Android Apps with Java

CS 50, Fall 2009
Preliminaries

• Install Eclipse (http://www.eclipse.org/)
• Install ADT for Eclipse (http://developer.android.com/sdk/eclipse-adt.html)
• Create an Android Virtual Device (AVD)  
  – android create avd --target 2 --name my_avd
Android App Vocabulary

• Activities
  – The visual user interface(s) associated with an app
  – Runs in the foreground, displayed with a View

• Services
  – A portion of the app running in the background
  – No user interface

• Events
  – Things to which an application can respond
  – Includes taps, button/keyboard presses, etc.
Activities

• Presents and manages the user interface for one task
• Only one activity can be in the foreground at a time, and only FG activities can do work
• Activities can spawn other activities to handle different tasks (selecting a recipient vs. writing a message, for example)
• The “activity lifecycle” defines how activities behave as different processes come to the foreground
The Activity Lifecycle

1. **Activity starts**
   - `onCreate()`
2. **Activity is running**
   - **User navigates back to the activity**
     - `onStart()`
   - **Process is killed**
     - **Another activity comes in front of the activity**
       - `onPause()`
   - **Other applications need memory**
3. **Activity is no longer visible**
   - `onStop()`
4. **Activity is shut down**
   - `onDestroy()`

- **onRestart()**
  - **The activity comes to the foreground**
  - **The activity comes to the foreground**
  - **The activity comes to the foreground**
Views

- Hierarchical organizations of UI elements
- ViewGroups are collections of View objects that can organize them in different ways (linearly, in a grid, etc.)
- Activities define the view they want to use, then the view is drawn from the top down
Services

• Processes that do work in the background without the help of a UI
• Counterpart to activities
• Services can run independently of applications, or applications can “bind” to services to interact with them
• Services won’t be automatically killed by the OS the way activities are
Event Handling

• Android is an “event-driven” operating system
• Special messages called events are passed to the activity, which it can choose to ignore or handle specially
• Events have two parts
  – The event’s name (what kind of event it is)
  – Additional data associated with the event (if the event was a keyboard keypress, which key was it?)
Saving and Restoring State

- The onCreate method can have a “bundle,” specified by a unique string from the resources
- Bundles are map objects, which can store various types of data
- By using onCreate and onResume to check/modify the bundle, state can be saved for the next invocation when your activity is paused or killed
Debugging

• Eclipse has a rich debugging component as good or better than gdb
• If you get a force close, try running your program in the emulator with breakpoints set
• Other useful features:
  – Variable list
  – Watch expressions