Fragments

Repo: https://github.com/karlmorris/AndroidFragments
Overview

- What are fragments and how are they used?
- Fragment lifecycle
- FragmentManager and FragmentTransaction
- Fragment Design Principles
- Implementing a Fragment
Fragments

- A modular portion of an Activity
- Introduced in Android SDK ver 11
- Multiple fragments can be combined in one activity
- They share a many to many relationship with activities
  - Can be reused
- Allows for the creation of dynamic UI's, as well as support for larger screens
Fragments

- Tablet
  - Selecting an item updates Fragment B
  - Activity A contains Fragment A and Fragment B

- Handset
  - Selecting an item starts Activity B
  - Activity A contains Fragment A
  - Activity B contains Fragment B
Fragments

- Think of fragments as building blocks
- They can be contained inside activities, or inside other fragments
- Each activity (and fragment) has its own FragmentManager object
- A FragmentManager allows us to manipulate fragments in the parent
Fragment Manager

- Allows actions such as adding, removing, and replacing fragments within its parent using Transactions
- Each action begins with retrieving a Transaction object: `fragmentManager.beginTransaction()`
- Any number of actions can be contained in a single transaction
- Transactions are executed by calling `commit()` on the transaction object
Fragment Layout

• When added to an activity, a fragment resides in a ViewGroup (an element within the parent's layout)

• Each fragment contains and manages its own layout

• A fragment needn't be attached to its parent's layout. In that case, the fragment is invisible and can act as a worker
Fragments

- Must be associated with an activity (or with a Fragment that is associated with an activity)
- Housed in the parent activity's back stack
- Possess their own lifecycle, but affected by lifecycle of parent
  - e.g. A paused activity will pause all child fragments
Fragments

- Can be added to an activity at design time or runtime
  - `<fragment>` or `<FrameLayout>` tag in activity's layout file
  - Using FragmentTransaction
Proper Fragment Design

- Fragments should be designed to be modular (reusable)
- Allow fragments to utilize the space available on the device
  - single fragment view on small screens, multiple fragments on larger screens
- Limit dependency among fragments
- Communicate only through parent activity
  - Fragment → Activity: interface
  - Activity → Fragment: public methods
Creating a Fragment

- Extend the Fragment class (or other subclass) and implement the lifecycle methods
- Unlike the Activity class, you use the onCreateView() method to create the layout
- Other methods can be implemented as needed for such things are saving and restoring state information