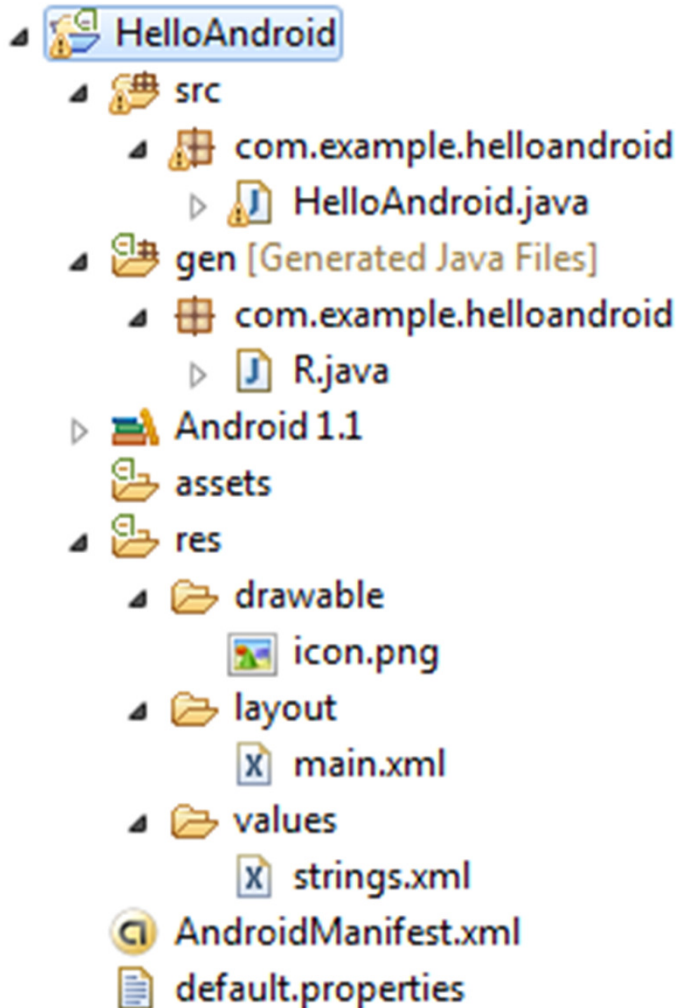


CS378 - Mobile Computing

Anatomy of an Android App and
the App Lifecycle

Hello Android Tutorial

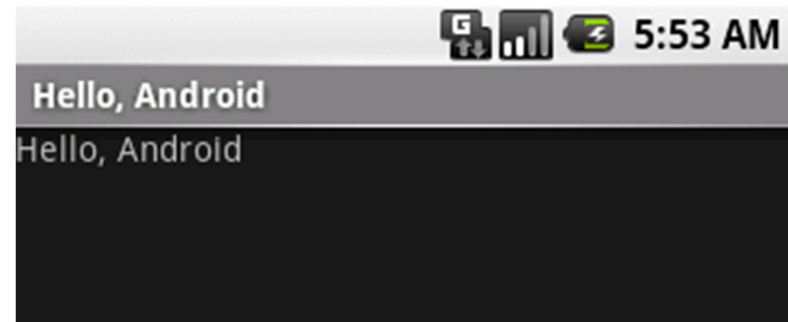


```
package com.example.helloandroid;

import android.app.Activity;
import android.os.Bundle;

public class HelloAndroid extends Activity {

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



<http://developer.android.com/resources/tutorials/hello-world.html>

Important Files

- **src/HelloAndroid.java**
 - Activity which is started when app executes
- **res/layout/main.xml**
 - Defines & lays out widgets for the activity
- **res/values/strings.xml**
 - String constants used by app
- **gen/R.java** (DO NOT MODIFY!)
 - Auto-generated, auto-updated file with identifiers from main.xml, strings.xml, and elsewhere
- **AndroidManifest.xml**
 - Declares all the app's components
 - Names libraries app needs to be linked against
 - Identifies permissions the app expects to be granted

src/HelloAndroid.java

- Activity which is started when app executes

```
package com.example.helloandroid;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class HelloAndroid extends Activity {

    private int mHelloCount = 0;

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

gen/R.java

- Auto-generated file with identifiers from main.xml, strings.xml, and elsewhere

```
public final class R {  
    public static final class attr {  
    }  
    public static final class drawable {  
        public static final int icon=0x7f020000;  
    }  
    public static final class id {  
        public static final int hello_button=0x7f050001;  
        public static final int my_button=0x7f050003;  
        public static final int my_check_box=0x7f050002;  
        public static final int name=0x7f050000;  
    }  
    public static final class layout {  
        public static final int main=0x7f030000;  
        public static final int second=0x7f030001;  
    }  
    public static final class string {  
        public static final int app_name=0x7f040001;  
        public static final int hello=0x7f040000;  
    }  
}
```

Do not
modify!

AndroidManifest.xml

- Declares all the app's components
- Names libraries app needs to be linked against
- Identifies permissions the app expects to be granted

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="scott.examples.hello2"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="10" /> ← min sdk version

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".Hello2Activity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

res/layout/main.xml

- layout of main activity
- xml view



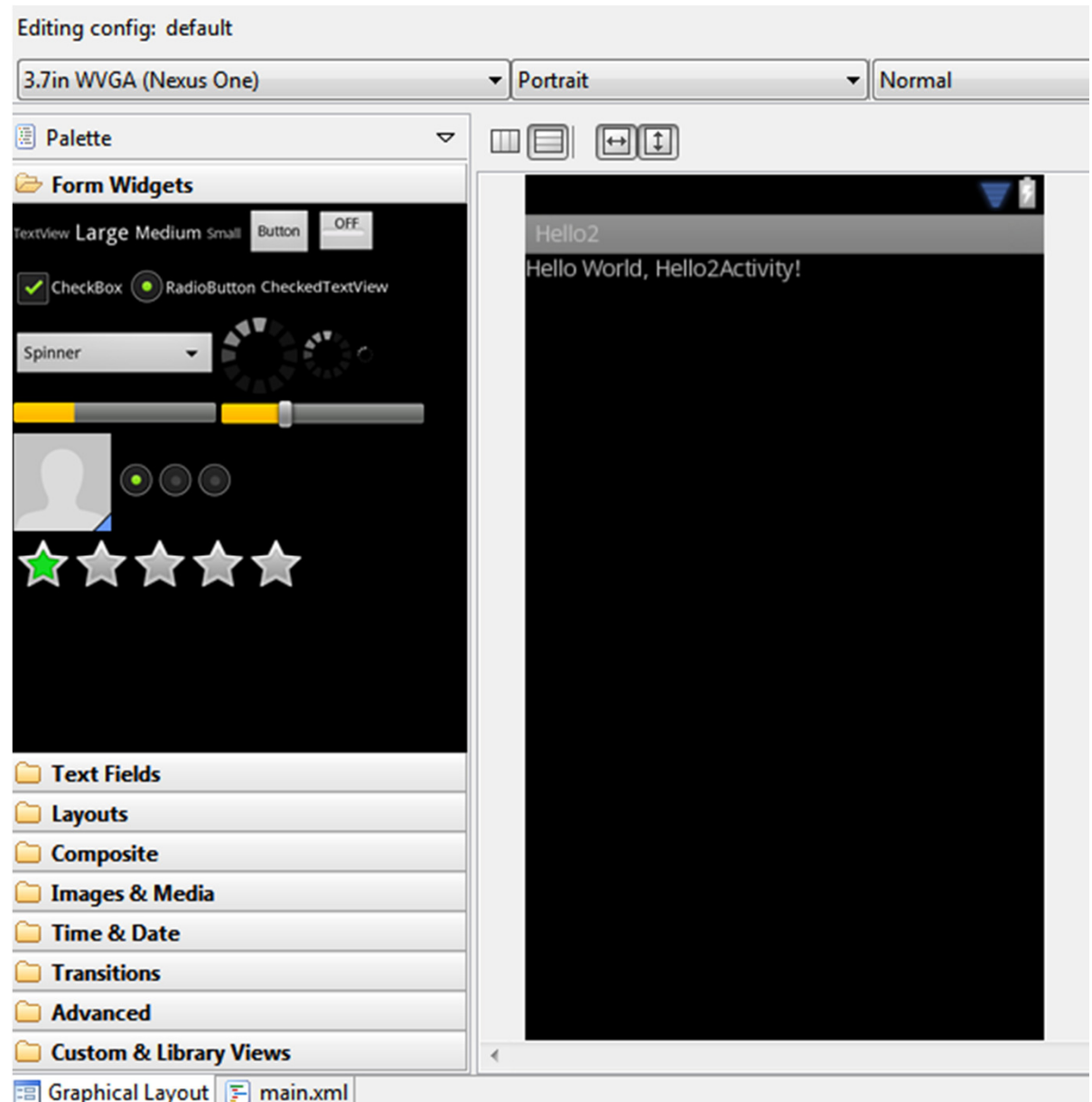
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

</LinearLayout>
```

res/layout/main.xml

- Drag and Drop UI Editor (your mileage may vary.)



res/layout/main.xml

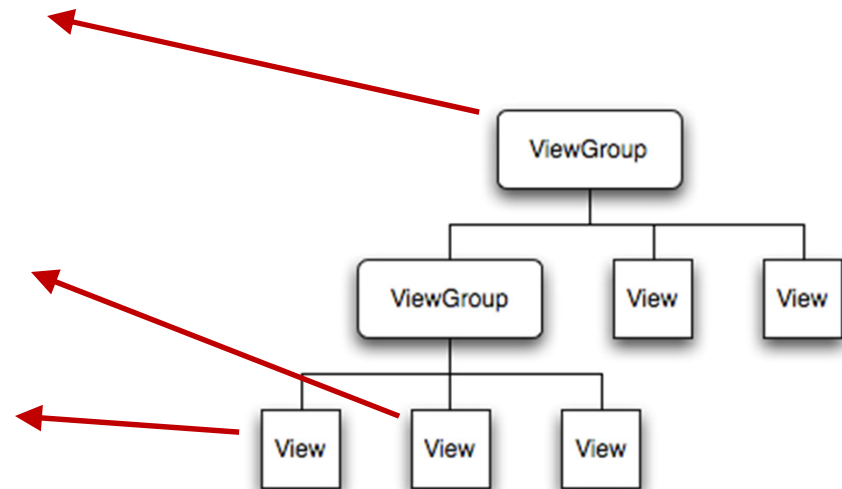
- Declares layouts & widgets for the activity

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <EditText
        android:id="@+id/name"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

    <Button
        android:id="@+id/hello_button"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="Press Me" />

</LinearLayout>
```

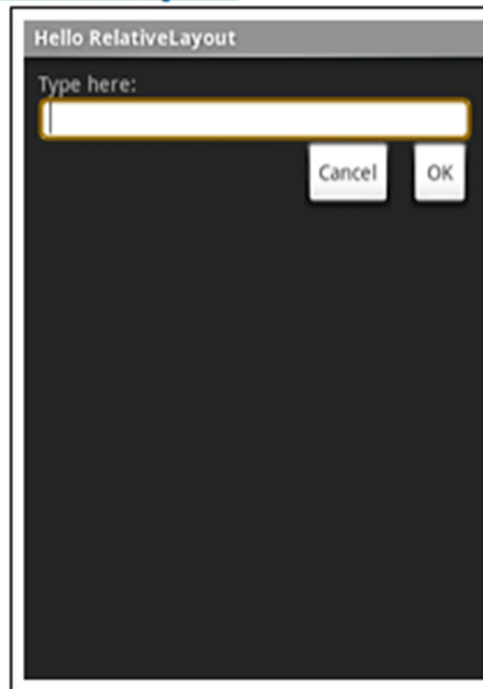


Available Layouts

LinearLayout



RelativeLayout

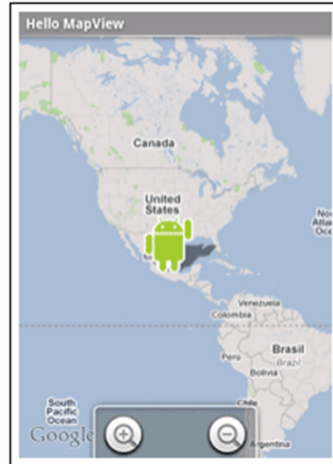


TableLayout

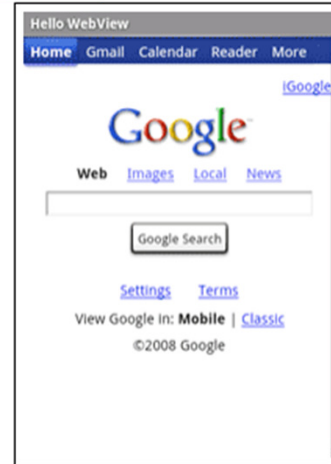


Available Widgets

MapView



WebView



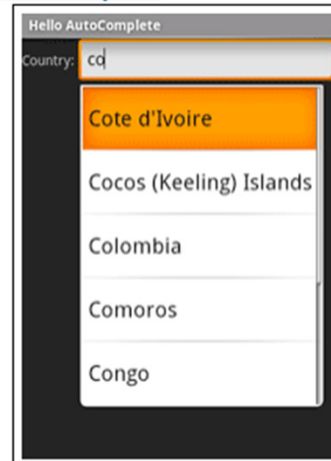
DatePicker



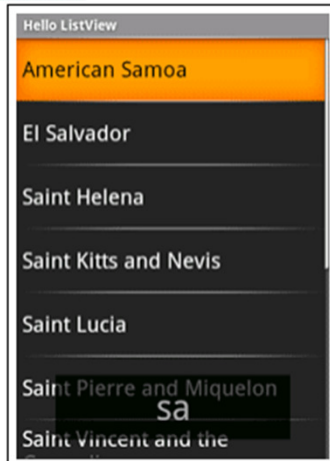
Spinner



AutoComplete



ListView



res/values/strings.xml

- String constants used by app

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Android</string>
    <string name="app_name">Hello Android</string>
</resources>
```

- Used for supporting Localization
 - res/values-es/values/strings.xml to support Spanish
 - res/values-fr/values/strings.xml to support French
 - Etc.

Four Application Components

1. Activities

- Presents a visual UI for a single endeavor
- Single app may be composed of several activities
- Examples: list of photos, buttons to start/stop a song

2. Services

- Performs background work (no UI)
- Examples: play background music, retrieve data over a network

3. Broadcast Receivers

- Receives and reacts to broadcast announcements (no UI)
- Broadcast examples: battery is low, pic is taken, lang pref changed

4. Content Providers

- Provides app data to other applications (no UI)
- Examples: share contact info from SQLite, image from the file system

Activity Stack

Most recently
created is at Top

Activity 1

User currently interacting with me

Activity 2

Pressing Back or destroying A1
will bring me to the top

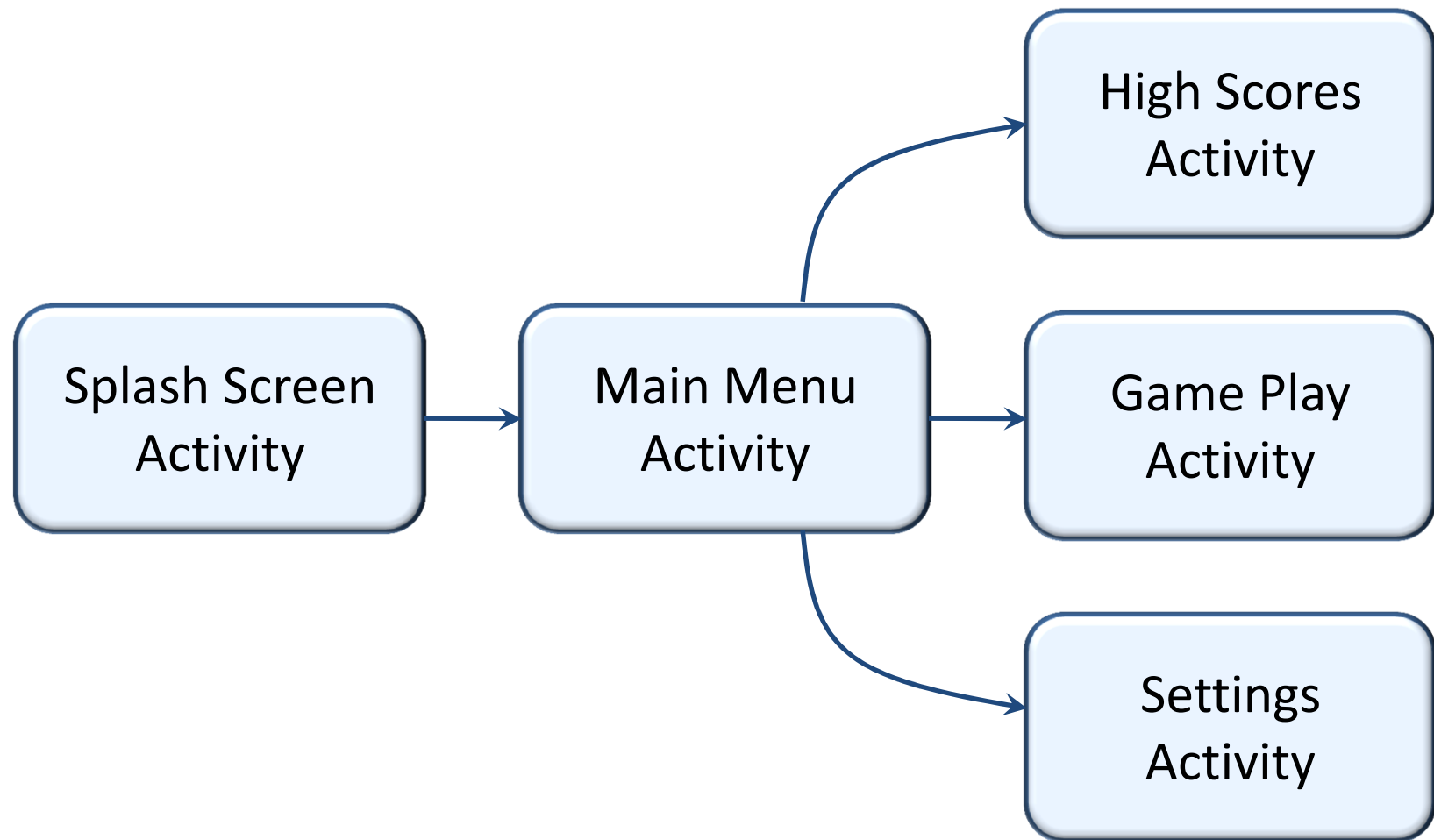
Activity 3

⋮

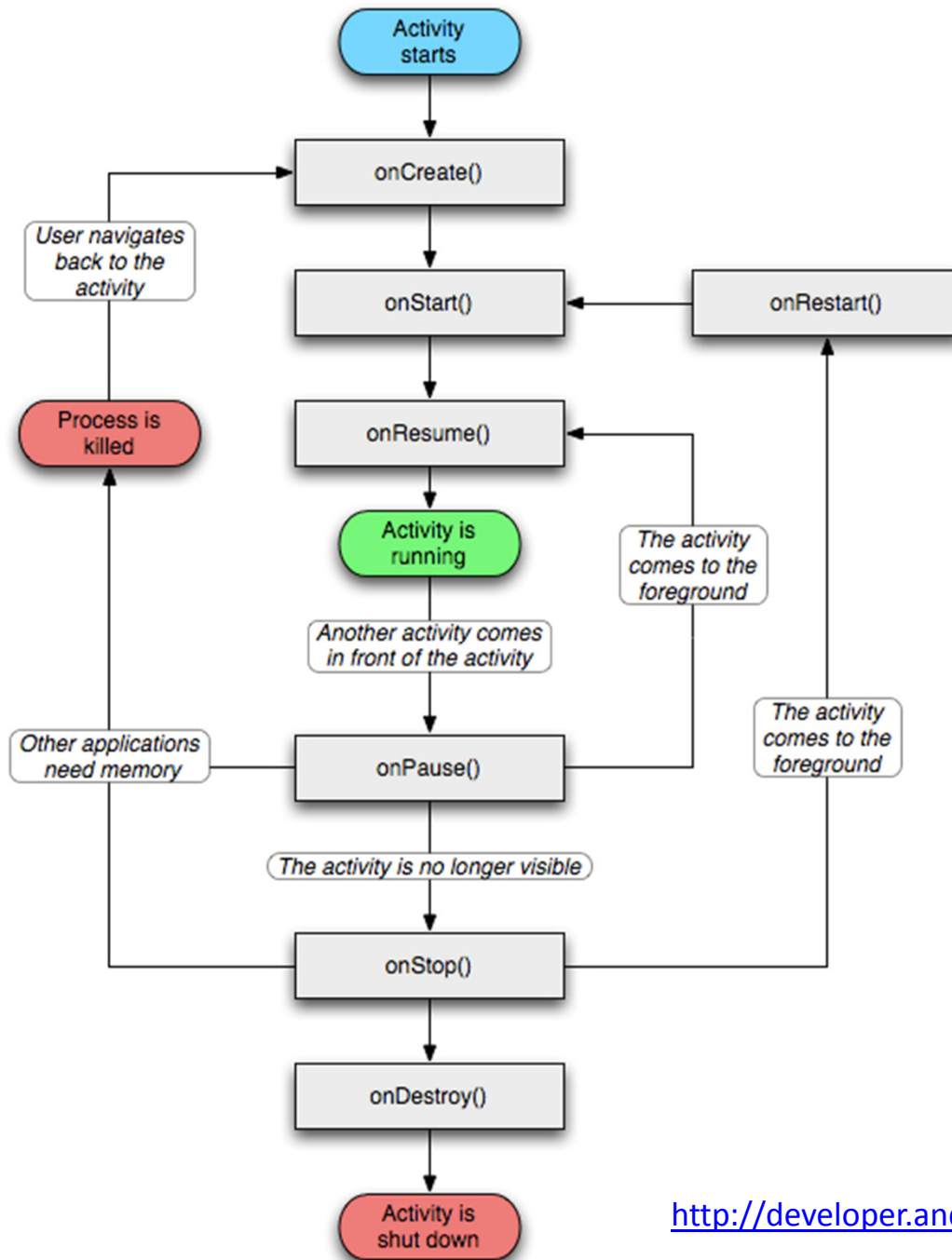
Activity N

If Activities above me use too
many resources, I'll be destroyed!

Typical Game



Activity Lifecycle



<http://developer.android.com/reference/android/app/Activity.html>

What is used for what?

- **Entire lifetime:** onCreate / onDestroy
 - Load UI
 - Could start and stop threads that should always be running
- **Visible lifetime:** onStart / onStop
 - Access or release resources that influence UI
- **Foreground lifetime:** onResume / onPause
 - Restore state and save state
 - Start and stop audio, video, animations

LifeCycleTest

- overload these methods from Activity:
 - onCreate(), onStart(), onResume(), onPause(), onStop(), onRestart, onDestroy()
 - Use the Log class to log activity
 - methods: v, d, i, w, e
 - VERBOSE, DEBUG, INFO, WARN, ERROR
 - Create a TAG so we can filter

LifeCycleTest

- Run the app and open the Logcat view.
 - Eclipse Window-> Show View -> Other -> Android -> Logcat or via DDMS

```
protected void onStart() {  
    super.onStart();  
    Log.d(TAG, "in onStart Method");  
}  
  
protected void onRestart() {  
    super.onRestart();  
    Log.d(TAG, "in onRestart Method");  
}  
  
protected void onResume() {  
    super.onResume();  
    Log.d(TAG, "in onResume Method");  
}  
  
protected void onPause() {  
    super.onPause();  
    Log.d(TAG, "in onPause Method");  
}  
  
protected void onStop() {  
    super.onStart();  
    Log.d(TAG, "in onStop Method");  
}  
  
protected void onDestroy() {  
    super.onDestroy();  
    Log.d(TAG, "in onDestroy Method");  
}
```

Logcat

- After app started

Time	Time	Time	Time	Time	Time
I	01-31 15:09:45.665	321		AndroidRuntime	NOTE: attach of threa
I	01-31 15:09:45.875	60	syste...	dalvikvm	Jit: resizing JitTabl
D	01-31 15:09:46.195	329	scott...	LIFECYCLE:	in onCreate Method
D	01-31 15:09:46.195	329	scott...	LIFECYCLE:	in onStart Method
D	01-31 15:09:46.195	329	scott...	LIFECYCLE:	in onResume Method
I	01-31 15:09:46.385	60	syste...	ActivityManager	Displayed scott.examp
I	01-31 15:09:46.385	60	syste...	ActivityManager	Displayed com.androic

Logcat

- Rotate emulator with CTRL+F-11

L...	Time	PID	Application	Tag	Text
D	01-31 15:09:46.195	329	scott...	LIFECYCLE:	in onCreate Method
D	01-31 15:09:46.195	329	scott...	LIFECYCLE:	in onStart Method
D	01-31 15:09:46.195	329	scott...	LIFECYCLE:	in onResume Method
I	01-31 15:09:46.385	60	syste...	ActivityManager	Displayed scott.examples.lifeCyc
I	01-31 15:09:46.385	60	syste...	ActivityManager	Displayed com.android.launcher/c
I	01-31 15:11:56.218	60	syste...	InputReader	Device reconfigured: id=0x0, nar
I	01-31 15:11:56.218	60	syste...	InputManager-Callbacks	No virtual keys found for device
I	01-31 15:11:56.625	60	syste...	ARMA assembler	generated scanline__00000177:03!
I	01-31 15:11:56.675	60	syste...	ARMA assembler	generated scanline__00000077:03!
I	01-31 15:11:56.745	60	syste...	ARMA assembler	generated scanline__00000177:03!
I	01-31 15:12:00.491	60	syste...	WindowManager	Setting rotation to 1, animFlags
I	01-31 15:12:00.495	60	syste...	ActivityManager	Config changed: { scale=1.0 ims:
D	01-31 15:12:00.535	329	scott...	LIFECYCLE:	in onPause Method
D	01-31 15:12:00.535	329	scott...	LIFECYCLE:	in onStop Method
D	01-31 15:12:00.535	329	scott...	LIFECYCLE:	in onDestroy Method
D	01-31 15:12:00.565	329	scott...	LIFECYCLE:	in onCreate Method
D	01-31 15:12:00.565	329	scott...	LIFECYCLE:	in onStart Method
D	01-31 15:12:00.565	329	scott...	LIFECYCLE:	in onResume Method
D	01-31 15:12:02.844	60	syste...	dalvikvm	GC_EXPLICIT freed 341K, 47% free

How to Stop an Activity?

- Generally, don't worry about it!
- "**Note:** In most cases, you should not explicitly finish an activity using these methods. As discussed in the following section about the activity lifecycle, the Android system manages the life of an activity for you, so you do not need to finish your own activities. Calling these methods could adversely affect the expected user experience and should only be used when you absolutely do not want the user to return to this instance of the activity."
- methods: `finish()`, `finishActivity()`

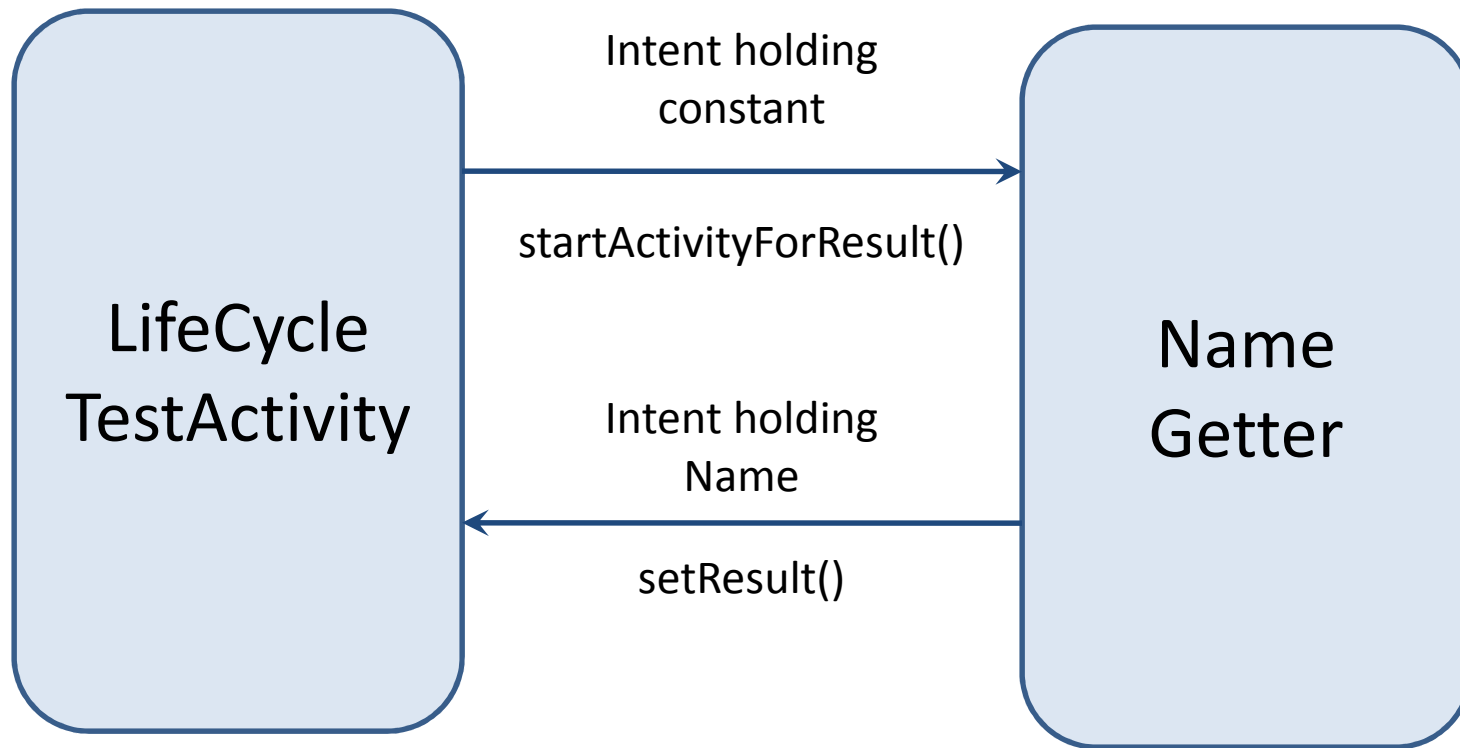
Saving State

- activities that are paused or stopped the state (instance vars) of the activity are retained
 - even if not in foreground
- When activity destroyed the Activity object is destroyed
 - can save information via onSaveInstanceState method. Write data to Bundle, Bundle given back when restarted

Starting You Own Activities

- You will often start new Activities within your Activity
 - accomplish a task
 - get some data
- Click Button to get name
 - on button click (look at xml)
 - create an intent
 - call startActivityForResult
 - override onActivityResult()
 - add new Activity to Manifest
 - add data to intent, setResult, finish

Intent Demo



References

- Android Introduction by Marko Gargenta,
<http://www.lecturemaker.com/2009/10/android-software-platform/>
- Android Dev Guide
<http://developer.android.com/guide/topics/fundamentals.html>
<http://developer.android.com/guide/topics/fundamentals/activities.html>
- *Pro Android* by Hashimi & Komatineni (2009)
- Frank McCown, Harding University