

移动互联网编程实践大作业

Monkey 测试的调研与实践

中山大学计算机学院 计算机科学与技术

19335174 施天予

目录

1	Monkey 测试调研	2
1.1	基础知识	2
1.2	基本用法	2
1.3	命令选项	3
1.4	日志分析方法	5
1.5	基础 Monkey 的缺陷	5
2	实验环境	6
2.1	设备平台	6
2.2	Monkey 环境	6
2.3	应用选择	7
3	基础 Monkey 测试	8
3.1	测试过程	8
3.2	日志分析	8
3.2.1	测试命令	8
3.2.2	伪随机事件流	9
3.2.3	异常信息	10
3.2.4	结果信息	11
3.3	测试结果	11
4	智能 Monkey 测试	12
4.1	AppetizerIO	12
4.1.1	Maxim 工具	12
4.1.2	测试过程	13
4.2	测试结果	15
5	总结	18

一、Monkey 测试调研

1. 基础知识

Monkey 是一个 app 的自动化命令行测试工具,可以运行在模拟器里或实际设备中。Monkey 在英文中是猴子的意思。顾名思义,它的自动化测试就类似猴子一样在软件上乱敲按键。原理上,它利用 socket 通信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、手势输入等操作),实现对正在开发的应用程序进行测试,检测程序长时间的稳定性,多久的时间会发生异常。

Monkey 工具存在于 Android 系统中,由 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 包含许多选项,主要分为以下四个类别:

- 基本配置选项,例如设置要尝试的事件数。
- 操作限制条件,例如将测试对象限制为单个软件包。
- 事件类型和频率。
- 调试选项。

Monkey 在运行时会生成事件并将其发送到系统,监视被测系统并查找三种特殊情况:

- 如果将 Monkey 限制为在一个或多个特定软件包中运行,它会监视并阻止转到任何其他软件包的尝试。
- 如果应用崩溃或收到任何未处理的异常,Monkey 会停止并报告错误。
- 如果应用生成“应用无响应”错误,Monkey 会停止并报告错误。
- 根据选择的详细程度级别,还会看到有关 Monkey 进度和所生成事件的报告。

2. 基本用法

可以使用开发计算机上的命令行启动 Monkey,也可以通过脚本启动。由于 Monkey 在模拟器/设备环境中运行,因此必须从该环境中通过 shell 启动它。为此,可以在每个命令前面加上 adb shell,或者直接进入 shell 并输入 Monkey 命令。

基本语法如下:

```
1 $ adb shell monkey [options] <event-count>
```

如果未指定任何选项,Monkey 将以静默(非详细)模式启动,并将事件发送到目标上安装的任何(及所有)软件包。下面是一个更典型的命令行,它会启动应用并向其发送 500 个伪随机事件:

```
1 $ adb shell monkey -p your.package.name -v 500
```

3. 命令选项

在具体使用时，可以使用 `adb shell monkey -help` 来查看提示命令参数帮助，获得完整的参数列表，如下图：

```
PS C:\Users\DELL> adb shell monkey -help
usage: monkey [-p ALLOWED_PACKAGE [-p ALLOWED_PACKAGE] ...]
              [-c MAIN_CATEGORY [-c MAIN_CATEGORY] ...]
              [--ignore-crashes] [--ignore-timeouts]
              [--ignore-security-exceptions]
              [--monitor-native-crashes] [--ignore-native-crashes]
              [--kill-process-after-error] [--hprof]
              [--pct-touch PERCENT] [--pct-motion PERCENT]
              [--pct-trackball PERCENT] [--pct-syskeys PERCENT]
              [--pct-nav PERCENT] [--pct-majornav PERCENT]
              [--pct-appswitch PERCENT] [--pct-flip PERCENT]
              [--pct-anyevent PERCENT] [--pct-pinchzoom PERCENT]
              [--pct-permission PERCENT]
              [--pkg-blacklist-file PACKAGE_BLACKLIST_FILE]
              [--pkg-whitelist-file PACKAGE_WHITELIST_FILE]
              [--wait-dbg] [--dbg-no-events]
              [--setup scriptfile] [--f scriptfile] [--f scriptfile] ...]
              [--port port]
              [-s SEED] [-v [-v] ...]
              [--throttle MILLISEC] [--randomize-throttle]
              [--profile-wait MILLISEC]
              [--device-sleep-time MILLISEC]
              [--randomize-script]
              [--script-log]
              [--bugreport]
              [--periodic-bugreport]
              [--permission-target-system]
              COUNT
```

事件

- `-s <seed>`

伪随机数生成器的种子值。使用相同的种子值重新运行 Monkey，会生成相同的事件序列。

- `-throttle <milliseconds>`

在事件之间插入固定的延迟时间。可以使用此选项减慢 Monkey 速度。如果未指定，则不延迟，系统会尽快地生成事件。

- `-pct-touch <percent>`

调整轻触事件所占百分比。（轻触事件是指屏幕上的单个位置上的按下/释放事件。）

- `-pct-motion <percent>`

调整动作事件所占百分比。（动作事件包括屏幕上某个位置的按下事件，一系列伪随机动作和一个释放事件。）

- `-pct-trackball <percent>`

调整轨迹球事件所占百分比。（轨迹球事件包括一个或多个随机动作，有时后跟点击。）

- `-pct-nav <percent>`

调整“基本”导航事件所占百分比。(导航事件包括向上/向下/向左/向右, 作为方向输入设备的输入。)

- `-pct-majornav <percent>`

调整“主要”导航事件所占百分比。(这些导航事件通常会导致界面中的操作, 例如 5 方向键的中间按钮、返回键或菜单键。)

- `-pct-syskeys <percent>`

调整“系统”按键事件所占百分比。(这些按键通常预留供系统使用, 例如“主屏幕”、“返回”、“发起通话”、“结束通话”或“音量控件”。)

- `-pct-appswitch <percent>`

调整 Activity 启动次数所占百分比。Monkey 会以随机间隔发起 `startActivity()` 调用, 以最大限度地覆盖软件包中的所有 Activity。

- `-pct-anyevent <percent>`

调整其他类型事件所占百分比。这包括所有其他类型的事件, 例如按键、设备上的其他不太常用的按钮等等。

约束条件

- `-p <allowed-package-name>`

指定一个或多个软件包, Monkey 将仅允许系统访问这些软件包内的 Activity。如果应用需要访问其他软件包中的 Activity (例如选择联系人), 还需要指定这些软件包。如果未指定任何软件包, Monkey 将允许系统启动所有软件包中的 Activity。

- `-c <main-category>`

指定一个或多个类别, Monkey 将仅允许系统访问其中一个指定类别中所列的 Activity。

调试

- `-dbg-no-events`

指定后, Monkey 将初始启动到测试 Activity, 但不会生成任何其他事件。

- `-hprof`

在 Monkey 事件序列之前和之后立即生成分析报告。

- `-ignore-crashes`

通常当应用崩溃或遇到任何类型的未处理异常时, Monkey 将会停止。如果指定此选项, Monkey 会继续向系统发送事件, 直到计数完成为止。

- `-ignore-timeouts`

通常当应用遇到任何类型的超时错误 (例如“应用无响应”对话框) 时, Monkey 将会停止。如果指定此选项, Monkey 会继续向系统发送事件, 直到计数完成为止。

- `-ignore-security-exceptions`

通常当应用遇到任何类型的权限错误（例如，如果它尝试启动需要特定权限的 Activity）时，Monkey 将会停止。如果指定此选项，Monkey 会继续向系统发送事件，直到计数完成为止。

- `-kill-process-after-error`

通常当 Monkey 因出错而停止运行时，出现故障的应用将保持运行状态。设置此选项后，它将会指示系统停止发生错误的进程。

- `-monitor-native-crashes`

监视并报告 Android 系统原生代码中发生的崩溃。

- `-wait-dbg`

阻止 Monkey 执行，直到为其连接了调试程序。

4. 日志分析方法

1. 初步分析

(a) 查找 Monkey 中出错的位置

(b) 查看 Monkey 里面出错前的一些事件动作，并手动执行该动作

(c) 若以上步骤还不能找出，可以使用之前执行的 Monkey 命令再执行一遍，注意 seed 值要一样

2. 一般结果分析

(a) ANR 问题：在日志中搜索 “ANR”。

(b) 崩溃问题：在日志中搜索 “Exception”。

3. 详细日志分析

首先需要查看 Monkey 测试中是否出现了 ANR 或者异常。将执行 Monkey 生成的 log，从手机中导出并打开查看该 log；在 log 的最开始都会显示 Monkey 执行的 seed 值、执行次数和测试的包名。然后我们要分析 log 中的具体信息，查看 log 中第一个 Switch，主要是查看 Monkey 执行的是那一个 Activity。在下一个 switch 之间的，如果出现了崩溃或其他异常，可以在该 Activity 中查找问题的所在。

5. 基础 Monkey 的缺陷

1. 遍历界面有限

因为 Monkey 是不断输入伪随机事件流来测试应用，所以无法控制输入的事件，这导致有些较深的界面很难进入。最终导致测试遍历的界面有限。

2. 路径回环

在随机测试的过程中，如果程序中存在大量的环路，就会容易出现一直重复相同路径的现象，造成路径循环。

3. 难以进行针对性测试

因为 Monkey 对于测试事件和数据都是随机的，自定义程度不高，无法对指定界面和指定动作进行测试，所以有很大的局限性。

4. 无意义动作过多

当某界面可操作按钮较少时，会反复执行一些无意义的动作。

5. 容易在某些界面卡死

在 Monkey 测试中，如果 app 在某个界面难以退出，或者难以找到进一步搜索的方法时，就会导致 app 在这个界面上卡死。

二、实验环境

1. 设备平台

- Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
- Windows10
- jdk 17
- android-sdk
- 夜神安卓模拟器

由于没有安卓手机，所以我使用了安卓模拟器进行 Monkey 测试。从官网下载夜神安卓模拟器，进入设置，双击“版本号”来启动开发者模式

版本号

beyond1qlteue-user 7.1.2 PPR1.190810.011 700211101 release-keys

之后在“开发者模式”中打开 USB 调试选项

USB调试

连接USB后启用调试模式



2. Monkey 环境

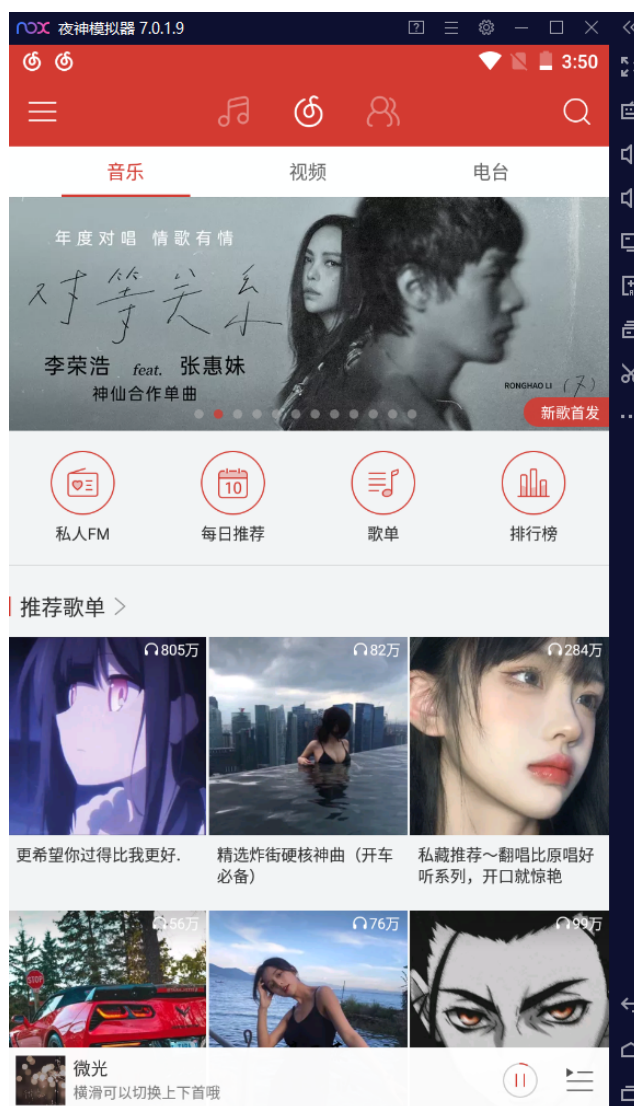
首先需要下载安装 JDK 和 SDK。

JDK 是 Java 语言的软件开发工具包，主要用于移动设备、嵌入式设备上的 java 应用程序。JDK 是整个 java 开发的核心，它包含了 JAVA 的运行环境（JVM+Java 系统类库）和 JAVA 工具。Android SDK 包含了许多可以帮助你开发 Android 平台应用的工具。这些工具分为两类：一是 SDK 工具；二是平台工具。SDK 工具独立于平台，任何开发 Android 应用的平台都需要配置。平台工具是定制的适应 7 最新的 Android 平台特性。

JDK 和 SDK 具体的安装过程参考了链接 [2] 和 [3] 的内容。

3. 应用选择

我选择了网易云音乐 APP 进行测试。网易云音乐有多种功能模块，如音乐、视频、电台等，且它的界面也特别多，所以在测试时有较高的价值意义。



三、基础 Monkey 测试

1. 测试过程

首先将之前安装的SDK\platform-tools下的 adb.exe 文件替换成夜神模拟器 bin 目录下的 adb.exe，打开安卓模拟器和网易云音乐。

在夜神模拟器 bin 目录下的终端输入

```
1 PS D:\Nox\bin> nox_adb.exe connect 127.0.0.1:62001
2 already connected to 127.0.0.1:62001
```

接下来在任意位置的终端输入

```
1 PS C:\Users\DELL> adb connect 127.0.0.1:62001
2 already connected to 127.0.0.1:62001
```

检查是否连接成功

```
1 PS C:\Users\DELL> adb devices
2 List of devices attached
3 already connected to 127.0.0.1:62001
```

查看测试的安装包名

```
1 PS C:\Users\DELL> adb shell logcat|findstr "Displayed"
2 12-10 17:10:12.213 2169 2189 I ActivityManager: Displayed com.android.settings/.
   ↳ FallbackHome: +356ms
3 12-10 17:10:14.722 2169 2189 I ActivityManager: Displayed com.android.launcher3/.
   ↳ launcher3.Launcher: +1s390ms
4 12-10 17:10:23.781 2169 2189 I ActivityManager: Displayed com.netease.cloudmusic/.
   ↳ activity.MainActivity: +840ms (total +4s219ms)
```

可以发现测试的 APP 相关内容，其中 com.netease.cloudmusic 是指包名，接下来在指定这个包，输入如下的命令，此时指定软件（网易云音乐）就会被打开然后开始各种随机操作。（-v-v-v 表示日志级别 2，是最详细的日志，300 表示随机进行 300 次操作）

```
1 PS C:\Users\DELL> adb shell monkey -p com.netease.cloudmusic -v-v-v 300
```

2. 日志分析

(i) 测试命令

这一部分的内容包括解析命令参数、随机种子值、可运行的应用列表、包含的 Category、各随机事件的百分比：

```
1 :Monkey: seed=1639264421487 count=300
2 :AllowPackage: com.netease.cloudmusic
3 :IncludeCategory: android.intent.category.LAUNCHER
```



```

4 :IncludeCategory: android.intent.category.MONKEY
5 // Event percentages:
6 // 0: 15.0%
7 // 1: 10.0%
8 // 2: 2.0%
9 // 3: 15.0%
10 // 4: -0.0%
11 // 5: -0.0%
12 // 6: 25.0%
13 // 7: 15.0%
14 // 8: 2.0%
15 // 9: 2.0%
16 // 10: 1.0%
17 // 11: 13.0%

```

其中各个序号对应的事件和操作如下：

- 0: 触摸事件百分比，参数-pct-touch
- 1: 滑动事件百分比，参数-pct-motion
- 2: 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: nav 导航事件，就是上下左右，参数-pct-nav
- 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: 系统事件百分比，参数-pct-syskeys
- 9: Activity 启动事件百分比，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

(ii) 伪随机事件流

测试开始后，其执行的随机事件流信息也会输入到日志文件中。比如在上面的事件百分比之后跟着的是启动应用事件：

```

1 :Switch: #Intent;action=android.intent.action.MAIN;category=android.intent.
    ↪ category.LAUNCHER;launchFlags=0x10200000;component=com.netease.cloudmusic/.
    ↪ activity.LoadingActivity;end
2 // 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

```

```

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

```

接下来是一系列点击、轨迹球等事件的日志，包括每一步点击的坐标等，信息十分完整。

```

1 :Sending Touch (ACTION_DOWN): 0:(671.0,1322.0)
2 :Sending Touch (ACTION_UP): 0:(699.0339,1317.7219)
3 :Sending Touch (ACTION_DOWN): 0:(398.0,1283.0)
4 :Sending Touch (ACTION_UP): 0:(388.94434,1267.6072)
5 :Sending Touch (ACTION_DOWN): 0:(509.0,1097.0)
6 :Sending Touch (ACTION_UP): 0:(510.136,1100.7361)
7 :Sending Touch (ACTION_DOWN): 0:(11.0,974.0)
8 :Sending Touch (ACTION_UP): 0:(8.18838,976.8228)
9 :Sending Trackball (ACTION_MOVE): 0:(2.0,-4.0)
10 :Sending Trackball (ACTION_MOVE): 0:(4.0,2.0)
11 :Sending Touch (ACTION_DOWN): 0:(417.0,1097.0)

```

在日志中也会有事件失败的信息，因为测试的随机性有时会发生一些无效的事件。

```

1 :Sending Touch (ACTION_DOWN): 0:(532.0,1076.0)
2 // Injection Failed
3 // Injection Failed
4 // Injection Failed
5 :Sending Touch (ACTION_UP): 0:(549.2026,1055.3192)
6 // Injection Failed

```

(iii) 异常信息

在 Monkey 测试的过程中也可能会发生各种异常或错误，Monkey 日志中常见的异常信息主要包括 ANR(应用程序无响应) 和 Crash(奔溃)。

- ANR：线程的响应超过 5 秒，或者 BroadcastReceiver 没有在 10 秒内作出响应，可以尝试把耗时的操作另起线程来处理。
- Crash：当应用程序出现错误时导致程序异常停止或退出的情况，通常会弹出对应的错误提示框，可以观察提示框之前的行为来定位日志并分析。

我们可以在日志中全局搜索是否含有这些异常的信息，如果测试中出现了错误，一般可以采取如下步骤进行排错：

1. 先找到出现错误的位置
2. 查看出现错误之前 2 个 switch 之间的 activity
3. 手动执行事件，复现问题
4. 若以上步骤还不能找出，输入与产生错误时相同的 seed 值，重新按照之前命令跑 Monkey，看是否能够再次复现。

经过多次测试，我发现我选择的网易云音乐并没有出现异常信息，说明其稳定性非常出色。

(iv) 结果信息

当所有事件执行结束后，日志末尾会出现结果信息，包括事件数量、旋转角度、丢失信息、网络状态等。

```

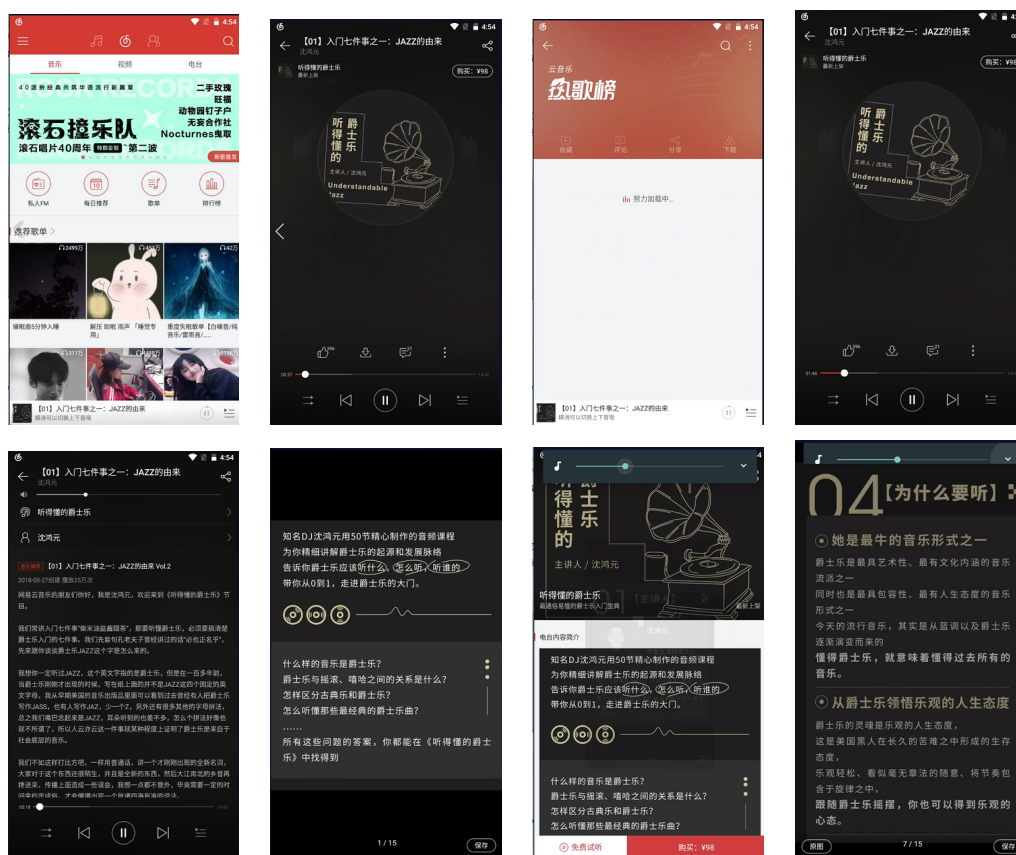
1 Events injected: 300
2 :Sending rotation degree=0, persist=false
3 :Dropped: keys=0 pointers=4 trackballs=0 flips=0 rotations=0
4 ## Network stats: elapsed time=5762ms (0ms mobile, 0ms wifi, 5762ms not connected)
5 // Monkey finished

```

在本次 Monkey 测试中，300 个随机事件全部完成，旋转角度为 0 表示未旋转，丢失事件数量中主要为 pointers 事件。如果发生了异常或其他错误而提前终止，此时的执行事件数量会小于指定数量，并且不会显示 Monkey finished。

3. 测试结果

基础 Monkey 随机测试时，网易云音乐的许多界面都会被遍历。然而也会遍历到某些界面后，就会执行一些无意义的操作：比如反复左滑右滑查看图片，一直在当前界面停留。具体如下图所示（执行一系列事件后的结果顺序为：第一行 1234，再第二行 1234）：



基础 Monkey 测试常常会发生这种情况，停留在一个界面的测试就没有意义，浪费时间。因此我们需要智能 Monkey 来进行“智能化”的测试。

四、智能 Monkey 测试

针对基础 Monkey 测试中容易在某一界面卡死、遍历界面有限、无意义动作过多的问题，有以下解决方案：在进行 Monkey 测试时选择软件内的所有有效按键部分进行点击，并且排除一些其他无效的事件，这样可以使得 Monkey 测试更加智能化且高效，大大减少了许多无意义的测试。另外我觉得可以记录过去尝试过的事件和结果，结合当前界面和过去的测试结果“智能化”地给出接下来的测试事件，使其得到更加充分的测试。

1. AppetizerIO

在智能 Monkey 的测试中，我使用了 AppetizerIO [5] 进行可视化的分析。AppetizerIO 是一个集成 App 测试以及数据分析平台，平台提供如下功能：

- 通过 USB 或者无线 ADB 方式管理设备，PC 端实时投屏
- 统一管理不同版本的测试包
- 提供 DEX 插桩技术，自动为 APK 自动增加收集错误、性能、网络抓包收集功能
- 提供智能编辑器编写 UI 自动化脚本
- 集成 Monkey 类以及一键启动专项测试等自动化测试方案

其自动化测试的工具是 Maxim [6]，是一个基于 Monkey 测试的可自定义并且封装了更多实用测试序列工具等第三方测试工具。

(i) Maxim 工具

Maxim 工具，是一种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，它相比于基础 Monkey 新增了许多特性：

- 防跳出及切回、防误点状态栏及下拉状态栏
- 防睡眠及睡眠唤醒、放假死及自拉活机制
- 场景细粒度控制，可以自定义 activity 黑白名单
- 多种随机测试模式
 - 模式 Mix (基于事件概率的压力测试)
 - 模式 DFS(深度遍历算法)
 - 模式 Troy(控件选择策略按 max.xpath.selector 配置的高低优先级来进行深度遍历)

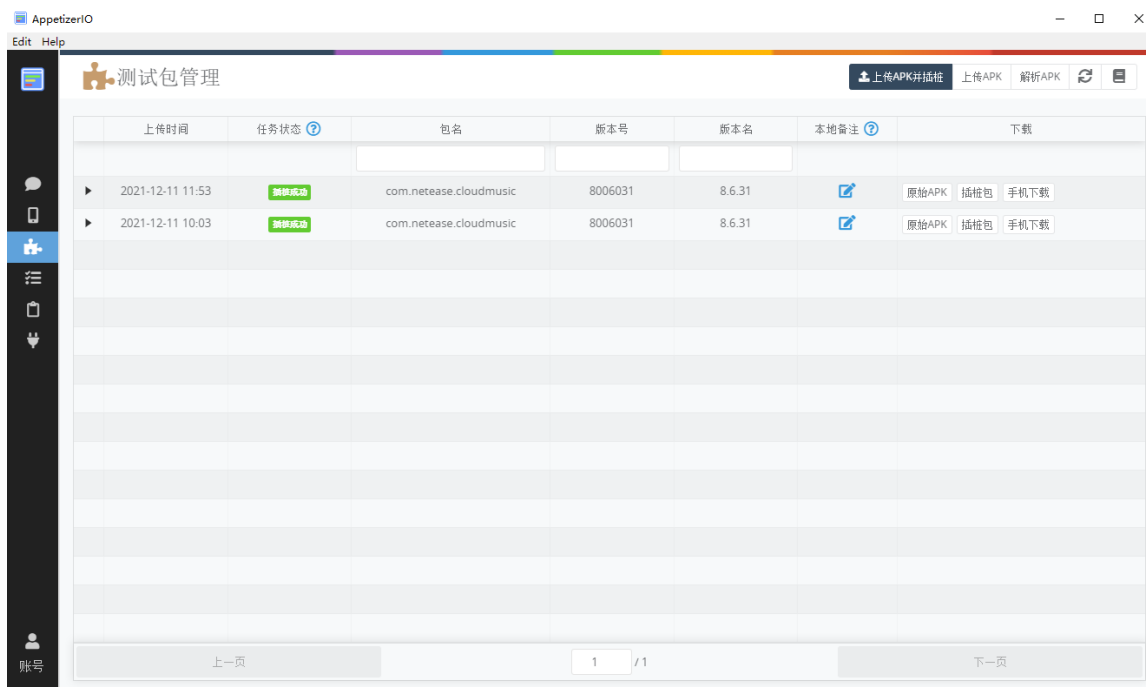
从 github [6] 上下载工具包后，我们首先要将其中的 monkey.jar 和 framework.jar push 到虚拟机安卓系统中的 /sdcard 目录。

```
1 adb push monkey.jar /sdcard/
```

```
2 adb push framework.jar /sdcard/
```

(ii) 测试过程

使用 AppetizerIO 进行数据分析，我们首先要将要测试的网易云音乐安装包上传进行插桩，添加数据收集功能：



接下来下载插桩后的安装包，在夜神安卓模拟器中卸载原来的网易云音乐客户端，安装插桩后的应用，准备进行 Maxim 的测试。

选定夜神安卓模拟器中的设备



设置待测 APP 为网易云音乐，可以设置界面的黑白名单以及是否允许访问的权限

待测APP (必选) 选择APK

网易云音乐 8.6.31

界面	操作
com.alibaba.security.biometrics.face.auth.FaceLivenessActivity	<input type="checkbox"/> 禁止遍历 <input type="checkbox"/> 允许遍历
com.alibaba.security.rp.activity.RPH5Activity	<input type="checkbox"/> 禁止遍历 <input type="checkbox"/> 允许遍历
com.alibaba.security.rp.activity.RPTakePhotoActivity	<input type="checkbox"/> 禁止遍历 <input type="checkbox"/> 允许遍历
com.alipay.sdk.app.AlipayResultActivity	<input type="checkbox"/> 禁止遍历 <input type="checkbox"/> 允许遍历
com.alipay.sdk.app.H5AuthActivity	<input type="checkbox"/> 禁止遍历 <input type="checkbox"/> 允许遍历
com.alipay.sdk.app.H5OpenAuthActivity	<input type="checkbox"/> 禁止遍历 <input type="checkbox"/> 允许遍历
com.alipay.sdk.app.H5PayActivity	<input type="checkbox"/> 禁止遍历 <input type="checkbox"/> 允许遍历
com.alipay.sdk.app.PayResultActivity	<input type="checkbox"/> 禁止遍历 <input type="checkbox"/> 允许遍历
com.baidu.mobads.AppActivity	<input type="checkbox"/> 禁止遍历 <input type="checkbox"/> 允许遍历
com.baidu.mobads.production.rewardvideo.MobRewardVideoActivity	<input type="checkbox"/> 禁止遍历 <input type="checkbox"/> 允许遍历

上一页 1 / 57 下一页

选择 Monkey 引擎为 Maxim，模式为 Mix 模式，指定测试时长为 1 分钟

Monkey引擎

Maxim maxim 1.4 文档

Fastbot-Android fastbot-android v1.0 文档

模式 Mix模式：识别控件并随机点击+随机点击屏幕位置

测试时长 指定测试时间 1 分钟

最后设置一系列参数，开始测试！

事件间延迟 1500 毫秒

日志级别 完整日志模式（有完整事件记录以及当前界面） ☐ 屏蔽stdout ☐ 屏蔽stderr

输出路径 选择输出目录 C:\Users\DELL\Desktop\log 打开输出目录

数据收集 ☐ 保存控制台输出 ☐ 保存logcat ☐ 保存Battery Historian数据 ☐ 保存内存快照（hprof）
☒ 保存插桩数据 ☐ Maxim日志与截图

测试前置步骤 测试开始前运行Shell命令
☐ 预授权所有的权限免除动态权限弹窗 ☐ 暂时禁用输入法

其他参数（高级功能） 请输入Monkey的其他参数

最终命令

```
CLASSPATH=/sdcard/monkey.jar:/sdcard/framework.jar exec app_process /system/bin tv.panda.test.monkey.Monkey -p com.netease.cloudmusic --running-
minutes 1 --throttle 1500 --uiautomatorMix -v -v --output-directory /sdcard/max-output
```

开始测试 保存测试配置 打开输出目录 退出编辑

2. 测试结果

测试开始后会显示完整的日志信息，这里选取部分进行说明。首先是完整的命令和一个安卓的 logo，接下来是随机种子和计数值，可以发现 Maxim 成功指定网易云音乐为测试目标。

```

1 CLASSPATH=/sdcard/monkey.jar:/sdcard/framework.jar exec app_process /system/bin tv
  ↪ .panda.test.monkey.Monkey -p com.netease.cloudmusic --running-minutes 1 --
  ↪ throttle 1500 --uiautomatormix -v -v --output-directory /sdcard/max-output
2 version:1.0.21.0530.2049
3
4      ..          ...
5      ..t..    Maxim    .h..
6      ..i.. . . . . . . . . .si
7      sandroidmonkeystreste
8      .sttoolMaxim.thistool.
9      .helpyoutofoundappcrash.thi
10     ..stooluserandomordfsalgorithm:.
11     .smonkeyMaxim machine gunmonkey..
12     ..monkey,..,monkeymonkeym,..,onkeym.
13     .onkeymon..keymon.keymonke..ymonkeym.
14     ,onkeymonkeymonkeymonkeymonkeymon.key.
15     .monkeymonkeymonkeymonkeymonkeymonkeym.
16     .onkeymonkeymonkeythxforphdguxiaotian.:
17     .anythingUcancontactpanda_zzaqq77227005
18     .
19
20     .
21 [Maxim] // Monkey: seed=1639389262119 count=1000
22 [Maxim] // Targeted packages and permissions:
23 [Maxim] //
24 [Maxim] // Fuzz Null Intent:
25 [Maxim] // com.netease.cloudmusic.activity.LoadingActivity
26 [Maxim] // com.netease.cloudmusic.wxapi.WXEntryActivity
27 [Maxim] // com.netease.cloudmusic.wxapi.WXPayEntryActivity
28 [Maxim] // com.netease.cloudmusic.wbapi.WBEntryActivity
29 [Maxim] // com.netease.cloudmusic.yxapi.YXEntryActivity
30 [Maxim] // com.netease.cloudmusic.activity.RedirectActivity
31 [Maxim] // com.netease.cloudmusic.activity.IdentifyShortcutActivity
32 [Maxim] // com.netease.cloudmusic.activity.RadioShortcutActivity
33 [Maxim] // com.tencent.tauth.AuthActivity

```

然后是各个随机事件的百分比，可以发现其百分比相比基础 Monkey 有了更科学的设计。因为触摸是我们在使用 APP 是最常用的，所以其百分比从 15.0% 提高到了 62.5%；基本导航和主要导航因为在 APP 中不会经常去使用，所以将其百分比从 25.0% 和 15% 直接降低为 0%；其他事件的百分比也做了一定的调整。

```

1 // Event percentages:
2 // TOUCH : 62.5%
3 // MOTION : 10.0%
4 // PINCHZOOM : 4.0%
5 // TRACKBALL : 5.0%
6 // ROTATION : 5.0%
7 // PERMISSION : -0.0%
8 // NAV : -0.0%
9 // MAJORNAV : -0.0%
10 // SYSOPS : -0.0%
11 // APPSWITCH : 1.0%
12 // FLIP : 1.0%
13 // ANYTHING : -0.0%
14 // BACK : 10.0%
15 // MUTE : 1.0%
16 // SCHEMA : -0.0%
17 // NULLINTENT : 0.2%
18 // RESET : 0.3%

```

接下来是一系列的日志信息，日志内容与 Monkey 测试十分类似。仔细观察发现，其有效点击的频率大大提高，并且在我的测试中没有出现 Injection Failed 这样的事件失败信息，对软件的有效组件的测试率大大提高，相比于原始的 Monkey 测试更加智能化。

```

1 [Maxim] // : debug, currentActivity is com.netease.cloudmusic.activity.
   ↳ PlayerProgramActivity
2 [Maxim] // event0, 2021-12-11 15:24:47.555
3 [Maxim] // Switch: #Intent;action=android.intent.action.MAIN;category=android.
   ↳ intent.category.LAUNCHER;launchFlags=0x10200000;component=com.netease.
   ↳ cloudmusic/.activity.LoadingActivity;end
4 [Maxim] // the top activity is com.netease.cloudmusic.activity.LoadingActivity,
   ↳ phone launcher activity is com.android.launcher3.launcher3.Launcher
5 [Maxim] // the top activity is com.netease.cloudmusic.activity.LoadingActivity,
   ↳ phone launcher activity is com.android.settings.FallbackHome
6 [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
7 [Maxim] // event1, 2021-12-11 15:24:47.561
8 [Maxim] // Sleeping for 1500 milliseconds
9 [Maxim] // event1, 2021-12-11 15:24:49.075
10 [Maxim] // Sleeping for 6000 milliseconds
11 [Maxim] // event1, 2021-12-11 15:24:55.082
12 [Maxim] // Sleeping for 1500 milliseconds
13 [Maxim] Random Number = 0.6975024
14 [Maxim] random hit motion

```

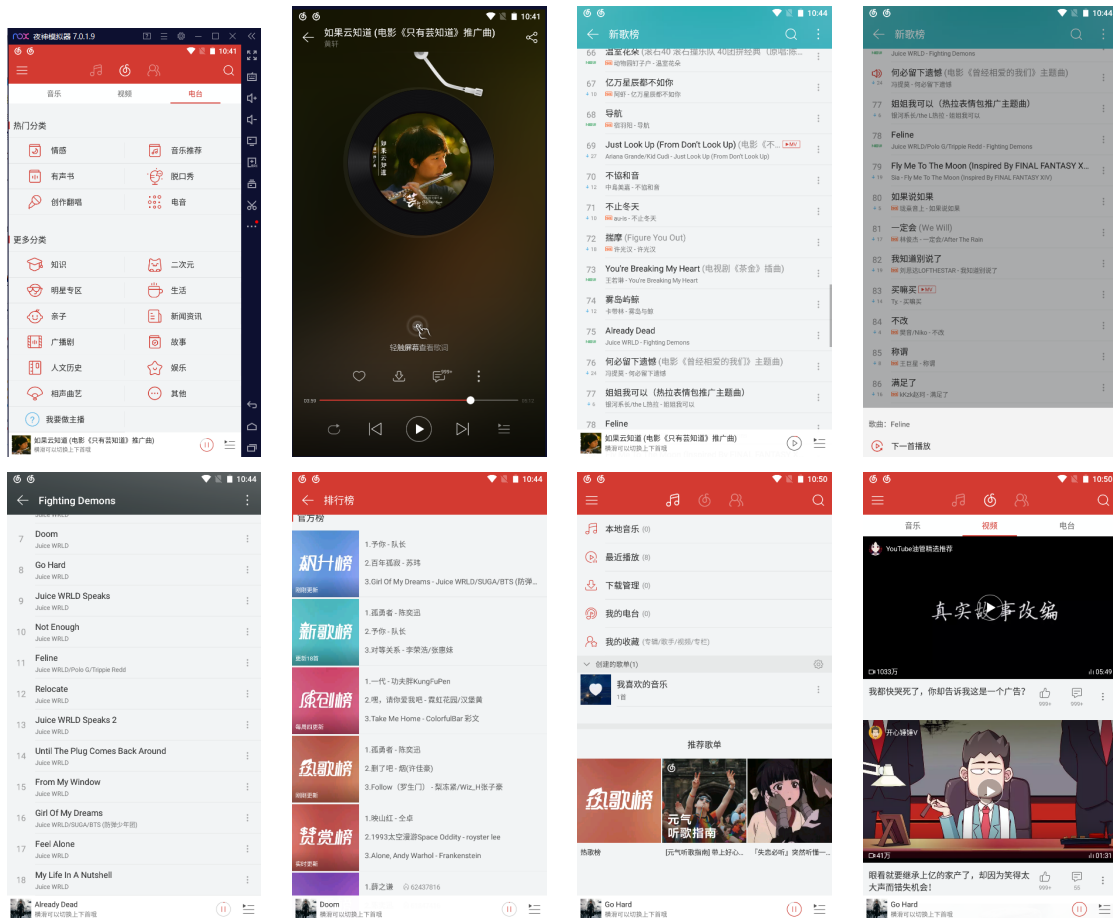


```

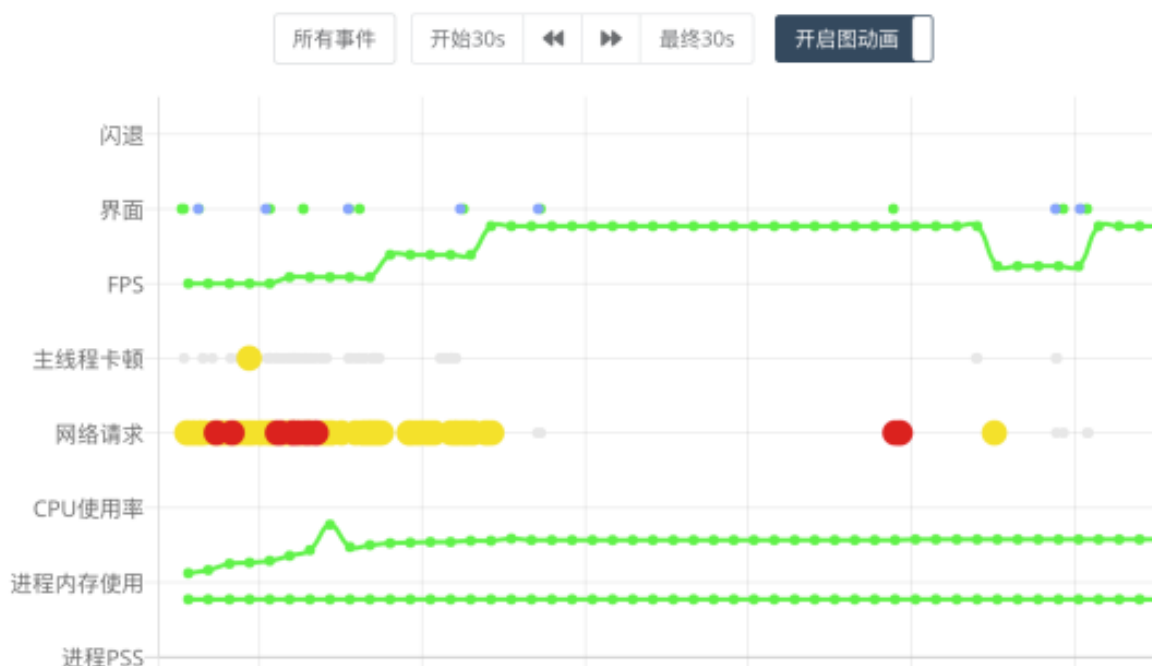
15 [Maxim] // Click: null, android.widget.TextView, (852.0,100.0)
16 [Maxim] // event1, 2021-12-11 15:24:56.643
17 [Maxim] :Sending Touch (ACTION_DOWN): 0:(852.0,100.0)
18 [Maxim] // event2, 2021-12-11 15:24:56.659
19 [Maxim] :Sending Touch (ACTION_MOVE): 0:(846.4106,88.52913)
20 [Maxim] // event3, 2021-12-11 15:24:56.662
21 [Maxim] :Sending Touch (ACTION_UP): 0:(845.42615,79.64305)

```

观察网易云音乐在测试时的画面可以发现，相比于基础 Monkey 测试总是在某些界面难以退出，导致一直在做无意义的动作、遍历界面过少等问题，智能 Monkey 的测试由于加入了智能控制，使得测试不会卡在某一界面长时间不变，并且通过调整动作比例来减少无意义动作，既不会停在几张图片中左划右划，也不会出现之前乱按音量键的情况。总而言之，智能 Monkey 测试的效率明显高于基础 Monkey。



测试完成后，我们将生成的详细的报告文件，导入到 AppetizerIO 中，即可得到如下可视化的性能数据分析：



可以发现整个测试过程中 FPS 一直比较稳定, CPU 使用率和进程内存使用也比较正常。但是在网络方面发生了一些错误, 包括 HTTP 错误和慢响应等问题, 观察测试时的网易云音乐界面发现其正在登陆一些不存在的关联应用账号, 可能这就是问题的来源。但网易云音乐总体的稳定性还是十分良好的, 可见网易公司的强大和制作的用心。

五、总结

本次实验是我第一次接触自动化测试的内容, 我首先调研学习了 Monkey 测试, 实现基础 Monkey 测试并思考其不足, 最后提出可改进的地方并实现智能化 Monkey 测试。

通过阅读详细的 Monkey 官方文档, 让我对整个 Monkey 测试有了丰富的认识, 其各式各样的命令可以使 APP 得到广泛的测试。然而基础 Monkey 也有许多不合理之处, 比如事件的比例设置就不太合理, 导致其经常进入某些界面就卡死, 因此我们就需要智能化 Monkey 测试“智能”地进行触摸、划动, 从而提高测试的效率。

在整个实验过程中, 我也遇到了许许多多的问题, 好在最后都逐一解决:

1. 配置 JDK 和 SDK 的环境变量时, 我本来将同个环境变量的多个路径通过分号设置为一条, 结果却配置失败了, 百思不得其解。最后将它们分成多条环境变量就配置成功了, 这可能是 Windows 系统的问题。
2. 在 adb 连接时我的 android sdk 与夜神安卓模拟器无法连接, 后来发现需要将 sdk 中的 platform-tools 文件夹下的 adb.exe 文件, 替换成夜神模拟器 bin 目录下的 adb.exe 文件, 就能成功连接了。

3. 开始在查找安卓模拟器的 sdcard 文件夹时在 Windows 中没有找到，后来发现安卓模拟器是通过一个 shared 文件夹与 Windows 电脑互通的，将电脑中的文件传到安卓模拟器可以先把文件传入 shared 文件夹，再通过安卓中的目录移动文件。

总而言之，通过本次实验，我对 Monkey 测试有了更加深入的了解，也通过自己实践的方式实现了基础 Monkey 和智能 Monkey 的测试，收获颇丰。

参考文献

- [1] UI/Application Exerciser Monkey.<https://developer.android.google.cn/studio/test/monkey>, 2021.
- [2] JDK 安装.<https://blog.csdn.net/write6/article/details/79136388>, 2018.
- [3] Android Monkey 测试入门.<https://blog.csdn.net/write6/article/details/79136388>, 2017.
- [4] Monkey 测试总结.https://blog.csdn.net/qq_36155340/article/details/78114040, 2017.
- [5] AppetizerIO.<https://www.notion.so/appetizerio/AppetizerIO-1383d971ded94388b8e76f3236fe1f4f>, 2021.
- [6] Maxim.<https://github.com/zhangzhao4444/Maxim>, 2021.