

The background of the slide is a grayscale photograph of a wide, multi-tiered stone staircase. The staircase leads up towards a large, ornate building with multiple windows and a central dome. Several people are visible walking on the stairs, providing a sense of scale. The overall atmosphere is academic and formal.

Technik Mobiler Systeme

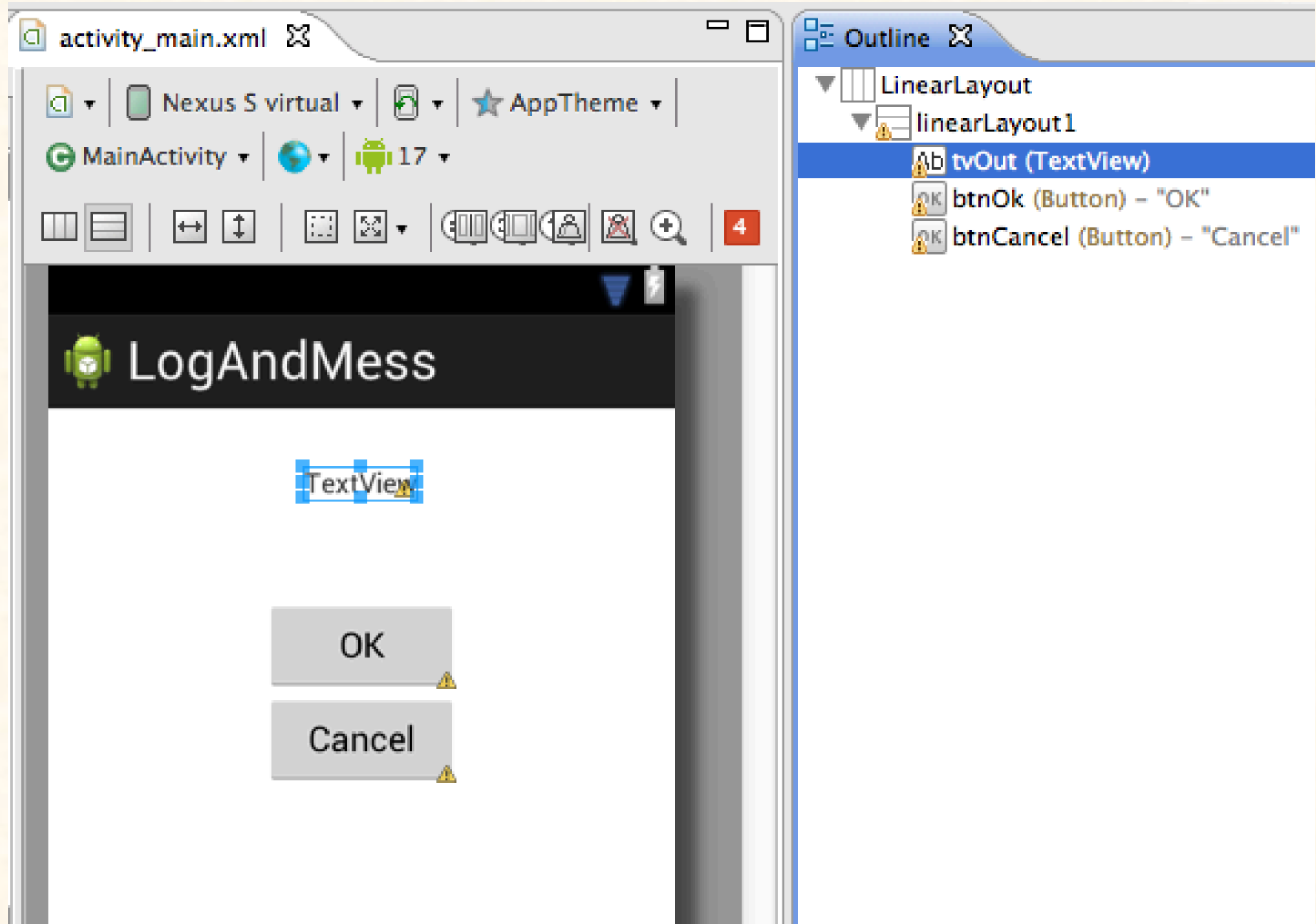
Debugging von Android Apps

Prof. Dr.-Ing. V.Brovkov

Debugging von Android Apps

- Um eine App zu testen, braucht man entsprechende Werkzeuge.
- Das Problem: die App läuft on einem separaten Gerät und auch auf einem anderen Prozessor-Typ (ARM/Cortex, kein Intel/AMD)
- In-Circuit Debugger ist in einem Smartphone nicht immer vorhanden 😊
- Die einfachsten Lösungen sind:
 - Ein TextView als Diagnostik Mittel
 - Eine Pop-Up Meldung (Class Toast)
 - Das Logcat (Systemlog)
- **Ein Projekt Beispiel: P0121_LogAndMess**

Debugging Example: Layout



Debugging mit TextView

- Ein TextView
 - Gut bekannt und offensichtlich
 - Das Text für Diagnostik kann entsprechend dem App-Ablauf geändert werden
 - Gut in Rahmen einer Activity
 - Benutzt die Nutzfläche der Activity

Debugging mit TextView: MainActivity.java

```
12 public class MainActivity extends Activity implements OnClickListener {
13
14     TextView tvOut;
15     Button btnOk;
16     Button btnCancel;
17     /** Called when the activity is first created. */
18     @Override
19     public void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22         // find Views
23
24         tvOut = (TextView) findViewById(R.id.tvOut);
25         btnOk = (Button) findViewById(R.id.btnOk);
26         btnCancel = (Button) findViewById(R.id.btnCancel);
27         // attach handler
28
29         btnOk.setOnClickListener(this);
30         btnCancel.setOnClickListener(this);
31     }
32     @Override
33     public void onClick(View v) {
34         // find signal source
35
36         switch (v.getId()) {
37             case R.id.btnOk:
38                 // OK
39
40                 tvOut.setText("OK pressed");
41
42                 break;
43             case R.id.btnCancel:
44                 // Cancel
45
46                 tvOut.setText("Cancel pressed");
47
48                 break;
49         }
50     }
51 }
```

1

2

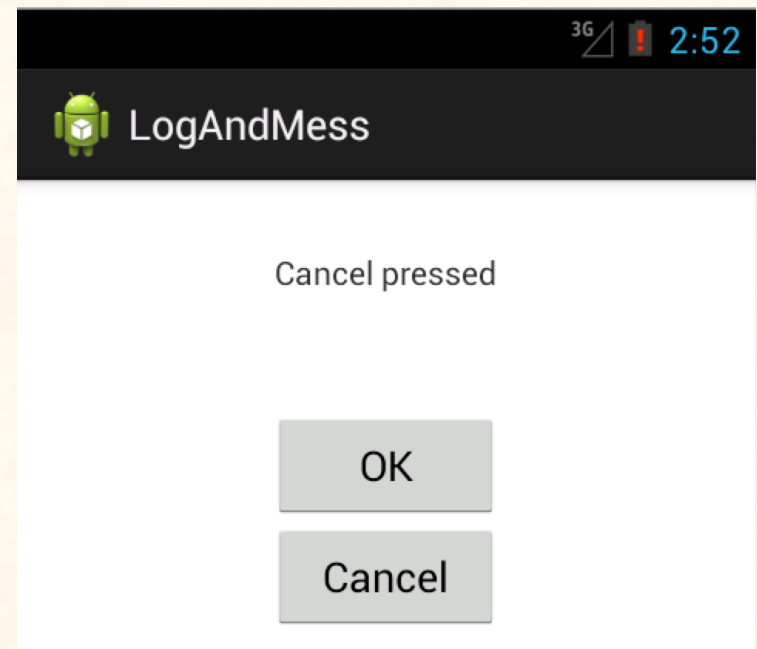
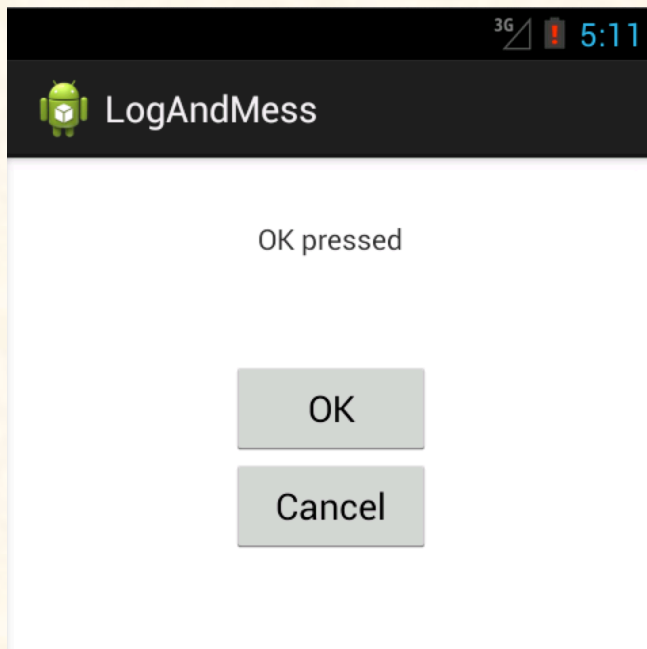
3

5

4

Debugging mit TextView: Summary

- in der Activity das Interface **OnClickListener** implementieren (1)
- Symbolische Namen für Objekte (2)
- Namen mit Objekten binden: **findViewById ()** (3)
- **onClick()** Method definieren und die passende Aktionen beschreiben (4)
- die Knöpfe mit den OnClickListener verbinden (5)



Logs und Toasts in Android Apps

- Eine Pop-Up Meldung (Class Toast)
 - braucht keine Extra-Nutzfläche der Activity
 - kann das Screen (kurz) überlagern
 - relativ kurze Meldungen und nur für die kurze Zeit sichtbar

Debugging mit Toast: MainActivity.java

```
12 public class MainActivity extends Activity implements OnClickListener {
13
14     TextView tvOut;
15     Button btnOk;
16     Button btnCancel;
17     /** Called when the activity is first created. */
18     @Override
19     public void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22         // find Views
23
24         tvOut = (TextView) findViewById(R.id.tvOut);
25         btnOk = (Button) findViewById(R.id.btnOk);
26         btnCancel = (Button) findViewById(R.id.btnCancel);
27         // attach handler
28
29         btnOk.setOnClickListener(this);
30         btnCancel.setOnClickListener(this);
31     }
32     @Override
33     public void onClick(View v) {
34         // find signal source
35
36         switch (v.getId()) {
37             case R.id.btnOk:
38                 // OK
39
40                 tvOut.setText("OK pressed");
41                 Toast.makeText(this, "OK pressed", Toast.LENGTH_LONG).show();
42                 break;
43             case R.id.btnCancel:
44                 // Cancel
45
46                 tvOut.setText("Cancel pressed");
47                 Toast.makeText(this, "Cancel pressed", Toast.LENGTH_LONG).show();
48                 break;
49         }
50     }
51 }
```

1

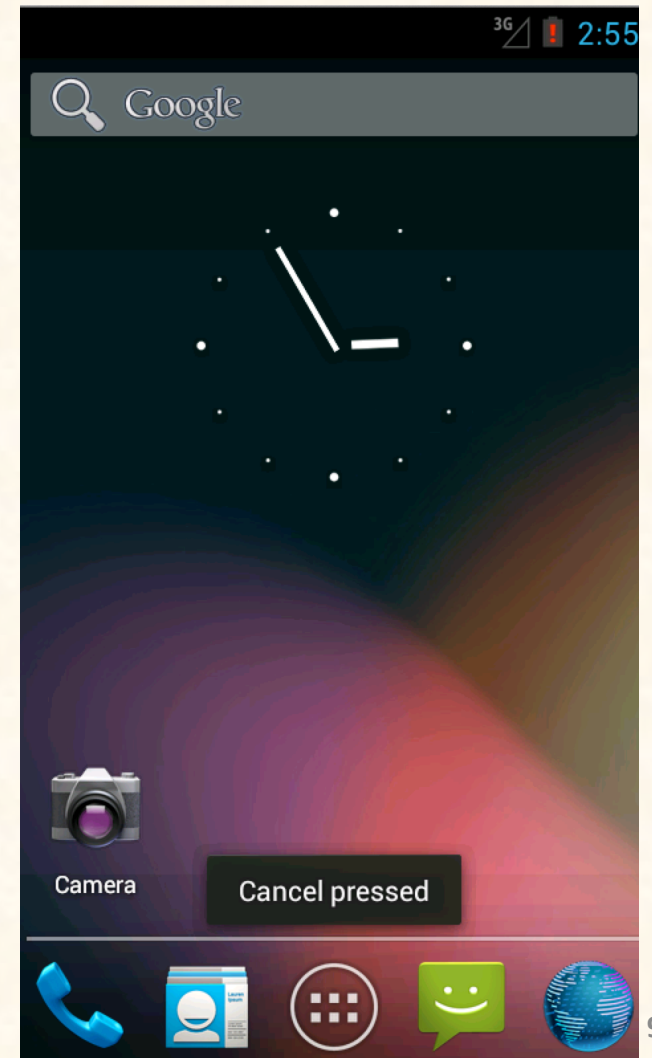
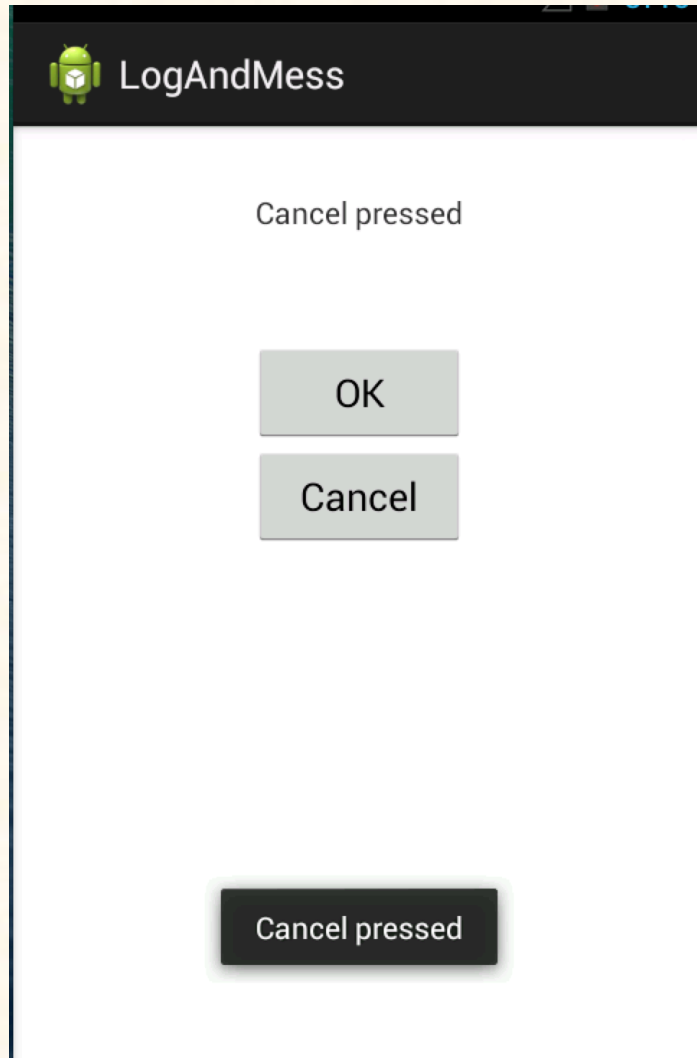
2

Debugging mit Toast: Summary

Ganz einfach:

Toast.makeText(...).show();

Nicht problemlos: beim
mehrfachen Klicken 😊



Logs und Toasts in Android Apps

- Das Logcat (Systemlog) des Android Smartphones
 - funktioniert als ein Datenfluss von Android Gerät (ein virtuelles oder einreelles) zur Programmierumgebung
 - mit Hilfe der Klasse **Log** können aus einer App die Textmeldungen zu einem Log-Fenster geliefert werden
 - Die Log-Datei kann ziemlich informative Meldungen empfangen und speichern
 - die Informationen werden nicht verloren
 - die Meldungen können gefiltert werden
 - jede Meldung enthält einen Timestamp
 - universelles Debuggingmittel für Android

Debugging mit LogCat: MainActivity.java

```
12 public class MainActivity extends Activity implements OnClickListener {
13     private static final String TAG = "myLogs";
14     TextView tvOut;
15     Button btnOk;
16     Button btnCancel;
17     /** Called when the activity is first created. */
18     @Override
19     public void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22         // find Views
23         Log.d(TAG, "find Views");
24         tvOut = (TextView) findViewById(R.id.tvOut);
25         btnOk = (Button) findViewById(R.id.btnOk);
26         btnCancel = (Button) findViewById(R.id.btnCancel);
27         // attach handler
28         Log.d(TAG, "attach handler");
29         btnOk.setOnClickListener(this);
30         btnCancel.setOnClickListener(this);
31     }
32     @Override
33     public void onClick(View v) {
34         // find signal source
35         Log.d(TAG, "find signal source");
36         switch (v.getId()) {
37             case R.id.btnOk:
38                 // OK
39                 Log.d(TAG, "OK pressed");
40                 tvOut.setText("OK pressed");
41                 Toast.makeText(this, "OK pressed", Toast.LENGTH_LONG).show();
42                 break;
43             case R.id.btnCancel:
44                 // Cancel
45                 Log.d(TAG, "Cancel pressed");
46                 tvOut.setText("Cancel pressed");
47                 Toast.makeText(this, "Cancel pressed", Toast.LENGTH_LONG).show();
48                 break;
49         }
50     }
51 }
```

1

2

3

4

5

6

Debugging mit LogCat: MainActivity.java

- Gans einfach:

private static final String TAG = "myLogs";

und dann überall wo es erforderlich ist:

Log.d(TAG, "OK pressed");

Lev	Time	PID	TID	Application	Tag	Text
D	12-29 17:19:17.710	2018	2018	ua.opu.brovkov	myLogs	find Views
D	12-29 17:19:17.710	2018	2018	ua.opu.brovkov	myLogs	attach handler
D	12-29 17:19:33.710	2018	2018	ua.opu.brovkov	myLogs	find signal source
D	12-29 17:19:33.710	2018	2018	ua.opu.brovkov	myLogs	OK pressed
D	12-29 17:19:35.400	2018	2018	ua.opu.brovkov	myLogs	find signal source
D	12-29 17:19:35.400	2018	2018	ua.opu.brovkov	myLogs	Cancel pressed

Fragen?