CS409
Software Testing

TAN, Shin Hwei
陈馨慧
Southern University of Science and Technology

Recap: Fuzz Testing (Fuzzing)

- A quality assurance technique used to discover coding errors and security loopholes in software, operating systems or networks
- Random Testing
  - Randomly generate input
  - Treat program as black box

Lab Today

You get to:

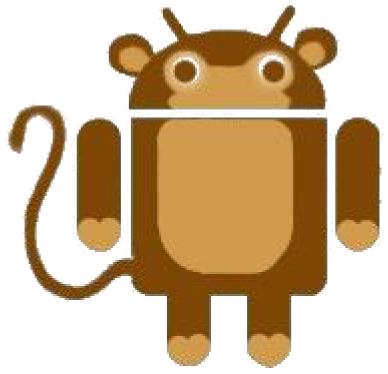
- Try android fuzz testing tool
- Try delta debugging
- Try the basic idea of a research paper:
  - Minimizing GUI Event Traces [FSE2016]

Android Testing

- Monkeyrunner
  - Allow you to record test script and replay test script
- Monkey
  - Fuzzing tool by Google

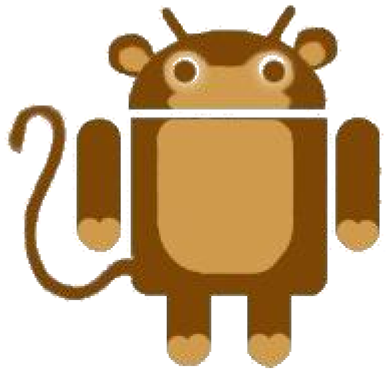
Monkey

- Runs on your emulator/device.
- Generates pseudo-random streams of user events such as clicks, touches, or gestures, and system-level events.
- Stress test your app in a random yet repeatable manner.
- Command line tool



Options in Monkey

- Includes a few options from four primary categories:
  - Basic configuration options, such as setting the number of events to attempt.
  - Operational constraints, such as restricting the test to a single package.
  - Event types and frequencies.
  - Debugging options.



Monkey: How it works?

- Generates events and sends them to the system
- Watch the system and looks for three conditions:
  - If you have constrained it to run in one or more specific packages, it watches for attempts to navigate to any other packages, and blocks them.
  - If your application crashes or receives any unhandled exception, it will stop and report the error.
  - If your application generates an *application not responding* error, it will stop and report the error.

Running Monkey

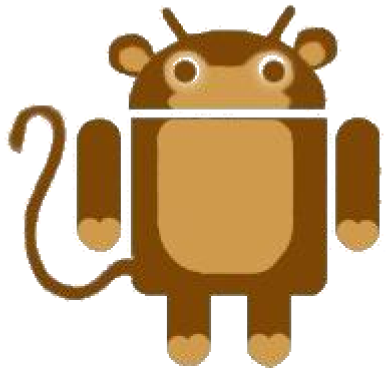
- Comes with Android SDK
- Launch it from a shell： sdk/bin/adb
- Basic syntax is:

```
adb shell monkey [options] <event-count>
```

- Example:

```
adb shell monkey -p your.package.name -v 500
```

Package name can be found using:
- ADB Logcat
- Check the full package name of your app in GitHub



Other useful options

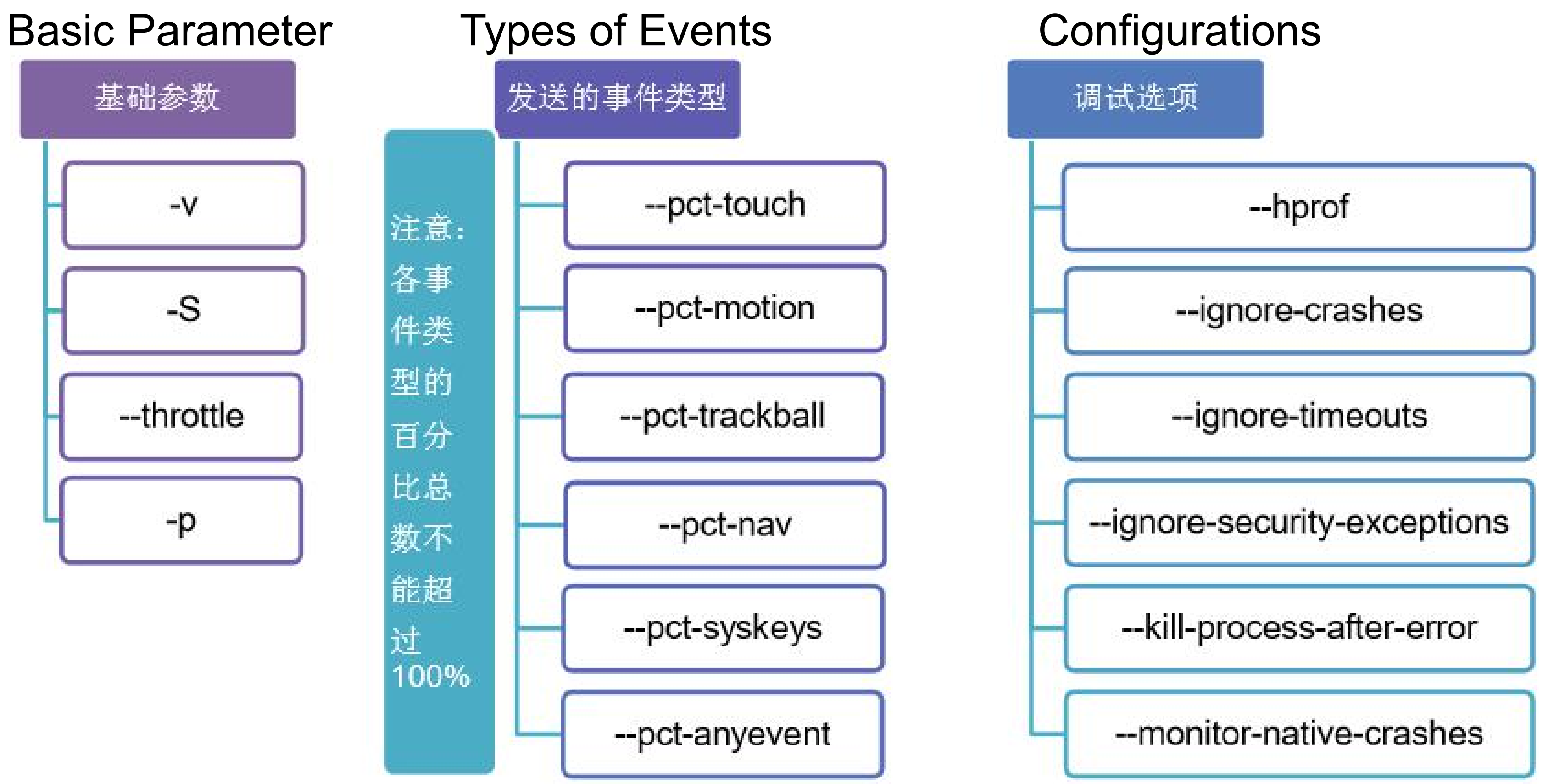
Read this for other options:

- https://developer.android.com/studio/test/monkey
- https://www.cnblogs.com/TankXiao/p/4815134.html (Chinese)

General	-v	Each -v on the command line will increment the verbosity level. Level 0 (the default) provides little information beyond startup notification, test completion, and final results. Level 1 provides more details about the test as it runs, such as individual events being sent to your activities. Level 2 provides more detailed setup information such as activities selected or not selected for testing.
Events	-s <seed>	Seed value for pseudo-random number generator. If you re-run the Monkey with the same seed value, it will generate the same sequence of events.
	--throttle <milliseconds>	Inserts a fixed delay between events. You can use this option to slow down the Monkey. If not specified, there is no delay and the events are generated as rapidly as possible.

Important for reproducing the events

Monkey Parameter 参数大全



Try it yourself in your app

1. Run monkey on you selected app using one seed (s=0)
2. See if it finds any crash
3. Run monkey on using another seed (s=12345)
4. Increase the event count if no crash found
5. Repeat

Pros and Cons of Monkey

Pros:

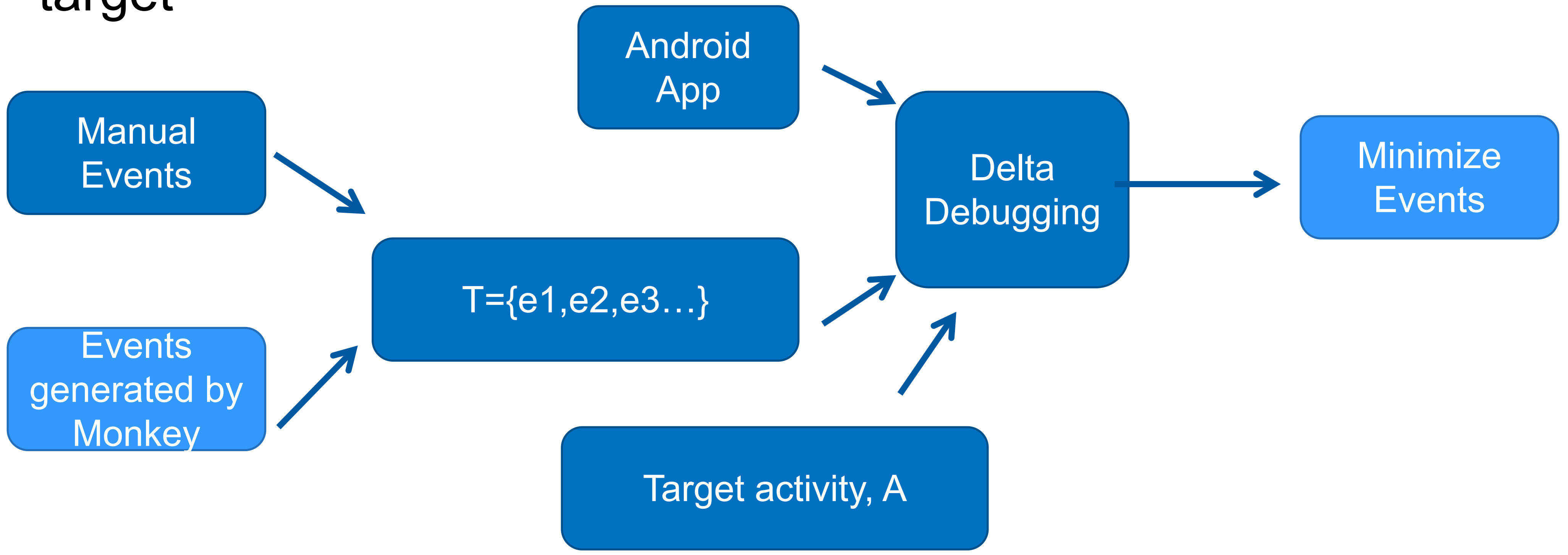
- Easy to run
- Fast and effective in testing app

Cons:

- Long event traces
- Often different to reproduce a crash

Minimizing GUI Event Traces [FSE2016]

- Basic Idea: Since event traces are long, minimize events to reach a target



Minimizing GUI Event Traces [FSE2016]

- Searches for 1-minimal subtrace
- Read the paper if you are interested:
  - https://web.stanford.edu/~lazaro/lazaro-fse2016-final.pdf

Try it yourself in your app: Monkey + Delta Debugging

1. After running monkey, if it crashes on T={e1,e2,e3…} , then use delta debugging to minimize the traces.

$$\{e1,e2,e3\ldots e8\} \xrightarrow{\text{minimizes}} \{e2,e3\ldots e5\}$$

Reproducing the crash

- Remember the starting screen
- Remember the seed value
- Use sendevent to stimulate the click
  https://blog.csdn.net/mldxs/article/details/8561424

This Lab

1. Accept the invitation link:
2. https://classroom.github.com/a/yq3B85aj
3. Test your selected app with Monkey
4. Check if Monkey detect any crash
5. If there is crash, then use delta debugging to localize the root cause of the crash