综述

该用户指南致力于帮助测试人员熟悉 RR_Tester 的使用方法以及一些使用技巧；同时也在介绍使用流程的同时，方便开发工程师参与进来改进 RR_Tester。

RR_Tester 的使用分为两个阶段：

首先，使用 record 模式用它在参考机上记录目标应用的操作过程以及期望的结果。

然后，使用 replay 模式在测试机上复现目标应用的操作，并且比较操作过程中的结果，以判断目标应用在测试机上的运行结果，并生成报告。

Host 设置

无论工作在 record 还是 replay 模式，工具都需要在用户主机上运行，所以 host 上的环境设置是必不可少的部分。

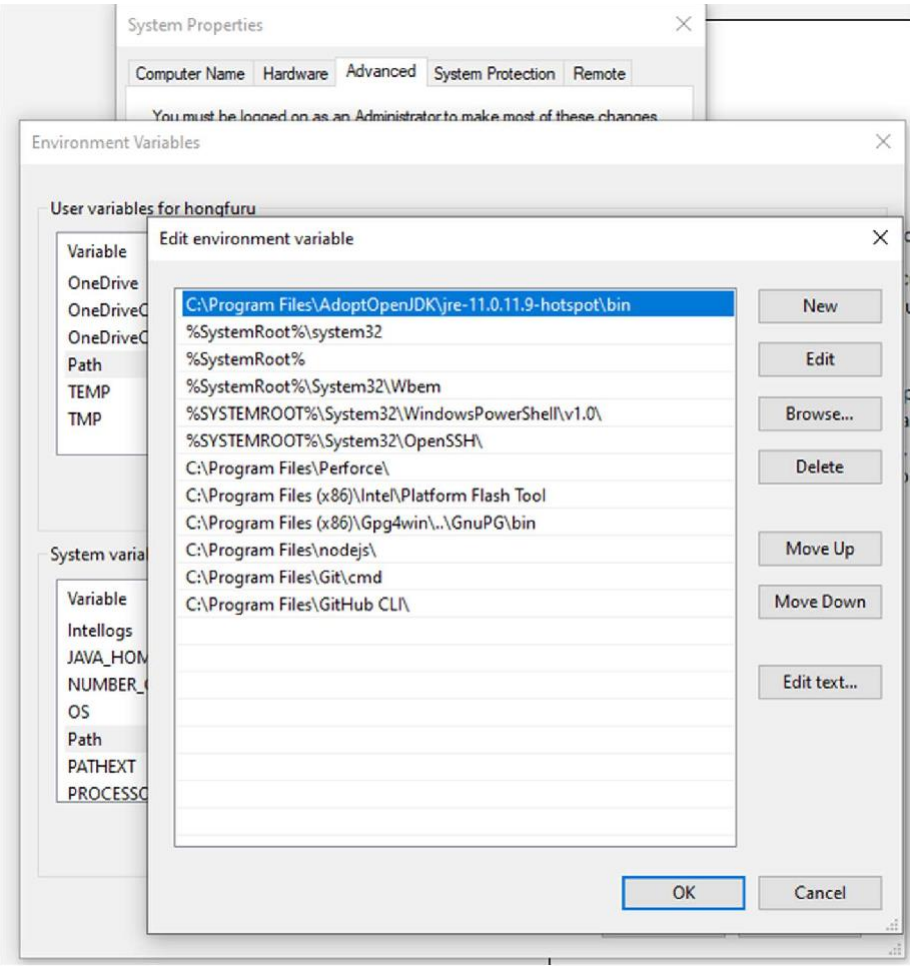
1. Python 设置

- Python3: https://www.python.org/downloads/
- 我们推荐使用 Python 3. 7+版本。其他Python3.x 版本应该都支持，但是没有完整验证过.
- 安装相关的 python 模块:
    - pip3 install readchar
    - pip3 install pillow
    - pip3 install colorama

2. ADB 设置

RR_Tester 依赖 adb 与设备连接，使用工具前请确保 Host 上 adb 路径存在于系统环境变量，可以在任意路径直接使用 adb 命令，而不是要指定 adb 的绝对路径。

系统环境变量设置参考：

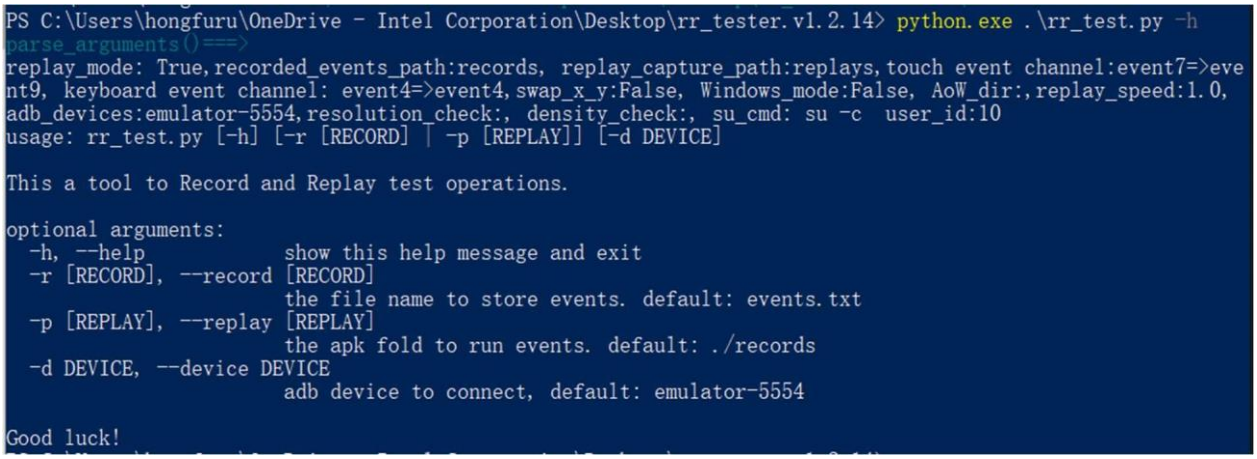


*当系统中存在不同版本的 adb 时，经常会遇到如下的警告，为了避免不必要的麻烦，我们建议系统中只保留一份 adb 命令：

```
adb server version (31) doesn't match this client (36)
adb server version (32) doesn't match this client (36); killing...
```

工作模式

目前，RR_Tester 是一个脚本工具，跑在 Windows 或者 Ubuntu 终端环境下。它通过命令行参数以及 config.json 来设置工作模式、传递参数。命令行的帮助信息如下：



config.json 里的设置内容依平台不同而不同，里面一般保存和测试平台相关的固定设置。参考例子如下：

{

"event_channel_record_touch": "event7", //对应 record 平台的触摸屏输入 channel

"event_channel_replay_touch": "event9", //对应 replay 平台的触摸屏输入 channel

"event_channel_record_keyboard": "event4", //对应 record 平台的键盘输入 channel

"event_channel_replay_keyboard": "event4", //对应 replay 平台的键盘输入 channel

"record_max_35": 32767, //record 平台输入 channel 的 0035 max

"record_max_36": 32767, //record 平台输入 channel 的 0036 max

"replay_max_35": 16383, //replay 平台输入 channel 的 0035 max

"replay_max_36": 9599, //replay 平台输入 channel 的 0036 max

"Windows_mode": true, //Windows 下运行会检查管理者模式

"swap_x_y": false, //record/replay 平台坐标原点如果不同，则需要 swap x/y

"user_id": "10", //多用户平台上的 user id

"apk_folder": "apk" //自动安装 apk 时对应的 apk 文件目录

“scan_phone”: “192.168.10.43” //用作扫码辅助的设备 adb 编号

“su_cmd_scan_phone”: “su -c” //扫码设备需要的 su 参数

“scan_apps”: { //扫码 app 对应的按键以及 package name

“w”: “com.tencent.mm”, //可以按需添加，如微信，qq，钉钉等

“q”: “com.tencent.mobileqq”,

"d": ""

},

"sms_phone_adb_device_name": "c6247a" //用来接收短信验证码的手机 adb 设备 id

}

- 查看平台输入 channel 的方法：adb shell getevent -l 然后通过触摸或者键盘产生输入事件
- 查看平台输入 channel 的 0035/0036 max：adb shell getevent -p, 在输出中找到对应 channel 下的 0035/0036 max 值

Record 模式

使用工具的第一阶段是在参考机上记录目标应用的操作和对应的结果。包括准备参考机、APK 安装以及运行工具几个部分。

参考机的准备

参考机的选择原则是它能尽量兼容我们需要测试的应用(ARM 平台)，能执行我们需要的命令（能 root）并且没有太多特殊化的地方方便测试机适配。我们选择的参考机是联想小新 Pad(Link)，解锁操作请参考 https://www.zui.com/iunlock

APK 安装

- 应用测试场景下，一般都有大量的 App 需要安装/卸载，建议使用脚本来批量操作
- 安装 App 时，为了统一避免 App 使用时弹出权限申请的窗口，需要指定-g 参数如：
    - adb install -g com.tencent.xxx.apk

运行 Record

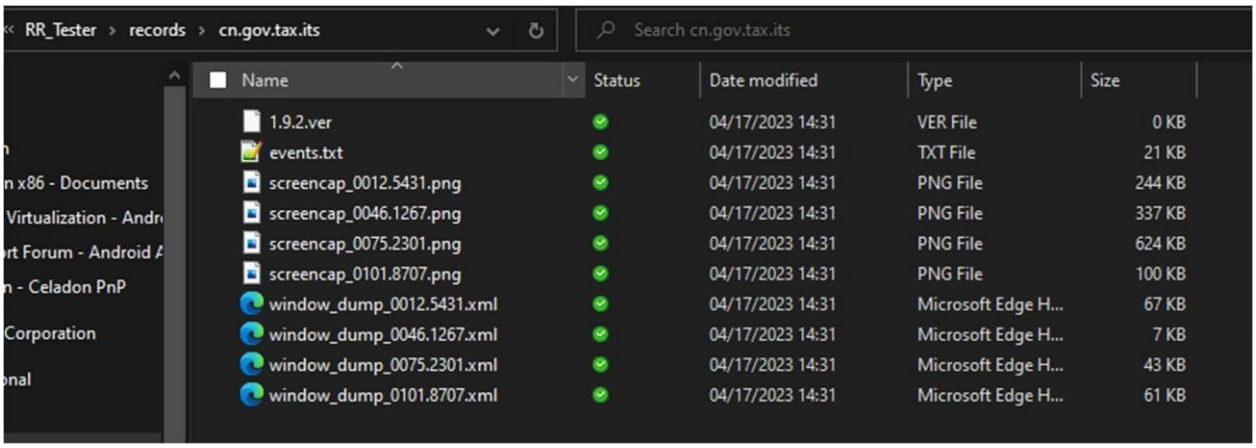
- adb 连接参考机，然后打开需要测试的 App。
- 用管理员模式运行 PowerShell 终端
- 根据 help 信息设定需要的参数启动工具。Record 模式下只需要-r 和-d 两个参数，用于设定 record 模式以及指定 device id：

```
python.exe rr_test.py -r -d HVA0763H
```

- 工具会 reset 应用让它回到一个初始状态以确保 App 的行为不受前面操作的影响，然后停在等待用户开始 record
- 在应用启动完成后，用户按 s (start)启动 record 操作。
    - 因为工具使用后台 eventrecorder 来记录用户输入，用户按 s 的同时需要触发一个 input 事件。建议的操作是按 s 的同时用手指在 App 无效区域滑动，这样可以确保 record start 后立刻就有 input event 产生，同时又不使 App 状态发生变化
- 启动后，工具会记录用户在应用上的一切输入操作。在操作过程中，用户可以选择一些画面按 c (capture)作为检查点(check point)，用于 replay 时候比较结果
    - 选择检查点的建议是选择不随时间/状态而改变的 activity，比如启动时的用户协议，App 的设置菜单等。避免会随着日期而变化的内容，避免有临时广告的界面。

- 用户选择checkpoint 后，record 目录下会有对应的 png/xml 文件产生。如果发现选的 checkpoint 不合适，将对应的 png/xml 文件删除即可取消该 checkpoint

- 在操作并记录 checkpoints 结束后，按 e (end)结束record 过程。产生的记录结果如下:



Replay 模式

在参考机上 record 之后，工具会被多次用在测试机上以 replay 模式运行以检查 App 的结果，生成的结果会保存在 replays 目录下，包括每个 App 对应的结果和一份综合报告。运行之前，测试机可能需要一些设置以适配参考机。不同的测试机需要适配的操作不同，AoW 的适配工作在后面章节详细介绍。

Replay 相对 record 操作来说比较简单，只需要在环境准备好的情况下运行一个命令即可完成 records 目录下所有 App 的 replay 并且生成测试报告。Replay 过程如下：

- 确保测试机适配完成，adb 连接正常，App 安装完成
- 在管理员模式打开 PowerShell 窗口，运行:

  Python.exe rr_test.py -d emulator-5554 -e event11 -p -a \"D:\Program Files\Tencent\Androws\Application\1.0.639.0\"

  其中: -d 指定测试机的 device id (通过 adb devices 可以获取)

  -e 指定测试机对应的输入 channel (通过 adb getevent -l 并在测试机上任意操作可以看到 channel 名称)

  -p 指定 replay 模式

  -a 指定应用宝启动目录，用于系统异常时重启应用宝

  -s 是可选项，用于调节 replay 的速度，方便在性能较差的机器上以较慢的速度重复 record 的操作，给机器更多的反应时间

启动 replay 后，工具会从 record 目录挨个读取对应目录里的内容，基于里面的 events.txt 在测试机上复现用户操作，并且在对应的检查点(checkpoint) dump 机器上的内容用以判断。最终结果会生成在 replay 目录下的csv 文件中。

- Replay 时工具会检查 app 目录里是否存在 events.txt 文件，如果没有就会跳过。因此如果用户想跳过 replay 某些 App 时，可以重命名相应目录里的 events.txt 即可
- Replay 完成后，工具会对比 records 和 replays 下对应的 app 目录里的 xml 文件，xml 文件结构一致则判断为结果符合，所有检查点的结果符合则判断 app 测试通过。如果有 App failed 的情况，可以手动对比 App 在 records/replays 下的目录，参考 png 文件看 App 的表现有什么差异。（检查点的 png 文件只用于人工参考，不是工具比较的依据）

Replay 结果

Replay 后生成的 report 里可能存在以下结果：

```
TEST_RESULT_PASSED = 0           # "Passed"
TEST_RESULT_XML_FAILED = -1      # "Failed"
TEST_RESULT_INVALID_EVENT = -2   # "Invalid"
TEST_RESULT_SYS_CRASH = -3       # "System Crash"
TEST_RESULT_APP_CRASH = -4       # "App Crash"
TEST_RESULT_NOT_EXISTING = -5    # "Not Exists!"
TEST_RESULT_WRONG_AOW = -6       # "AoW is not right configured"
TEST_RESULT_WRONG_VERSION = -7   # "Version mismatched"
```

Passed: 意味着 App 在测试机上运行正常，在每个检查点上获取的 App 状态信息和参考机一致

Failed: App 在 replay 过程中至少有一处检查点和参考机对应不上

Invalid: record 目录下 App 子目录里的 events.txt 文件异常

System Crash: replay 过程中测试机 adb 连接断开，Android 系统需要重启

App Crash: 测试过程中 App 退出，根据 FocusedApp 信息判断

Not Exists: 说明 App 在测试机上不存在

TEST_RESULT_WRONG_AOW: 不会存在 report 中，它的发生是 AoW 平台未正常配置，会中断 replay
TEST_RESULT_WRONG_VERSION: 意味着测试机和参考机上用的 App 版本不同，目前只输出警告，不中断 replay，不会记录到 report。

二维码扫码功能

在测试过程中，常常会碰到需要扫二维码的场景。无论是扫码登录还是扫码支付，replay 和 record 时的二维码都是动态生成而不是固定不变的。这种场景不能用普通的 events replay 方法继续下去。为此，tool 针对扫二维码添加了特别的功能。

首先是 record 时，在遇到二维码场景需要在终端输入 w(微信二维码)或者 q(QQ 二维码)，tool 会在这个时间点在 record 目录生成一个 qrcode.xxxx.xxxx.wechat(微信二维码)或者 qrcode.xxxx.xxxx.qq(QQ 二维码)的 png 文件。文件内容不重要，关键是文件名以及文件名中的 xxxx.xxxx 时间戳信息。

然后在 replay 时，当 tool replay 到 xxxx.xxxx 这个时间点，它会根据文件名格式得知这里需要扫码一个微信/QQ 二维码。Tool 会把当前屏幕上的二维码截图下来发给 scan server，并且出发 scan server 上的扫码动作。顺利的话扫码成功，replay 继续后面的操作。

关于 scan server，这是任意一个安卓设备并且满足以下条件：

1. 和运行 rr_tester 机器 adb 互通：无论是 USB 连接或者网络连接均可，因为 rr_tester 需要把动态二维码通过 adb 方式 push 到 scan server 上去扫码
2. 能访问外网：微信/QQ 扫码都需要外网环境
3. /data/local/tmp 下有 scan_qr_replay (eventrec rename 而来)和事先录制好的微信/QQ 扫码登录过程的 events 文件：rr_tester replay 过程检测到二维码之后，它会触发 scan server 去 replay 这个扫码登录的 events。
    a. 扫码登录的 events 文件可以用“eventrec -r wecat.scan.events”录制
    b. 在微信/QQ 已经登录的情况下准备这样的扫码 event 比较方便，可以避免账号密码被记录，而且 events 文件也比较简单

Tips:

1. Scan server 尽量设置成行为一致，比如屏幕不会息屏；录制 scan events 和 replay scan events 时方向一致；等等
2. 录制 scan events 时，为了避免微信/QQ 有新消息干扰，要增加冗余 event，比如打开后多点几次空白处，让微信/QQ 处在一个固定的初始状态

短信验证码登录功能

有的 App 需要短信验证登录。Tool 在 replay 过程中如果连接并配置 sms_phone_adb_device_name 手机，那么在需要输入短信验证码的 checkpoint，它会从手机中读取短信，从短信中找到格式匹配的验证码并正确输入到 App 的窗口。

应用宝设置

安装 AoW

- 从腾讯应用宝官网下载&安装应用宝 PC 版：https://sj.qq.com/

- 或者从项目组取得下载器并安装特定版本的应用宝.

测试模式

因为使用工具时需要和参考机保持一致的显示设置，测试应用宝时需要一些调整：

1. 应用宝默认不能通过 IP 连接 ADB，需要通过 IP 访问的话需要修改 Program Files\Tencent\Androws\Image\androws.abox 里 ADB_PORT 的设置，将默认”127.0.0.1”改成”0.0.0.0”即可。
2. 退出应用宝然后到 Androws.exe 目录(如`D:\Program Files\Tencent\Androws\Application\1.0.510.0`)用以下命令启动应用宝并制定显示参数:

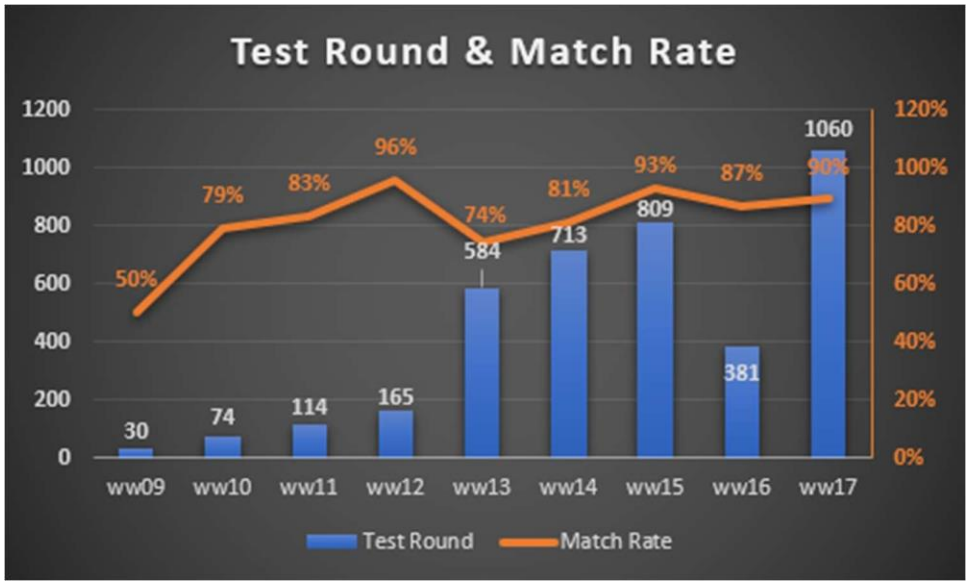
```
.\Androws.exe --launch-pkg-name com.android.settings --vm-pscreen-width 1200 --vm-pscreen-height 2000 –vm-screen-width 2000 –vm-screen-height 1200
```

启动完成后，应用宝会停留在 Setting 界面，等待用户执行 replay 操作。

总结

使用情况

目前工具测试 120+App 超过 3000+轮次，准确度大概在 85~90%之间:



待改进

有一些已知的会影响准确度的因素如：

- App 临时弹窗
- App 在参考机和测试机上 layout 有差异
- 其他

我们会持续改进工具让它更方便使用，以及准确度更高。同时也欢迎用户提意见，需求。能参与提供 Patch 更是欢迎。

源代码

https://github.com/tingai1/os.android.aow.rr_tester-tool/