

Android 应用录制与回放工具设计^①

闫 凯, 孙军梅, 刘雪娇, 朱 敏

(杭州师范大学 杭州国际服务工程学院, 杭州 311121)

通讯作者: 孙军梅, E-mail: junmeisun@hznu.edu.cn

摘 要: 智能手机以及平板凭借其便捷的丰富的输入功能越来越受到用户的欢迎, 但是它们丰富的输入功能增加了测试的复杂性. 现有的基于 GUI 级别的录制与回放工具满足不了 Android 应用程序可以从设备上的各种传感器获取输入, 识别 GUI 手势以及一些对时间有精确要求的录制和回放需求. 本文设计并且实现了一个工具: RARA. RARA 可通过直接捕获手机底层的 GUI 事件以及传感器事件进行应用的录制, 并且以微秒时间精度回放. 最后通过实验验证了: (1) RARA 录制与回放功能是有用的; (2) 回放的时间开销只有 1% 左右, 不会对宿主 APP 的性能产生影响; (3) 通过 RARA 的录制回放功能, 可在多款应用程序的测试中重现应用 Bug.

关键词: 软件测试; Android; 自动化测试; 录制; 回放

引用格式: 闫凯, 孙军梅, 刘雪娇, 朱敏. Android 应用录制与回放工具设计. 计算机系统应用, 2018, 27(9): 249-255. <http://www.c-s-a.org.cn/1003-3254/6505.html>

Design of Recording and Replay Tools for Android Application

YAN Kai, SUN Jun-Mei, LIU Xue-Jiao, ZHU Min

(Hangzhou Institute of Service Engineering, Hangzhou Normal University, Hangzhou 311121, China)

Abstract: Smart phones and tablets are gaining popularity with their rich and powerful input, but their rich input features add to the complexity of the testing. Existing GUI-based recording and playback tools are inadequate for capturing the input of sensor devices, GUI gestures, and have precise timing requirements. This study designs and implements a tool named RARA. RARA can be recorded and replay both GUI events and sensor events, and replaying with microsecond accuracy. Finally, it is verified by experiment: (1) RARA is effective; (2) The playback time overhead is only about 1%, will not affect the performance of the host APP; (3) The application bug can be reproduced through the RARA.

Key words: software testing; Android; automatic testing; record; replay

随着移动互联网的日益成熟, 移动设备已经成为了许多应用程序的载体, 从办公自动化到游戏, 再到社交工具, 无不依托于移动设备的便捷. 移动平台的吸引力主要源自于用户可以通过触摸屏实现与系统的交互, 以及通过多样化的传感器 (如加速度传感器, 罗盘, GPS) 驱动程序的行为, 实现多样化的功能. 然而, 这些新特性也给软件开发、测试和维护带来了重大挑战.

录制和回放对于许多软件工程的任務都是很有价值的, 可以进行程序的调试、测试以及对应用程序的分析. 然而, 现有的基于 GUI 的应用程序的录制与回放工具却不能满足目前智能手机平台的要求. 基于 GUI 的录制回放工具主要是通过离散的捕获用户的高级动作, 不能够详细的表达复杂的手势动作, 例如滑动, 手指的缩放等, 也不支持传感器事件的回放.

① 基金项目: 国家自然科学基金 (61502134); 杭州市科技计划项目 (20170533B04)

Foundation item: National Natural Science Foundation of China (61502134); Science and Technology Plan Project of Hangzhou Municipality (20170533B04)

收稿时间: 2018-01-11; 修改时间: 2018-01-31; 采用时间: 2018-02-06; csa 在线出版时间: 2018-08-16

本文设计实现了一个 Android 平台的录制回放工具 RARA (Record And Replay for Android). RARA 通过直接捕获和重现用户在使用 Android 应用过程中的底层事件流, 不仅能捕获和回放应用 GUI 的触屏事件, 如点击, 输入等, 而且能捕获交互的手势事件, 如滑动、手指的缩放动作; 此外 RARA 也能捕获传感器事件, 如加速度传感器、光传感器、指南针等.

1 Android 事件介绍

Android 是基于 Linux 的开放源代码的操作系统, 在 Linux 系统上增加了 Dalvik 虚拟机, 每一个 Android 应用都运行在一个 Dalvik 虚拟机实例里, 而每一个虚拟机实例都是一个独立的进程空间. 当用户与 Android 应用交互时, Android 设备的传感器通过 /dev/input/event* 设备文件生成事件并且发送到系统内核. 常见的触摸屏操作主要包括: 快速点击-释放、长按、滑动、缩放四种操作. 其中三项涉及到手势: 长按, 滑动, 缩放. 触摸屏手势需要大量的底层触摸屏事件来提供它们的感知功能.

事件采用五元组格式 (timestamp device: type code value) 表示, 其中 timestamp 表示距上一次系统重启的时间或者当前时间的戳, 原生 Android 系统表示距上一次系统重启的时间, 部分厂商经过修改表示当

前时间的戳; device 表示已创建事件的设备, 其余字段取决于具体的事件类型. 如:

[100.119269]/dev/input/event2: 0002 0035 000000ef 表示一条事件五元组编码, 其中:

[100.119269] 字段表示该事件距离手机最近一次启动的时间间隔为 100.119 269 s.

/dev/input/event4: 字段表示输入设备为 event2, 对应于本文的设备是触摸屏.

0002 0035 000000ef 分别指事件的 type(输入事件类型), code(键码类别), value(具体数值), 这三个字段表示了坐标信息, 0035 对应于事件的 X 坐标, 具体值为 000000ef(十六进制) 对应于屏幕的坐标为 239. Android 其它复杂的事件都是由上述标准的事件流格式组合而成. 一个触摸屏手势由若干个事件构成, 形成事件流. 例如, 单个按键操作通常涉及大约 18 个触摸屏事件, 而滑动通常涉及大约 80 个触摸屏事件. 图 1 所示为一个滑动的事件流详细信息, 此次滑动一共有 80 条基本事件流, 表示该次滑动在 154 546.327 764 s 至 154 546.588 003 s 时间段完成; 1-7 行是滑动事件的开始信息, 第 4 行, 5 行分别表示滑动开始时手指所按屏幕位置的横坐标以及纵坐标. 8-75 行表示在滑动过程中持续变化的位置的坐标信息; 76-80 行表示该次滑动结束时手指释放位置的横坐标以及纵坐标.



图 1 一次滑动事件流

时间序列是回放成功至关重要的因素. 来自传感器的事件以及触摸屏的事件都必须以高吞吐量捕获,

某一事件的缺失都会影响到事件回放的行为. 为了评估传感器以及触摸屏的事件量, 我们对 4 款常见应用

程序在同一台设备上 5 分钟内三个输入设备的事件量做了统计, 如表 1 所示, 其中触摸屏事件占绝大多数, 传感器也有大量的事件.

表 1 事件量统计

| 应用名称 | 屏幕触摸 | 光传感器 | 加速度传感器 |
|------|-------|------|--------|
| QQ | 6,199 | 7010 | 6 |
| 微信 | 5,895 | 6753 | 3 |
| 高德地图 | 4,723 | 2730 | 15 |
| 手机淘宝 | 4,831 | 7790 | 4 |

1.1 Android 应用录制与回放工具介绍

目前比较常见的录制与回放工具主要是采用基于关键字的录制与回放技术. 如: Android GUITAR^[1], Abbot^[2], HP WinRunner^[3], IBM Rational Robot^[4]和 GUI crawler^[5]就是采用关键字技术对用户的操作过程进行录制, 然后进行回放. 基于关键字技术的录制与回放工具将输入的事件从具体的 GUI 级别中抽象出来, 使用 GUI 对象在系统中的句柄或者名称与其进行交互. 例如 click buttonMenu 表示点击控件 ID 为 buttonMenu 的 button. 然而, 像愤怒的小鸟^[6], 王者荣耀^[7], 割绳子^[8]此类游戏主界面只有一个 Activity, 没有布局文件, 获取不到 GUI 组件对象. 所以基于关键字的录制技术并不能适用于此类场景. 另外一些如 Monkey^[9], MonkeyRunner^[10]能够模拟用户触摸屏幕、滑动 Trackball、按键等操作来对设备上的程序进行操作回

放. 但它们的脚本不支持触摸屏手势的操作. 文献[11]给出了一个基于手势的录制与回放工具 RERAN. 该工具能够录制和回放涉及手势和传感器的 Android 事件. 但该工具没有对不同厂家类型的手机时间戳表示进行统一. 对于不同类型的手机厂家收集的事件流有时不能正常回放. RARA 扩充了 RERAN 的功能. 在回放时用到了 RERAN 的回放代理工具. 本文的主要贡献如下:

- (1) 使用统一的事件流格式来表示 Android 事件流.
- (2) 对录制生成的事件流文件进行排序及关键信息提取, 生成可回放的脚本文件.
- (3) 录制与回放的时间精确到微妙, 时间增量只有 1% 左右.
- (4) 设计并实现了一个能录制并回放不同类型 Android 手机事件流的工具 RARA.

2 RARA 框架

图 2 为 RARA 的整体架构图, 录制和回放可以从任何一台 Android 设备开始, 可以使用 Android 提供的模拟器, 或者通过 USB 连接到计算机的手机. 通过使用 Android SDK 中的 ADB (Android Debug Bridge) 命令作为 RARA 与手机之间的接口. 通过 ADB 发送 shell 命令到手机系统进行录制与回放.

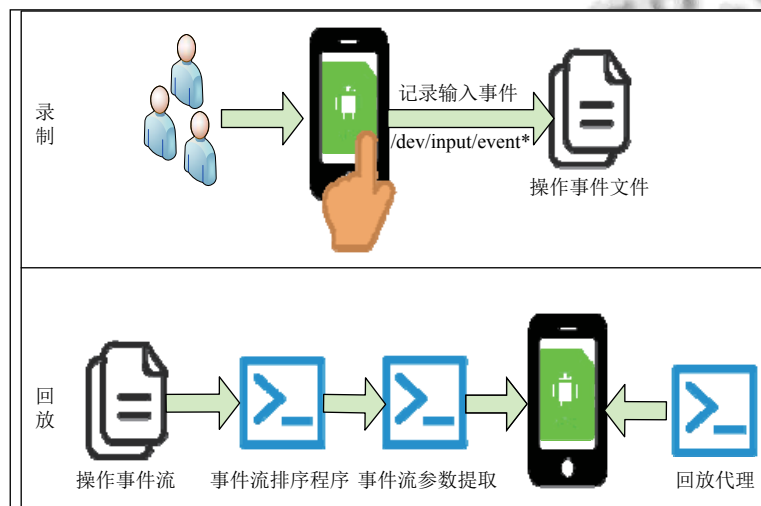


图 2 RARA 架构图

2.1 录制

RARA 使用 Android SDK 的 `getevent` 工具, 读取

`/dev/input/event *` 文件, 可以记录手机上输入事件的实时日志, 如图 2 录制部分, 在录制过程中, 首先利用

ADB (Android Debug Bridge) 启动收集命令, 将手机的输入实时的储存到录制文件中, 应用程序的性能在日志记录过程中不受影响; 一旦记录结束, 就会生成一个操作事件文件, 供回放时使用。

2.2 事件流格式统一

Android 平台的开源性导致 Android 系统的碎片化问题。RARA 提出的基于事件流的录制方法也受到了 Android 碎片的影响, 事件流格式不统一, 不同的品牌不同版本的 Android 操作系统捕获的事件流有所差异。

本文通过三星手机与小米手机来分析事件流之间的差异, 如图 3, 图 4 所示, 分别为三星设备事件流内容的样例以及小米手机录制到的事件流样例。它们都是标准的五元组事件流格式, 但是通过比较可以发现存在两处差异, 首先, 两份事件流文件中的时间字段格式不统一, 如三星事件流的时间字段为[100.839976], 代表手机距离上次启动的时间间隔为 100.839 976 s, 而小米手机事件流的时间字段为[149256649.144810], 代表捕获事件当时的时间戳; 时间字段不论从格式, 还是所代表的意义都有一定的差异; 其次, 三星手机相邻两条事件流中都是空白行, 而小米手机是隔行输出。虽然这些差异很细微, 但是这些差异给回放带来了巨大的障碍。

```
[ 100.659810]/dev/input/event4: 0002 0005 00000004
[ 100.659810]/dev/input/event4: 0000 0000 00000000
[ 100.839976]/dev/input/event4: 0002 0006 0000015b
[ 100.839976]/dev/input/event4: 0002 0007 000001d8
[ 100.659810]/dev/input/event4: 0002 0005 00000004
[ 100.839976]/dev/input/event4: 0002 0008 00000221
[ 100.839976]/dev/input/event4: 0002 0009 000004fd
```

图 3 三星设备事件流内容样例

```
[1491396430.240089]/dev/input/event2: 0002 0001 00000001
[1491396430.240095]/dev/input/event2: 0000 0000 00000000
[1491396430.300769]/dev/input/event2: 0000 0000 00000000
[1491396430.360168]/dev/input/event2: 0002 0001 00000001
[1491396430.360177]/dev/input/event2: 0000 0000 00000000
[1491396430.420200]/dev/input/event2: 0002 0001 00000001
[1491396430.300760]/dev/input/event2: 0002 0001 00000001
[1491396430.420209]/dev/input/event2: 0000 0000 00000000
[1491396430.480199]/dev/input/event2: 0002 0001 00000001
```

图 4 小米手机事件流内容样例

本文针对数据流格式差异的问题, 实现了时间格式转换器程序, 将事件流信息转换为统一格式, 将

时间字段的数字单独提取出来, 统一为 149256649.144810/dev/input/event2: 0002 0001 00000001, 用于重现, 使得回放程序的兼容性更强, 统一后的格式如图 5 所示。

```
1492566448.604453 /dev/input/event2: 0002 0001 00000001
1492566448.604459 /dev/input/event2: 0000 0000 00000000
1492566448.604442 /dev/input/event2: 0002 0001 00000001
1492566448.664447 /dev/input/event2: 0000 0000 00000000
1492566448.724421 /dev/input/event2: 0002 0001 00000001
1492566448.724424 /dev/input/event2: 0000 0000 00000000
1492566448.784410 /dev/input/event2: 0002 0001 00000001
1492566448.784414 /dev/input/event2: 0000 0000 00000000
1492566448.844412 /dev/input/event2: 0002 0001 00000001
1492566448.904429 /dev/input/event2: 0002 0001 00000001
1492566448.904434 /dev/input/event2: 0000 0000 00000000
```

图 5 统一格式以后的事件流

2.3 回放

Android SDK 提供了一个 sendevent 工具, 允许开发人员将单个输入事件命令发送到手机, 但是, 在实验过程中发现, 当输入一系列连续的事件流时, sendevent 会有时间滞后, 事件之间的时间间隔不精确, 无法真实的回放录制的事件流, 对于涉及手势或者传感器交互时, 就需要精确的物理坐标和事件之间的时间间隔; 如果回放期间时间没有严格控制, 则会产生许多错误的结果, 例如, 滑动中间的延迟将会导致滑动变为一长串长按; 所以, sendevent 不能够准确的回放事件流程。以准确的时间回放事件是一个关键技术, 比如, 回放滑动通常需要 80 或者更多的事件, 而对于每次滑动通常时间小于 1 s, 并且滑动中的各个事件之间的时间间隔通常小于 60 μ s。

2.3.1 基于时间节点的事件流排序器

在录制过程中, 录制程序实时对用 Android 的所有输入设备进行采集并且写入文件, 时间精确到 μ s; 但是, 当数据写到文件中时, 由于终端无缓冲, 高频输出事件流, 导致事件流文件的时间节点会乱序, 图 3, 图 4 中框选的事件流为乱序的时间节点。在回放过程中, 提取事件的关键信息时, 相邻事件间出现时间间隔为负数, 导致回放屡屡失败。因此, RARA 实现了基于时间节点的操作日志排序器, 在对事件流文件进行转换之前, 首先根据时间节点对录制到的操作事件流按照时间戳升续重新排序, 保证事件执行过程中的时间间隔不能为负数, 使得事件流信息能够完整的实时回放。

2.3.2 事件流关键参数提取 (生成回放的脚本)

如图 2 所示, 在回放阶段, 首先对事件流文件进行

格式统一,其次通过基于时间节点的事件流排序器对事件流文件进行排序.在完成上述步骤以后,RARA实现了对事件流文件关键信息的提取,提取出每条事件流的关键信息,以此生成回放的脚本.通过图2中的回放代理对该脚本的读取进行事件的重现.如:

```
[100.119269]/dev/input/event2: 0002 0035 000000ef
```

该条事件流提取完的信息为 2, 2, 35, 289 分别代表事件的输入设备为 event2, type (输入事件类型为 2), code (键码类别为 35), value (具体参数数值为 289).

事件流文件提取完事件参数生成的重现脚本如图6所示,每个脚本的第一行的数字表示该脚本总共包含的事件数目,图6所示的事件流包含 1549 条事件流信息.从第二行开始,所有的偶数行只有一个数字,代表与上一个事件的时间间隔,单位为毫秒;所有的奇数行有四个数字,分别代表事件的输入设备,事件 type, 事件 code 码,以及事件 value 值.脚本执行的过程中就是循环读取事件的参数,然后进行事件注入.

```
1549
0
2,2,1,1
5888
2,0,0,0
59983104
2,2,1,1
4864
2,0,0,0
59974144
2,2,1,1
2816
2,0,0,0
59986176
2,2,1,1
3840
2,0,0,0
```

图6 重现脚本样例

2.3.3 回放代理

RARA 采用 RERAN 的回放代理技术进行回放.回放代理工具由 C 语言编写,并且用交叉编译 arm-elf-gcc 进行编译,通过 Android 的 sendevent 工具源代码作为指导,确定如何与设备进行交互,回放代理工具通过读取事件回放脚本,把事件流信息写入到/dev/input/event*中,对手机设备进行事件注入,通过手机设备生成事件.回放代理运行在手机设备上,可以满足回放手势所需的微秒级别的精度.

2.3.4 回放

如图2所示,通过 ADB 命令将回放脚本以及回放

代理工具发送到手机设备的/data/local/目录下面,通过 ADB shell 进入手机系统,首先利用 su 命令获取 root 权限,其次,输入命令行/data/local/Replay.exe(回放代理工具)/data/local/translateEvent.txt(回放脚本)用于完整的事件回放.

3 实验验证及分析

本节主要对 RARA 的功能以及性能进行了实验验证,主要验证了如下三个问题:

- (1) RARA 工具是否有效?
- (2) RARA 在录制回放时对宿主 APP 的影响?
- (3) RARA 是否有助于发现软件中的缺陷?

实验设备采用表2所列的 10 台设备,详细的配置信息如表2所示.

表2 实验设备信息表

| 设备型号 | 操作系统 | CPU | RAM 容量 |
|-----------------|---------------|-------|--------|
| 红米 | Android 4.4.2 | ARMv7 | 1 GB |
| 三星 note4 | Android 5.0.2 | ARMv7 | 3 GB |
| Galaxy S6 edge+ | Android 5.1.1 | ARMv7 | 4 GB |
| Smartisan T1 | Android 4.4.2 | ARMv7 | 2 GB |
| Vivo Y31A | Android 5.1.1 | ARMv7 | 1 GB |
| 华为 P8 | Android 5.0.1 | ARMv8 | 3 GB |
| 荣耀平板 2 | Android 5.0.1 | ARMv7 | 2 GB |
| 三星 note3 | Android 4.3 | ARMv7 | 3 GB |
| 小米 Note | Android 5.0.2 | ARMv7 | 4 GB |
| 小米 4C | Android 5.1.1 | ARMv7 | 2 GB |

3.1 录制回放实验验证

实验选取了豌豆荚应用市场上 20 款不同类型的应用程序进行录制与回放,应用包括了通讯社交类、视频影音类、游戏类、系统工具类应用.每款应用的录制不少于 3 分钟,实验步骤如下:

- (1) 通过 USB 数据线连接实验设备到计算机,输入 adb devices 验证连接是否成功;
- (2) 启动要录制的的应用,同时启动 getevent 工具录制事件流信息;
- (3) 操作应用不少于三分钟,结束录制,保存事件流信息到文件夹;
- (4) 利用编译完成的程序对事件流的格式进行统一;
- (5) 基于时间节点的事件流排序器对步骤 4 完成的事件流信息进行排序;
- (6) 经过脚本生成器对事件流的关键参数进行提

取,生成回放脚本;

(7) 利用 ADB push 命令将生成的重现脚本与重现代理工具传送到录制的 Android 设备上,启动回放。

通过上述步骤的录制与回放,实验结果成功完整地回放了其中的 14 款,表 3 为成功录制回放的 14 款应用名称,表 4 为失败的 6 款应用名称。RARA 成功的录制与重现了割绳子与愤怒的小鸟此类对时间精确度要求很高的游戏类应用。首先,这两款游戏的实现用到传感器以及物理计算,时间精确度非常高,每次手势操作的时机不同,结果也是有所差别的。其次,这两款游戏的布局文件实现是通过在运行时绘制的,游戏界面没有控件元素,基于对象的录制方法是无能为力的。通过对 6 款未成功录制回放的应用程序进行分析,原因主要有两点:1) 需要 Android 传感器服务。例如,无法回放音频录制类应用,因为此类应用需要从麦克风捕获音频数据,在 Android 中,这些类型的输入来自传感器,不像触摸屏和加速度计(使用/dev/input/event*)一样。2) 界面元素动态或者随机出现的应用,RARA 不能捕获非确定性的界面,例如,通过动态布局或弹出窗口,网络连接变化或者使用了非确定性的算法(如使用随机数生成器)。通过表 4 可以看出这些界面动态或者随机变化的应用大部分是游戏类应用,比如像游戏 2048,每次游戏开始的初态是动态变化的,游戏数字的产生都是随机的。

表 3 成功录制与回放的应用

| 应用类别 | 应用名称 |
|-------|-------------------------|
| 通讯社交类 | 微信、QQ、微博、陌陌 |
| 视频影音类 | 腾讯视频、爱奇艺、虾米音乐、 |
| 系统工具类 | UC 浏览器、QQ 浏览器、搜狗输入法、计算器 |
| 游戏类 | 割绳子、愤怒的小鸟、一笔画 |

表 4 未成功回放的应用名称

| 失败原因 | 应用名称 |
|------------------|----------------------|
| 需要 Android 传感器服务 | 微会、有信 |
| 动态或者随机界面 | 2048、逃离神庙、开心消消乐、地铁跑酷 |

3.2 时间与空间开销

本文通过对表 5 所示的 10 款应用程序的时间与空间开销进行了测量,每款应用分别由 5 位不同的用户操作,分别对 5 位用户的操作进行录制与回放。实验结果如表 5 所示。首先,我们将原始执行的完成时间与重播执行的完成时间进行了比较。第 2 和第 3 列显示

了原始运行和重播运行的执行时间的平均运行时间;第 4 列显示了所有情况下的时间开销,接近 1%。回放开销很大程度上是由于事件注入不是即时的。在捕获期间,有时输入设备将事件同时传递,然而,由于事件是以程序方式注入的,而不是由物理设备生成,捕获的同时事件只能在 500 μs 内重播。尽管如此,它在应用程序行为上没有明显的差异。最后一列显示捕获事件的平均日志大小。

表 5 录制回放时间与空间开销

| 应用名称 | 平均录制时间 (s) | 平均回放时间 (s) | 时间增量 (%) | 日志大小 (KB) |
|--------|------------|------------|----------|-----------|
| QQ | 230.21 | 231.61 | 0.61 | 247 |
| 微信 | 253.13 | 254.45 | 0.52 | 232 |
| 微博 | 224.1 | 226.05 | 0.87 | 184 |
| 割绳子 | 233.15 | 234.04 | 0.38 | 176 |
| 计算器 | 186.25 | 186.64 | 0.21 | 86 |
| 一笔画 | 291.23 | 293.99 | 0.95 | 135 |
| 腾讯视频 | 246.33 | 247.64 | 0.53 | 196 |
| QQ 浏览器 | 192.26 | 194.37 | 1.1 | 103 |
| 虾米音乐 | 244.05 | 244.95 | 0.37 | 156 |
| 愤怒的小鸟 | 253.31 | 254.48 | 0.46 | 318 |

3.3 RARA 重现 Bug 能力的测试

重现是错误修复的关键。RARA 通过记录执行,直到遇到错误,然后重播录制,为重现错误提供了重要帮助。我们使用常见的 Android 应用程序中的实际错误来测试 RARA 的 Bug 再现功能。首先在 GitHub 上收集带有 Bug 的 Android 程序,通过 Issue 中提供的重现步骤,进行手工重现,同时使用 RARA 记录输入操作,当手动能够重现 Bug 时,通过 RARA 进行重放收集的事件输入,手动重现的 2 种错误,在回放期间,全部重现了。重现的 Bug 具体如下:

(1) Android 空指针异常 (NullPointerException) 造成的闪退,在 GeekNews V1.01 版本中, listView 加载网络数据的时候出现了空指针异常,由于在 listView 的适配器 getView() 方法中,把 convertView 返回 null 导致。

(2) Android 中数组越界异常 (IndexOutOfBoundsException) 造成闪退,在 Calculator 5.4 版本中,由于计算器历史结果数目超过了 listView 的长度,导致从计算器的主界面跳转到历史结果界面时出现闪退。

RARA 可以录制回放由于手势操作而触发的闪退,而其他 Android 测试工具无法处理,如 AndroidGUITAR^[2]

就无法重现 GeekNews V1.01 版本中的空指针异常造成的用户手势操作时触发的闪退现象. Android GUITAR 不支持触摸屏手势, 例如滑动和缩放, 或其他输入设备, 例如加速度计和罗盘.

4 结论与展望

本文设计并实现了一个 Android 应用录制与回放工具: RARA. RARA 通过直接捕获手机设备上底层的事件流信息, 并进行统一格式处理、排序、关键元素提取处理后, 生成可进行回放的脚本, 实现 Android 应用的录制与回放功能. 实验验证 RARA 能成功回放豌豆荚应用市场上的多款应用程序. 与传统的基于关键字技术的录制回放工具相比, RARA 可以广泛的应用于复杂 GUI 手势的精确回放. 而基于关键字技术的录制回放工具当涉及到触摸屏手势的很多维度的关键信息时, 显得无能为力. 因此, RARA 对于触摸屏的手势以及传感器事件录制回放具有很大的优势. 此外, 本文的方法可以用于软件测试时 Bug 的重现, 通过录制回放有效的提高软件测试人员的测试效率.

参考文献

- 1 SourceForge, Android GUITAR. August 2012, [http://sourceforge.net/apps/mediawiki/guitar/index.php?title=Android GUITAR](http://sourceforge.net/apps/mediawiki/guitar/index.php?title=Android+GUITAR).
- 2 Timothy Wall. Abbot framework for automated testing of Java GUI components and programs. <http://abbot.sourceforge.net>. [2012-08].
- 3 Hewlett-Packard Company. HP Functional Testing. [2012-08].
- 4 IBM. Rational Robot. <http://www.ibm.com/software/awdtools/tester/robot/>. [2012-08].
- 5 Amalfitano D, Fasolino AR, Tramontana P. A GUI crawling-based technique for android mobile application testing. 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops. Berlin, Germany. 2011. 252–261. [doi: 10.1109/ICSTW.2011.77]
- 6 Angrybirds. <http://www.wandoujia.com/apps/com.xk001.angrybirds>.
- 7 王者荣耀. <http://pvp.qq.com/>.
- 8 割绳子. http://www.wandoujia.com/apps/com.yodo1.ctr2.UC_02.
- 9 Monkey. <http://developer.android.com/tools/help/monkey.html>.
- 10 Monkey Runner. http://developer.android.com/tools/help/monkeyrunner_concepts.html.
- 11 Gomez L, Neamtii I, Azim T, et al. RERAN: Timing- and touch-sensitive record and replay for Android. 2013 35th International Conference on Software Engineering (ICSE). San Francisco, CA, USA. 2013. 72–81. [doi: 10.1109/ICSE.2013.6606553]