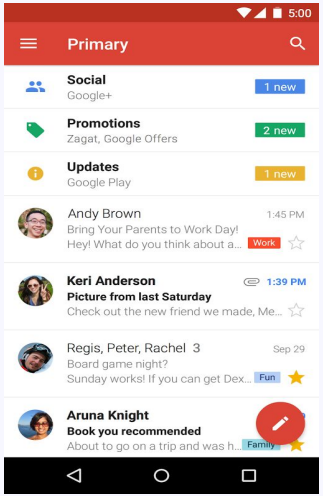
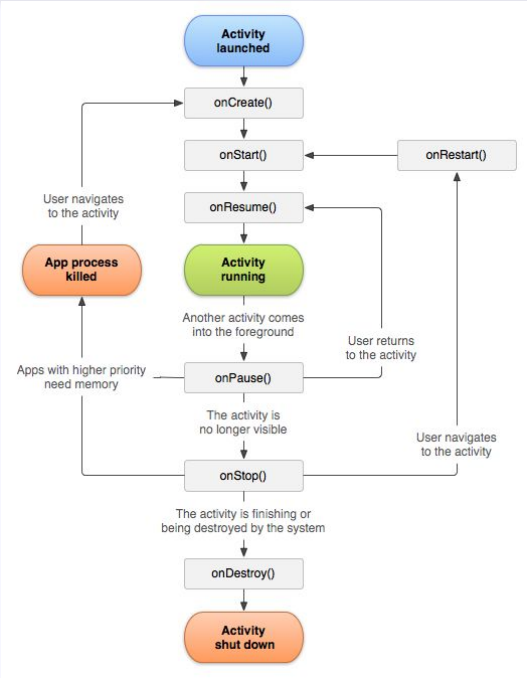
Activities and Fragments

What are Activities?

- "A single, focused thing that the user can do"
  - The Activity class creates a window for you to place your UI
  - Normally presented to the user as a full-screen window
- "An activity is an application component that provides a screen with which users can interact in order to do something"
  - Dial the phone
  - Take a photo
  - Send an email
  - View a map



Activity Lifecycle



Activity Lifecycle

- Different States
  - Active and running
    - Activity is in the foreground of the screen
  - Paused
    - Activity has lost focus but is still visible
    - Activity is still alive (maintains current state)
  - Stopped
    - Activity is obscured by another activity; no longer visible
    - Retains all state and member info

Lifecycle Callback Methods

- onPause():
  - Called when activity is about to lose focus to another
  - Use this to save any persistent state
- onResume():
  - Called for your activity to start interacting with the user
  - Good time to begin animations, open camera
- onStop():
  - Called when you are no longer visible to the user
- onRestart():
  - Called after onStop() when the user has navigated back to an activity. Followed by an onStart() and onResume()

How to Create an Activity

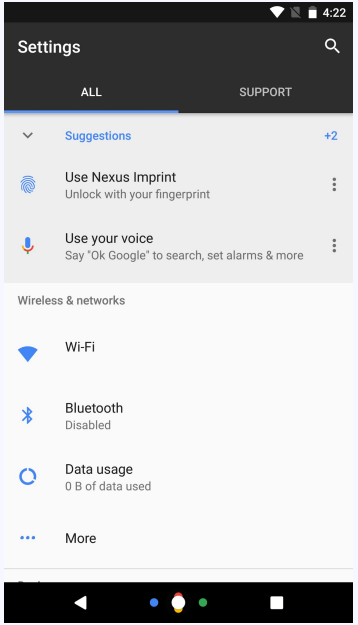
- Create an Activity class that extends an existing subclass of Activity
  - In most cases, extend AppCompatActivity
- Two methods that need to be implemented:
  - onCreate(Bundle)
    - Where the activity is initialized
    - setContentView(int) is called here to define the layout
  - onPause()
    - This method is called when the user leaves the activity
    - Normally used to persist any changes that the developer wants

Creating Activities continued

- After writing code for an activity, make sure to add it to the manifest file
  - If you don't, your app will crash
  - The activity won't be accessible to the system
- Intent Filters
  - Intent filters declare how other application components activate it
    - Launcher filter declares that certain activity should appear in app launcher
    - Send filter declares that the activity handles "send" actions

Different Activity Types

- ListActivity
  - An activity that displays a list of items by binding to a data source (array)
  - Hosts a ListView object
- PreferenceActivity
  - Used to make the settings page for apps
  - Shows a hierarchy of preferences to the user

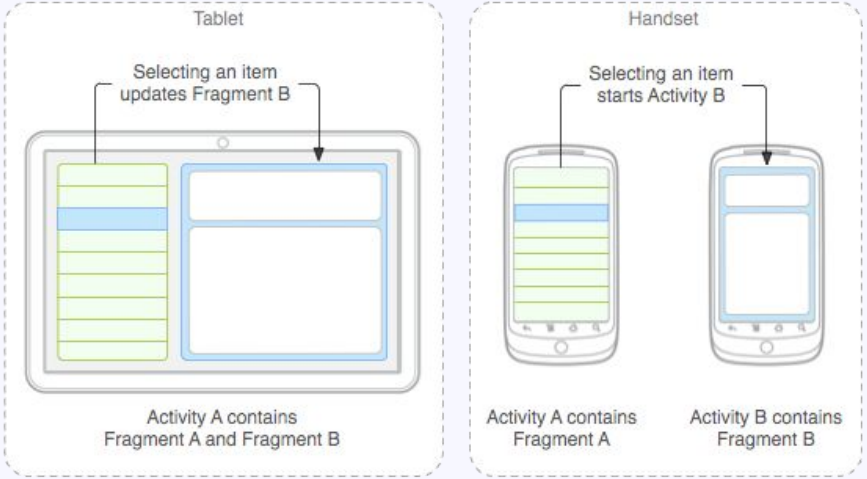


Intents

- A messaging object you can use to request an action from another app component
- You can use intents to transition between activities
- You can use intents to start services and perform one time operations

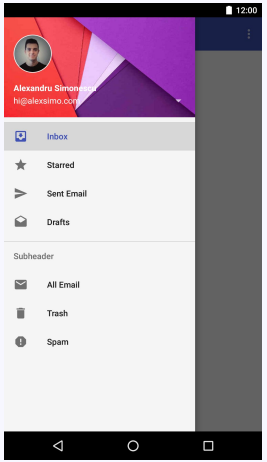
What are Fragments?

- “A fragment represents a behavior or a portion of user interface in an Activity”
- Used to build multi-plane UIs
- Reusable
- Introduced in Android 3.0 Honeycomb
    - Very useful for tablet optimized apps



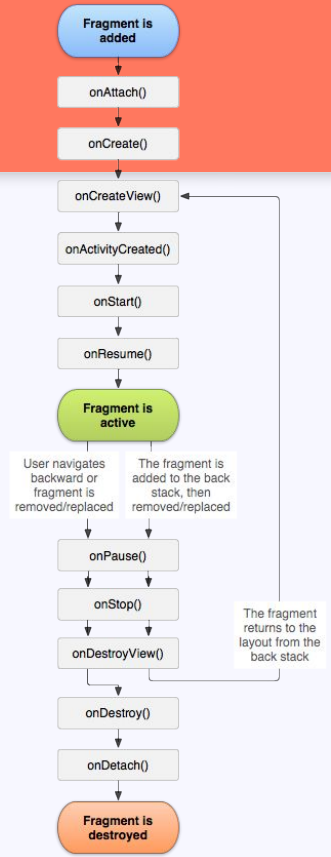
More on Fragments

- Must be embedded within an Activity
  - Fragment lifecycle is affected by the activity's lifecycle
  - When an activity is paused so are all the fragments within it
- Activities can manipulate individual fragments
- Fragments live within a ViewGroup inside of the host activity's view hierarchy



Fragment Lifecycle

- onCreate()
  - Called when creating the fragment
  - Initialize essential components of the fragment here
- onCreateView()
  - You will most likely be changing this
  - Similar to the onCreate for activities (setup UI here)
- onPause()
  - Called when the user is leaving the fragment
  - Similar to onPause for activities



Creating Fragments

- Start out by creating a subclass of Fragment (similar to activities)
- Declare an onCreateView() method that overrides its parent class' behavior
  - Inflate your layout resource file here
  - Return the view at the end
- Add the fragment to an activity
  - One option is to declare the fragment inside the activity's layout file
    - Treat the fragment similar to a view
  - Second option is to programmatically add it
    - Use a FragmentTransaction to replace a ViewGroup with the fragment

Different Fragment Types

- DialogFragment
  - Displays a floating dialog
  - Better that creating a normal dialog because the user can navigate back to the dismissed fragment
- ListFragment
  - Displays a list of items that are managed by an adapter
  - Similar to listActivity
- PreferenceFragment
  - Similar to PreferenceActivity