Android - Overview

- What is different about Android?
- Application Components
- Inter Application Communication
- Activity Life Cycles
What is Android?
Application Components

- Activities
  - Visual user interface
  - Hierarchy of Views

- Services
  - Background processes (playing music, etc..)

- Broadcast Receivers
  - Low battery, time zone change, etc..

- Content Providers
  - Allows data sharing between applications
Activating Components

- **ContentProvider**
  - Activated when targeted by a ContentResolver

- **Intents**
  - Start: Activities, Services, BroadcastReceivers
  - Activities, services: names the action and the data
  - BroadcastReceivers: names the action being announced.
Example intent

Intent i = new Intent(android.provider.MediaStore.Audio.Media.RECORD_SOUND_ACTION);
i.putExtra(android.provider.MediaStore.EXTRA_OUTPUT, “/sdcard/odk/sounds”);
startActivityForResult(i, AUDIO_RECORDING);

protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
    super.onActivityResult(requestCode, resultCode, intent);

    switch (requestCode) {
        case AUDIO_RECORDING:
            ......
    }
}
Activities vs. Tasks

- **Activity** is a screen
- **Task** is a group of Activities
  - Not necessarily defined in the same Application.
  - Stack of activities. Activities can only be pushed and popped.
  - All activities in a task move as one, i.e. all go to background and or all to foreground at once.
Activity Lifecycle

1. Activity starts
2. onCreate()
3. onStart()
4. onResume()
5. onPause()
6. onStop()
7. onDestroy()

- User navigates back to the activity
- Process is killed
- Other applications need memory
- Activity is running
- Another activity comes in front of the activity
- The activity comes to the foreground
- The activity is no longer visible
Activities Lifecycle

- Screen rotation will completely kill and restart your program.

A new instance of your application is created
Activities – Saving State

- Primitives, parcelables, serialized objects
  - `onSaveInstanceState(Bundle outState)`
  - `onRestoreInstanceState()` or manually in `onCreate(Bundle savedInstanceState)`

- Objects
  - `onRetainNonConfigurationInstance()`
  - `getLastNonConfigurationInstance()`
Activities - Threads

- **UI thread**
  - Must be quick. Respond in less than 9 seconds.

- **Background Threads**
  - For long activities, downloading, etc..
  - Use AsyncTask
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <Button
        android:id="@+id/add_button"
        android:text="@string/add_file"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:padding="15px"
        android:textSize="8pt"
        android:layout_weight="1"/>

    <ListView
        android:id="@android:id/list"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_above="@id/upload_button"
        android:layout_alignParentTop="true" />
</RelativeLayout>
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent">
  
  <Button android:id="@+id/add_button"
android:text="@string/add_file"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:padding="15px"
android:textSize="8pt"
android:layout_weight="1"/>

  <ListView android:id="@android:id/list"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:layout_above="@id/upload_button"
android:layout_alignParentTop="true" />
</RelativeLayout>
```java
onCreate() {
    setContentView(R.layout.myLayout);
    // where myLayout is in {project}/res/layout/myLayout.xml

    Button b = (Button) findViewById(R.id.add_button);
b.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            // do something interesting;
        }
    });
}
```
onCreate() {
    setContentView(R.layout.myLayout);
    // where myLayout is in {project}/res/layout/myLayout.xml

    Button b = (Button) findViewById(R.id.add_button);
    b.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            // do something interesting;
        }
    });
}
Important things…

- Lots o’ java
  - Though, there is an NDK

- Intents
  - Applications can call other applications

- Activities
  - Can get garbage collected whenever not showing
  - Need to manage own state
  - Can run within another application
Tons more...