package:com.shark.jizhang
package:com.mayi.xiaoyi
package:com.android.bankabc
package:com.strivexj.timetable
package:com.sankuai.meituan
package:com.shanbay.sentence
package:com.tencent.qqmusic
package:com.tencent.androidqqmail
package:com.jingyao.easybike
package:com.nfyg.hsbb
package:com.camppmobile.snowcamera
package:com.tencent.gamehelper.smoba
package:com.chinatelecom.bestpayclient
package:com.sitech.ac
package:me.ele
package:com.hupu.games
package:com.xingin.xhs
package:com.fotiaoqiangvpn.android
package:com.tencent.mm
package:com.sankuai.meituan.takeoutnew
package:com.chinamworld.bocmbci
```

最终我们选择网易云音乐 APP 进行测试。

```
package:com.intsig.camscanner
package:com.yingyonghui.market
package:com.galaxylab.ss
package:com.baidu.netdisk
package:com.taobao.taobao
package:com.netease.cloudmusic
package:tv.danmaku.bili
package:com.fenbi.android.solar
package:com.eg.android.AlipayGphone
package:cn.wps.moffice_eng
```

文件安装名为: com.netease.cloudmusic

输入：adb shell monkey -p com.netease.cloudmusic -v -v -v 1000 >
C:\Users\93508\Desktop\test.out

代表打开网易云音乐 APP 并进行随机操作 1000 次，同时将日志 log 命名为 test.out 文件存储到对应路径中。

具体日志结果和录制视频可见压缩包中文件夹 Origin Monkey。

3.2 结果分析

我们对于操作日志进行分析：

最开始输出了相关参数 bash arg 以及 Monkey 执行的 seed 值、
执行次数 count 和测试包名 AllowPackage。

:IncludeCategory: android.intent.category.LAUNCHER 和:IncludeCategory: android.intent.category.MONKEY 是 Monkey 默认的参数，如果需要添加其它的 Category，使用-c 参数添加。

```
bash arg: -p
bash arg: com.netease.cloudmusic
bash arg: -v
bash arg: -v
bash arg: -v
bash arg: 1000
:Monkey: seed=1639659218185 count=1000
:AllowPackage: com.netease.cloudmusic
:IncludeCategory: android.intent.category.LAUNCHER
:IncludeCategory: android.intent.category.MONKEY
```

带有 NOT USING 的均代表该安装包未被测试。

```
// Selecting main activities from category
android.intent.category.LAUNCHER
// - NOT USING main activity com.android.deskclock.AlarmsMainActivity
(from package com.android.deskclock)
```



```
// - NOT USING main activity com.huawei.gallery.app.GalleryMain (from
package com.android.gallery3d)
// - NOT USING main activity com.android.mediacenter.PageActivity
(from package com.android.mediacenter)
// - NOT USING main activity com.android.mms.ui.ConversationList
(from package com.android.mms)
// - NOT USING main activity com.android.settings.HWSettings (from
package com.android.settings)
// - NOT USING main activity
com.huawei.android.findmyphone.ui.findphone.FindMyPhoneActivity (from
package com.huawei.android.findmyphone)
```

带有+ USING 的代表该安装包被测试。

```
// + Using main activity
com.netease.cloudmusic.activity.LoadingActivity (from package
com.netease.cloudmusic)
```

包含了 Seed 种子序号以及各种操作的占比。

```
// Seeded: 1639659218185
// 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%
```

查看源码可知各序号代表操作如下：

```
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;

```

- 0: 触摸事件百分比, 参数--pct-touch
- 1: 手势事件百分比, 参数--pct-motion
- 2: 二指缩放事件百分比, 参数--pct-pinchzoom
- 3: 轨迹球事件百分比, 参数--pct-trackball
- 4: 屏幕旋转事件百分比, 参数--pct-rotation
- 5: 指定（权限）事件百分比, 参数--pct-permission
- 6: 基本导航事件百分比, 参数--pct-nav
- 7: 主要导航事件百分比, 参数--pct-majornav
- 8: 系统导航事件百分比, 参数--pct-syskeys
- 9: 启动 activity 事件百分比, 参数--pct-appswitch
- 10: 键盘翻转事件百分比, 参数--pct-flip
- 11: 其他事件百分比, 参数--pct-anyevent

Switch 表示当前执行的 activity, 注释中的 Allowing 则代表 activity 成功跳转（切换）。

```

:Switch:
#Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10200000;component=com.netease.cloudmusic/.activity.LoadingActivity;end

```

```
// Allowing start of Intent { act=android.intent.action.MAIN
cat=[android.intent.category.LAUNCHER]
cmp=com.netease.cloudmusic/.activity.LoadingActivity } in package
com.netease.cloudmusic
```

接下来的内容就是各种操作的具体信息：

```
Sleeping for 0 milliseconds
:Sending Trackball (ACTION_MOVE): 0:(4.0,1.0)
:Sending Trackball (ACTION_MOVE): 0:(-3.0,0.0)
:Sending Trackball (ACTION_MOVE): 0:(3.0,2.0)
:Sending Trackball (ACTION_MOVE): 0:(1.0,4.0)
:Sending Trackball (ACTION_MOVE): 0:(3.0,1.0)
:Sending Trackball (ACTION_MOVE): 0:(-2.0,3.0)

:Sending Touch (ACTION_DOWN): 0:(28.0,2142.0)
:Sending Touch (ACTION_MOVE): 0:(28.323025,2137.5586)
:Sending Touch (ACTION_MOVE): 0:(28.37308,2134.9907)
:Sending Touch (ACTION_MOVE): 0:(28.741571,2127.8022)
:Sending Touch (ACTION_MOVE): 0:(29.245035,2118.1257)
:Sending Touch (ACTION_MOVE): 0:(29.711208,2117.1511)
```

Monkey 测试结束后，日志会在最后打印出当前执行事件的次数 Events injected 和执行时间 elapsed time。Dropped 代表执行失败的操作次数。

```
Events injected: 1000
:Sending rotation degree=0, persist=false
:Dropped: keys=0 pointers=0 trackballs=0 flips=5 rotations=0
## Network stats: elapsed time=10614ms (0ms mobile, 0ms wifi, 10614ms
not connected)
// Monkey finished
```

同时对测试结果的分析还需要搜索以下几个关键词：

ANR、Exception、Null、Error、crash (Fatal)

① ANR 问题：在日志中搜索 “ANR”

ANR=Application Not Responding

在 Android 上，若应用程序有一段时间响应不够灵敏(5 秒内没有输入响应事件)，则系统会向用户显示一个对话框，这个对话框称作应用程序无响应（ANR：Application Not Responding）对话框。

② 崩溃问题：在日志中搜索 “Exception”

Null 指针异常：空指针异常主要是有 NullPointerException 异常提示；

Debug 异常：出现的是 IllegalStateException 异常；

低内存异常：出现的是 OutOfMemoryError 异常；

操作无响应异常：TimerOut（KeyDispatchingTimedOut 提示）；

RuntimeException 异常（操作无响应或应用的服务无法启动或连接）；

StaleDataException/readException 异常（android 数据库出现异常）；

IllegalArgumentException 异常（向函数传递了一个不正确或不合法的参数）；

4 Monkey 测试优缺点

（1）优点：

Monkey 的优点是可以对 APP 进行各种混乱操作，能有效的对 APP 进行压力测试，观察 APP 是否有 Crash 或者异常日志。

(2) 缺点:

Monkey 虽然可以根据一个指定的命令脚本发送按键消息, 但它不支持条件判断, 也不支持读取待测界面的信息来执行验证操作。

5 智能 Monkey 测试

5.1 Maxim & UICrawler

常规的 Monkey 测试存在着若干问题, 比如遍历界面有限, 路径回环等。

所以我对于更人性化, 效率更高的智能 Monkey 测试进行了调研。最终在 Github 上发现了对于智能 Monkey 测试第三方库/软件的总结列表 https://github.com/GitDzreal93/dev-tester/blob/fdc4c02569976b099e1a6546aa77878b38f579ad/page/app_testing/basics.md

稳定性测试工具

1. 安卓原生Monkey

```
adb shell monkey -p com.lianjia.beike --throttle 100 --pct-touch 50 --pct-motion 50 -v 1000 > c:\monkey.txt
```

2. Maxim

- 智能monkey
- [GitHub](#)

3. UICrawler

- 支持Android、iOS和微信小程序, 依赖Appium
- [GitHub](#)

4. AppCrawler

- 支持Android和iOS, 依赖Appium
- [GitHub](#)

经过筛选对比, 最终我选择 Maxim & UICrawler 来进行第二部分的实验。(其中 Maxim 是基于遍历规则的高性能 Android Monkey, 适用于真机/模拟器的 APP UI 压力测试, UICrawler 则是基于

Appium 实现，可面向 Android 和 IOS 移动 App，或 H5 或微信等应用的，基于元素识别的 UI 自动化遍历测试。)

原始 Monkey 有两大缺点亟需改善：

① 有可能点击到页面的顶部，下拉通知栏，或者点击到底部返回键菜单栏（尤其是在模拟器里执行的时候），会有各种误点操作；

② 纯随机，没有任何遍历算法加持。

而 Maxim 是基于原始 Monkey 实现的二次开发，相较于原始 Monkey，Maxim 新增以下功能：

① 多种随机测试模式：DFS（深度遍历），MIX 模式（Monkey 随机测试+控件识别），troy 模式（按照控件选择器进行遍历）；

② 可进行执行时长设定；

③ 可以自定义 activity 黑白名单；

④ 兼容多个 Android 版本；

⑤ 防跳出，防休眠，防假死。

同时 UICrawler 相较于原始 Monkey，它也具有以下优势：

① 基于深度优先的原则，点击 UI 上的元素。当发现 Crash 时会提供操作步骤截图及相应的 log；

② 元素遍历结束或按下 Ctrl + C 键会生成 HTML 测试报告。测试报告中汇集了测试统计、log 及截图的相应信息；

③ 同一个元素只会点击一次(白名单中的元素可能会被多次点击)；

④ 支持对输入框的文本输入(需在文件中进行配置 INPUT_TEXT_LIST)；

⑤ 支持更高级的配置，可配置用户名、密码等。

5.2 Maxim 环境配置

① 下载 <https://github.com/zhangzhao4444/Maxim> 中的压缩包；

② USB 连接安卓机，打开 cmd，先输入 `adb devices` 检验设备是否连接成功。确认成功后执行以下 2 条命令将 `monkey.jar` 和 `framework.jar` 安装到安卓机的 `/sdcard` 文件夹中：

```
adb push C:\Users\93508\Desktop\Maxim-master\monkey.jar /sdcard/
adb push C:\Users\93508\Desktop\Maxim-master\framework.jar /sdcard/
```

```
C:\Users\93508>adb push C:\Users\93508\Desktop\Maxim-master\monkey.jar /sdcard/
C:\Users\93508\Desktop\Maxim-master\monkey.jar: 1 file pushed, 0 skipped. 6.2 MB/s (186992 bytes in 0.029s)
C:\Users\93508>adb push C:\Users\93508\Desktop\Maxim-master\framework.jar /sdcard/
C:\Users\93508\Desktop\Maxim-master\framework.jar: 1 file pushed, 0 skipped. 31.5 MB/s (1149240 bytes in 0.035s)
C:\Users\93508>_
```

至此环境配置成功，我们接下来可以使用命令行来运行 Maxim。

5.3 命令行运行 Maxim

打开 cmd，执行如下命令：

```
adb shell CLASSPATH=/sdcard/monkey.jar:/sdcard/framework.jar exec app_process /system/bin tv.panda.test.monkey.Monkey -p com.netease.cloudmusic --uiautomatormix --running-minutes 1 -v -v > C:\Users\93508\Desktop\test2.out
```

其中：


```
.anythingUcancontactpanda_zzaqq77227005
.
.
.
.
[Maxim] // Monkey: seed=1639829207807 count=1000
```

这里与原始 Monkey 同理，带有 NOT USING 的均代表该安装包未被测试。

```
[Maxim] // Selecting main activities from category
android.intent.category.LAUNCHER
[Maxim] // - Not Using main activity
com.android.deskclock.AlarmsMainActivity (from package
com.android.deskclock
[Maxim] // - Not Using main activity com.huawei.gallery.app.GalleryMain
(from package com.android.gallery3d
[Maxim] // - Not Using main activity
com.android.mediacenter.PageActivity (from package
com.android.mediacenter
[Maxim] // - Not Using main activity
com.android.mms.ui.ConversationList (from package com.android.mms
[Maxim] // - Not Using main activity com.android.settings.HWSettings
(from package com.android.settings
[Maxim] // - Not Using main activity
com.huawei.android.findmyphone.ui.findphone.FindMyPhoneActivity (from
package com.huawei.android.findmyphone
[Maxim] // - Not Using main activity com.huawei.browser.Main2 (from
package com.huawei.browser
```

带有+ USING 的代表该安装包被测试。

```
[Maxim] // + Using main activity
com.netease.cloudmusic.activity.LoadingActivity (from package
com.netease.cloudmusic)
```

这里显示了安装包的白名单和黑名单，安卓机基本信息以及随机测试的策略。（这里为默认的混合模式 uiautomatormix，70%控件解

析随机点击，其余 30%按原 Monkey 事件概率分布)

```
[Maxim] // Add Package To PackageWhiteList : com.oppo.camera
[Maxim] // Add Package To PackageWhiteList :
com.android.packageinstaller
[Maxim] // Add Package To PackageWhiteList : com.android.documentsui
[Maxim] // AllowPackage: com.netease.cloudmusic
[Maxim] // AllowPackage: com.oppo.camera
[Maxim] // AllowPackage: com.android.packageinstaller
[Maxim] // AllowPackage: com.android.documentsui
[Maxim] // IncludeCategory: android.intent.category.LAUNCHER
[Maxim] // IncludeCategory: android.intent.category.MONKEY
[Maxim] // Special Event:
[Maxim] // Black Widget List:
[Maxim] // InputMethod ID:
InputMethodInfo{com.baidu.input_huawei/.ImeService, settings:
com.baidu.input.ImeMainConfigActivity}.id
[Maxim] // Device Info: HUAWEI-TAS-AN00, Android Version: 10
[Maxim] strategy : uiautomator-Mix.
```

这里显示了各类操作的占比。

```
// Event percentages:
// TOUCH : 62.5%
// MOTION : 10.0%
// PINCHZOOM : 4.0%
// TRACKBALL : 5.0%
// ROTATION : 5.0%
// PERMISSION : -0.0%
// NAV : -0.0%
// MAJORNAV : -0.0%
// SYSOPS : -0.0%
// APPSWITCH : 1.0%
// FLIP : 1.0%
// ANYTHING : -0.0%
// BACK : 10.0%
// MUTE : 1.0%
// SCHEMA : -0.0%
// NULLINTENT : 0.2%
// RESET : 0.3%
```

这里显示了各操作事件的执行开始时间，并且从 Switch 正式开始各种随机操作。

```
[Maxim] // event0, 2021-12-16 18:12:04.539
[Maxim] // Switch:
#Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10200000;component=com.netease.cloudmusic/.activity.LoadingActivity;end
[Maxim] // the top activity is
com.netease.cloudmusic.activity.LoadingActivity, phone launcher
activity is com.huawei.android.launcher.unihome.UniHomeLauncher
[Maxim] // the top activity is
com.netease.cloudmusic.activity.LoadingActivity, phone launcher
activity is com.android.settings.FallbackHome
[Maxim] // Allowing start of Intent { act=android.intent.action.MAIN
cat=[android.intent.category.LAUNCHER]
cmp=com.netease.cloudmusic/.activity.LoadingActivity } in package
com.netease.cloudmusic
[Maxim] // : debug, currentActivity is
com.netease.cloudmusic.activity.LoadingActivity
[Maxim] // event1, 2021-12-16 18:12:04.562
[Maxim] // Sleeping for 0 milliseconds
[Maxim] // event1, 2021-12-16 18:12:04.563
[Maxim] // Sleeping for 6000 milliseconds
[Maxim] // event1, 2021-12-16 18:12:10.566
[Maxim] // Sleeping for 0 milliseconds
[Maxim] Random Number = 0.44295365
[Maxim] random hit touch
[Maxim] // Click: null, android.widget.FrameLayout, (540.0,1071.0)
[Maxim] // event1, 2021-12-16 18:12:10.626
[Maxim] :Sending Touch (ACTION_DOWN): 0:(540.0,1071.0)
[Maxim] // event2, 2021-12-16 18:12:10.636
[Maxim] :Sending Touch (ACTION_UP): 0:(545.8,1077.7322)
```

日志的最后部分对本次智能 Monkey 随机测试进行了数据总结：操作总数，Dropped 操作数，Activity of Coverage（Activity 生命周期）等等。

```
[Maxim] // Try to stop package com.netease.cloudmusic
[Maxim] // event1312, 2021-12-16 18:13:00.369
```

```
[Maxim] // Sleeping for 6000 milliseconds
[Maxim] // realtime=143321757, mEndTime=143319909
[Maxim] // Events injected: 1312
[Maxim] // Monkey is over!
[Maxim] // Sending rotation degree= 0, false
[Maxim] // Unaccounted for: 61884
[Maxim] // Tested activities 17
[Maxim] // 1 - com.netease.cloudmusic.activity.LoadingActivity
[Maxim] // 2 - com.netease.cloudmusic.activity.MainActivity
[Maxim] // 3 - com.netease.cloudmusic.activity.PlayerActivity
[Maxim] // 4 - com.netease.cloudmusic.activity.ArtistActivityV3
[Maxim] // 5 -
com.netease.cloudmusic.music.biz.rn.activity.CloudMusicRNActivity
[Maxim] // 6 - com.netease.cloudmusic.activity.PlayListActivity
[Maxim] // 7 -
com.netease.cloudmusic.music.biz.mlogdetail.MLogVideoActivity
[Maxim] // 8 -
com.netease.cloudmusic.music.biz.rn.activity.MainProcessRNActivity
[Maxim] // 9 - com.netease.cloudmusic.activity.H5CustomViewActivity
[Maxim] // 10 - com.netease.cloudmusic.activity.RedirectActivity
[Maxim] // 11 - com.netease.cloudmusic.activity.ImageBrowseActivity
[Maxim] // 12 - com.netease.cloudmusic.activity.AlbumActivity
[Maxim] // 13 -
com.netease.cloudmusic.music.biz.comment.activity.CommentActivity
[Maxim] // 14 - com.netease.cloudmusic.activity.AddToPlayListActivity
[Maxim] // 15 -
com.netease.cloudmusic.activity.MLogAggregationMusicActivity
[Maxim] // 16 - com.netease.cloudmusic.activity.SharePanelActivity
[Maxim] // 17 -
com.netease.cloudmusic.module.musicshare.PlayerMusicShareActivity
[Maxim] // Activity of Coverage: 3.1858406
[Maxim] // How many Events Dropped: keys=0, pointers=100, trackballs=0,
flips=0, rotations=0
[Maxim] // Monkey finished
```

本想继续结合 Appetizer 来进行图形化界面的 Monkey 测试。但插桩后的安装包在手机上安装完成后，一直打开就闪退。虽然网站指引说将安装包重新签名即可。但我用了各种方式对其进行重新签名（Jarsigner, apksigner, Android Studio），均无法打开。只好作罢。

所以为了更好的体现智能 Monkey 测试的优越性，我决定继续调研 UICrawler。

5.5 UICrawler 环境配置

(1) 安装 Appium

① 进入官网 <http://appium.io/> 下载 Windows 客户端的 GUI，按照默认配置安装即可；

② 打开 Appium 并设置 IP 为本机 IP：127.0.0.1。

(2) 打开 https://pan.baidu.com/s/1mNci6SWNHPUlj_mvrfglbg?errormsg=Auth+Login+Params+Not+Corret&errno=2&ssnerror=0#list/path=%2F，下载 UICrawler.jar（内含配置文件 config.yml）

(3) USB 线连接安卓机和电脑，打开 cmd，输入：
adb shell logcat|findstr "Displayed" 查看打开网易云音乐所对应的 activity。

```
C:\Users\93508\Desktop\UICrawler>adb shell logcat|findstr "Displayed"
12-17 23:53:57.074 1677 1721 I WindowManager: Check opening app=AppWindowToken{2c243c9 token=Token{30cf67c ActivityRecord{2e84e09 u0 com.netease.cloudmusic/.activity.LoadingActivity:t552}}}: allDrawn=false startingDisplayed=false startingMoved=false isRelaunching()=false startingWindow=Window{2420129 u0 Splash Screen com.netease.cloudmusic}
12-17 23:53:57.100 1677 1721 I WindowManager: Check opening app=AppWindowToken{2c243c9 token=Token{30cf67c ActivityRecord{2e84e09 u0 com.netease.cloudmusic/.activity.LoadingActivity:t552}}}: allDrawn=false startingDisplayed=true startingMoved=false isRelaunching()=false startingWindow=Window{2420129 u0 Splash Screen com.netease.cloudmusic}
12-17 23:53:57.555 1677 1735 I ActivityTaskManager: Displayed com.netease.cloudmusic/.activity.LoadingActivity: +558ms
12-17 23:54:01.116 1677 2107 I WindowManager: Check opening app=AppWindowToken{2c24209 token=Token{2fd3ab8 ActivityRecord{2789583 u0 com.netease.cloudmusic/.activity.MainActivity:t552}}}: allDrawn=false startingDisplayed=false startingMoved=false isRelaunching()=false startingWindow=null
12-17 23:54:01.605 1677 5149 I WindowManager: Check opening app=AppWindowToken{2c24209 token=Token{2fd3ab8 ActivityRecord{2789583 u0 com.netease.cloudmusic/.activity.MainActivity:t552}}}: allDrawn=false startingDisplayed=false startingMoved=false isRelaunching()=false startingWindow=null
12-17 23:54:01.766 1677 1721 I WindowManager: Check opening app=AppWindowToken{2c24209 token=Token{2fd3ab8 ActivityRecord{2789583 u0 com.netease.cloudmusic/.activity.MainActivity:t552}}}: allDrawn=true startingDisplayed=false startingMoved=false isRelaunching()=false startingWindow=null
```

所以可将 config.yml 中内容修改如下：

```
CRITICAL_ELEMENT:

#---Android---
#Android App测试时，需指定App包名和启动的Activity
ANDROID_PACKAGE: com.netease.cloudmusic
ANDROID_MAIN_ACTIVITY: .activity.LoadingActivity
#通过xpath过滤出要点击的元素，注意值两侧要加单引号"
ANDROID_CLICK_XPATH_HEADER: '(string-length(@text)<30 or contains(@text,"允许"))'
#若App顶、底含有TabBar，需指定TabBar的id，若无TabBar注释下行。多个id时用or拼接：如 @resource-id="id1" or @resource-id="id2"
# ANDROID_BOTTOM_TAB_BAR_ID: '@resource-id="android:id/tabs"'
```

同时对于其他操作方面的内容进行以下修改（见下页）：

```

#*****Monkey测试时根据需要更改下面的值，调节事件发生概率*****
#每项为事件随机发生的次数，各项之和为100
MONKEY:
#滑动
SWIPE_RATIO: 7
#点击
CLICK_RATIO: 50
#点击 MONKEY_SPECIAL_POINT_LIST 中的坐标 主要用于触发“返回”功能
CLICK_SPECIAL_POINT_RATIO: 2
#重启app
RESTART_APP_RATIO: 0
#长按
LONG_PRESS_RATIO: 1
#触发Home键 (Android Only)
HOME_KEY_RATIO: 0
#触发Back Key
BACK_KEY_RATIO: 20
#手势操作
#双击
DOUBLE_TAP_RATIO: 2
#缩小(两指)
PINCH_RATIO: 2
#放大(两指)
UNPINCH_RATIO: 2
#拖拽
DRAG_RATIO: 4
#CLICK_ITEM_XPATH_LIST中指定元素的点击次数
CLICK_ITEM_BY_XPATH_RATIO: 1

```

#一些全局的配置

GENERAL:

#截屏数量限制

SCREENSHOT_COUNT: 5000

#达到截屏限制数量时删除所有截屏

ENABLE_DELETE_SCREEN: false

#Crash时截图显示步骤数量

CRASH_PIC_COUNT: 10

#遍历深度

MAX_DEPTH: 80

#点击次数

MAX_CLICK_COUNT: 10000

#查找元素的时间

DEFAULT_WAIT_SEC: 1

#查找元素的间隔时间

DEFAULT_POLLING_INTERVAL_SEC: 1

#是否忽略Crash,设为true时,crash后会重启app然后继续遍历

IGNORE_CRASH: true

#遍历的过程中进行向上划动操作 !!! 此项会加长遍历时间!!!

ENABLE_VERTICAL_SWIPE: false

#运行时间限制(分钟)

CRAWLER_RUNNING_TIME: 72000

#控制是否生成视频

GENERATE_VIDEO: true

#控制视频方向，竖屏或横屏

VIDEO_VERTICAL: true

#每个UI变化X次时，检测是否需要自动登录

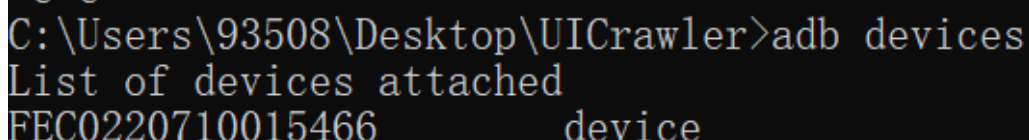
USER_LOGIN_INTERVAL: 5

至此，环境配置已结束。

5.6 UICrawler 运行结果

USB 线连接安卓机和电脑后，在 UICrawler-2.3.7.jar 所在的文件夹打开 cmd，执行如下命令：

adb devices 查看设备 ID：



```
C:\Users\93508\Desktop\UICrawler>adb devices
List of devices attached
FEC0220710015466      device
```

再输入以下指令进行智能 Monkey 测试：

```
java -jar UICrawler-2.3.7.jar -u FEC0220710015466 -m
```

第一次执行时，手机端会自动安装两个 Appium 的相关 APK：Appium Settings 和 io.appium.uiautomator2.server，同意即可。

之后程序会进行各种随机操作，UICrawler 相较于原始 Monkey 测试的优势之处在于：它会自动截图每次随机操作（以红点表示点击），并对全过程进行录屏，同时生成 log 日志以及 HTML 版本的测试报告。使我们能够更加详细的了解到此次随机测试的各种信息。

截图，视频，log 日志，HTML 测试报告可见压缩包中的文件夹 UICrawler\crawler_output。

HTML 测试报告如下（见下页）：

执行结果信息汇总 Testing result summary

基本信息 - Basic information

手机系统 - Mobile operating system	ANDROID
系统版本 - OS version	10
设备UUID - Device UUID	FEC0220710015466
测试开始时间 - Testing start time	2021-12-17 21:19:38
包名 - Package name	com.netease.cloudmusic
主活动 - Main Activity	.activity.LoadingActivity
总执行时间 - Total running time	0小时5分29秒
系统日志 - System log	C:\Users\93508\Desktop\UICrawler\crawler_output\FEC0220710015466-2021-12-17-21_19_38\FEC0220710015466-2021-12-17_21-19-45.log
Crash数量 - Crash count	0
测试类型 - Test type	Monkey随机测试

Crash信息 - Crash information

No crash found during testing.

Activity Clicked List

Activity	Detail
.music.biz.voice.player.PlayerProgramActivity	{CLICK_SPECIAL_POINT=1, DRAG=1, CLICK=10}
.music.biz.voice.voicelist.VoiceListDetailActivity	{SWIPE=3, DRAG=1, CLICK=1}
.activity.MainActivity	{SWIPE=2, PINCH=2, BACK_KEY=2, DRAG=2, UNPINCH=2, CLICK_ITEM_BY_XPATH=1, CLICK=3, LONG_PRESS=1, DOUBLE_TAP=2}
.music.biz.oldcomment.activity.ResourceCommentActivity	{CLICK=1}
.activity.ArtistActivityV3	{SWIPE=1, BACK_KEY=1, CLICK=55}
.music.calendar.view.activity.MusicCalendarActivity	{CLICK_SPECIAL_POINT=1}

可见该测试报告比原始 Monkey 测试的 log 日志更为简洁易懂。

同时从视频可看出，UICrawler 控制的 Monkey 测试遍历界面更多，无效操作更少，效率高于原始 Monkey 测试。

6 实验感想

本次实验让我对于移动端的稳定性测试有了初步且全面的了解。

明白了 Monkey 测试的本质以及当前互联网上更为优秀的一些第三方开发的智能 Monkey 测试并能灵活运用。

本次实验主要困难集中在前期环境的配置。由于我之前对于安卓环境的开发配置一无所知，所以踩了很多坑。好在通过大量的时间花费和资料查询逐一解决。唯一遗憾的是本次实验没有实现 Appetizer 图形化界面与 Maxim 的结合。一直卡在插桩安装包闪退这一步，即使重新生成签名也无济于事，最终被迫放弃。但我仍然实

现了命令行调用 Maxim 来执行智能 Monkey 测试，并继续调研了另一种方法 UICrawler 来补充自己对于本次实验的理解。

通过此次实验，我对于安卓环境开发，移动端稳定性测试均有了更深一步的了解，这为我以后的研究打下了基础。