EC601 Hw5: unit test

Date test run: Dec 9th 2017 Student name: Zhiwei Tang

1. The APP tested in this homework: MyFirstApp

(Github page: https://github.com/tzwk/EC601_HW2/tree/master)

2. Environment Description:

I used my laptop as the hardware platform. The software platforms I used in this test were Android Studio and Amazon Web Services.

3. Software Version:

Android Studio 3.0

4. Information about the test:

- The test done on the Android Emulator assumed the application was run on a Pixel XL device.
- The automated test is done on both monkey test as well as AMS test.
- The test done on monkey is

  1) On level 1 verbose;

  2) The application will be launched and the tester will send 500 pseudo-random events to it. The events are basic user events such as clicks, touches, or gestures, as well as a number of system-level events;

  3) There are no delays between the events.

- The test done on AMS test is

  1) The tester tested the apk file (app-debug.apk) of the application;

  2) Done on all the devices that is compatible with the application (Samsung Galaxy S5 (T-Mobile), Samsung Galaxy S6 (Verizon), LG G Pad 7.0" (AT&T), Samsung Galaxy Tab 4 10.1" (WiFi), Amazon Kindle Fire HDX 7 (2013).);

  3) The radio states are set to enable WiFi, Bluetooth, GPS and NFC, the device location is set to 47.6204 N, 122.3491 W;

  4) Host machine is $WORKING_DIRECTORY, the Device locale is en_US and the predefined network profile is used;

  5) The execution timeout is set to 10mins.

Test results:

Part 1: Test on the android studio emulator

Test ID	Test Scenario	Components Involved	Platform	Test Steps and Descriptions	Actual Results	Pass/Fail	Priority
			For example: Samsung S7, Android iOS Safari Browser IE, etc.				Critical High Medium Low
Unique ID for every test case	Title that describes what you are testing	if needed, describe comonents involved in yoru test		Step by step description for the test in order to be able to reproduce results	Description of results. This can include media to show what happened.		
1	Type random words	Android Studio	Android Studio Emulator	1. open the emulator 2. Input random words like “Hello” 3. Press “send” button	1. A new window is opened 2. The words inputted is displayed on the head of the new page.	Pass	High
2	Send without any input	Android Studio	Android Studio Emulator	1. open the emulator 2. Input nothing 3. Press “send” button	1. A new window is opened 2.Nothing displayed on the head of the new page.	Pass	Medium
3	Goes back to the input window after the app goes to the display window	Android Studio	Android Studio Emulator	2. open the emulator 2. Input nothing 3. Press “send” button 4. Press the arrow at the top left	1. A new window is opened 2.Nothing displayed on the head of the new page. 3. The app goes back to the input window	Pass	Critical
4	Type words in different languages(e.g. Chinese)	Android Studio	Android Studio Emulator	2. open the emulator 2. Input a Chinese word 3. Press “send” button 4. Press the arrow at the top left	1. A new window is opened 2.The words whatever inputted are displayed on the head of the new page. 3. The app goes back to the input window	Pass	Medium

Part 2: Test on the monkey

```
** Error: Unable to connect to activity manager; is the system running?
blamon@blamon-ThinkPad-X230:~$ adb shell monkey -p com.example.blamon.myfirstapp -v 500
  bash arg: -p
  bash arg: com.example.blamon.myfirstapp
  bash arg: -v
  bash arg: 500
args: [-p, com.example.blamon.myfirstapp, -v, 500]
 arg: "-p"
 arg: "com.example.blamon.myfirstapp"
 arg: "-v"
 arg: "500"
data="com.example.blamon.myfirstapp"
:Monkey: seed=1513080187838 count=500
:AllowPackage: com.example.blamon.myfirstapp
:IncludeCategory: android.intent.category.LAUNCHER
:IncludeCategory: android.intent.category.MONKEY
// 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%
:Switch: #Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10200000;component=com.example.blamo
n.myfirstapp/.MainActivity;end
    // Allowing start of Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.example.blamon.myfirstapp/.Main
Activity } in package com.example.blamon.myfirstapp
:Sending Touch (ACTION_DOWN): 0:(84.0,909.0)
:Sending Touch (ACTION_UP): 0:(88.52291,917.63165)
:Sending Trackball (ACTION_MOVE): 0:(1.0,-2.0)
```

Figure 1 The test properties

```
:Sending Trackball (ACTION_MOVE): 0:(3.0,2.0)
    // Allowing start of Intent { act=android.intent.action.MAIN cat=[android.intent.category.HOME] cmp=com.google.android.apps.nexuslauncher/.
NexusLauncherActivity } in package com.google.android.apps.nexuslauncher
:Sending Trackball (ACTION_MOVE): 0:(-5.0,1.0)
:Sending Trackball (ACTION_UP): 0:(0.0,0.0)
:Sending Touch (ACTION_DOWN): 0:(925.0,1713.0)
Events injected: 500
:Sending rotation degree=0, persist=false
    // Rejecting start of Intent { act=android.intent.action.SHOW_ALARMS cmp=com.android.deskclock/.HandleApiCalls } in package com.android.des
kclock
:Dropped: keys=0 pointers=0 trackballs=0 flips=0 rotations=0
## Network stats: elapsed time=3160ms (0ms mobile, 0ms wifi, 3160ms not connected)
// Monkey finished
blamon@blamon-ThinkPad-X230:~$
```

Figure 2 The test results

Part 3 Test result on AWS

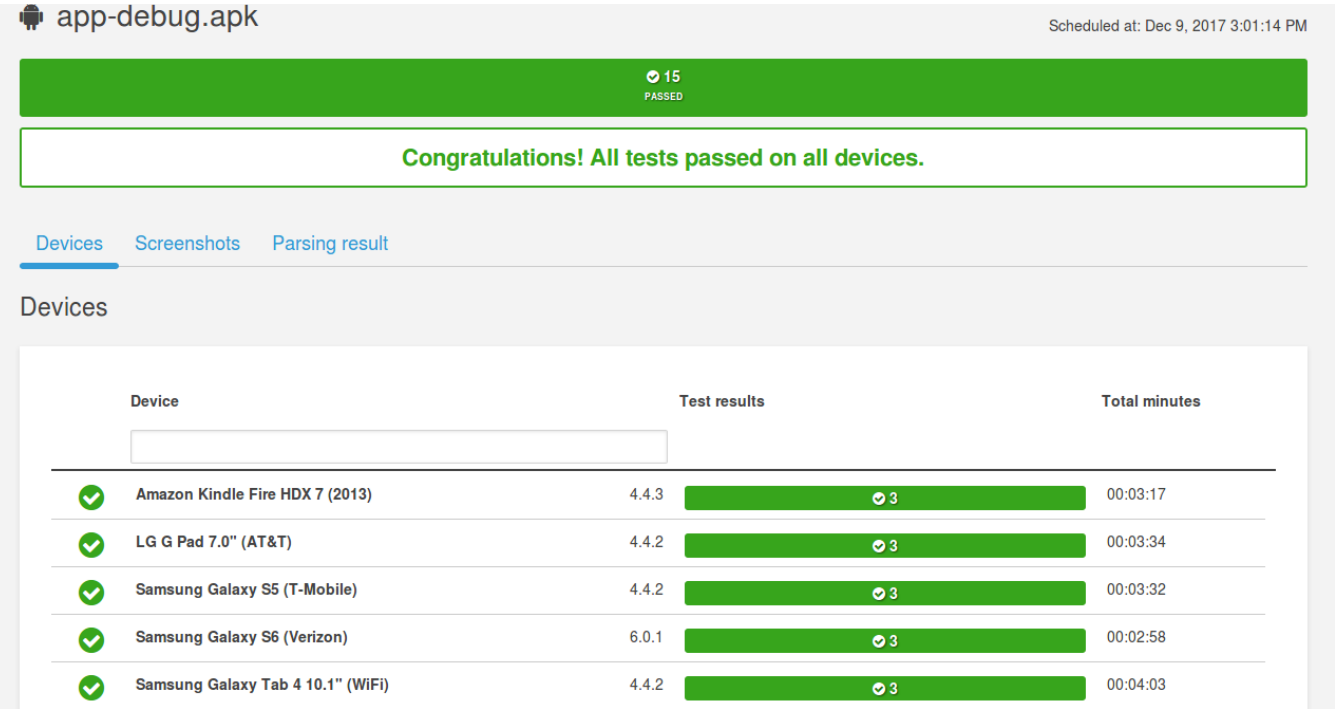

Figure 3 Test results of AWS