1.5 Debugging

system log

One tool for debugging apps is the system log, which shows messages from apps running on a device.

Log

Android apps send messages to the system log using static methods of the Log class.

Log.d()

Log.d() sends a debug message to the system log.

Log.e()

Log.e() sends an error message to the system log.

Log.i()

Log.i() sends an information message to the system log.

Log.v()

Log.v() sends a verbose message to the system log.

Log.w()

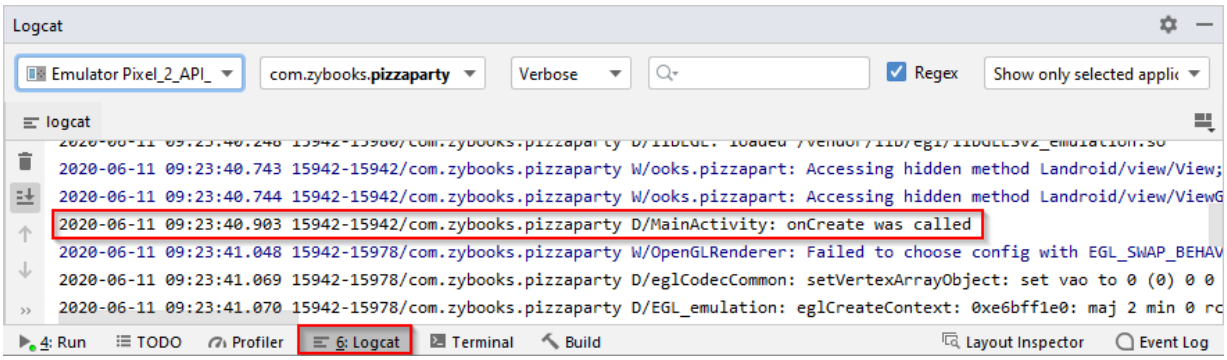
Log.w() sends a warning message to the system log.

Logcat

Logcat displays the system log in Android Studio, which is visible after clicking the Logcat tab.

Figure 1.5.1: Logcat displays a debug message from Log.d().

```
import android.util.Log;
...

public class MainActivity extends Activity {

   private final static String TAG = "MainActivity";

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      Log.d(TAG, "onCreate was called");
      ...
   }

   ...
}
```



Feedback?

stack trace

A stack trace lists the methods and code line numbers that were executing when the uncaught exception was thrown.

PARTICIPATION ACTIVITY 1.5.2: Uncaught exceptions are displayed in the Logcat.

Start 2x speed

```
public void calculateClick(View view) {
   // Get the text that was typed into the EditText
```

```
    String numAttendStr = mNumAttendEditText.getText().toString();
    Log.d(TAG, "number is " + numAttendStr);

    // Convert the text into an integer
    int numAttend = Integer.parseInt(numAttendStr);

    // etc.
}
```

Pizza Party keeps stopp
App info
Close app

Logcat

```
D/MainActivity: number is 7
D/MainActivity: number is
E/AndroidRuntime: Caused by: java.lang.NumberFormatException
```

Captions

1. To determine why the Pizza Party app crashes when the number of people is empty, the developer can call Log.d() from calculateClick().
2. When the user types a number and presses Calculate, the Log.d() method sends a debug message to the system log, and the number string is converted into an integer.
3. When the number of people is left blank, Integer.parseInt() throws an exception because numAttendStr is an empty string.
4. The exception is displayed in the Logcat, and the app crashes and closes abruptly. After several crashes, a dialog may appear with options to see app info or close the app.

Feedback?

Figure 1.5.2: Example stack trace in Logcat produced by uncaught NumberFormatException.

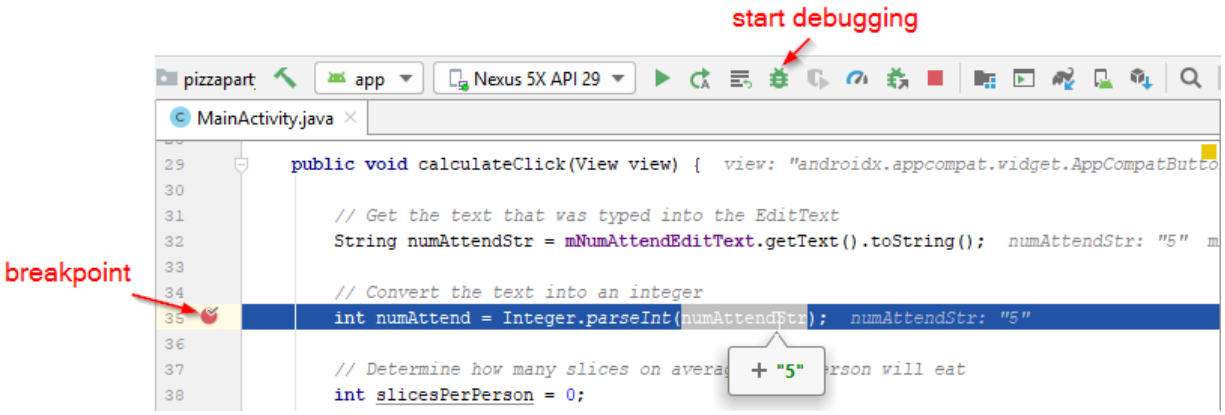
```
03-02 16:53:49.134 18515-18515/com.zybooks.pizzaparty E/AndroidRuntime: FATAL EXCEPTION: main
    Process: com.zybooks.pizzaparty, PID: 18515
    java.lang.IllegalStateException: Could not execute method for android:onClick
        at android.view.View$DeclaredOnClickListener.onClick(View.java:4452)
        at android.view.View.performClick(View.java:5198)
        at android.view.View$PerformClick.run(View.java:21147)
        at android.os.Handler.handleCallback(Handler.java:739)
        at android.os.Handler.dispatchMessage(Handler.java:95)
        at android.os.Looper.loop(Looper.java:148)
        at android.app.ActivityThread.main(ActivityThread.java:5417)
        at java.lang.reflect.Method.invoke(Native Method)
        at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:726)
        at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:616)
    Caused by: java.lang.reflect.InvocationTargetException
        at java.lang.reflect.Method.invoke(Native Method)
        at android.view.View$DeclaredOnClickListener.onClick(View.java:4447)
        at android.view.View.performClick(View.java:5198)
        at android.view.View$PerformClick.run(View.java:21147)
        at android.os.Handler.handleCallback(Handler.java:739)
        at android.os.Handler.dispatchMessage(Handler.java:95)
        at android.os.Looper.loop(Looper.java:148)
        at android.app.ActivityThread.main(ActivityThread.java:5417)
        at java.lang.reflect.Method.invoke(Native Method)
        at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:726)
        at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:616)
    Caused by: java.lang.NumberFormatException: Invalid int: ""
        at java.lang.Integer.invalidInt(Integer.java:138)
        at java.lang.Integer.parseInt(Integer.java:358)
        at java.lang.Integer.parseInt(Integer.java:334)
        at com.zybooks.pizzaparty.MainActivity.calculateClick(MainActivity.java:38)
        at java.lang.reflect.Method.invoke(Native Method)
        at android.view.View$DeclaredOnClickListener.onClick(View.java:4447)
        at android.view.View.performClick(View.java:5198)
        at android.view.View$PerformClick.run(View.java:21147)
        at android.os.Handler.handleCallback(Handler.java:739)
        at android.os.Handler.dispatchMessage(Handler.java:95)
        at android.os.Looper.loop(Looper.java:148)
        at android.app.ActivityThread.main(ActivityThread.java:5417)
        at java.lang.reflect.Method.invoke(Native Method)
        at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:726)
        at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:616)
etc...
```

Feedback?

breakpoint

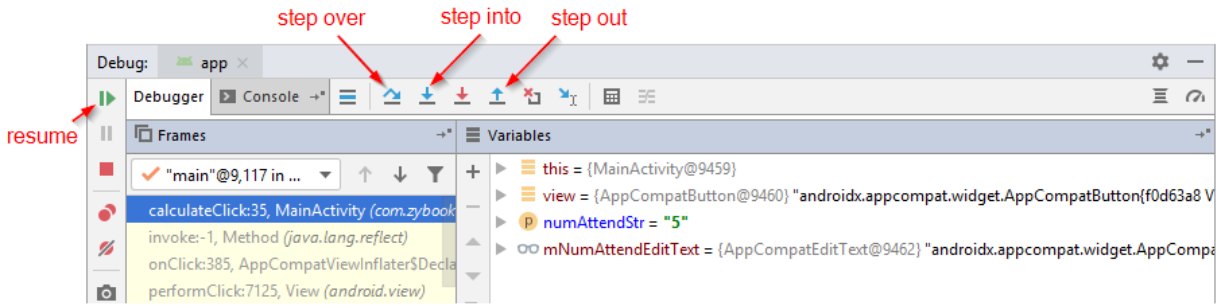
The debugger halts execution of the program when encountering a breakpoint.

Figure 1.5.3: Android Studio debugger halts execution at a breakpoint.



Feedback?

Figure 1.5.4: Buttons for controlling the debugger and variables displayed in the Variables list.



Feedback?

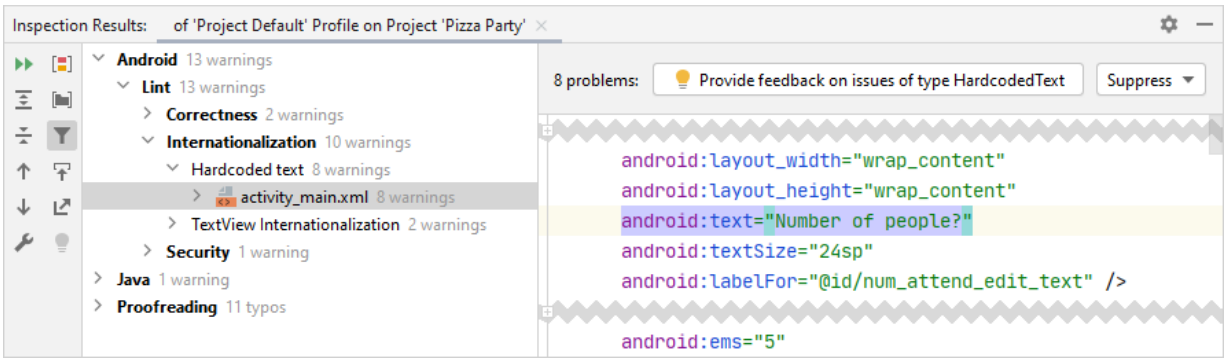
lint

Android Studio includes a tool called lint that identifies structural problems in an app's code and suggests fixes.

Figure 1.5.5: Left: Lint identifies an unused variable. Right: Lint suggests removing an unused variable.

Feedback?

Figure 1.5.6: Inspection Results produced by lint.



Feedback?

This section does not contain presentation elements.

How was this section? Provide feedback