Lecture 04

Debugging Android Apps

CMSC 4303/5303 Mobile Apps Programming

Hong K. Sung, Ph.D.
Department of Computer Science
University of Central Oklahoma

1

Objectives

- How to debug Android apps
  - use of LogCat
  - use of Eclipse debug tool
  - use of Android Lint

2

Debugging tools

- DDMS – Dalvik Debug Monitor Service
  - LogCat
- Android Lint – static (not runtime) analyzer
- the debugger in Eclipse

Types of errors

- compile errors: the program cannot run
- runtime errors: the program crashes
- logic errors: the program does not produce desired results

3

Runtime errors

- Program crashes – throws exceptions
  - LogCat displays the top level exception and its stack trace

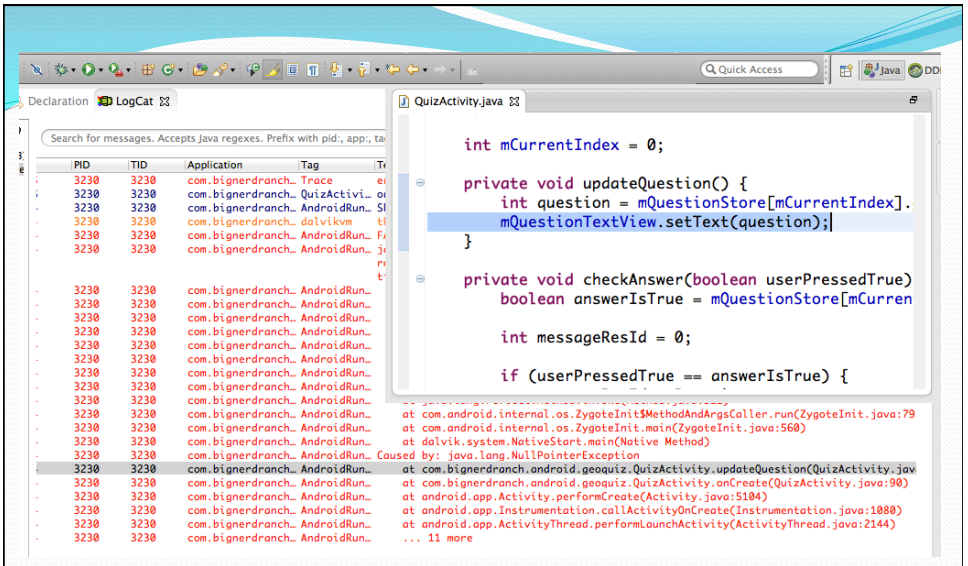
Listing 4.1 Comment out a crucial line (QuizActivity.java)

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    Log.d(TAG, "onCreate() called");
    setContentView(R.layout.activity_quiz);

    mQuestionTextView = (TextView)findViewById(R.id.question_text_view);
    //mQuestionTextView = (TextView)findViewById(R.id.question_text_view);

    mTrueButton = (Button)findViewById(R.id.true_button);
    mTrueButton.setOnClickListener(new View.OnClickListener() {
        ...
    });
```

4



The last exception with no cause (java.lang.NullPointerException) is the first thing to look into.
Double click the line just below to locate the statement which threw the exception.

5

Logic errors – Diagnosing misbehaviors

- Tools
  - Logging stack traces
  - Setting breakpoints – Eclipse debug tool

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    ...

    mNextButton = (Button)findViewById(R.id.next_button);
    mNextButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            mCurrentIndex = (mCurrentIndex + 1) % mAnswerKey.length;
            //mCurrentIndex = (mCurrentIndex + 1) % mAnswerKey.length;
            updateQuestion();
        }
    });

    ...
}
```

logic error: "Next" button has no effect

6

Logging stack traces

```
public class QuizActivity extends Activity {
    ...

    public void updateQuestion() {
        Log.d(TAG, "Updating question text for question #" + mCurrentIndex,
          new Exception());
        int question = mAnswerKey[mCurrentIndex].getQuestion();
        mQuestionTextView.setText(question);
    }
```

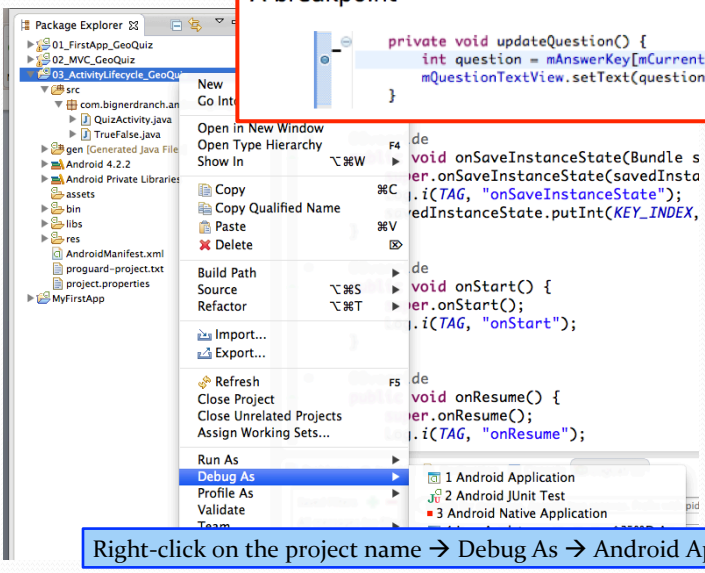
Tag	Text
	eNotFoundException: /proc/net/xt_qtaguid/iface_stat_all: o
	irectory)
SizeAdaptiveLa…	com.android.internal.widget.SizeAdaptiveLayout@41ccc060chi
	cd2c30 measured out of bounds at 95px clamped to 96px
QuizActivity	Updating question text for question #0
QuizActivity	java.lang.Exception
QuizActivity	at com.bignerdranch.android.geoquiz.QuizActivity.updat
QuizActivity	at com.bignerdranch.android.geoquiz.QuizActivity.acces
QuizActivity	at com.bignerdranch.android.geoquiz.QuizActivity$3.onC
QuizActivity	at android.view.View.performClick(View.java:4084)
QuizActivity	at android.view.View$PerformClick.run(View.java:16966)
QuizActivity	at android.os.Handler.handleCallback(Handler.java:615)
QuizActivity	at android.os.Handler.dispatchMessage(Handler.java:92)
QuizActivity	at android.os.Looper.loop(Looper.java:137)
QuizActivity	at android.app.ActivityThread.main(ActivityThread.java
QuizActivity	at java.lang.reflect.Method.invokeNative(Native Method
QuizActivity	at java.lang.reflect.Method.invoke(Method.java:511)
QuizActivity	at com.android.internal.os.ZygoteInit$MethodAndArgsCal
QuizActivity	at com.android.internal.os.ZygoteInit.main(ZygoteInit.
QuizActivity	at dalvik.system.NativeStart.main(Native Method)

7

Eclipse Debugger

A breakpoint

```
private void updateQuestion() {
    int question = mAnswerKey[mCurrentIndex].getQuestion();
    mQuestionTextView.setText(question);
}
```



Right-click on the project name → Debug As → Android Application

8

Figure 4.8 The Debug view

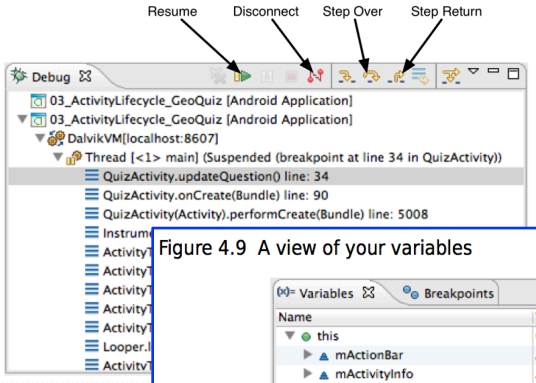
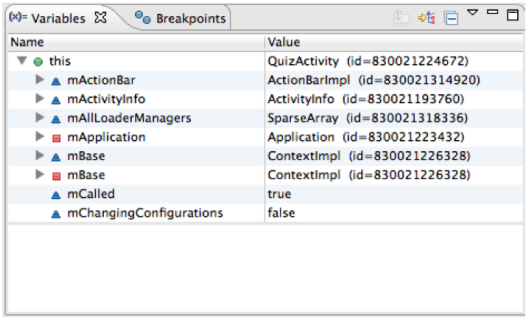
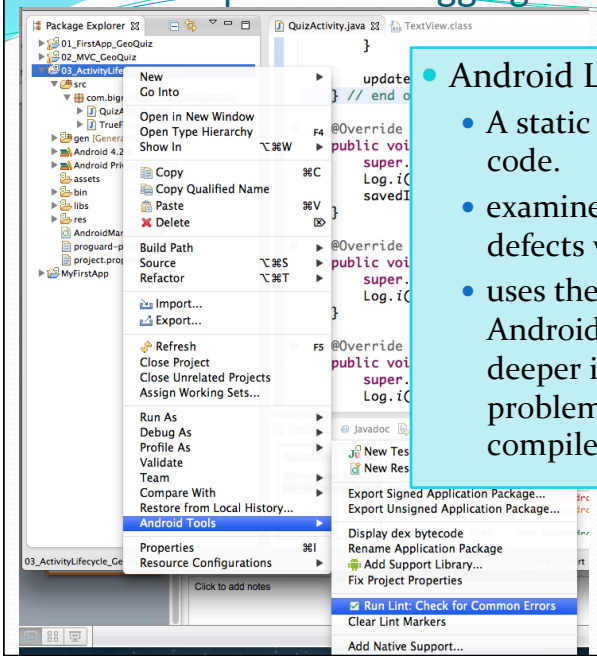


Figure 4.9 A view of your variables



9

Android-Specific Debugging



- Android Lint
  - A static analyzer for Android code.
  - examines the code to find defects without running it.
  - uses the knowledge of Android frameworks to look deeper into the code and find problems that the Java compiler cannot.

10