

# ANDROID APPS (NOW WITH JELLY BEANS!)

---

Jordan Jozwiak

November 11, 2012

# AGENDA

- Android v. iOS
- Design Paradigms
- Setup
- Application Framework
- Demo
- Libraries
- Distribution



# ANDROID V. IOS

## Android

- \$25 one-time developer account
- No restrictions for submitting apps to the Google Play Store
- Develop with Eclipse
- Java
- 68% worldwide market share
- 52% US market share
- Extreme customizability
- Open-source

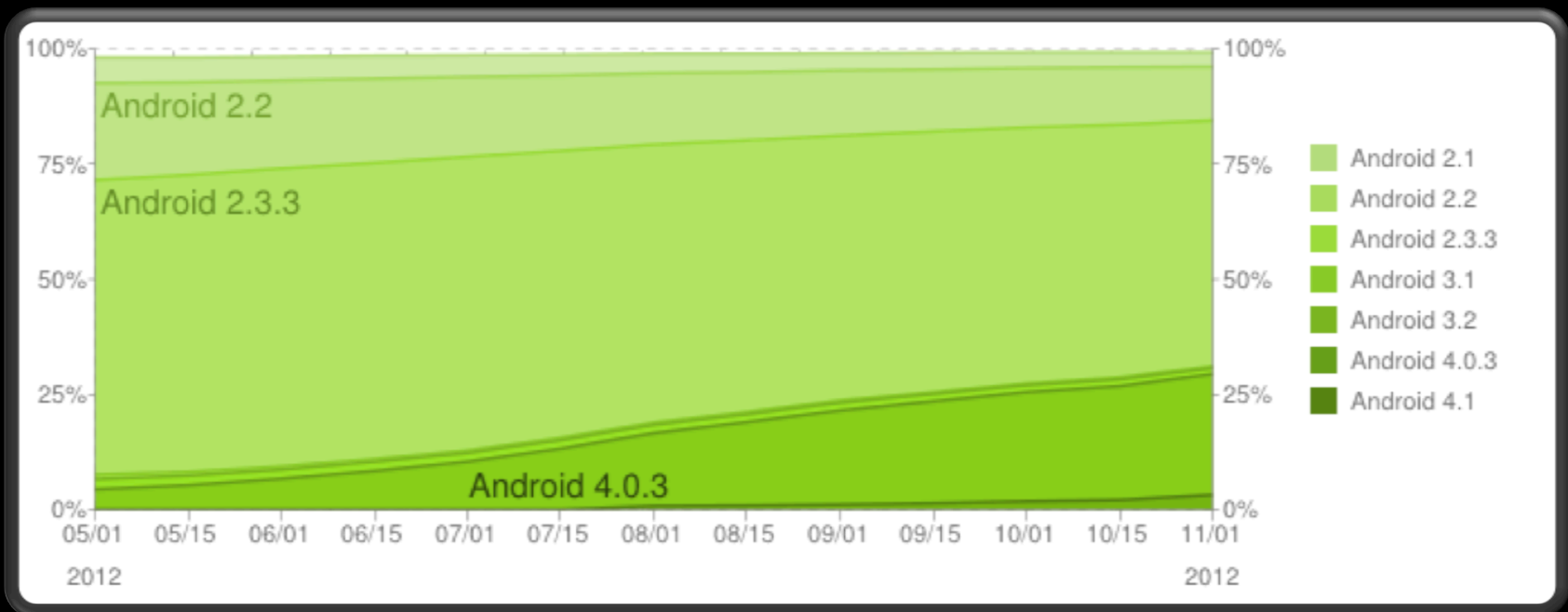
## iOS

- \$99/year developer program
- Apps and updates for App Store must go through a review process
- Develop with Xcode
- Objective-C
- 17% worldwide market share
- 34% US market share
- Customization limited by Apple
- Closed-source



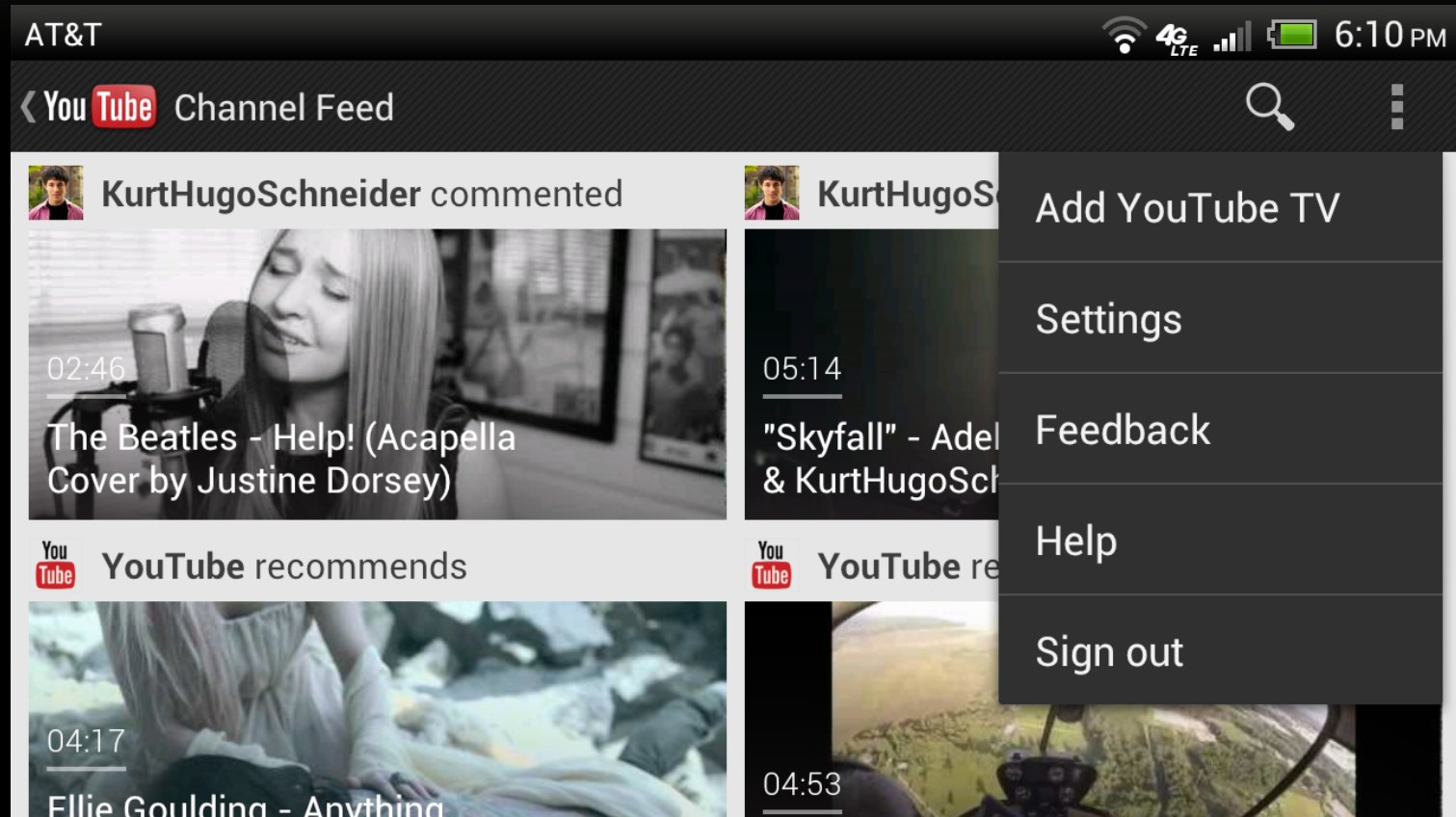
# DESIGN PARADIGMS

# DESIGN PARADIGMS FRAGMENTATION



# DESIGN PARADIGMS

## ACTIONBAR



# SETUP

- Follow the steps at <http://developer.android.com/sdk/installing/index.html>
  - Install the Android SDK (Software Development Kit)
  - Install “Eclipse Classic” version of Eclipse
  - Install the ADT (Android Development Tools) Plugin for Eclipse
- Building your first app
  - <http://developer.android.com/training/basics/firstapp/index.html>



# APPLICATION FRAMEWORK



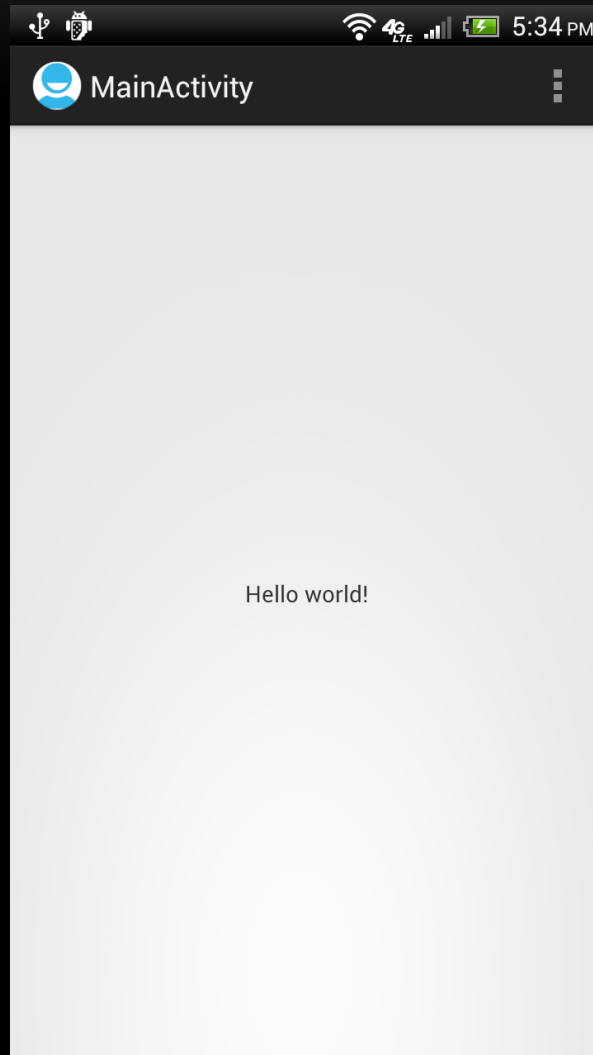
# APPLICATION FRAMEWORK

## SOME JARGON

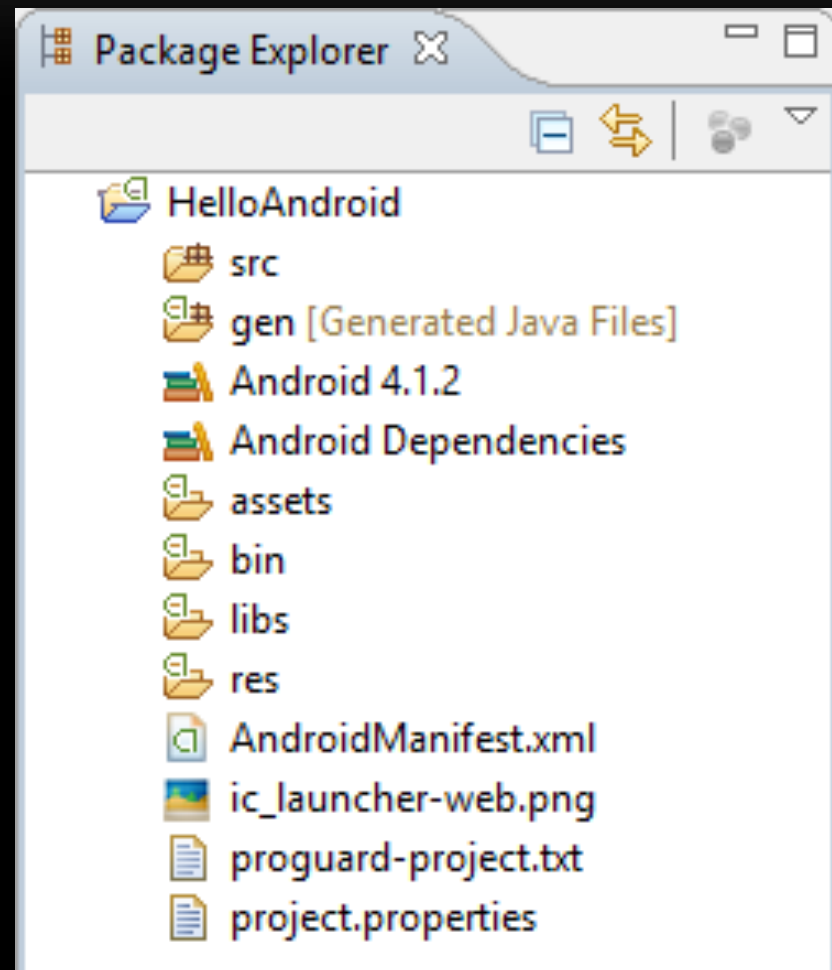
- Activity – visible screens controlled by a file in the source code that extends an Android Activity
- Fragments – represents a behavior or a portion of user interface in an activity
- Service – background processes such as playing music or uploading photos
- Content provider – shared data, such as contacts information
- Intent – helper function used to launch new activities

# APPLICATION FRAMEWORK

## HELLOANDROID



# APPLICATION FRAMEWORK ECLIPSE PROJECT



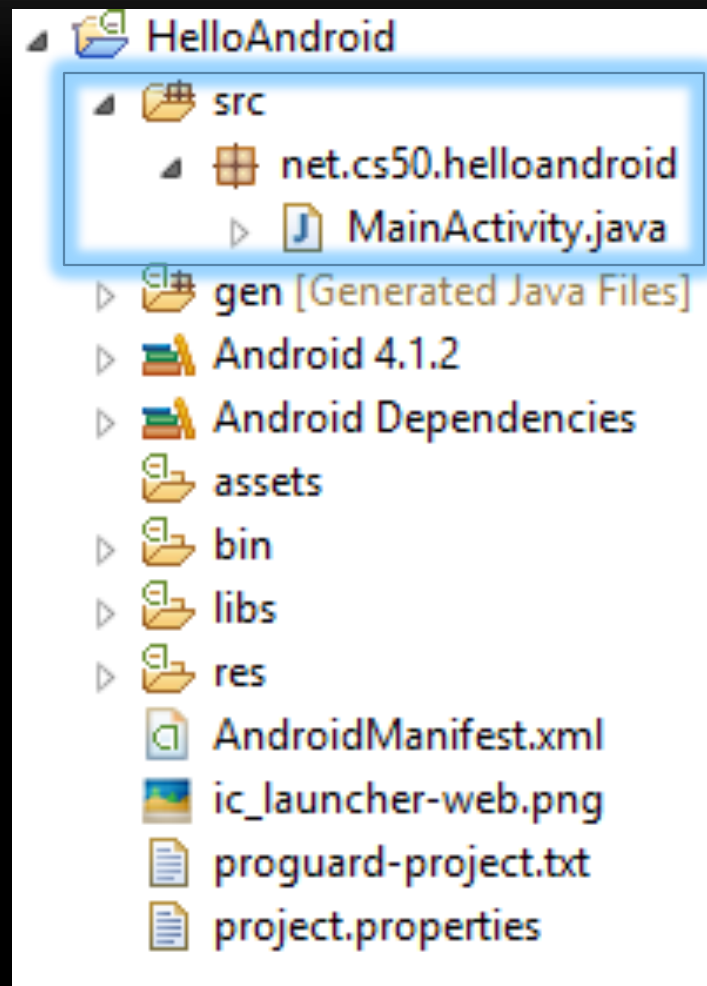
# APPLICATION FRAMEWORK

## FILE STRUCTURE

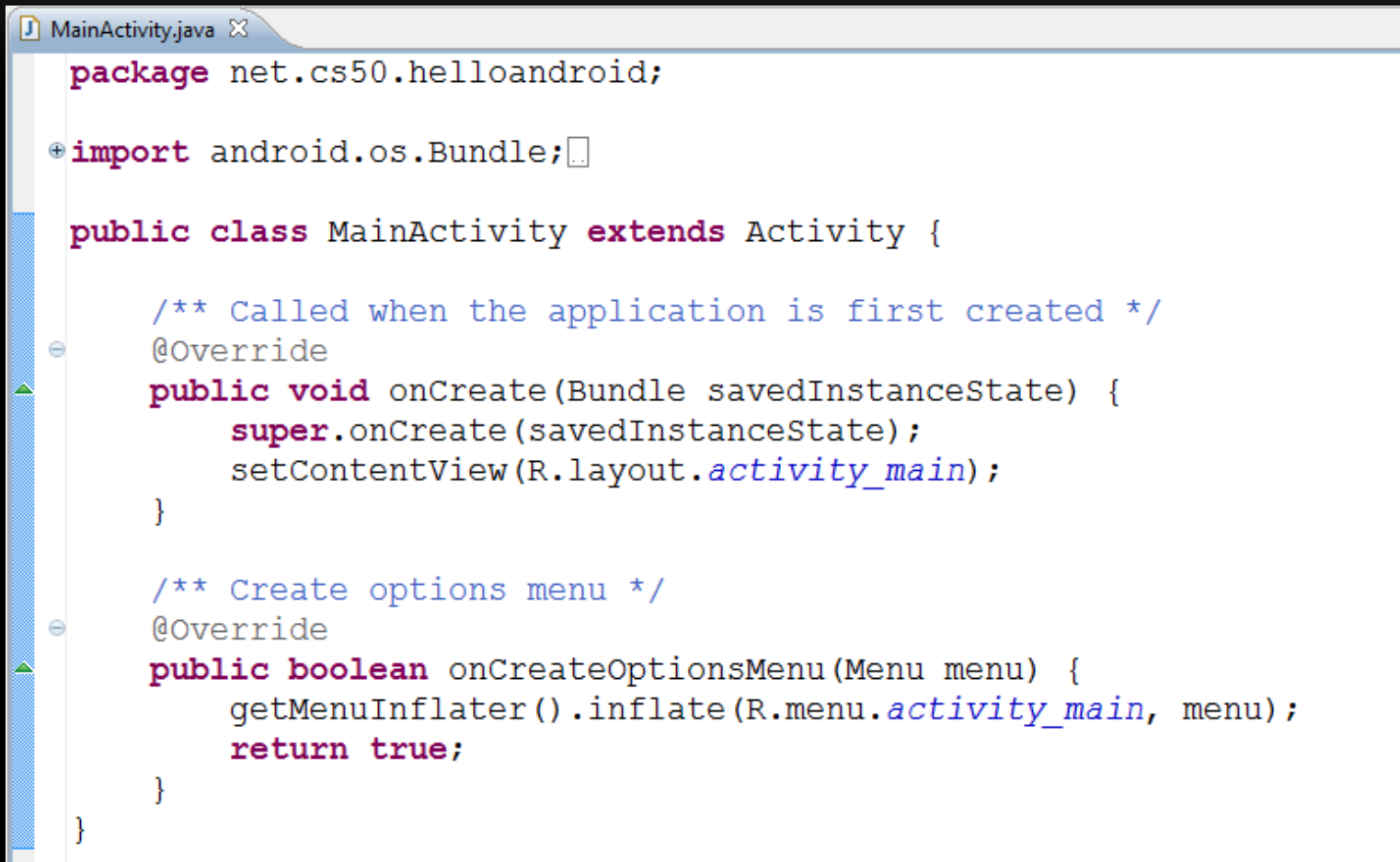
- src – java files where the logic of your program exists (similar to the PHP controllers in pset7)
  - res – collection of resources for screen layouts, images, sounds, text, animations and more
  - libs – any libraries you include in your project
  - Manifest – represents essential information about the application to the Android system
-

# APPLICATION FRAMEWORK

## SOURCE FOLDER



# APPLICATION FRAMEWORK SOURCE FILES

A screenshot of an IDE window titled 'MainActivity.java'. The code is written in Java and defines a 'MainActivity' class that extends 'Activity'. It includes two overridden methods: 'onCreate' and 'onCreateOptionsMenu'. The 'onCreate' method calls 'super.onCreate' and 'setContentView' with 'R.layout.activity\_main'. The 'onCreateOptionsMenu' method calls 'getMenuInflater().inflate' with 'R.menu.activity\_main' and returns 'true'.

```
package net.cs50.helloandroid;

import android.os.Bundle;

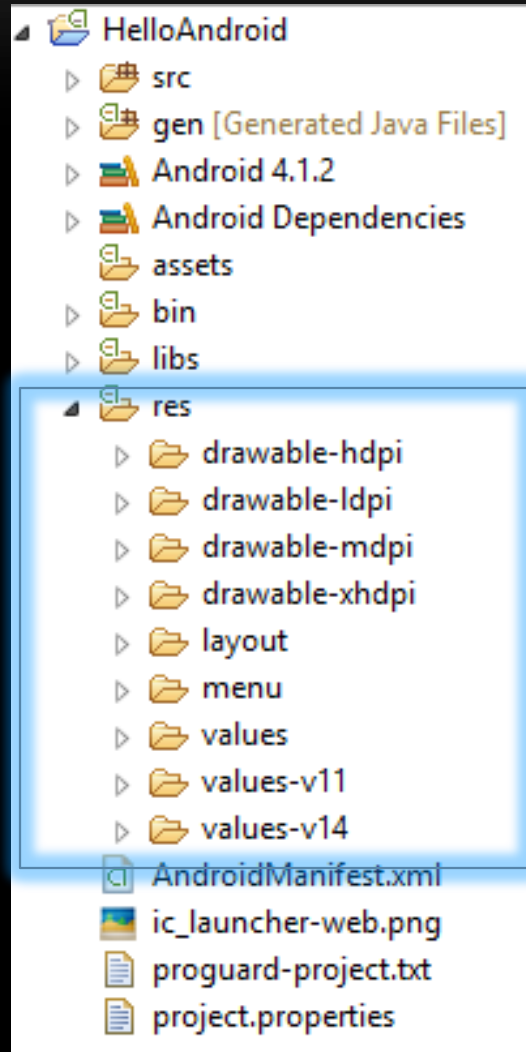
public class MainActivity extends Activity {

    /** Called when the application is first created */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    /** Create options menu */
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

# APPLICATION FRAMEWORK

## RESOURCE FOLDER



# APPLICATION FRAMEWORK

## RESOURCE FOLDER

- res/drawable – logo and all images
  - Customizable for different pixel densities
- res/raw – sounds and music
- res/layout – the XML layout for activities, dialogs, etc.
  - Customizable for different screen sizes and orientations
- res/values – saves all strings and styles
  - Customizable for different versions
  - Can be localized for different languages



# APPLICATION FRAMEWORK

## LAYOUT RESOURCE

```
activity_main.xml ✕  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" >  
  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_centerHorizontal="true"  
        android:layout_centerVertical="true"  
        android:text="@string/hello_world"  
        tools:context=".MainActivity" />  
  
</RelativeLayout>
```

# APPLICATION FRAMEWORK MANIFEST

- Presents essential information about the application to the Android system
  - Application package name (unique identifier for application)
  - Components: declares the activities, services, broadcast receivers and content providers used by the app
  - Permissions: these must be declared for the appropriate APIs to be used and they will also show up to users when downloading through the Google Play Store
  - SDK Version: allows developer to set minimum SDK version

# APPLICATION FRAMEWORK MANIFEST

```
HelloAndroid Manifest X
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.cs50.helloandroid"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/title_activity_main" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

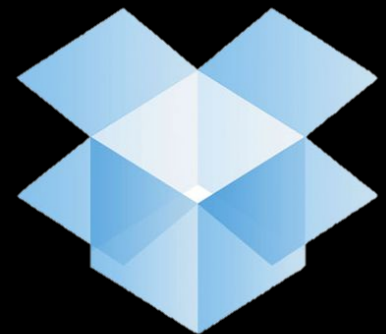
</manifest>
```

# DEMO

- In the main activity view
  - Add a text field; give it an id
  - Add a button; give it text, an id, and an onClick listener
- In the source code for the main activity
  - Create a method that handles the button click
  - Create an intent to launch a new activity
  - Create a new activity; add it to the manifest
  - Receive the intent in the new activity and display the message

# LIBRARIES

- Integrate cool features and save yourself some extra coding by including third party libraries



# APPLICATION DISTRIBUTION



<http://developer.android.com/distribute/googleplay/publish/index.html>

# WHERE TO GO FROM HERE

- Google Guides (for UI, best practices, etc.)
  - <http://developer.android.com/guide/components/index.html>
- Look at Google-provided sample projects in Eclipse
- Search for online tutorials
- Stackoverflow is your friend!

HAPPY CODING! :)

