Android Concepts

Fragments

Introduction

A fragment is a reusable class implementing a portion of an activity

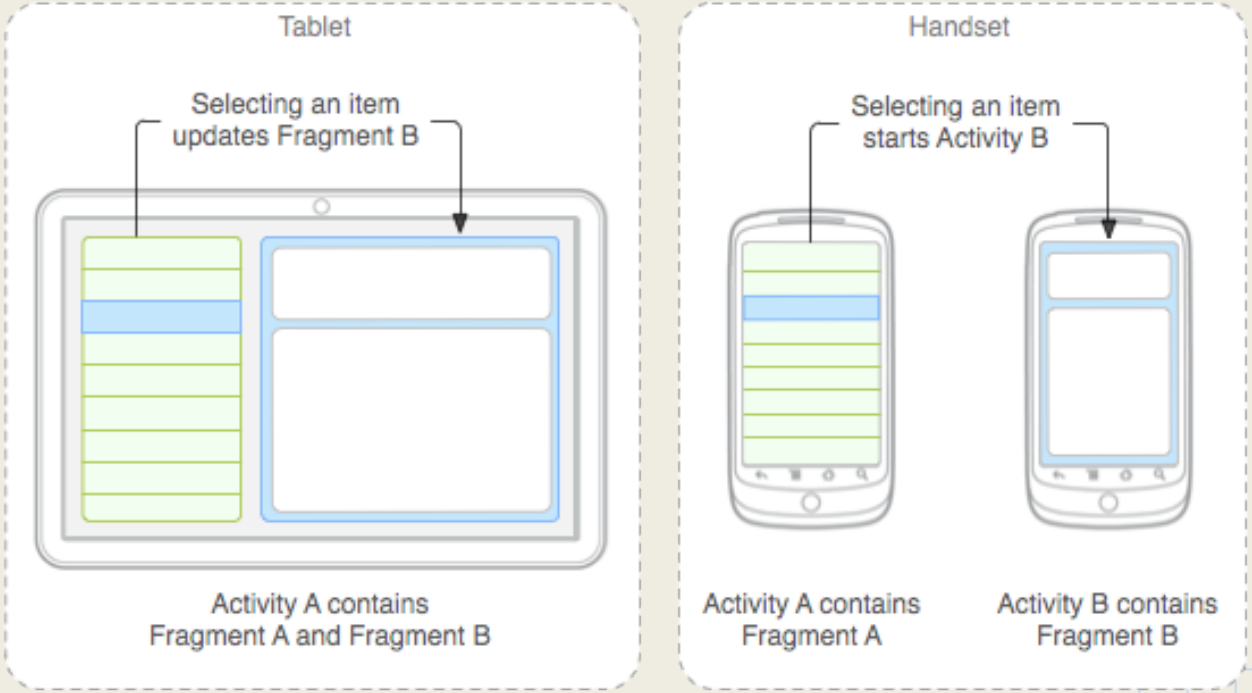
Fragments must be embedded in activities; they cannot run independently of activities.
To interact with fragments we use Fragment manager which can be obtained via Activity.getFragmentManager() and Fragment.getFragmentManager()

A fragment has a separate layout XML file

A Fragment encapsulates functionality so that it is easier to reuse within activities and layouts.

Fragments can be dynamically (Java) or statically (XML) added to an activity layout.

Reusing Fragments



Defining a Fragment

A fragment just like an activity has an xml layout file and a java class that represents a Fragment controller

```
import android.support.v4.app.Fragment;

public class FooFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
      Bundle savedInstanceState) {
      // Defines the xml file for the fragment
      View view = inflater.inflate(R.layout.foo, container, false);
      // Setup handles to view objects here
      // etFoo = (EditText) view.findViewById(R.id.etFoo);
      return view;
    }
}
```

Inserting a Fragment in an Activity

There are two ways to add a fragment in an activity: dynamically and statically.

Any activity using fragment should extend AppCompatActivity or FragmentActivity.

Adding Fragment: Statically

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <fragment
        android:name="com.example.android.FooFragment"
        android:id="@+id/fooFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

Adding Fragment:Dynamically

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

  <FrameLayout
      android:id="@+id/your_placeholder"
      android:layout_height="match_parent">
  </FrameLayout>

</LinearLayout>
```

In your activity's layout, you add a "placeholder" container usually a FrameLayout where a fragment is inserted at runtime.

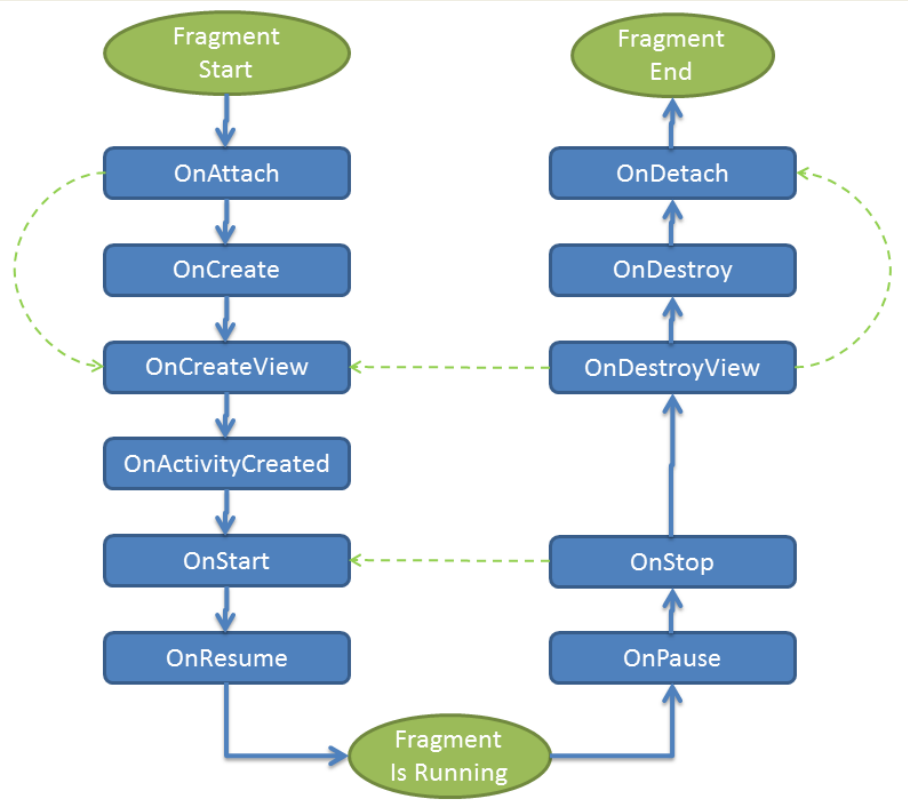
More on FrameLayout: http://developer.android.com/reference/android/widget/FrameLayout.html

Adding Fragment:Dynamically Cont'd

```
// Begin the transaction
FragmentTransaction ft = getSupportFragmentManager().beginTransaction();
// Replace the contents of the container with the new fragment
ft.replace(R.id.your_placeholder, new FooFragment());
// or ft.add(R.id.your_placeholder, new FooFragment());
// Complete the changes added above
ft.commit();
```

FragmentManager class and the FragmentTransaction class allow you add, remove and replace fragments in the layout and replace fragments in your activity's layout.

Fragment Lifecycle



Fragment Methods

- onAttach() - called when a fragment is connected to an activity
- onCreate() - is called to do initial creation of a fragment
- onCreateView() - called to inflate view i.e. the xml layout
- onActivityCreated() - called when host activity has completed its onCreate() method
- onStart()
- onResume()
- onPause()
- onDestroyView()
- onDestroy()
- onDetach()

Layout Weight

This attribute assigns an importance value to a view in terms of how much space it should occupy on the screen.

NB: A larger weight value allows it to expand to fill any remaining space in the parent view.

Resourceful Links

Fragments: https://github.com/codepath/android_guides/wiki/Creating-and-Using-Fragments

Layout weight: http://developer.android.com/guide/topics/ui/layout/linear.html#Weight