


Android User Interface  
Android Smartphone Programming

University of Freiburg

Matthias Keil / Tim Aicher  
Institute for Computer Science  
Faculty of Engineering  
University of Freiburg  
20. Oktober 2017



Notizen

---

---

---

---

---

---



---

---

Outline

University of Freiburg

- 1 Android User Interface
- 2 Multi-Language Support
- 3 Summary

Matthias Keil / Tim Aicher    Android User Interface    20. Oktober 2017    2 / 19

Notizen

---

---

---

---

---

---

---

---

Android User Interface



From activity to widgets

University of Freiburg

**Activity** Application component that provides a screen [1].  
User interface of an activity is build using View and ViewGroup objects [5].

**View** Basis unit for user interface, base for subclasses called *widgets*.

**ViewGroup** Base for subclasses called *layouts*.

Matthias Keil / Tim Aicher    Android User Interface    20. Oktober 2017    3 / 19

Notizen

---

---

---

---

---

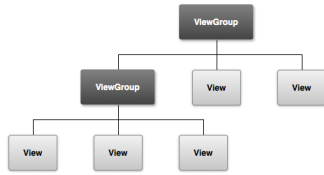
---

---

---

## Android User Interface

View Hierarchy  
University of Freiburg



Android View Hierarchy containing ViewGroup objects as nodes and View objects as leafs.



Notizen

---

---

---

---

---

---

---

---

## Android User Interface

Layout  
University of Freiburg



- Can be defined in an **XML** layout file [7].
- Similar to HTML layout development.
- Each element is a View or ViewGroup object or a subclass of these.
- ViewGroup objects contain more Views or ViewGroup objects.



Notizen

---

---

---

---

---

---

---

---

## Android User Interface

Example XML layout  
University of Freiburg



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.
  android.com/apk/res/android"
3   android:layout_width="fill_parent"
4   android:layout_height="fill_parent"
5   android:orientation="vertical" >
6   <TextView android:id="@+id/text"
7     android:layout_width="wrap_content"
8     android:layout_height="wrap_content"
9     android:text="Hello, _I_u_a_m_a_TextView" />
10  <Button android:id="@+id/button"
11    android:layout_width="wrap_content"
12    android:layout_height="wrap_content"
13    android:text="Hello, _I_u_a_m_a_Button" />
14 </LinearLayout>
```



Notizen

---

---

---

---

---

---

---

---

## Android User Interface

Widget  
University of Freiburg



- Subclass of View.
- Serves as interaction interface with user.
- Many fully implemented widgets available.
  - Examples: *Button*, *Checkbox*, *EditText* and many more.
  - Advanced Example *WebView*: Displays web pages and can use JavaScript [6].
- Own implementation enables full customization of behavior.



Matthias Keil / Tim Aicher

Android User Interface

20. Oktober 2017

7 / 19

Notizen

---

---

---

---

---

---

---

---

## Android User Interface

Input Events  
University of Freiburg



- Many ways to intercept events from user interaction.
- Approach for user interface events: Capture events from View objects the user interacts with [2].
- Two ways of implementation:
  - Overwrite existing callback method.
  - Define own event listener.
- Mostly used: Defining event listeners.



Matthias Keil / Tim Aicher

Android User Interface

20. Oktober 2017

8 / 19

Notizen

---

---

---

---

---

---

---

---

## Android User Interface

Example: Overwriting Callback Method  
University of Freiburg



```
1 public class MyActivity extends Activity {  
2     ...  
3     @Override  
4     public boolean onKeyDown (int keyCode ,  
5         KeyEvent event) {  
6         // Do something.  
7     }  
}
```



Matthias Keil / Tim Aicher

Android User Interface

20. Oktober 2017

9 / 19

Notizen

---

---

---

---

---

---

---

---

## Android User Interface

Example: Defining own Event Listener  
University of Freiburg



```
1 public class MyActivity extends Activity {
2     private OnClickListener myListener = new
      OnClickListener() {
3         public void onClick(View v) {
4             // Do something.
5         }
6     };
7
8     public void onCreate(Bundle state) {
9         ...
10        Button button = (Button)findViewById(R.id.
      myButton);
11        button.setOnClickListener(myListener);
12        ...
13    }
14 }
```



Notizen

---

---

---

---

---

---

---

---

## Android User Interface

Intents and Broadcast Receivers  
University of Freiburg



**Intent** Message to communicate between components. [3].  
Can connect components in the same or in different applications.  
Starts activities, background processes or notifies broadcast receivers.

**Broadcast Receiver** Can be registered to receive certain intents.  
Example: Intent sent from system indicates incoming call and application stops playing music.



Notizen

---

---

---

---

---

---

---

---

## Android User Interface

Example: Intent to call telephone number  
University of Freiburg



- Intent starts activity by specifying what action should be performed.
- Note: Activity only implicitly given though action.

```
1 Intent intent = new Intent(Intent.ACTION_DIAL,
      Uri.parse("tel:5905-5635"));
2 startActivity(intent);
```



Notizen

---

---

---

---

---

---

---

---

## Android User Interface

Example: Broadcast receiver to react to phone calls  
University of Freiburg



- Step 1: Create broadcast receiver as a new class.

```
1 public class MyPhoneReceiver extends  
  BroadcastReceiver {  
2   @Override  
3   public void onReceive(Context context, Intent  
    intent) {  
4     // Do something.  
5   }  
6 }
```



Notizen

---

---

---

---

---

---

---

---

## Android User Interface

Example: Broadcast receiver to react to phone calls  
University of Freiburg



- Step 2: Extend *AndroidManifest.xml* to register broadcast receiver to intents.

```
1 <application ... >  
2   <receiver android:name="MyPhoneReceiver" >  
3     <intent-filter>  
4       <action android:name="android.intent.  
        action.PHONE_STATE" >  
5     </action>  
6   </intent-filter>  
7 </receiver>  
8 </application>
```



Notizen

---

---

---

---

---

---

---

---

## Multi-Language Support

Overview  
University of Freiburg



- Done though localization: Switch language according to locale settings of the device [4].
- Helps reaching more users.
- Easy though separation of string resources and application code.
- Refer to string names in code and define strings in resource files.



Notizen

---

---

---

---

---

---

---

---

## Multi-Language Support

Resources  
University of Freiburg



- Default resources in `res/values/strings.xml` provides all strings used.
- Special language resource files like e.g. `res/values-de/strings.xml` provides adjusted strings.
- If no special resource file exists, default is used.



Matthias Keil / Tim Aicher

Android User Interface

20. Oktober 2017

16 / 19

Notizen

---

---

---

---

---

---

---

---

## Multi-Language Support

Example  
University of Freiburg



- In Activity

```
1 tv = new TextView(this);  
2 tv.setText(R.string.example);
```

- In `res/values/strings.xml`

```
1 <string name="example">Example</string>
```

- In `res/values-de/strings.xml`

```
1 <string name="example">Beispiel</string>
```



Matthias Keil / Tim Aicher

Android User Interface

20. Oktober 2017

17 / 19

Notizen

---

---

---

---

---

---

---

---

## Summary

University of Freiburg



- User interfaces of activities are build through *View* and *ViewGroup* objects.
- *ViewGroup* subclasses are *layouts* that group other *ViewGroup* or *View* objects.
- *View* subclasses are *widgets* for user interaction like button, checkbox and so on.
- Enabling user interaction is implemented by *capturing input events*.
- *Intents* are messages and can be received through broadcast receivers.
- Multi-language support is implemented through *resource files* for strings.



Matthias Keil / Tim Aicher

Android User Interface

20. Oktober 2017

18 / 19

Notizen

---

---

---

---

---

---

---

---

# Bibliography

University of Freiburg



- ANDROID DEVELOPERS.  
Activities.  
<http://developer.android.com/guide/topics/fundamentals/activities.html>.
- ANDROID DEVELOPERS.  
Input Events.  
<http://developer.android.com/guide/topics/ui/ui-events.html>.
- ANDROID DEVELOPERS.  
Intents and Intent Filters.  
<http://developer.android.com/guide/topics/intents/intents-filters.html>.
- ANDROID DEVELOPERS.  
Localization.  
<http://developer.android.com/guide/topics/resources/localization.html>.
- ANDROID DEVELOPERS.  
User Interface.  
<http://developer.android.com/guide/topics/ui/index.html>.
- ANDROID DEVELOPERS.  
WebView.  
<http://developer.android.com/reference/android/webkit/WebView.html>.
- ANDROID DEVELOPERS.  
XML Layouts.  
<http://developer.android.com/guide/topics/ui/declaring-layout.html>.



## Notizen

---

---

---

---

---

---

---

---

## Notizen

---

---

---

---

---

---

---

---

## Notizen

---

---

---

---

---

---

---

---