

Android 101 - Developer Workshop



6th October, 2012

Thank You for not using iOS 6 Maps

2



Goals of Workshop

3

- Help you understand what Android is
- Try out some Android APIs via Android Eclipse Projects
- Build a simple Application (Quotes)
- Build enough enthusiasm for you to take your next leap into Android Programming

A problem has been detected and windows has been shut down to prevent damage to your computer.

DRIVER_IRQL_NOT_LESS_OR_EQUAL

If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical information:

*** STOP: 0x00000001 (0x0000000C,0x00000002,0x00000000,0xF86B5A89)

*** gv3.sys - Address F86B5A89 base at F86B5000, DateStamp 3dd991eb

Beginning dump of physical memory
Physical memory dump complete.

Contact your system administrator or technical support group for further assistance.

Participant Requirements

5

- Basic working knowledge of Java or any other OOP language
- Eclipse IDE
- Android SDK and Tools
- We shall run all examples on the Android Emulator
- A device is important to have but not necessary for this course



What is Android ?



What is Android?

7

- A software stack for mobile devices
- Includes Linux Kernel, OS Middleware, Application Framework & Applications.
- Provides SDK for developers
- From Google
- Open Source and provided to Handset manufacturers
- <http://www.android.com>
- <http://developer.android.com>

OS Version	Name	API Level	Date
Beta, 1.0 , 1.1	-	-, 1 , 2	Nov 2007, Sep 2008, Feb 2009
1.5	Cupcake	3	April 2009
1.6	Donut	4	September 2009
2.1	Eclair	7	October 2009
2.2	Froyo	8	May 2010
2.3	Gingerbread	10	December 2010
3.0	Honeycomb	11	February 2011
3.1.x, 3.2	Honeycomb	12,13	
4.0, 4.0.1, 4.0.2	Icecream Sandwich	14	October 2011
4.0.3, 4.0.4	Icecream Sandwich	15	December 2011
4.1	JellyBean	16	June 2012

Android Device Dashboard

9

Version	Codename	API	Distribution
1.5	Cupcake	3	0.1%
1.6	Donut	4	0.4%
2.1	Eclair	7	3.4%
2.2	Froyo	8	12.9%
2.3 - 2.3.2	Gingerbread	9	0.3%
2.3.3 - 2.3.7		10	55.5%
3.1	Honeycomb	12	0.4%
3.2		13	1.5%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	23.7%
4.1	Jelly Bean	16	1.8%

~ 72%

~ 25%

October 1, 2012 : <http://developer.android.com/resources/dashboard/platform-versions.html>

Android Features

10

- Handset Layouts, Multiple Languages
- Storage
- Connectivity
- Messaging
- Media Support
- Hardware support
 - ▣ Cameras, touch screens, GPS, accelerometer, etc.
- Reuse of Data
- Reuse of Functionality

Why Android?

11

- ❑ Android has garnered more than 50% of the current market share in Smartphone
- ❑ Android by its nature is available across various devices (Phone, Tablet, TV, Navigation Devices, etc)
- ❑ Robust Ecosystem : 300+ Hardware/Software partners
- ❑ Open Source (<http://git.android.com>)
- ❑ Good set of Developer Tools
- ❑ 25 Billion+ Downloads



Android Developer Tools

12

- Java SDK
- Eclipse
- Android SDK
 - ▣ Android Emulator
 - ▣ Command Line Tools
 - ▣ Documentation + Samples
 - ▣ Platform Libraries
- Android Development Tools (ADT) for Eclipse
(Recommended to begin with)



Development Environment Setup



Tools Needed

14

- Java SDK

- ▣ <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

- Eclipse

- ▣ <http://www.eclipse.org/downloads>

- Android SDK

- ▣ <http://developer.android.com/sdk/index.html>

- Android Development Tools (ADT) for Eclipse

- ▣ <http://developer.android.com/sdk/eclipse-adb.html>

(Recommended to begin with)

Android Virtual Device

15

- ❑ SDK includes an Emulator - a virtual mobile device
- ❑ Emulator helps you test your applications without a physical device
- ❑ It mirrors the System Image and functionality as closely as possible
- ❑ You need to create an Android Virtual Device for the Android OS that you wish to target. E.g. 2.2, 2.3, 3.x, 4.x

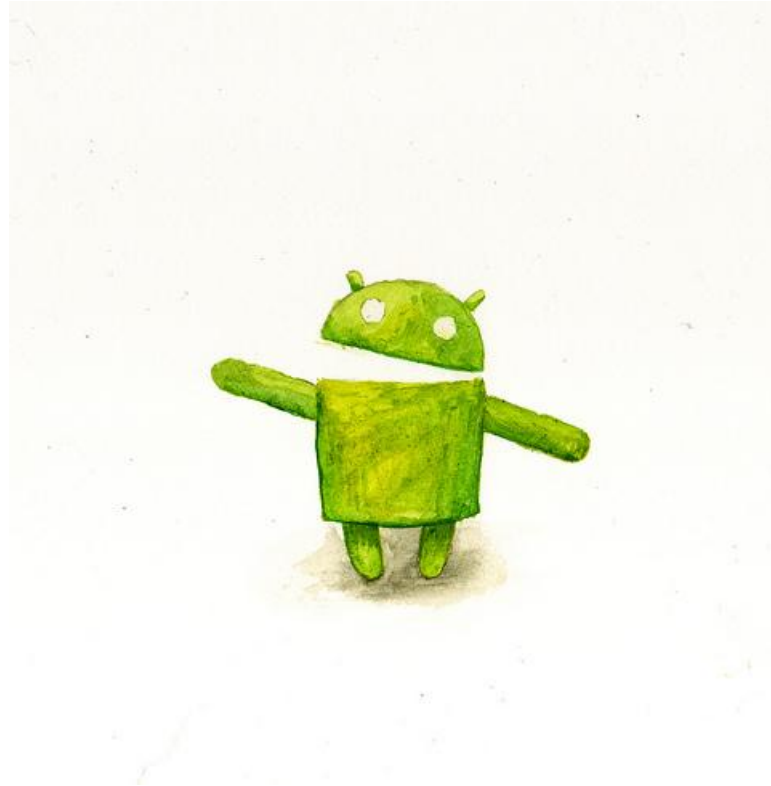
Android Emulator

16

- Mimics a Normal Android Device
- Use it to get familiar with the functions

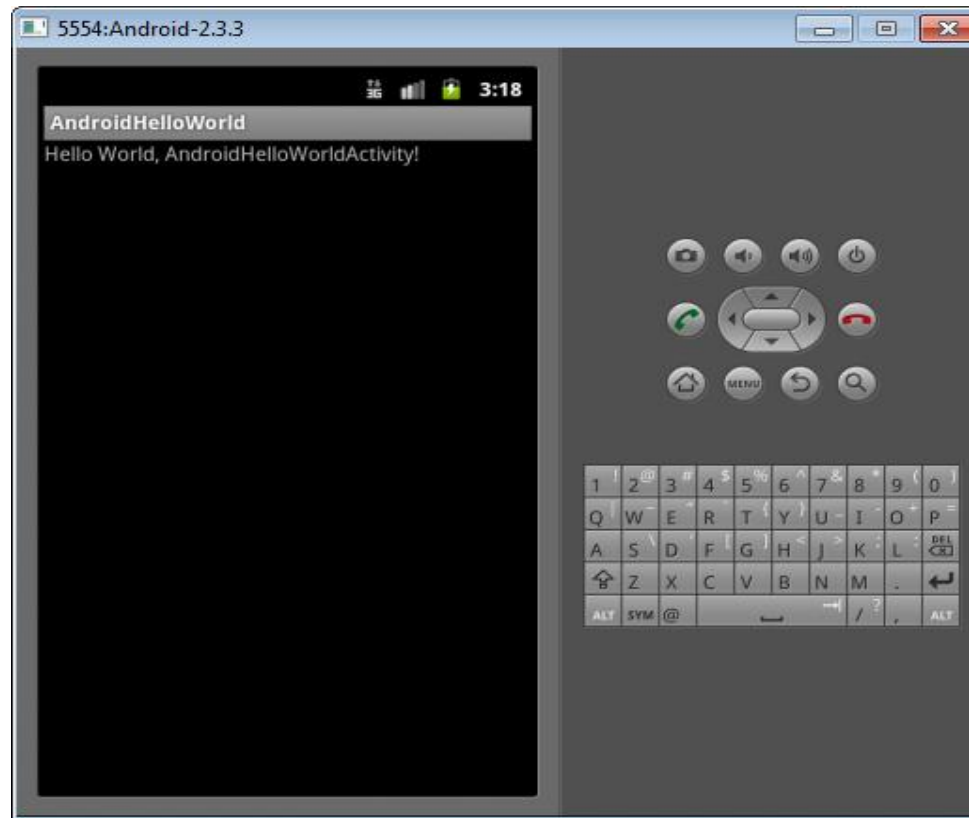


Your First Android App



Hello World In Android

18



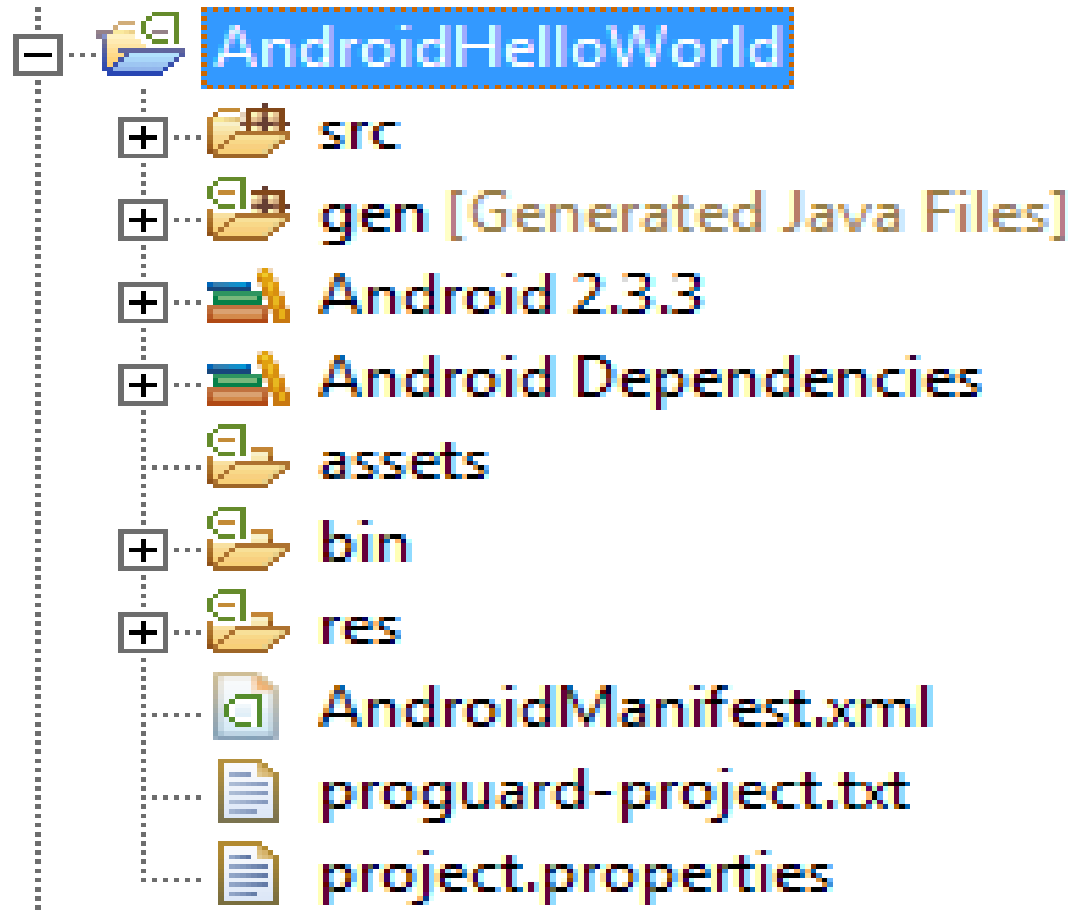
Steps to First Hello World

19

- Development Environment !
- Create the AVD and start the Device
- Use Eclipse to create the Android Project
 - ▣ Code, Code, Code
- Run the Application either the AVD or a real device

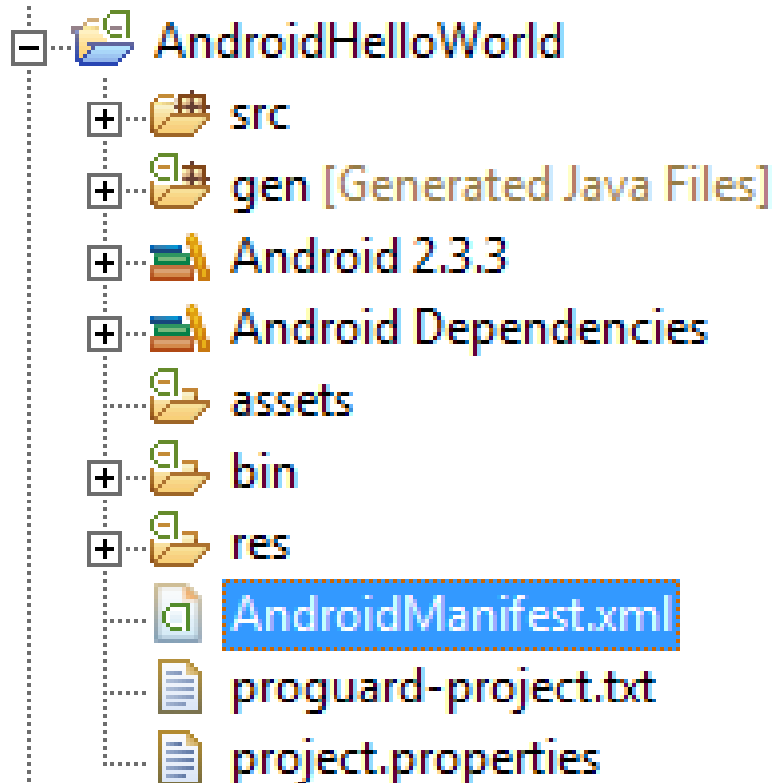
Project Structure

20



Android Manifest XML

21



- Describes the Application
- Activities and other components
- Application Name
- Application Version
- Permissions
- Android SDK Version

Android Manifest XML

22

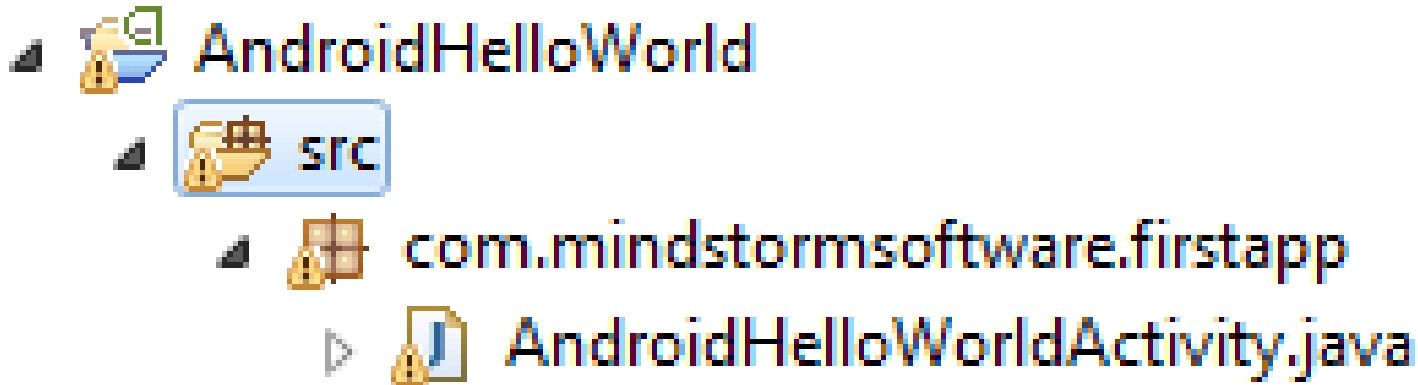
AndroidHelloWorld Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.mindstormsoftware.firstapp"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".AndroidHelloWorldActivity"
            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>
```

Project Structure - Source

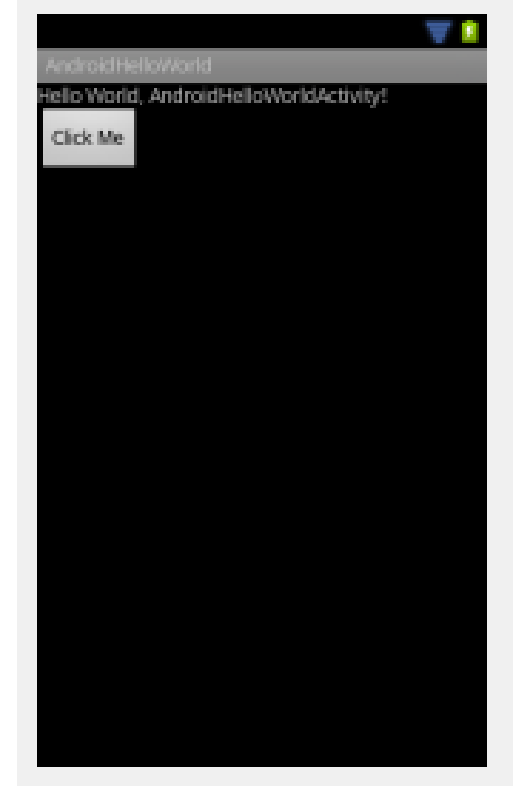
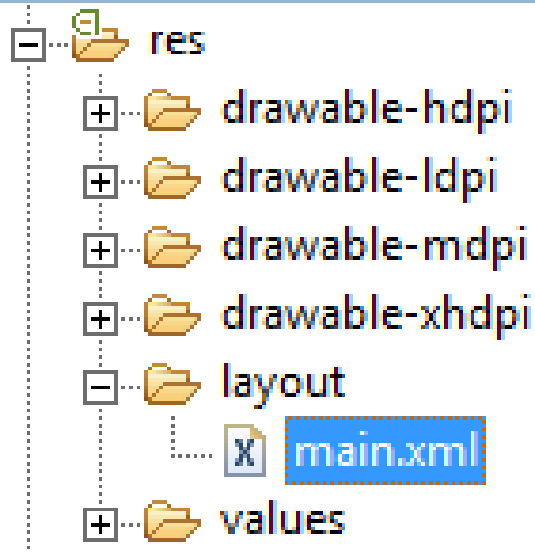
23



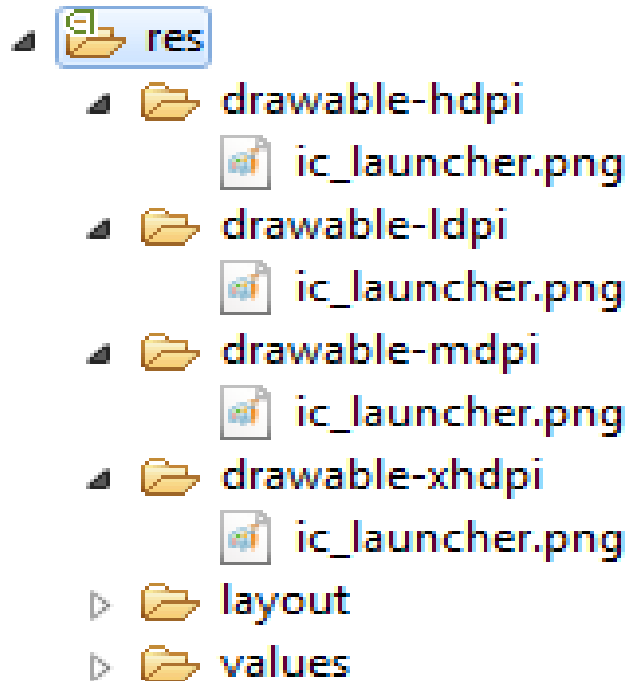
- Contains all the Java Source files

Project Structure - Layout

24



- Defines how the User Interface is laid out.

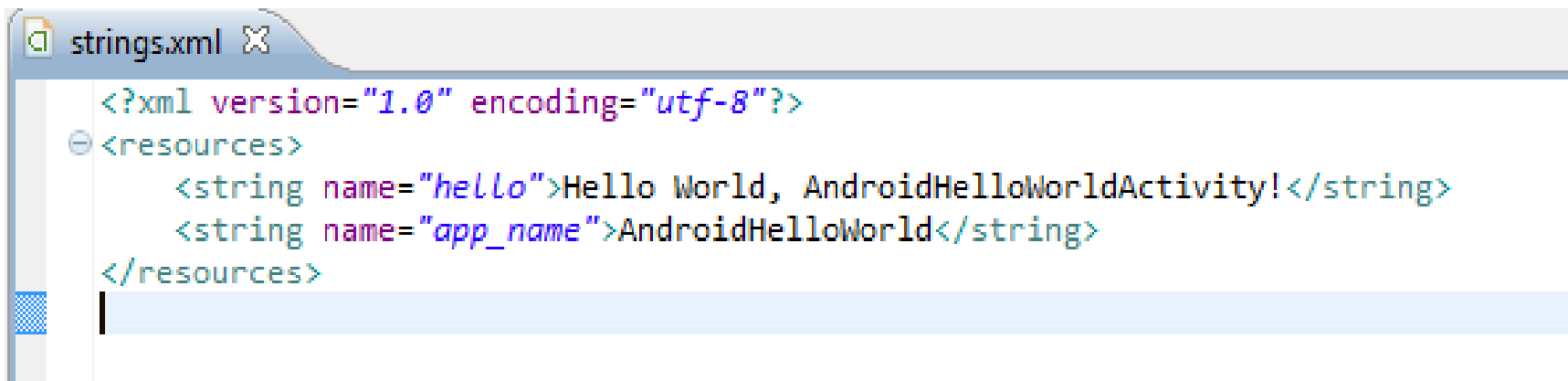


- Define Image Resources for different resolutions
- The runtime takes care of using the right image
- If not present, it scales it accordingly.

Project Structure – strings.xml

26

- ❑ Externalize all String values
- ❑ strings.xml is just one kind of resource you can put
- ❑ Other examples are:
 - ❑ colors.xml
 - ❑ dimens.xml
 - ❑ arrays.xml



```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, AndroidHelloWorldActivity!</string>
    <string name="app_name">AndroidHelloWorld</string>
</resources>
```


Project Structure – R.java file

27

```
R.java X
/* AUTO-GENERATED FILE. DO NOT MODIFY.

package com.mindstorm.helloworldapp;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int hello=0x7f040000;
    }
}
```

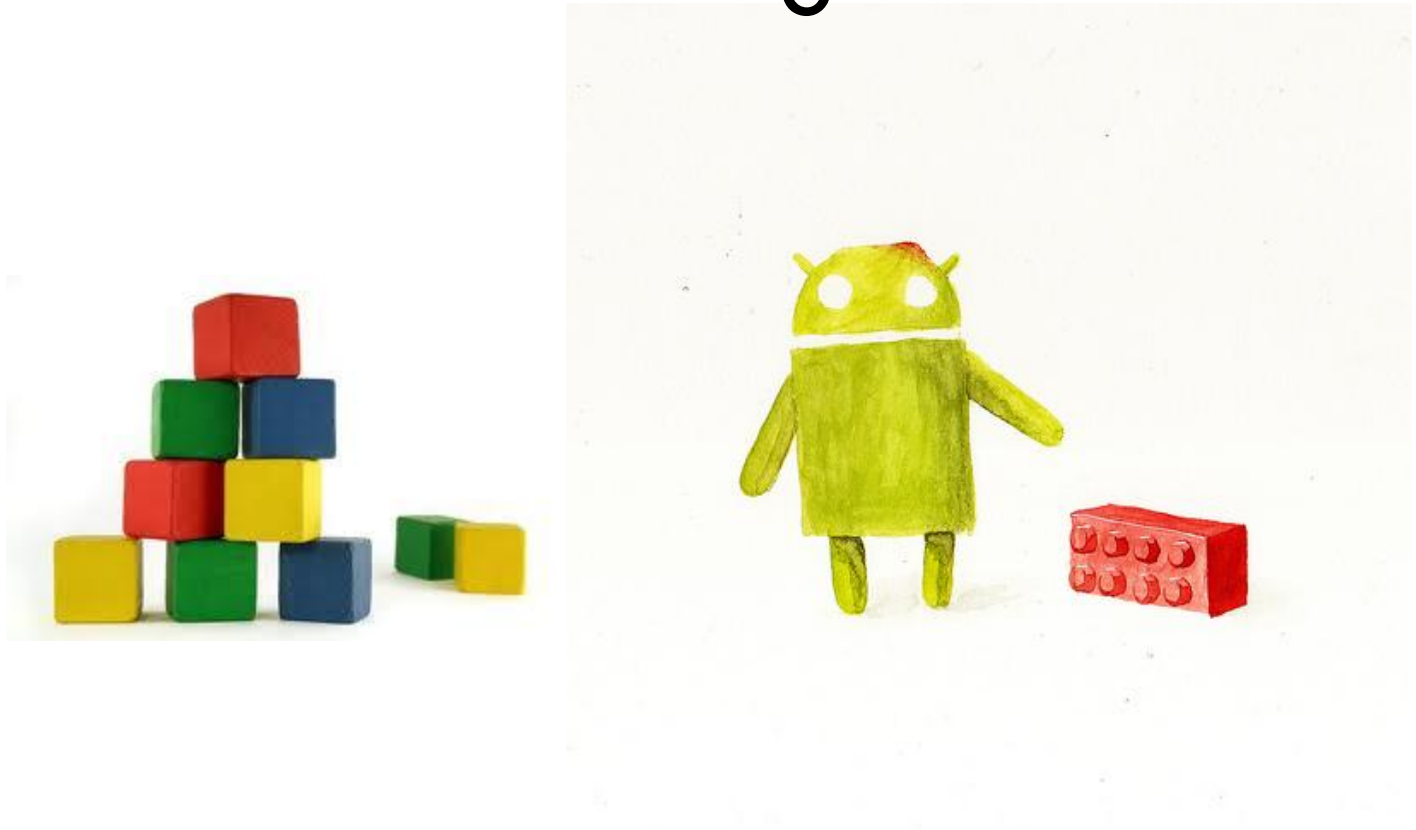
Hands On Exercise

28

- Hands On Exercise : **ex01 – Hello World.doc**
 - ▣ Hello World
- Check List
 - ▣ Developer Tools are setup
 - ▣ Android Virtual Device has been created



Android Building Blocks



Building Blocks

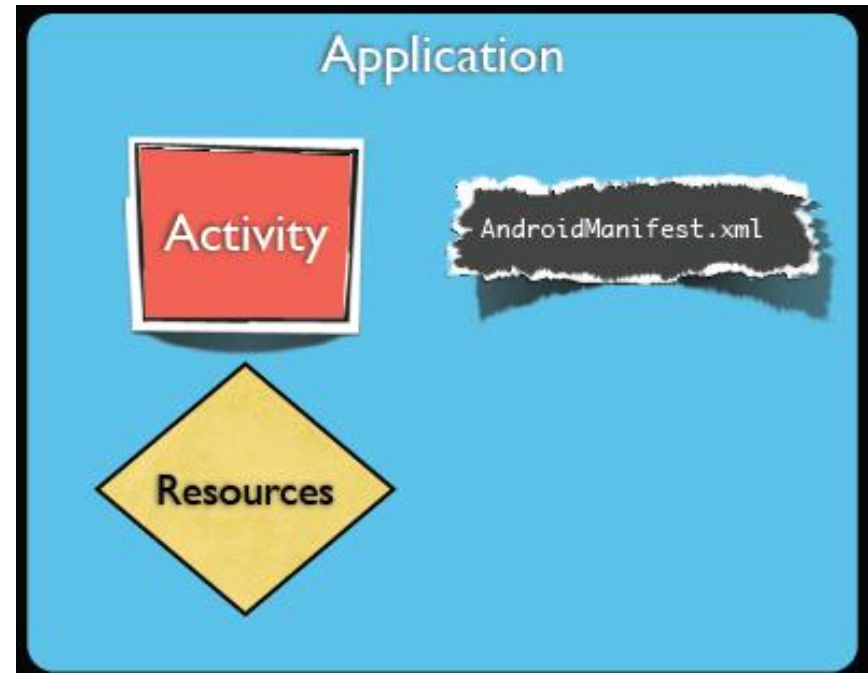
30

- Important to understand what an Android Application is made up of
- Android defines few buildings blocks that make up an application
 - ▣ Activity
 - ▣ Manifest File
 - ▣ Resources
 - ▣ Content Provider
 - ▣ Broadcast Receiver
 - ▣ Service

Android Application - Minimal

31

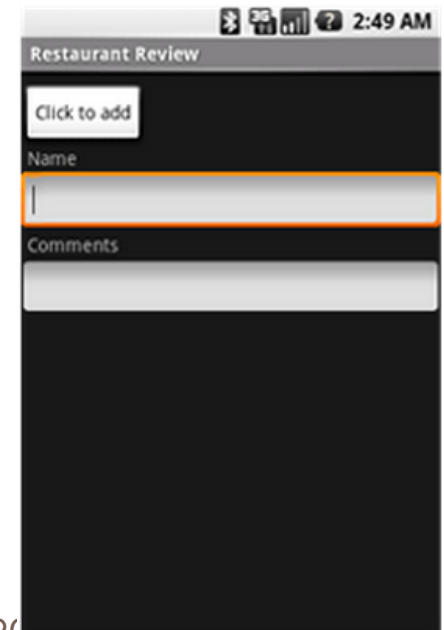
- You need 3 things at least
 - ▣ Activity
 - ▣ Android Manifest XML
 - ▣ Resources



Android Activity

32

- ❑ Most Basic Building Block
- ❑ It is a Screen that users can interact with
- ❑ Example: Filling out a text field, dialing a number, etc.
- ❑ An Activity is typically written by extending the **Activity** class.
- ❑ You can implement the call back methods to get notified of Activity life cycle (`onCreate()`, `onDestroy()`, etc)



Android Activity

33

- Creating an Activity
 - ▣ Extend the Activity class
 - ▣ Implement the following method
 - onCreate() : In this method, you can specify the layout (UI) for the activity and write any other initialization functions

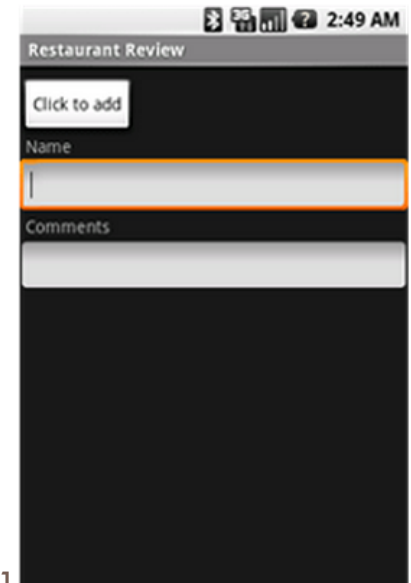
Android Activity

34

- An Application can have one or more Activities
- One Activity is designated as the Main Activity that is launched when the Application Starts

```
<application android:icon="@drawable/icon" android:label="@string/app_name">
    <activity android:name=".AndroidHelloWorldActivity"
        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>
```

- ❑ An Activity can implement its User Interface by defining a XML file.
- ❑ The User Interface will consist of ViewGroups (Layout) and Views.
- ❑ You can set the View by calling the setContentView() method in the onCreate() method of the Activity
- ❑ Pass the layout in the onCreate()



Android UI

36

- Android supports
 - ▣ Description of UI in XML
 - ▣ Programming UI creation
- Design Principle : Separate the UI Design from Behaviour (Event Handling e.g. Click)
- Views
- ViewGroups

Android UI - Views

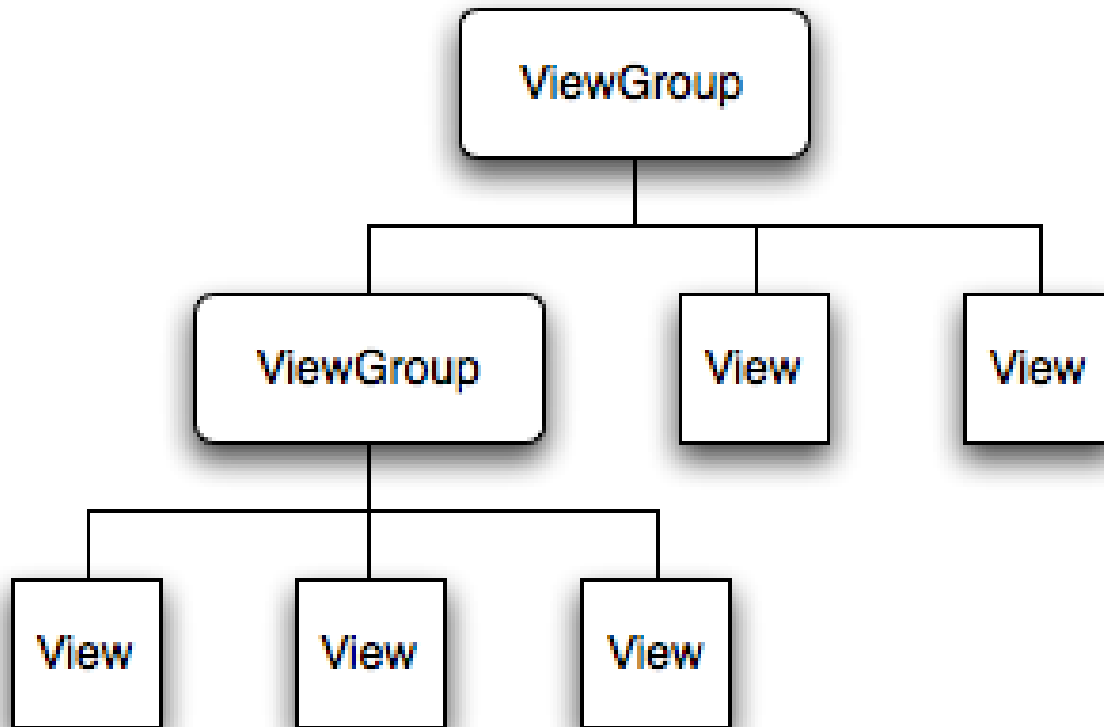
37

- View
- Basic Unit of UI
- Base class for all widgets
- Examples : TextField, Button, etc.

Android UI - ViewGroup

38

- ViewGroup serves as the base class for “Layouts”
- Layout
 - ▣ Allows to group Views
 - ▣ Lay them out in various architectural ways
 - Tabular
 - Linear
 - Relative
- Layouts can be nested to create complex screens



Android UI – View Types

40

- Android supports variety of views that help capture various kinds of user input
- What kind of data + User Experience

- Button
- EditText
- TextView
- ImageButton

- CheckBox
- RadioButton, RadioGroup
- ToggleButton
- SeekBar
- ProgressBar

Android UI – Basic Views

41

Date Picker



Time Picker



Form Stuff



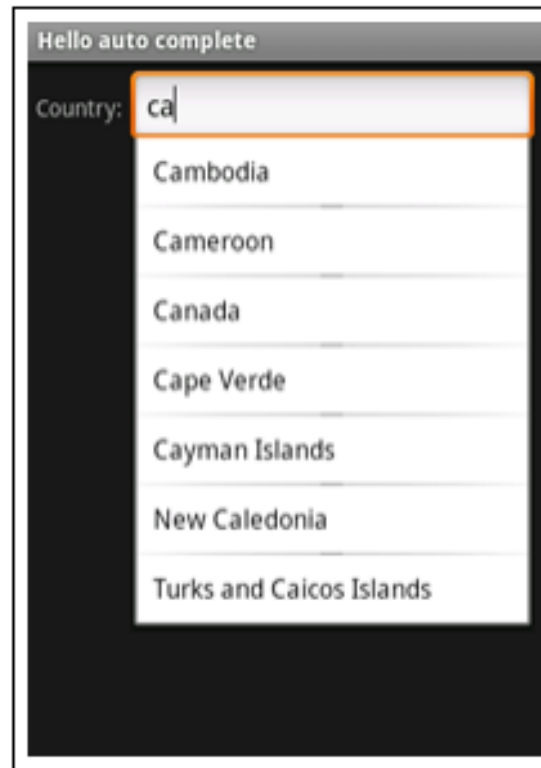
Android UI – Basic Views

42

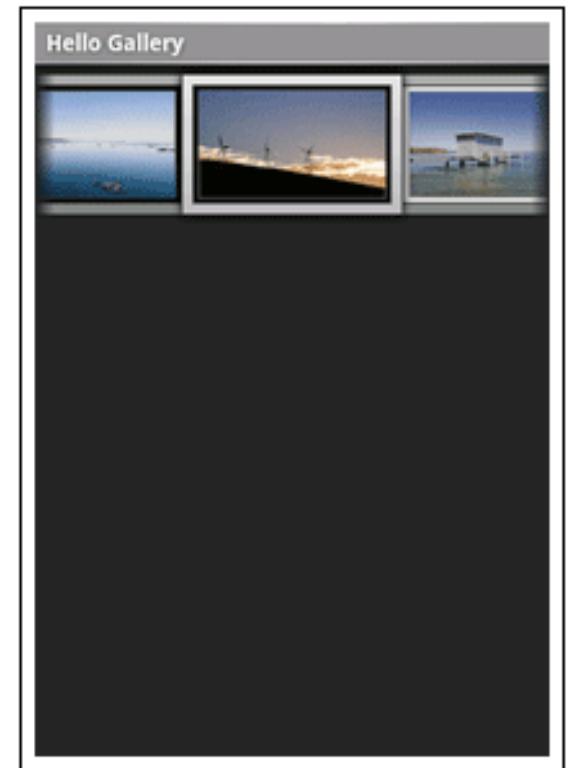
Spinner



Auto Complete



Gallery



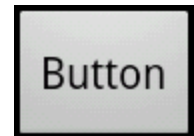


- XML
- Common attributes
 - ▣ Id , text , layout_width, layout_height
 - ▣ Various other fields

```
<Button android:text="Button"
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
```

```
</Button>
```

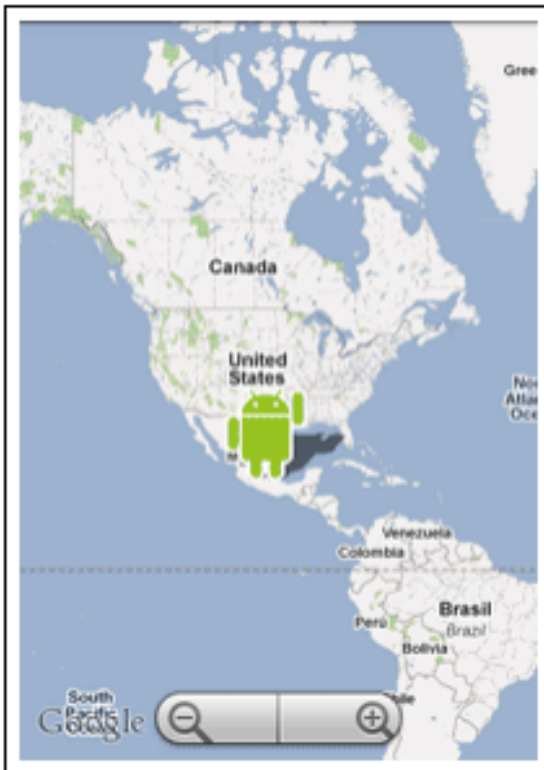
```
<CheckBox android:text="CheckBox"
          android:id="@+id/checkbox1"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:checked="true">
```



Android UI – Basic Views

44

Google Map View



Web View



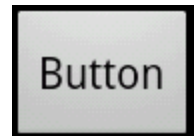


- XML
- Common attributes
 - ▣ Id , text , layout_width, layout_height
 - ▣ Various other fields

```
<Button android:text="Button"
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
```

```
</Button>
```

```
<CheckBox android:text="CheckBox"
          android:id="@+id/checkbox1"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:checked="true">
```



Android UI – Event Handling

46

- ❑ Pattern for working with Widgets
- ❑ Get the instance to the view
 - ❑ `findViewById(R.id.<viewid>)`
- ❑ Set up Event Handlers
 - ❑ Example : `onClick`
- ❑ Provide Event Handler implementation
- ❑ Study the Java class for different methods and events that you can handle on the Widgets

Android UI – Layout

47

- Layout – Recommended way is to define the Viewgroup hierarchy in an XML file
- Different Layouts are supported. Common ones are:
 - ▣ LinearLayout
 - ▣ Framework
 - ▣ TableLayout
 - ▣ RelativeLayout
- Other ViewGroups: Gallery, GridView, ScrollView, TabHost, etc.

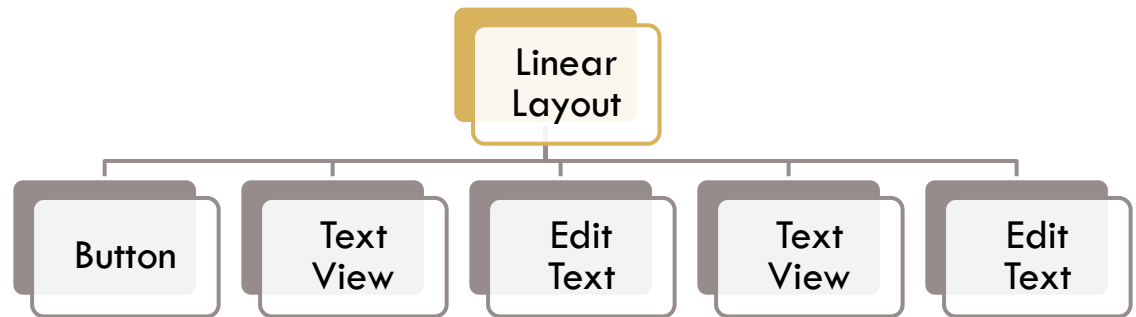
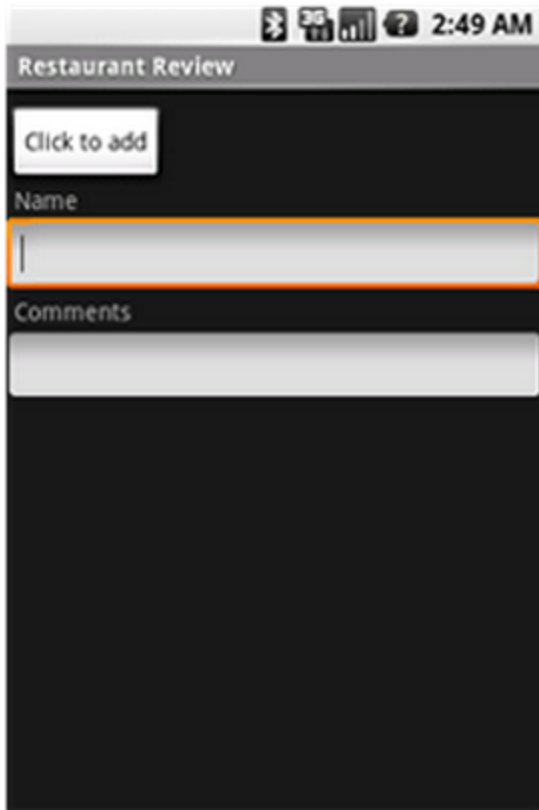
Android UI – Linear Layout

48

- ❑ Is a ViewGroup
- ❑ Displays Child View Elements in a linear fashion
- ❑ Linear = One After Another
- ❑ Orientation attribute
 - ▣ Horizontal
 - ▣ Vertical

Android UI – Linear Layout

49



Android UI – Linear Layout

50

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <Button .... />
    <TextView ..... />
    <EditText ..../>
    <TextView ..../>
    <EditText ..../>

</LinearLayout>
```

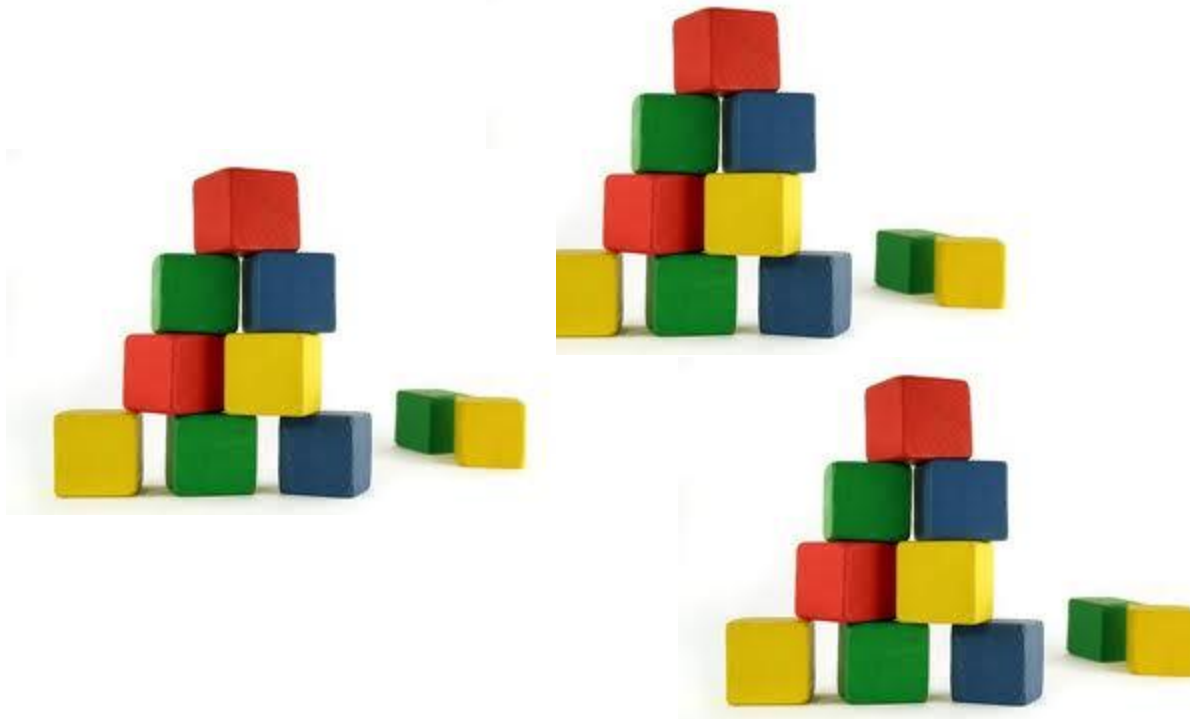
Hands On Exercise

51

- Hands On Exercise
 - ▣ **ex02.docx** – View and Layout Basics



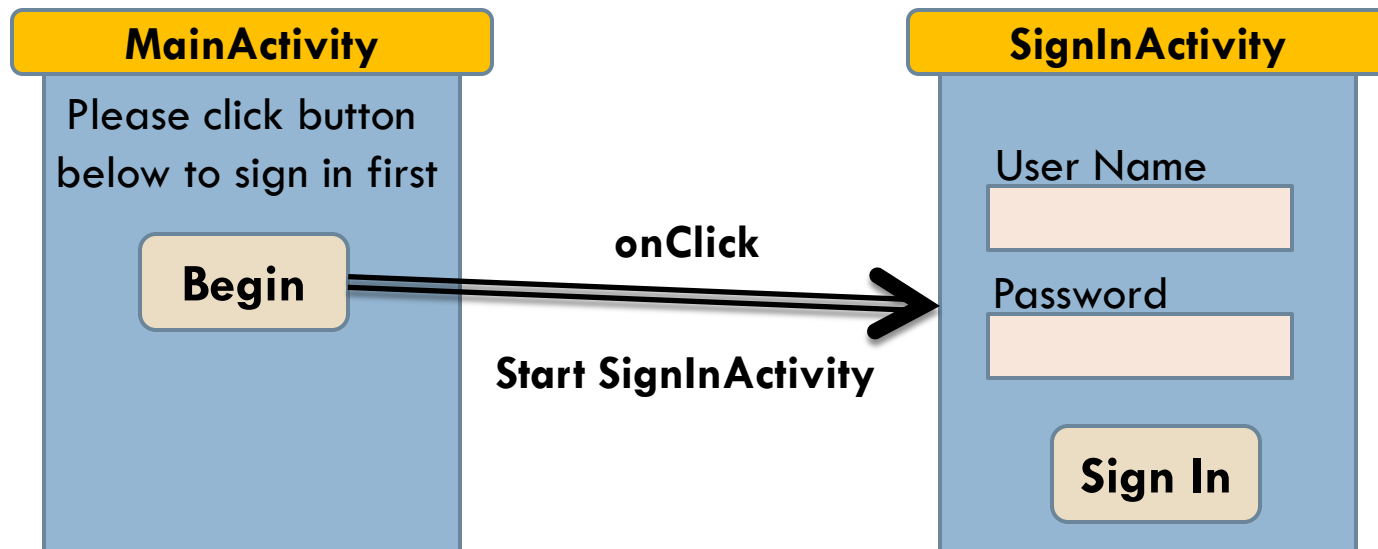
Multiple Activities



Multiple Activities

53

- You can define more than one Activity in an Application
- Think of : Activity \leftrightarrow User Screen
- You can launch an Activity from within an Activity



Android - Intent

54

- An Intent can be loosely termed as what you want to do.
- Examples:
 - ▣ Start a particular Activity
 - ▣ Send Email
 - ▣ Send SMS
 - ▣ View URL
- Intents : Explicit and Implicit

Android – Explicit Intent

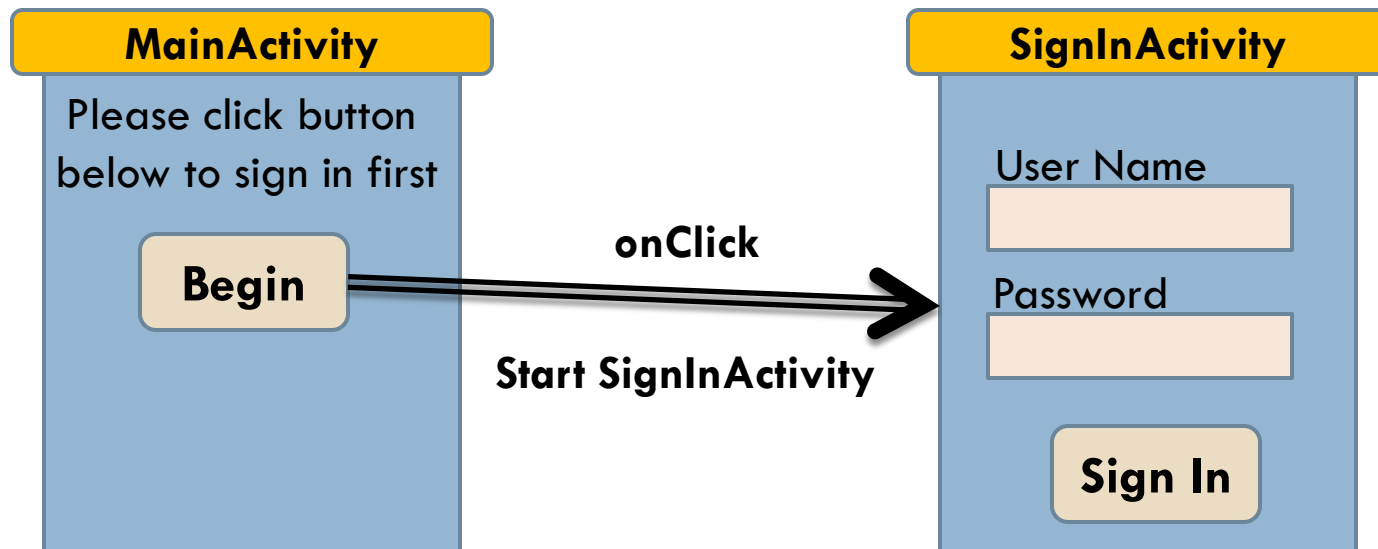
55

- Explicit Intent
 - ▣ Start Activity 2
 - ▣ You need to know the class for Activity 2
 - ▣ For e.g. if you have a screen (Activity 1) which has one button saying “Begin”
 - ▣ On clicking “Begin”, you want to display another form (start Activity) where the user enters userid/password

Android – Explicit Intent

56

```
Intent intent = new Intent(this, SignInActivity.class);  
  
startActivity(intent);
```



Android - Intent

57

- An Intent can be loosely termed as what you want to do.
- Examples:
 - ▣ Start a particular Activity
 - ▣ Send Email
 - ▣ Send SMS
 - ▣ View URL
- Intents : Explicit and Implicit

Android – Explicit Intent

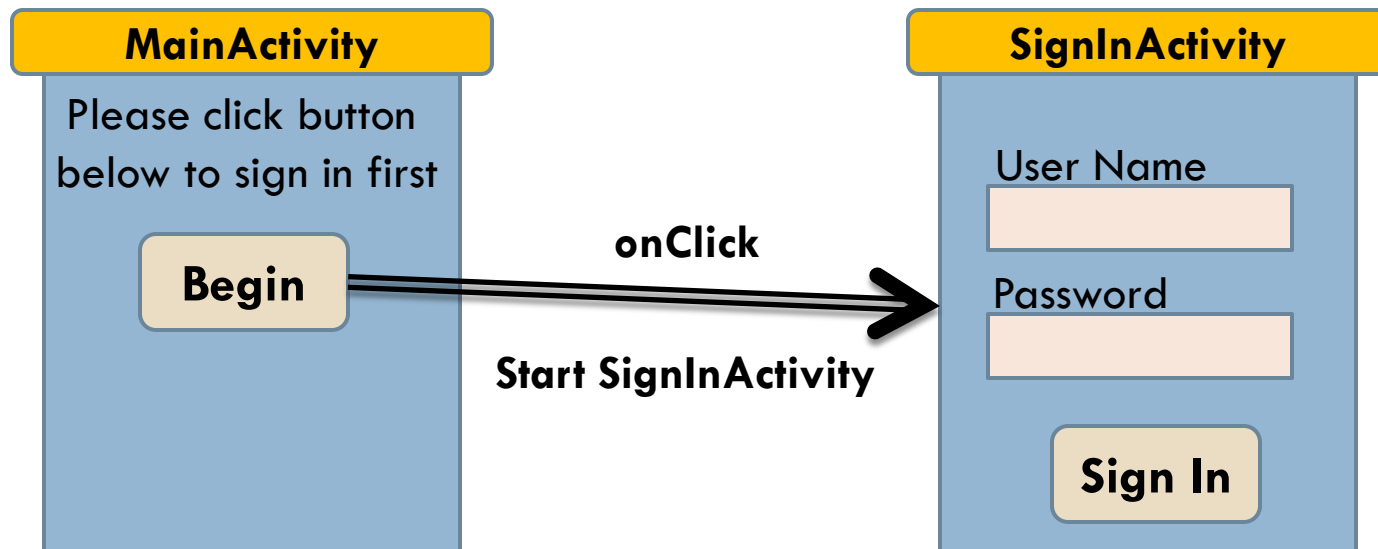
58

- Explicit Intent
 - ▣ Start Activity 2
 - ▣ You need to know the class for Activity 2
 - ▣ For e.g. if you have a screen (Activity 1) which has one button saying “Begin”
 - ▣ On clicking “Login”, you want to display another form (start Activity) where the user enters userid/password

Android – Explicit Intent

59

```
Intent intent = new Intent(this, SignInActivity.class);  
  
startActivity(intent);
```



Android Activity - Manifest

60

- All Activities need to be defined in the Android Manifest XML file

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.mindstorm.android"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".Activity1"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Activity2" android:label="@string/app_name"></activity>

    </application>
</manifest>
```

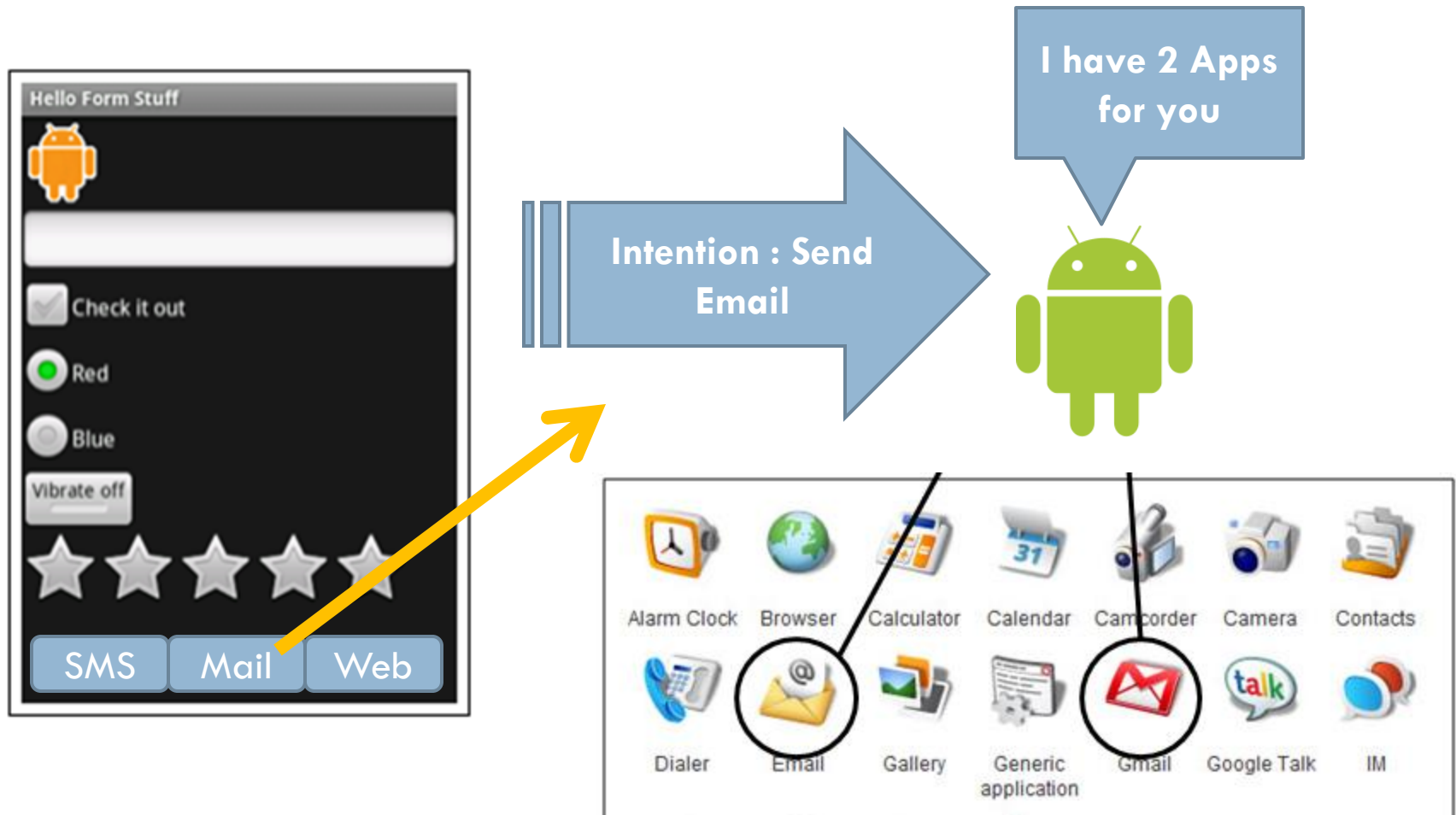
Android – Implicit Intent

61

- Android supports various standard activities that are commonly clubbed under Implicit Intent
- Examples:
 - ▣ Send Email
 - ▣ Send SMS
 - ▣ Dial / Call a number
 - ▣ View a web page
 - ▣ View a Map
 - ▣ Launch Camera
 - ▣ Share Content
- Android will look at all applications that are capable of handling the Intent and then it will launch that

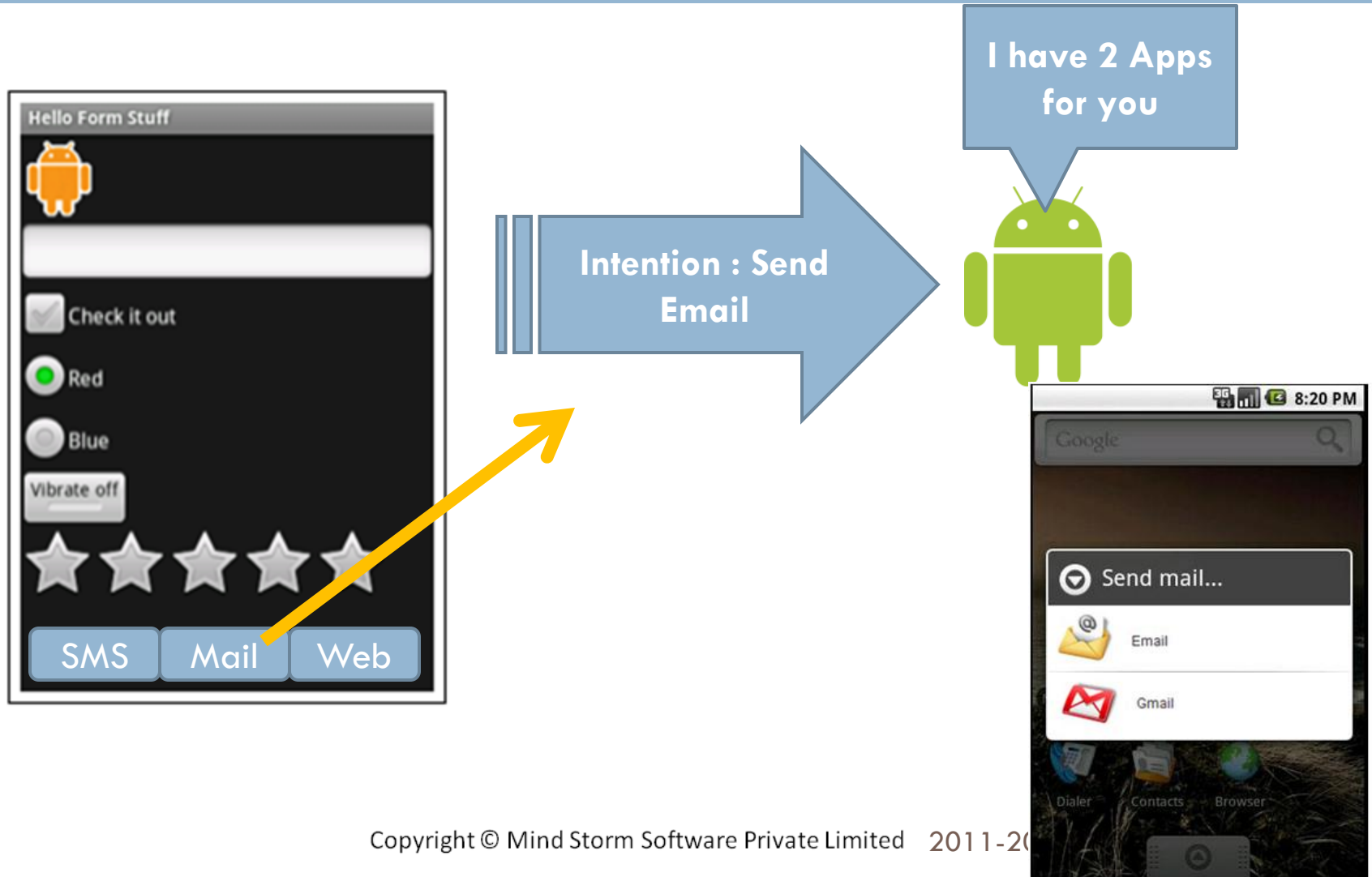
Android – Implicit Intent

62



Android – Implicit Intent

63



Android – Implicit Intents - 1

64

□ To Launch a Browser

```
Intent i = new  
    Intent(android.content.Intent.ACTION_VIEW,  
        Uri.parse("http://www.amazon.com"));  
startActivity(i);
```


Android – Implicit Intents - 2

65

- Built In Intent (ACTION_DIAL)
- This will show the Dial Screen but not start calling

```
Intent i = new Intent  
        (android.content.Intent.ACTION_DIAL,  
         Uri.parse("tel:+9810020011"));  
  
startActivity(i);
```

Android – Implicit Intents - 3

66

- Built In Intent (ACTION_CALL)
- This will start calling the number
- Requires the following permission:
**<uses-permission
android:name="android.permission.CALL_PHONE">**

```
Intent i = new Intent  
        (android.content.Intent.ACTION_CALL,  
         Uri.parse("tel:+9810020011"));  
  
startActivity(i);
```

Android – Send Email

67

```
Intent emailIntent = new Intent(Intent.ACTION_SEND);  
emailIntent.setType("text/plain");  
  
String[] to = {"romin.k.irani@gmail.com"};  
emailIntent.putExtra(Intent.EXTRA_EMAIL, to);  
emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Email Subject");  
emailIntent.putExtra(Intent.EXTRA_TEXT, "Email message");  
  
startActivity(Intent.createChooser(emailIntent, "Send Email via..."));
```

Android – Send SMS

68

```
Intent i = new  
    Intent(android.content.Intent.ACTION_VIEW);  
i.putExtra("address", "9821111111;9822211111");  
i.putExtra("sms_body", "Hello");  
i.setType("vnd.android-dir/mms-sms");  
startActivity(i);
```

Hands On Exercise

69

- Hands On Exercise
 - ▣ **ex03.docx** -- Multiple Activities
 - ▣ **ex04.docx** -- Built In Intents



Building a Complete Android Application



Quotes Application

71

- Functional Specifications
 - ▣ Displays Random Quotes
 - ▣ User can share quotes via SMS
 - ▣ **Future Extensions**
 - User can view quotes by category
 - User can mark some quotes as favourites

Quotes Application

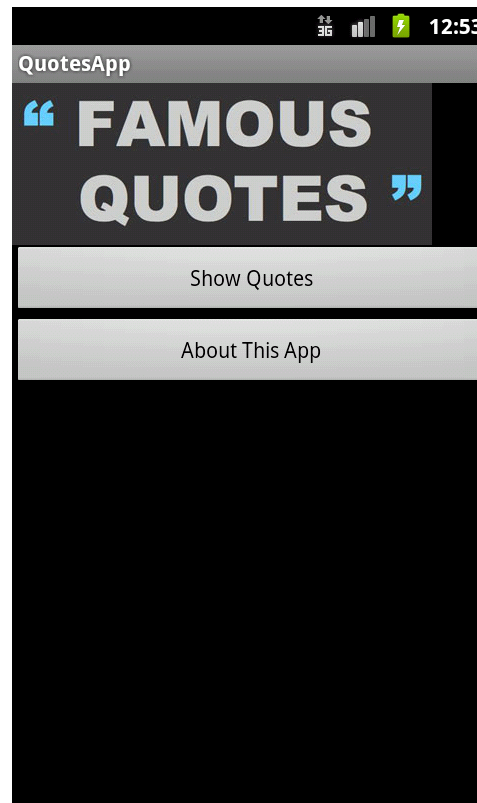
72

- Android Features to be used
 - ▣ Activities [**Multiple Screens**]
 - ▣ Layouts for all activities [**LinearLayout**]
 - ▣ **Built in Intents**
 - Launch SMS App to share Quote via SMS
 - Launch Email App to share Quote via Email
 - ▣ File based database to store quotes

Quotes App in Action

73

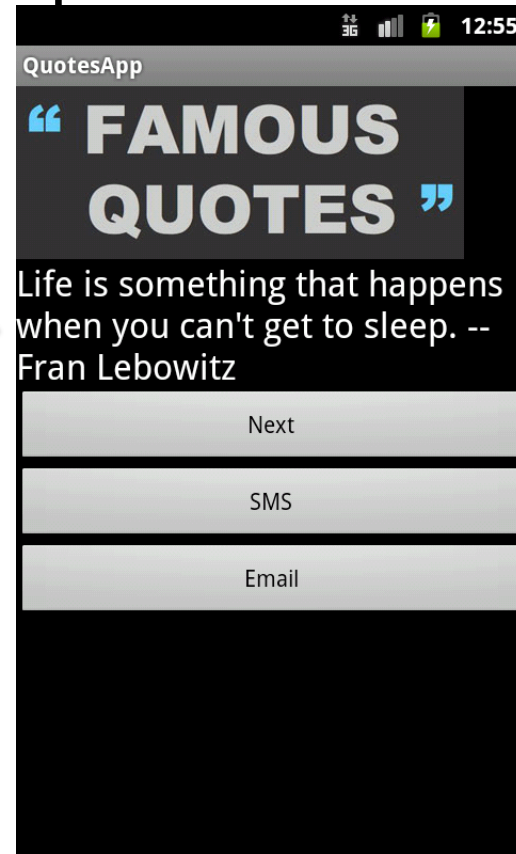
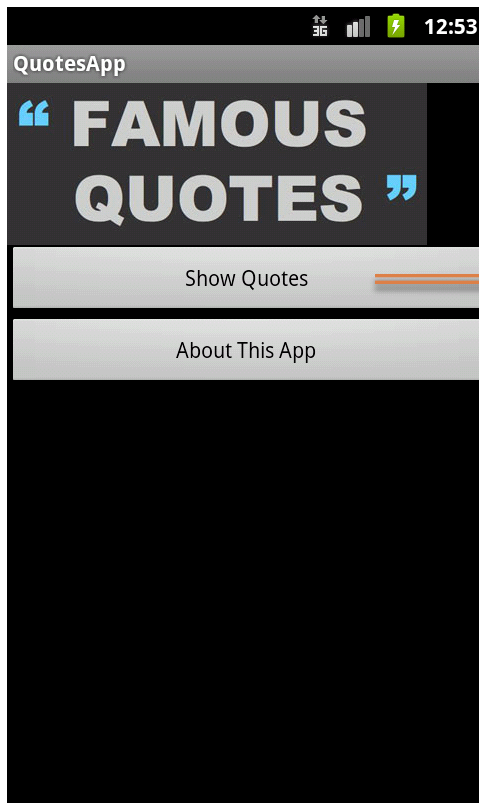
- QuoteActivity – Shows Main Menu



Quotes App in Action

74

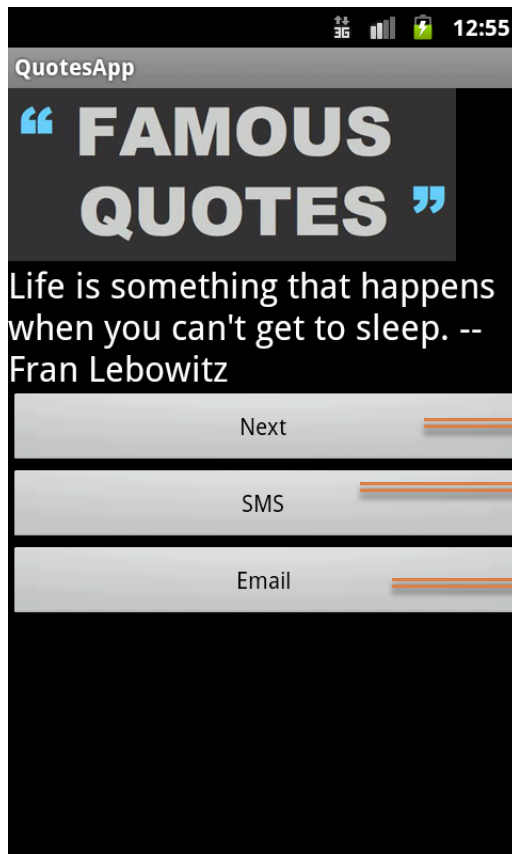
- Show Quotes retrieves a quote from Database



Quotes App in Action

75

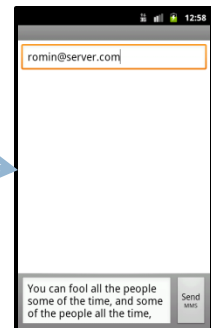
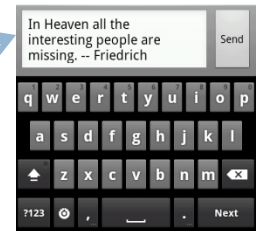
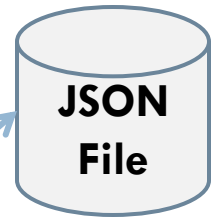
□ Show Quote Activity – Other Functions



Retrieves another random quote from database and displays it

Launches SMS App on phone to share this quote

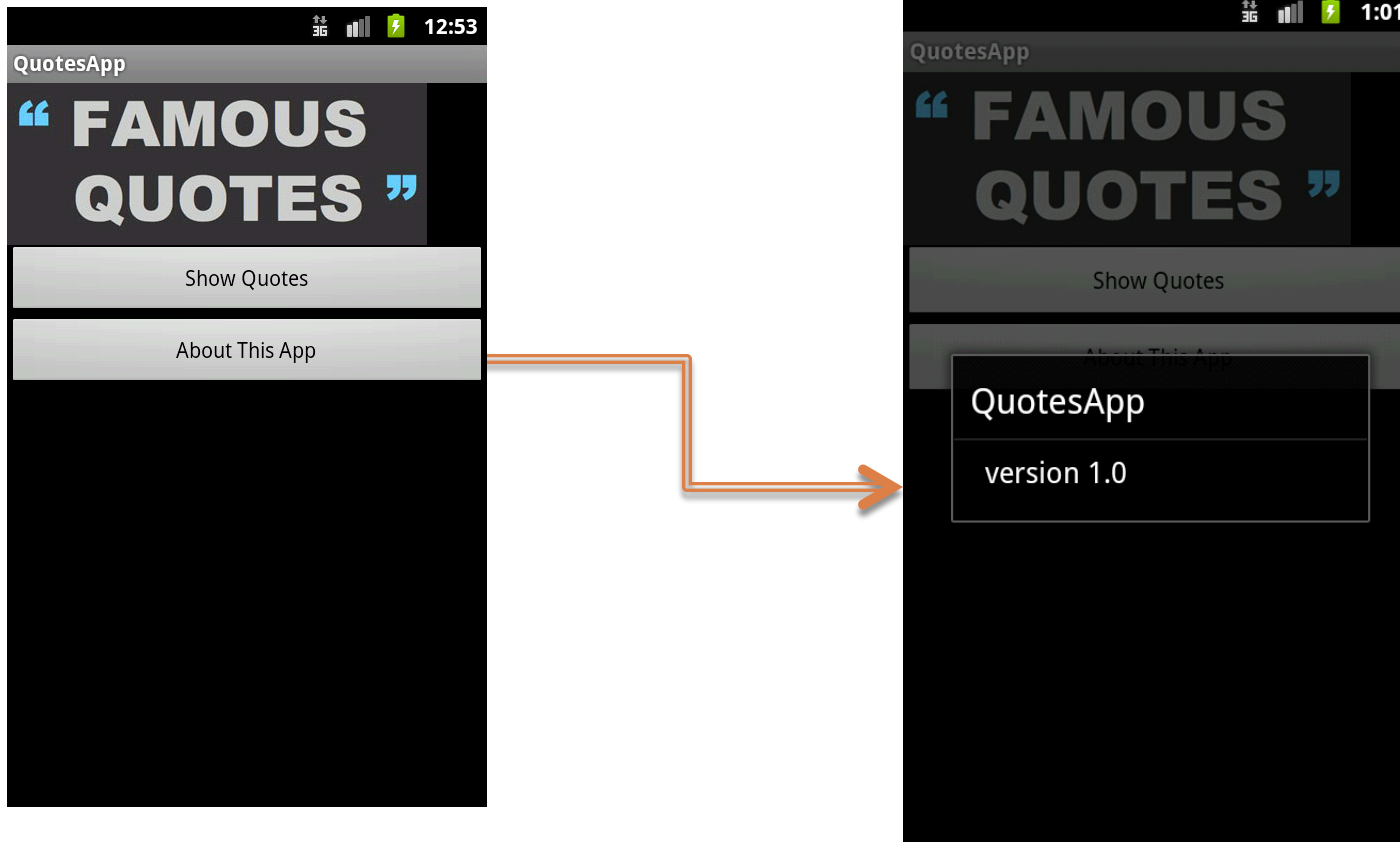
Launches Email App on phone to share this quote



Quotes App in Action

76

□ About App – Displays a Dialog box



Quotes App – Step by Step

77

- Quotes App – Step 1 - Navigation
- Quotes App – Step 2
 - ▣ Random Quotes
 - ▣ Share a Quote via SMS
 - ▣ Share a Quote via Email
- *Future - Quotes App – Step 3 - Favourites*
- *Future - Quotes App – Step 4 – Category Quotes*

Android Marketplace



Android Marketplace (Google Play)

79

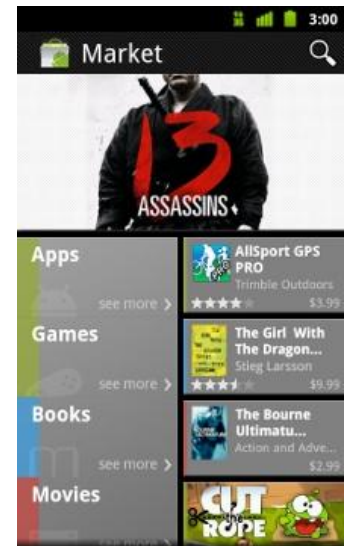
□ <http://play.google.com>



Android Marketplace

80

- ❑ Web application + Android Application
- ❑ Single Marketplace for eBooks, Music, Movies and Android Applications
- ❑ Filters out the above categories if services are not available in particular country
- ❑ Publisher Account
 - ❑ Google Account
 - ❑ Free Apps and Paid Apps (30% commission)
 - ❑ Selling a Paid App currently not available for India
 - ❑ Download Free Apps
 - ❑ Download Paid Apps
 - ❑ \$25 Registration Fee
 - ❑ Application Approval - Immediate










Android Marketplace

81

- For the Publisher – Login and visit <https://play.google.com/apps/publish/Home>

All Android Market listings

	Blood Donor Contact Manager 2.0 Applications: Social	(6)★★★★★ Comments	317 total installs 54 active installs (17 %) Statistics	Free	Errors	✓ Published Advertise this app
	Cows And Bulls 1.0 Games: Brain & Puzzle	(1)★★★★★ Comments	592 total installs 80 active installs (13 %) Statistics	Free	Errors	✓ Published Advertise this app
	Daily Channel 1.0 Applications: Entertainment	(1)★☆☆☆☆ Comments	54 total installs 8 active installs (14 %) Statistics	Free	Errors	✓ Published Advertise this app
	Game Score 1.0 Applications: Sports	(1)★★★★★ Comments	292 total installs 108 active installs (36 %) Statistics	Free	Errors	✓ Published Advertise this app
	Tambola Nite 3.0 Games: Casual	(3)★★★★☆ Comments	264 total installs 119 active installs (45 %) Statistics	Free	Errors	✓ Published Advertise this app
	Time Table Lite 2.0 Applications: Education	(0)☆☆☆☆☆ Comments	504 total installs 116 active installs (23 %) Statistics	Free	Errors	✓ Published Advertise this app
	Time Table Pro 1.0 Applications: Education	(9)★★★★★ Comments	2187 total installs 820 active installs (37 %) Statistics	Free	Errors	✓ Published Advertise this app

 Upload Application



- Test Your Application
- Review your AndroidManifest.xml file
 - ▣ Permissions (<uses-permission>)
 - ▣ Icon and Label
 - ▣ Application Version
 - ▣ API Level : <uses-sdk>, minSDKVersion, targetSDKVersion
 - ▣ Other Filters : <supports-screens>, etc
 - ▣ <http://developer.android.com/guide/appendix/market-filters.html>
- Compile Your Application
- Export Signed Application (.APK)

Build the APK

83

- APK = Android Application Package
- File Format to distribute and install software on Android OS
- Android Marketplace allows only signed APKs
- APK = Archive File
 - ▣ META-INF directory
 - ▣ res
 - ▣ AndroidManifest.xml
 - ▣ classes.dex
 - ▣ resources.arsc



- Upload the APK
- **Hands On Exercise : Build the APK : ex15.docx**
- Provide 2 screenshots
- Provide 1 High Resolution Image Icon (512 x 512)
- Title , Description, Recent Changes
- Application Type
- Category
- Contact Details

Android APIs

85

- We have seen just a fraction of what Android phones can do
- APIs for
 - ▣ Media
 - ▣ Networking
 - ▣ Location Based Programming (GPS, Maps)
 - ▣ Storage (File, Database)
- Rich UI Widgets



Next Steps

86

- ❑ Go back and absorb the material
- ❑ Look at the examples again
- ❑ Github:
<https://github.com/rominirani/Nitrodroid-2012-Android101-Workshop>
- ❑ Android Developer
<http://developer.android.com>
- ❑ Android Training
<http://developer.android.com/training>
- ❑ Get the Google Play Account...
- ❑ ... unleash your Apps
- ❑ All The Best !





THANK YOU!!!



- Q & A
- Website : <http://www.mindstormsoftware.com>
- Email : romin.irani@mindstormsoftware.com
- Twitter: iRomin
- 2-Day Android, HTML5 and Cloud Computing Hands-on Developer Courses

