# 0 Files overview

The files provided by us are organized into 3 parts:

1.source code: an eclipse workspace containing the project of our reduction tool and the project of modified Monkey for Android 5.1.0.

2.monkey tool: modified Monkey files for Android version 4.4.4, 5.1.0, and 6.0.1.

3.test case: 92 crash traces from 8 real-life Android application as example.

# 1 The use of reduction tools

The reduction tools provided by us are consist of two parts: the part on testing phone and the part on testing host. To use our reduction tools, both these two parts should be set appropriately.

**For the part on testing phone**: The modified Monkey files should be put into the testing phone. Modified Monkey files include *monkey* and *monkey.jar*. These two files should be put into a same local path within testing phone. */data/local/* is a recommended path. Other legal local path is also allowed but just remember to modify the variable *DEVICE\_MONKEY\_PATH* to the path of monkey files. Note that different modified Monkey files may be needed in different Android system version. More details about where the differences between modified Monkey and original Monkey files are as well as how to generate the modified Monkey files will be introduced in **section 3**. The modified Monkey files for Android version 4.4.4, 5.1.0, and 6.0.1 are contained in our tools.

Besides, a folder named *eventScript* should be created in the root directory of external storage within testing phone to ensure that the modified Monkey can run and output log files correctly.

**For the part on testing host**: First, a series of folders should be created within testing host to storage the input and output test case files, and the hierarchy of folders should be organized as followed. For example, if use *C:\Reduction* as the root directory, then subdirectory folders as many as the number of crash traces named from *1* to *n* should be created in root directory, i.e. *C:\Reduction\1\* to *C:\Reduction\10\* for scene that there are ten crash traces supposed. Afterwards, folders named *origin* and at least one of *resultdd*, *resulthdd*, *resultbhdd*, *resultlhdd* should be created within each subdirectory, i.e. *C:\Reduction\1\origin\*, *C:\Reduction\1\resultlhdd\*, *C:\Reduction\2\origin\, C:\Reduction\2\resultlhdd\*, etc. for scene that use LHDD as the reduction algorithm. Then, after putting log files *execute.log* and *script.log* of each crash trace into each *origin* folder the reduction process would be able to begin by running the *ReductionRunner.java* in our project. Note that there are some variables in *ReductionRunner.java* that can be adjusted according to test environment and specific needs. More details about these variables will be introduced in **section 2.**

# 2 The arrangement of project MonkeyReduction

The arrangement of source files in our project is shown in the following table.

|  |  |  |  |
| --- | --- | --- | --- |
| src | cn.edu.buaa | |  |
|  |  | ReductionRunner.java | entrance class |
|  | cn.edu.buaa.command | |  |
|  |  | AdbOpeartion.java | Adb controller class |
|  | cn.edu.buaa.reduction | |  |
|  |  | HddTreeNode.java | hierarchy tree node class |
|  |  | DeltaDebugging.java | DD algorithm class |
|  |  | HierarchicalDeltaDebugging.java | HDD algorithm class |
|  |  | BalancedHierarchicalDeltaDebugging.java | BHDD algorithm class |
|  |  | LocalHierarchicalDeltaDebugging.java | LHDD algorithm class |
|  | cn.edu.buaa.state | |  |
|  |  | EventState.java | state record class |
|  |  | EventQueueOperation.java | event and state operator class |
|  | cn.edu.buaa.util | |  |
|  |  | MD5.java | MD5 calculator class |

Particular introductions of each file are as followed.

**ReductionRunner.java**: The entrance of our project. Main parameters that can be adjusted are also defined in this files. They are:

*LOCAL\_PATH*: The root path of test case files within testing host, with default value *C:\Reduction\*.

*DEVICE\_MONKEY\_PATH*: The directory of monkey files within testing phone, with default value */data/local/*.

*DEVICE\_TMP\_PATH*: The root directory of external storage within testing phone. Note that this directory may be a path of SD card like */sdcard/* or a path of virtual storage like */storage/emulated/0/*, which is depend on the model of testing phone.

*DEVICE\_ID*: The ID to distinguish testing phones used by ADB Shell, which can be obtained through using command “adb devices”.

*PACKAGE\_NAME*: The package name of the application to run test case on.

*THROTTLE\_TIME\_MS*: The time interval between events in test case while executed, with default value 800ms. A time interval longer than 500ms is recommended to ensure that each event has enough time to be responded.

*TRACE\_NUMBER*: The number of crash traces to run reduction on, which should be a positive integer and be corresponding to the number of valid folders in *LOCAL\_PATH.*

*DO\_DD*: The Boolean variable for whether use Delta Debugging algorithm to run the reduction on each crash trace. While set to *true*, the folder *resultdd* must be created in each subdirectory.

*DO\_HDD*: The Boolean variable for whether use Hierarchical Delta Debugging algorithm to run the reduction on each crash trace. While set to *true*, the folder *resulthdd* must be created in each subdirectory.

*DO\_BHDD*: The Boolean variable for whether use Balanced Hierarchical Delta Debugging algorithm to run the reduction on each crash trace. While set to *true*, the folder *resultbhdd* must be created in each subdirectory.

*DO\_LHDD*: The Boolean variable for whether use Local Hierarchical Delta Debugging algorithm to run the reduction on each crash trace. While set to *true*, the folder *resultlhdd* must be created in each subdirectory.

*NEED\_EXTRACHECK*: The Boolean variable for whether utilize extra check in LHDD algorithm. Note that this parameter will work only with *DO\_LHDD* set to *true* and can only influence the process of LHDD.

**AdbOpeartion.java**: The controller of Adb commands to be executed. Being responsible to translate operations on testing phone into Adb commands.

**HddTreeNode.java**: The definition of the node used in hierarchy tree as well as some node operation methods.

**DeltaDebugging.java**: The realization of algorithm Delta Debugging.

**HierarchicalDeltaDebugging.java**: The realization of algorithm Hierarchical Delta Debugging.

**BalancedHierarchicalDeltaDebugging.java**: The realization of algorithm Balanced Hierarchical Delta Debugging.

**LocalHierarchicalDeltaDebugging.java**: The realization of algorithm Local Hierarchical Delta Debugging.

**EventState.java**: The definition of the storage element of each event within *script.log* and its corresponding state within *execute.log*.

**EventQueueOperation.java**: The operator of the list of event and state. Being responsible to read or print the log files and get event and state information from them.

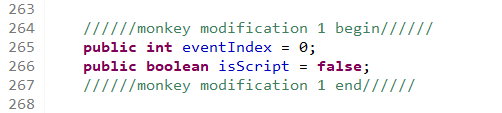
**MD5.java**: The calculator of MD5 value of given test case.

# 3 The modification of Monkey

The modified or added source files in Monkey is shown in the following table.

|  |  |  |  |
| --- | --- | --- | --- |
| src | com.android.commands.monkey | |  |
|  |  | Monkey.java | modification 1 to 7 |
|  |  | MonkeySourceRandom.java | modification 8 to 20 |
|  | com.android.commands.monkey.logger | | addition source files |
|  | com.android.commands.monkey.scriptevent | | addition source files |

Considering that the original Monkey provided by Android system does not support the function that print the events stream into a script file, which make it difficult to re-run or do reduction on a crash trace, we modify and add some source files in Monkey to extend its functions. Within source file **Monkey.java**, there are 7 modifications, and each one is annotated “*//////monkey modification x//////*” as shown in the following figure. Within source file **MonkeySourceRandom.java**, there are other 13 modifications with the same annotation. Within package **logger** and **scriptevent**, there are several source files added.



For different Android system version, the source files of Monkey may be different as well. The modified Monkey source files in our project as example are based on native Monkey from Android 5.1.0. If there is requirement for Monkey files used in Android system of other version, just migrate the above modifications and additions into Monkey source files for that Android version and re-package all the source files into a jar package then the *monkey.jar* would be able to obtained.

# 4 The use and generation of test cases

Considering that most events of Monkey are based on the coordinate on screen while coordinate could be meaningful only within certain screen resolution, the screen resolution of either physical phone or emulator must be the same as the testing phone used when the test case was generated if our example test cases are supposed to use. Therefore, the model of device used and its screen resolution are informed in the readme document of each application.

Of course new test cases can also be generated with modified Monkey, whose command and method of use are familiar to native Monkey, and would output log files into external storage within testing phone at */sdcard/execute.log* and */sdcard/eventSctipt/script.log*. These two files are all what doing reduction need.