Automation testing project for CleanCalculator

1. How to use it
   1. Make sure java and android sdk is set up correctly in your computer (java, adb, jarsigner works fine)
   2. Put the “CleanCalculator.apk” file that you want to test into to folder “APK”
   3. Connect your android phone/tablet to your computer in debug mode
   4. Run “Run.bat”, it just runs “Controller” which does following steps:
      1. Step1, push folder “CCTest” to device (instrumentation test will use those files)
      2. Step2, uninstall target and test app on phone/tablet
      3. Step3, re-sign target and test apk, then install them
      4. Step4, start instrumentation test and wait
      5. Step5, get the result file from phone/tablet
   5. After testing is done, there will be a result file “TestResult.xml”, it’s based on assertion information.
2. Structure of CleanCalculatorTest project
   1. TestSuite structure
      1. Action: a set of manipulations which compose a meaningful function. For example, GoToSetting, GoBack, ChangePrecision…
      2. TestCase: a sequence of action. For example, GoToSetting->ChangPrecision->GoBack->EnterBasicExpression
      3. TestSuite: a collection of test case

TestSuite

Action

Action

… …

TestCase

Action

Action

… …

TestCase

… …

* 1. How to maintain test suite

The definition of TestSuite is stored in “TestSuite.xml”. The definition of TestCase is stored in xml files in “TestCase” folder. The action classes are in CleanCalculatorTest.apk

To modify TestSuite is easy.

To modify TestCase is easy.

Only to modify actions require changing code in CleanCalculatorTest project.

1. Advantages of my framework
   1. Test cases are independent. At the beginning of every case, app will be restart.
   2. Changing TestSuite/TestCase is easy
   3. As we know, string id or text are very likely to be changed in the future. I use “Constants.xml” to maintain this string id and ui texts. And in the project, it is used to generate a hash map which stores these value. In this way, even if the string id or text of one view is changed, we don’t need to change code. We just need to change the “Constants.xml” file.
   4. Get testing result info from device for analysis
   5. After testing started, even if computer lost connection to phone/tablet, all test cases will be executed. After testing is done, we can manually pull the result to computer.
   6. No product source code is needed to run the test. Apk file is enough.
   7. Robotium is powerful. It supports both native and web view. For future extension, it’s a good choice.
2. To improve

As time limited,

* 1. Almost directly used robotium API, actually we should customize our own APIs based on it. APIs like getViewByClass will be useful.
  2. The result report is simple, not much information. We should generate a more elegant report.
  3. Actions, Constants are just examples, to test real product, define enough and good actions are very important.
  4. Can build GUI tool to create test case. Edit xml by notepad is not so efficient.
  5. Do not compare screenshot, sometimes for GUI testing screenshot is also important.