Activities

von Mattis Küper

Agenda

- Was sind Activities
- Struktur von Apps
  - Back Stack Diagramm
  - Manifestdatei
  - build.gradle-Datei
  - Trennung von Programmlogik und Ressourcen
  - strings.xml
  - Auf Ressourcen zugreifen
- Lebenszyklus von Activities
  - Zustand speichern und wiederherstellen
  - Wichtige Callback-Methoden
  - Activity Lifecycle Diagramm

Was sind Activities?

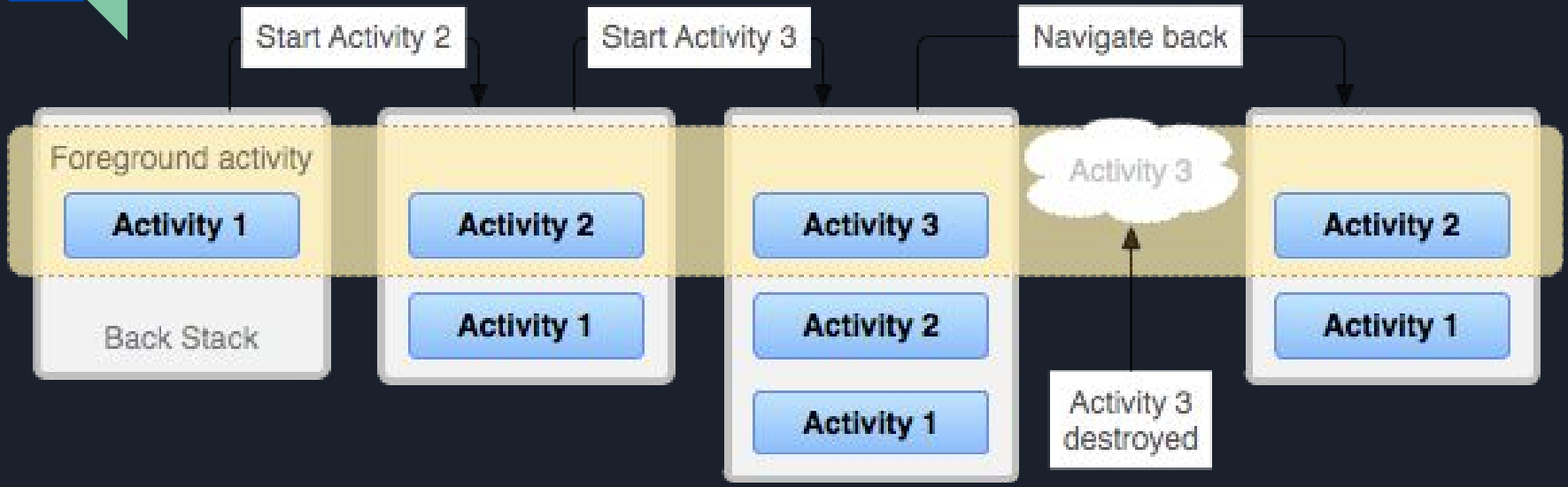
Definition:

Eine Activity ist eine Aktion die den ganzen oder einen Teil des Bildschirms ausfüllt.

Struktur von Apps

- eine App besteht aus mehreren Activities
- hat eine Hauptaktivität die beim Start ausgeführt wird
- jede weitere Activity ist eine Folgeaktivität
- beim ausführen einer Folgeaktivität wird die aktuelle angehalten und auf den Back Stack gelegt
- Zurück-Taste beendet die aktuelle Activity und reaktiviert die im Back Stack nächst Höchste

Back Stack Diagramm



Quelle: https://developer.android.com/guide/components/activities/tasks-and-back-stack.html

Manifestdatei

- jede App hat eine AndroidManifest.xml
- hat Liste der App-Komponenten, Berechtigungen, Anforderungen und Bibliotheken
- jede Activity benötigt android:name
- android:label und android:icon stellen Namen und Icon der App dar
- durch <intent-filter> können Activities zur Haupt- oder Startaktivität gemacht werden
  - <action android:name=”android.intent.action.MAIN” /> ist die Hauptaktivität der App
  - <action android:name=”android.intent.category.LAUNCHER” /> ist die Startaktivität der App
- ein führender . bei dem name-Attribut ersetzt den . durch das im package eingetragene Paket

build.gradle-Datei

- enthält Informationen zu Versionen
- minSdkVersion ist die Android Version die mindestens vorhanden sein muss
- targetSdkVersion ist die Android Version, für die die App optimiert und getestet wurde
- versionCode und versionName geben die Version der App an
- versionName z.B. 1.0, 1.1.2 usw.
- versionCode sollte bei jeder neuen Version um 1 erhöht werden (Google Standard)

Trennung von Programmlogik und Ressourcen

- alle Strings werden in strings.xml gespeichert und nur referenziert
  - einfacheres Übersetzen der App in mehrere Sprachen
  - leichteres finden von identischen Texten
- path der strings.xml = res/values/strings.xml
  - values ist Standardsprache (empfohlen ist englisch)
  - für weitere Sprachen werden weitere values-Ordner angelegt, die mit dem entsprechenden ISO-Sprachschlüssel enden (z.B. values-de)
- Vektorgrafiken, Bilder und ähnliches ist im drawable Ordner gespeichert
  - werden mit @drawable/<Dateiname> referenziert
  - weitere drawable Ordner für andere Auflösungen (z.B. drawable-hdpi und drawable-ldpi)

strings.xml

- besteht aus <string name=”stringname”> in denen der entsprechende String steht
- leichter zu bearbeiten mit Translation Editor
  - nicht übersetzte Strings werden rot dargestellt
  - nicht übersetzbare Strings können mit Untranslatable gekennzeichnet werden

Auf Ressourcen zugreifen

- für Strings deren Inhalte zur Laufzeit bestimmt werden lege in strings.xml z.B. <string name=”example”>Welcome %1$s, your value is: %2$d</string> an
- die Platzhalter werden dann mit getString() gefüllt, bei unserem Beispiel wie folgt: getString(R.string.example, “User”, 42)
  - %<Zahl> ist welcher Parameter wo eingesetzt wird
  - $d ist für Zahlen, $s für Zeichenketten
- einfache Resource wie Booleans, Integer oder Farben werden in diverses.xml gespeichert
  - werden z.B. mit getResources().getInteger(R.integer.name) referenziert

Lebenszyklus von Activities

- beginnt immer mit OnCreate(), hat vier Aufgaben
  - Aufrufen der gleichnamigen Elternmethode
  - Initialisieren von Instanzvariablen
  - Setzen der Benutzeroberfläche
  - Wiederherstellen eines gespeicherten Zustandes
- setContentView() setzt die Benutzeroberfläche
  - übergebene Layouts werden in XML-Dateien gespeichert

Zustand speichern und wiederherstellen

- werden in savedInstanceState gespeichert
  - ist ein Bundle, welche aus Key-Value-Paaren bestehen
- Aktionen wie das drehen des Smartphones stoppen die Activity und starten sie erneut
- mit onSaveInstanceState() können savedInstanceState-Bundles gesetzt werden
- put… Methoden um Werte zu setzen, get… um Werte zu erhalten
- Inhalte, wie Texte, und Status der Bedienelemente werden automatisch gespeichert
- sollte nicht für das speichern von Nutzdaten verwendet werden, dafür onPause()
  - Nutzdaten: Alles was in Dateien/Datenbanken gespeichert werden soll

Wichtige Callback-Methoden

- Nutzdaten sollten bei onPause() gespeichert werden
- Lifecycle verwendet den bereits vorgestellten Back Stack (last in, first out)

- onCreate(): Wird beim Erstellen einer Activity ausgeführt
- onStart(): Wird beim Start einer Activity ausgeführt
- onResume(): Wird vor dem Ausführen einer Activity ausgeführt
- onPause(): Wird ausgeführt, wenn andere Activity in den Vordergrund gerät
- onStop(): Wird ausgeführt, wenn Activity nicht mehr sichtbar ist
- onRestart(): Wird ausgeführt, wenn die Activity wieder Sichtbar wird
- onDestroy(): Wird bei der Terminierung der Activity ausgeführt

Quelle: https://developer.android.com/reference/android/app/Activity.html#ActivityLifecycle

Vielen Dank für eure Aufmerksamkeit!

Noch fragen?