

IT-ELEC3C

PRELIMS

Intents

- messaging object used to request an action from another app component
- facilitates communication between different components

Types of Intents

- Explicit Intent
 - communicates between two activities inside the same application
- Implicit Intent
 - communicates between two activities of different application
 - e.g., Facebook launching the Chrome browser

Intents are used to

- start an activity
- start a service
 - open email
 - open web browser
 - open camera
- pass data in same application or different application

Starting an Activity with Intent

```
1 Intent myIntent = new Intent (this,NextActivity.class);
2 startActivity(myIntent);
```

Android Fragments

- baby activity
 - attachable and detachable dynamically
- represents a behavior or a portion of user interface in an Activity
- modular section of an activity
 - has its own lifecycle
 - receives its own input events
 - which you can add or remove while the activity is running
- must always be embedded in an activity and the fragment's lifecycle is directly affected by the host activity's lifecycle

- need not have any UI (just a UI-less helper service for an activity)
- transactions can also be added to the activity backstack. So when you hit the back button, fragment is "undone"
- first added in Android 3.0 (usable in older versions if necessary)

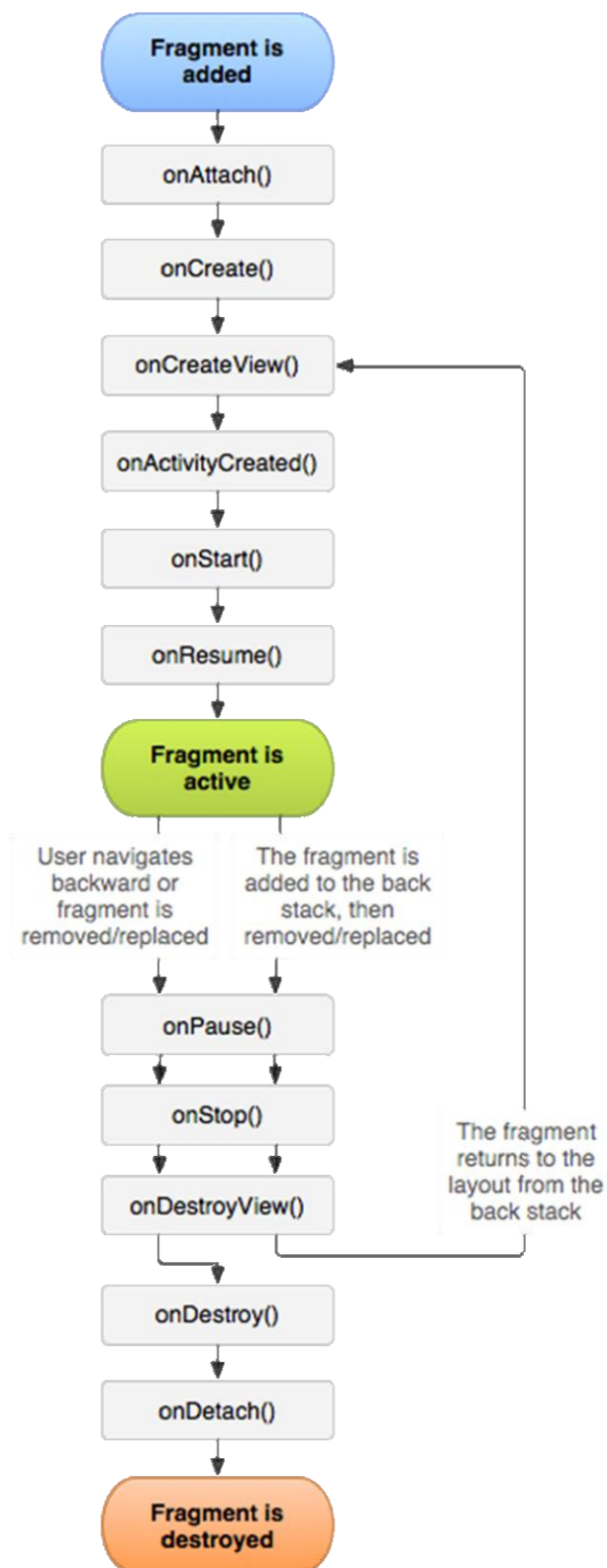
Advantages

- Dynamic Views/Layouts
 - two UI modules defined by fragments can be combined into one activity for a tablet design, but separated for a handset design
- Reusable Layouts
 - situational layout causes redundancy
 - code should behave situationally
 - in portrait mode, clicking a button should launch a new activity
 - in landscape mode, clicking a button should launch a new view
- Same layouts can be used in multiple devices
 - fragment is a reusable segment of Android UI that can appear in an activity
 - can help handle different devices and screen sizes
 - can reuse a common fragment across multiple activities

Creating a Fragment

1. In Android studio, right-click app
2. Click New > Fragment > Fragment (blank)
3. Uncheck boxes that include methods
4. Create XML layout and Java event code

Fragment Lifecycle



Fragment Lifecycle

- fragments have a similar lifecycle and events as activities
- important methods:
 - `onAttach`
 - to glue fragment to its surrounding activity
 - `onCreate`
 - when fragment is loading
 - `onCreateView`
 - method that must return fragment's root UI view
 - `onActivityCreated`
 - method that indicates the enclosing activity is ready
 - `onPause`
 - when fragment is being left/exited
 - `onDetach`
 - just as fragment is being deleted

Activity State	Fragment Callbacks
Created	<code>onAttach()</code>
	<code>onCreate()</code>
	<code>onCreateView()</code>
	<code>onActivityCreated()</code>
Started	<code>onStart()</code>
Resume	<code>onResume()</code>
Paused	<code>onPause()</code>
Stopped	<code>onStop()</code>
Destroyed	<code>onDestroyView()</code>
	<code>onDestroy()</code>
	<code>onDetach()</code>

Ways of Adding Fragments

- XML Layout
 - `<fragment>` element
 - when you add a fragment to an activity layout by defining the fragment in the XML layout file, you cannot remove the fragment at runtime
- Activity at runtime
 - rather than defining the fragments for an activity in the

layout file, you can add a fragment to the activity during runtime

- this is necessary if you plan to change fragments during the lifetime of the activity
- each set of changes to the activity is called a transaction

Perform a Transaction

- use `FragmentManager` to create a `FragmentTransaction`
- provides APIs to:
 - add
 - remove
 - replace
- inside your activity, call `getSupportFragmentManager()` to get a `FragmentManager` using the Support Library APIs
- then call `beginTransaction()` to create a `FragmentTransaction`; and
- call `add()` to add a fragment
- you can perform multiple fragment transaction for the activity using the same `FragmentTransaction`
- when you're ready to make the changes, you must call `commit()`
- an important rule when dealing with fragments (especially when adding fragments at runtime) is that your activity layout must include a container *View* in which you can insert the fragment