

Android Development Tutorial

Human-Computer Interaction (COMP 4020)
Winter 2014

Today: Android Tutorial

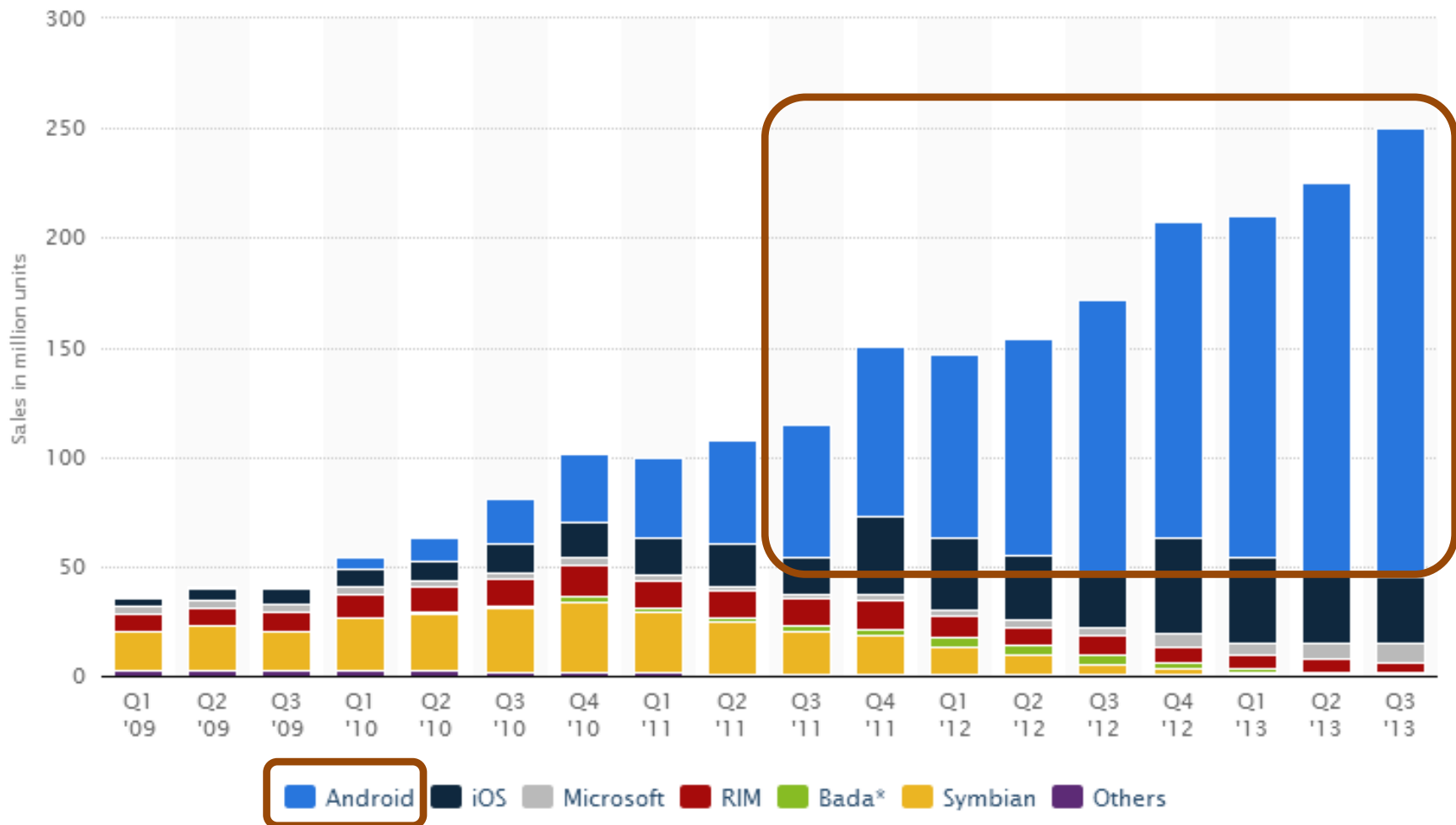
- ① Installation Issues
- ② Folder Structure
- ③ Core Components
- ④ Sample Applications

Mobile Phone OS

- Symbian
- iOS
- BlackBerry
- Windows Phone
- Android
-



World-Wide Smartphone Sales (In Million Units)



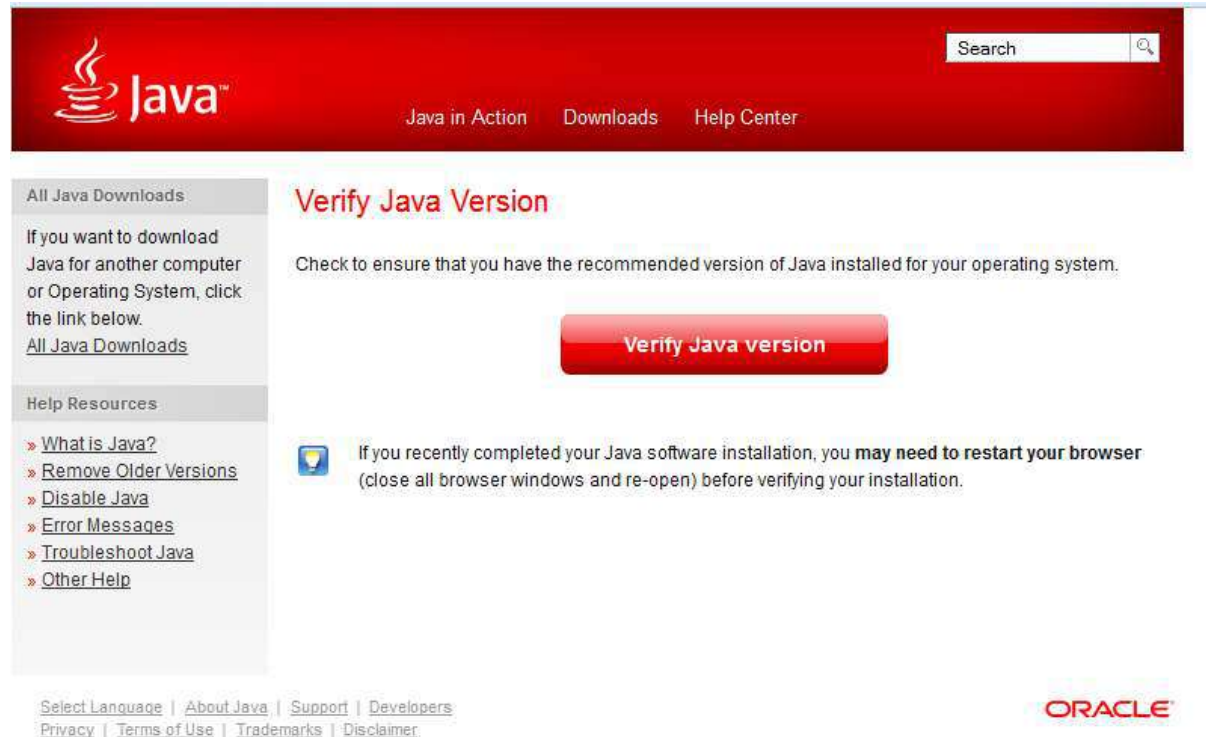
Source: Smartphone Sales, by operating system, Q1 2009-Q3 2013 <http://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operating-system/>

Installation Requirements

- ① Java Runtime Environment (JRE)
- ② Java Development Kit (JDK)
- ③ Android Developer Tools Bundle (ADT Bundle)
 - Eclipse + ADT plugin
 - Android SDK Tools
 - Android Platform-tools
 - The latest Android platform
 - The latest Android system image for the emulator

Verify Java Version

<http://www.java.com/en/download/installed.jsp>



The screenshot shows the Java website's 'Verify Java Version' page. At the top, there is a red navigation bar with the Java logo on the left, a search box on the right, and links for 'Java in Action', 'Downloads', and 'Help Center' in the center. Below the navigation bar, the page is divided into a left sidebar and a main content area. The sidebar contains two sections: 'All Java Downloads' with a link to 'All Java Downloads' and 'Help Resources' with a list of links: 'What is Java?', 'Remove Older Versions', 'Disable Java', 'Error Messages', 'Troubleshoot Java', and 'Other Help'. The main content area features the title 'Verify Java Version' in red, followed by the instruction 'Check to ensure that you have the recommended version of Java installed for your operating system.' Below this is a prominent red button labeled 'Verify Java version'. Further down, there is a blue shield icon and a note: 'If you recently completed your Java software installation, you **may need to restart your browser** (close all browser windows and re-open) before verifying your installation.' At the bottom of the page, there are links for 'Select Language', 'About Java', 'Support', 'Developers', 'Privacy', 'Terms of Use', 'Trademarks', and 'Disclaimer', along with the Oracle logo on the right.

Verify Java Version

<http://www.java.com/en/download/installed.jsp>

The screenshot shows the Java website's 'Verify Java Version' page. The header is red with the Java logo on the left and navigation links 'Java in Action', 'Downloads', and 'Help Center' in the center. A search bar is located in the top right corner. The main content area has a red box around the text 'No working Java was detected on your system' and a 'Download Java Now' button. A sidebar on the left contains links for 'All Java Downloads' and 'Help Resources'. The footer includes 'ORACLE' and various legal links.

Verify Java Version

No working Java was detected on your system. Install Java by clicking the button below.

[Download Java Now](#)

If you recently completed your Java software installation, you may need to **restart your browser** (close all browser windows and re-open) or there is a problem with your installation.

» [Check the troubleshooting FAQ.](#)

ORACLE

[Select Language](#) | [About Java](#) | [Support](#) | [Developers](#)
[Privacy](#) | [Terms of Use](#) | [Trademarks](#) | [Disclaimer](#)

Install JRE


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

Oracle Technology Network > Java > Java SE > Downloads


Overview Downloads Documentation Community Technologies Training

Java SE Downloads

Next Releases (Early Access) Embedded Use Previous Releases

 [DOWNLOAD](#)

Java Platform (JDK) 7u51

 [DOWNLOAD](#)

JDK 7u51 & NetBeans 7.4

Java Platform, Standard Edition

Java SE 7u51

This release includes important security fixes. Oracle strongly recommends that all Java SE 7 users upgrade to this release.
[Learn more](#)

Which Java package do I need?

- **JDK:** (Java Development Kit). For Java Developers. Includes a complete JRE plus tools for developing, debugging, and monitoring Java applications.
- **Server JRE:** (Server Java Runtime Environment) For deploying Java applications on servers. Includes tools for JVM monitoring and tools commonly required for server applications, but does not include browser integration (the Java plug-in), auto-update, nor an installer. [Learn more](#)
- **JRE:** (Java Runtime Environment). Covers most end-users needs. Contains everything required to run Java applications on your system.

JDK DOWNLOAD	Server JRE DOWNLOAD	JRE DOWNLOAD
--	---	--

JDK 7 Docs

- Installation Instructions
- ReadMe

Server JRE 7 Docs

- Installation Instructions
- ReadMe

JRE 7 Docs


- Installation Instructions
- ReadMe

Java SDKs and Tools

- Java SE
- Java EE and Glassfish
- Java ME
- JavaFX
- Java Card
- NetBeans IDE
- Java Mission Control

Java Resources

- Java APIs
- Technical Articles
- Demos and Videos
- Forums
- Java Magazine
- Java.net
- Developer Training
- Tutorials
- Java.com

 [Subscribe Today](#)

Install JRE

Java SE
Java EE
Java ME
Java SE Support
Java SE Advanced & Suite
Java Embedded
JavaFX
Java DB
Web Tier
Java Card
Java TV
New to Java
Community
Java Magazine

Overview

Downloads

Documentation

Community

Technologies

Training

Java SE Runtime Environment 7 Downloads

Do you want to run Java™ programs, or do you want to develop Java programs? If you want to run Java programs, but not develop them, download the Java Runtime Environment, or JRE™.

If you want to develop applications for Java, download the Java Development Kit, or JDK™. The JDK includes the JRE, so you do not have to download both separately.

JRE MD5 Checksum

Java SE Runtime Environment 7u51

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux x86	31.5 MB	jre-7u51-linux-i586.rpm
Linux x86	46.09 MB	jre-7u51-linux-i586.tar.gz
Linux x64	32.01 MB	jre-7u51-linux-x64.rpm
Linux x64	44.73 MB	jre-7u51-linux-x64.tar.gz
Mac OS X x64	48.44 MB	jre-7u51-macosx-x64.dmg
Mac OS X x64	44.44 MB	jre-7u51-macosx-x64.tar.gz
Solaris x86	51.99 MB	jre-7u51-solaris-i586.tar.gz
Solaris x64	16.05 MB	jre-7u51-solaris-x64.tar.gz
Solaris SPARC	54.7 MB	jre-7u51-solaris-sparc.tar.gz
Solaris SPARC 64-bit	18.09 MB	jre-7u51-solaris-sparcv9.tar.gz
Windows x86 Online	0.88 MB	jre-7u51-windows-i586-iftw.exe
Windows x86 Offline	27.79 MB	jre-7u51-windows-i586.exe
Windows x86	39.66 MB	jre-7u51-windows-i586.tar.gz
Windows x64	29.37 MB	jre-7u51-windows-x64.exe
Windows x64	41.46 MB	jre-7u51-windows-x64.tar.gz

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [JavaFX](#)
- [Java Card](#)
- [NetBeans IDE](#)
- [Java Mission Control](#)

Java Resources

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)
- [Forums](#)
- [Java Magazine](#)
- [Java.net](#)
- [Developer Training](#)
- [Tutorials](#)
- [Java.com](#)



Installation Requirements

- ① Java Runtime Environment (JRE) ✓
- ② Java Development Kit (JDK)
- ③ Android Developer Tools Bundle (ADT Bundle)
 - Eclipse + ADT plugin
 - Android SDK Tools
 - Android Platform-tools
 - The latest Android platform
 - The latest Android system image for the emulator

Install JDK


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

Oracle Technology Network > Java > Java SE > Downloads


Overview Downloads Documentation Community Technologies Training

Java SE Downloads

Next Releases (Early Access) Embedded Use Previous Releases

 [DOWNLOAD](#)

Java Platform (JDK) 7u51

 [DOWNLOAD](#)

JDK 7u51 & NetBeans 7.4

Java Platform, Standard Edition

Java SE 7u51

This release includes important security fixes. Oracle strongly recommends that all Java SE 7 users upgrade to this release.
[Learn more](#)

Which Java package do I need?

- **JDK:** (Java Development Kit). For Java Developers. Includes a complete JRE plus tools for developing, debugging, and monitoring Java applications.
- **Server JRE:** (Server Java Runtime Environment) For deploying Java applications on servers. Includes tools for JVM monitoring and tools commonly required for server applications, but does not include browser integration (the Java plug-in), auto-update, nor an installer. [Learn more](#)
- **JRE:** (Java Runtime Environment). Covers most end-users needs. Contains everything required to run Java applications on your system.

JDK DOWNLOAD	Server JRE DOWNLOAD	JRE DOWNLOAD
--	---	--

JDK 7 Docs

- [Installation Instructions](#)
- [ReadMe](#)

Server JRE 7 Docs

- [Installation Instructions](#)
- [ReadMe](#)

JRE 7 Docs


- [Installation Instructions](#)
- [ReadMe](#)

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [JavaFX](#)
- [Java Card](#)
- [NetBeans IDE](#)
- [Java Mission Control](#)

Java Resources

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)
- [Forums](#)
- [Java Magazine](#)
- [Java.net](#)
- [Developer Training](#)
- [Tutorials](#)
- [Java.com](#)

 [Subscribe Today](#)

Install JDK

Java SE
Java EE
Java ME
Java SE Support
Java SE Advanced & Suite
Java Embedded
JavaFX
Java DB
Web Tier
Java Card
Java TV
New to Java
Community
Java Magazine
Java Advanced

- Overview
- Downloads
- Documentation
- Community
- Technologies
- Training

Java SE Development Kit 7 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

Looking for JavaFX SDK?

JavaFX SDK is now included in the JDK for Windows, Mac OS X, and Linux x86/x64.

See also:

- [Java Developer Newsletter](#) (tick the checkbox under Subscription Center > Oracle Technology News)
- [Java Developer Day hands-on workshops \(free\) and other events](#)
- [Java Magazine](#)

Java SE Development Kit 7u11

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux x86	106.61 MB	jdk-7u11-linux-i586.rpm
Linux x86	92.95 MB	jdk-7u11-linux-i586.tar.gz
Linux x64	104.75 MB	jdk-7u11-linux-x64.rpm
Linux x64	91.7 MB	jdk-7u11-linux-x64.tar.gz
Mac OS X x64	143.72 MB	jdk-7u11-macosx-x64.dmg
Solaris x86 (SVR4 package)	135.54 MB	jdk-7u11-solaris-i586.tar.Z
Solaris x86	91.92 MB	jdk-7u11-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	22.52 MB	jdk-7u11-solaris-x64.tar.Z
Solaris x64	14.95 MB	jdk-7u11-solaris-x64.tar.gz

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [JavaFX](#)
- [Java Card](#)
- [NetBeans IDE](#)

Java Resources

- [New to Java?](#)
- [APIs](#)
- [Code Samples & Apps](#)
- [Developer Training](#)
- [Documentation](#)
- [Java.com](#)
- [Java.net](#)
- [Student Developers](#)
- [Tutorials](#)

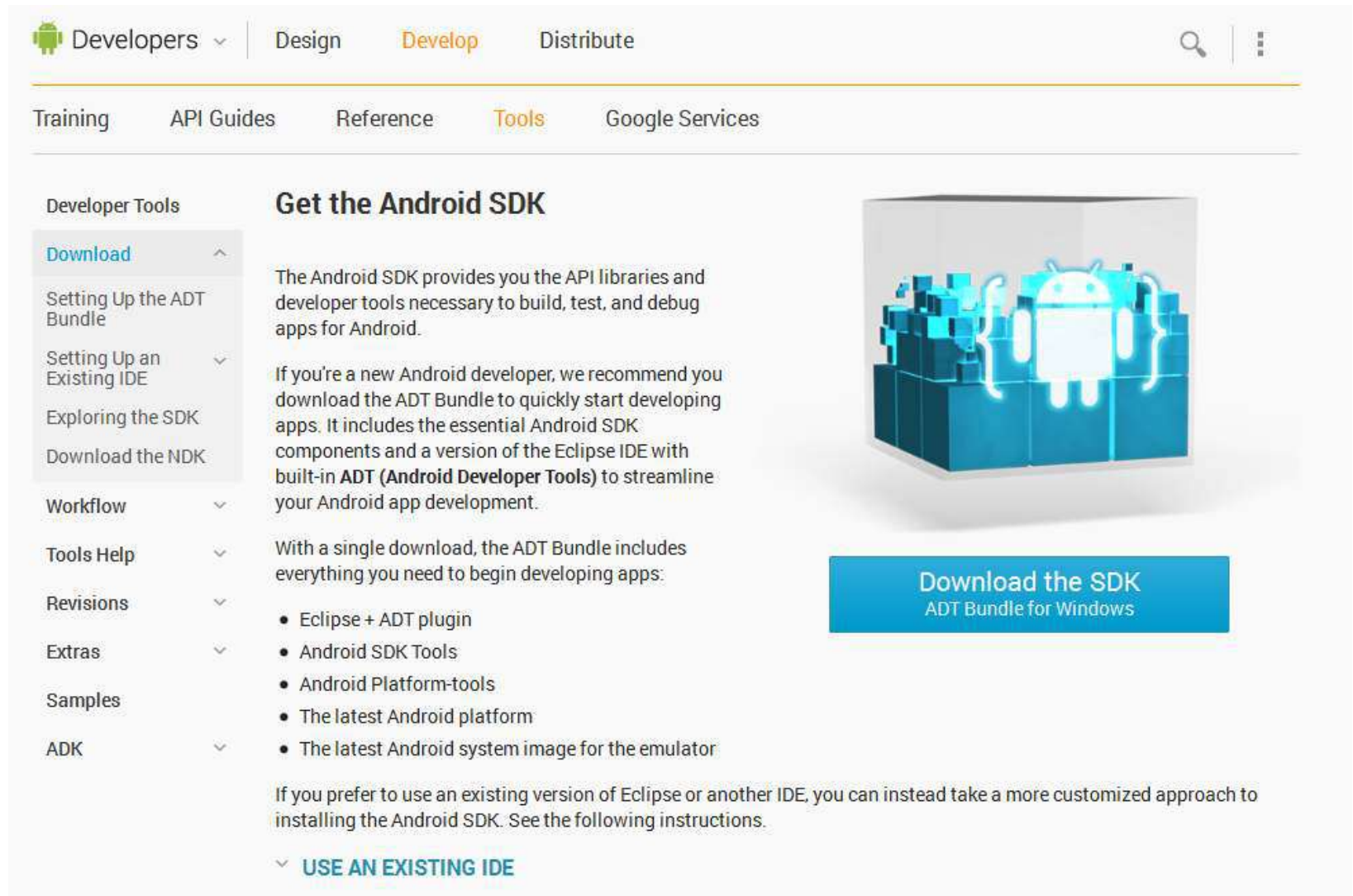


Installation Requirements

- ① Java Runtime Environment (JRE) ✓
- ② Java Development Kit (JDK) ✓
- ③ Android Developer Tools Bundle (ADT Bundle)
 - Eclipse + ADT plugin
 - Android SDK Tools
 - Android Platform-tools
 - The latest Android platform
 - The latest Android system image for the emulator

Install ADT Bundle

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



The screenshot shows the Android Developer website's 'Get the Android SDK' page. The navigation bar includes 'Developers', 'Design', 'Develop', and 'Distribute'. Below it are 'Training', 'API Guides', 'Reference', 'Tools', and 'Google Services'. The left sidebar lists 'Developer Tools' with sub-items like 'Download', 'Setting Up the ADT Bundle', 'Setting Up an Existing IDE', 'Exploring the SDK', and 'Download the NDK'. The main content area features the heading 'Get the Android SDK' and a description: 'The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.' It then recommends downloading the ADT Bundle for new developers, listing its components: Eclipse + ADT plugin, Android SDK Tools, Android Platform-tools, the latest Android platform, and the latest Android system image for the emulator. A blue button labeled 'Download the SDK ADT Bundle for Windows' is positioned below an image of the SDK components. At the bottom, there is a link to 'USE AN EXISTING IDE'.

Developers ▾ | Design | **Develop** | Distribute

Training | API Guides | Reference | **Tools** | Google Services

Developer Tools

- Download** ▴
- Setting Up the ADT Bundle
- Setting Up an Existing IDE ▾
- Exploring the SDK
- Download the NDK

Workflow ▾

Tools Help ▾

Revisions ▾

Extras ▾

Samples

ADK ▾

Get the Android SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.


If you're a new Android developer, we recommend you download the ADT Bundle to quickly start developing apps. It includes the essential Android SDK components and a version of the Eclipse IDE with built-in **ADT (Android Developer Tools)** to streamline your Android app development.

With a single download, the ADT Bundle includes everything you need to begin developing apps:

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator

If you prefer to use an existing version of Eclipse or another IDE, you can instead take a more customized approach to installing the Android SDK. See the following instructions.

▾ [USE AN EXISTING IDE](#)



Download the SDK
ADT Bundle for Windows


Eclipse

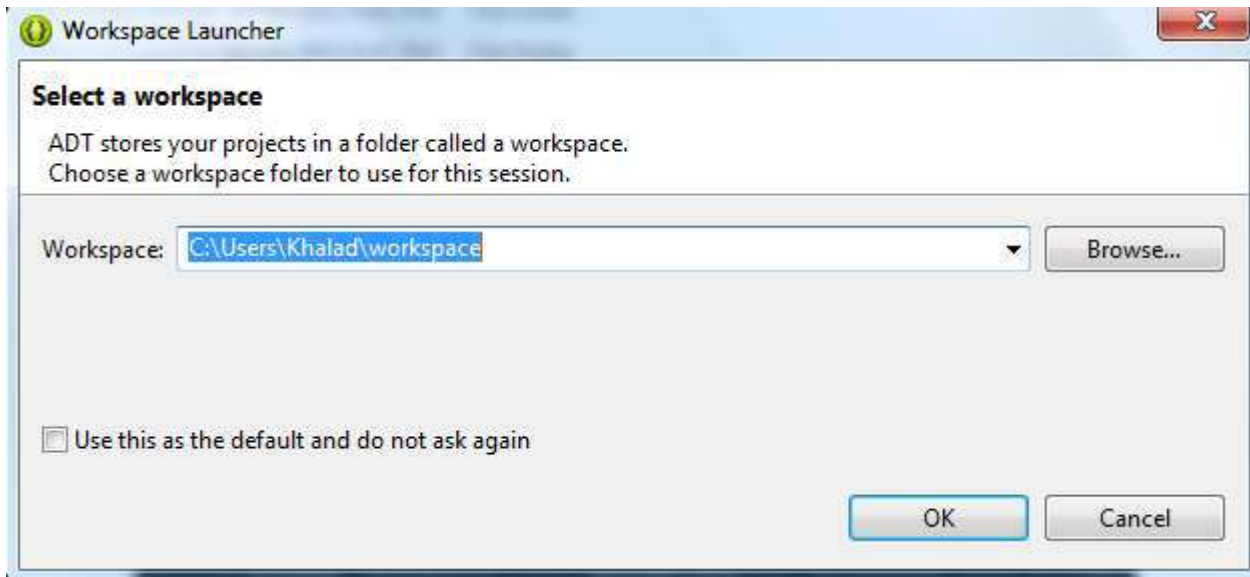
1

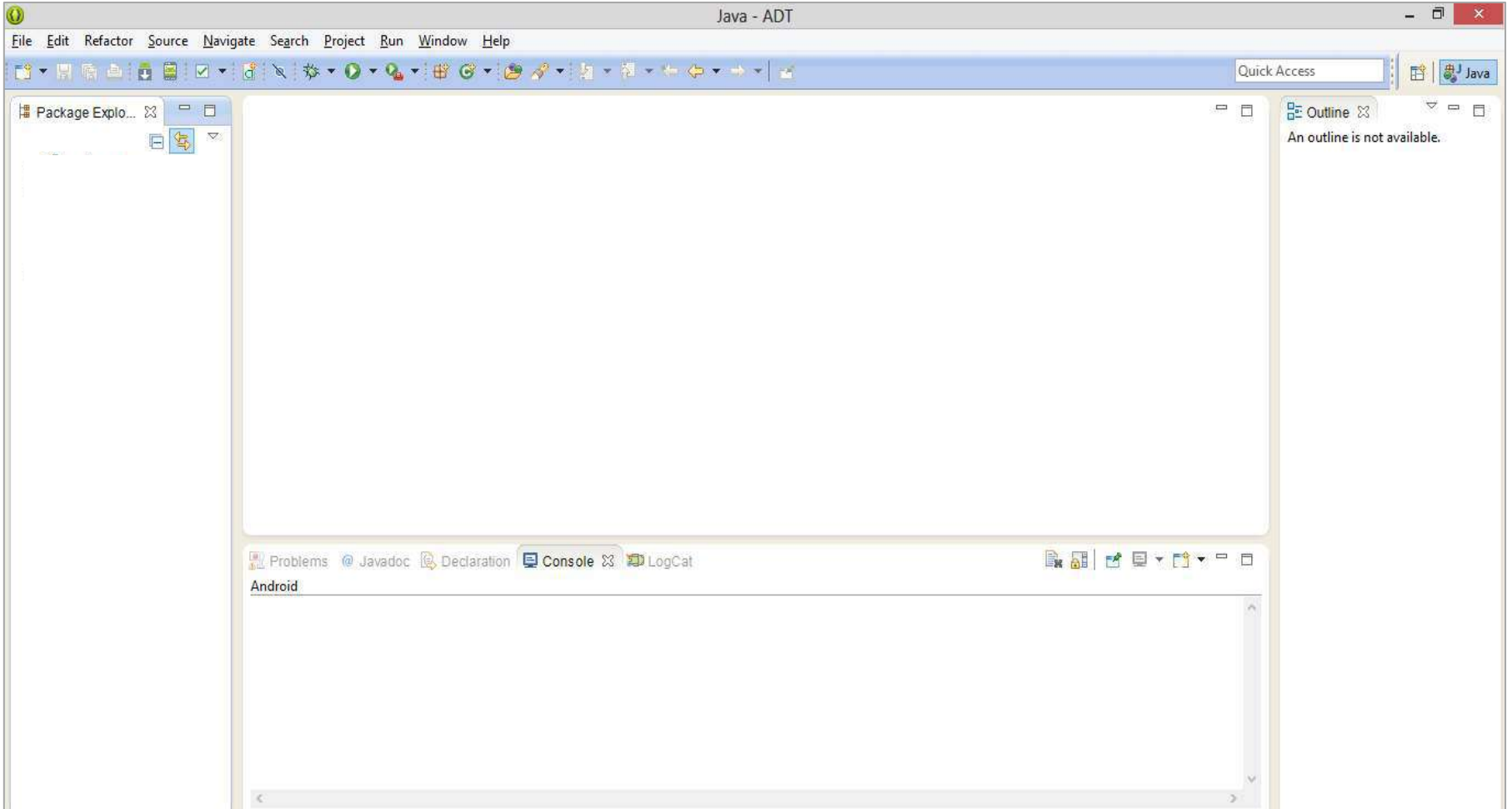
Name	Date modified	Type	Size
eclipse	17/01/2013 6:42 PM	File folder	
sdk	17/01/2013 6:44 PM	File folder	
SDK Manager	06/12/2012 11:09 ...	Application	350 KB

2

Name	Date modified	Type	Size
configuration	18/01/2013 1:47 AM	File folder	
dropins	05/12/2012 7:05 PM	File folder	
features	17/01/2013 6:41 PM	File folder	
p2	17/01/2013 6:41 PM	File folder	
plugins	17/01/2013 6:42 PM	File folder	
readme	17/01/2013 6:42 PM	File folder	
.eclipseproduct			
artifacts			
<input checked="" type="checkbox"/> eclipse			
eclipse			
eclipsec			
epl-v10			
notice			







Android SDK Manager



Android SDK Manager

The screenshot shows the Android SDK Manager interface. At the top, the title bar reads "Android SDK Manager". Below it, the "Packages" tab is active, and the "SDK Path" is set to "C:\Users\Khalad\Dropbox\Android Tutorials\adt-bundle-windows-x86\adt-bundle-windows-x86\sdk".

The main area displays a table of packages with columns for Name, API, Rev., and Status. The packages are organized into folders: Tools, Android 4.2 (API17), Android 4.1.2 (API16), Android 4.0.3 (API15), Android 4.0 (API14), Android 3.2 (API13), Android 3.1 (API12), Android 3.0 (API11), Android 2.3.3 (API10), Android 2.2 (API8), Android 2.1 (API7), Android 1.6 (API14), Android 1.5 (API3), and Extras.

Name	API	Rev.	Status
Tools			
Android SDK Tools		21.0.1	Installed
Android SDK Platform-tools		16.0.1	Installed
Android 4.2 (API17)			
Documentation for Android SDK	17	1	Installed
SDK Platform	17	1	Installed
Samples for SDK	17	1	Not installed
ARM EABI v7a System Image	17	1	Installed
MIPS System Image	17	1	Not installed
Google APIs	17	1	Not installed
Sources for Android SDK	17	1	Not installed
Android 4.1.2 (API16)			
Android 4.0.3 (API15)			
Android 4.0 (API14)			
Android 3.2 (API13)			
Android 3.1 (API12)			
Android 3.0 (API11)			
Android 2.3.3 (API10)			
Android 2.2 (API8)			
Android 2.1 (API7)			
Android 1.6 (API14)			
Android 1.5 (API3)			
Extras			
Android Support Library	11		Installed
Google AdMob Ads SDK	8		Not installed
Google Analytics SDK	2		Not installed
Google Cloud Messaging for Android Library	3		Not installed
Google Play services	4		Not installed
Google Play APK Expansion Library	2		Not installed
Google Play Billing Library	3		Not installed
Google Play Licensing Library	2		Not installed
Google USB Driver	7		Installed
Google Web Driver	2		Not installed
Intel x86 Emulator Accelerator (HAXM)	2		Not installed

At the bottom of the window, there are controls for filtering and sorting packages. The "Show:" section includes checkboxes for "Updates/New" (checked), "Installed" (checked), and "Obsolete" (unchecked). The "Sort by:" section has radio buttons for "API level" (selected) and "Repository" (unselected). There are also buttons for "Install packages...", "Delete packages...", and "Deselect All". A status bar at the very bottom indicates "Done loading packages."

Android SDK Manager

The screenshot shows the Android SDK Manager interface. At the top, the window title is "Android SDK Manager". Below the title bar, there are tabs for "Packages" and "Tools". The "SDK Path" is displayed as "C:\adt-bundle-windows-x86_64-20130522\sdk".

The main area is a table of packages. The "Real3D" package is highlighted with a red box. The table has columns for Name, API, Rev., and Status.

Name	API	Rev.	Status
Tools			
Android 4.4.2 (API 19)			
Android 4.3 (API 18)			
Android 4.2.2 (API 17)			
Android 4.1.2 (API 16)			
Android 4.0.3 (API 15)			
Android 4.0 (API 14)			
Android 3.2 (API 13)			
Android 3.1 (API 12)			
Android 3.0 (API 11)			
Android 2.3.3 (API 10)			
SDK Platform	10	2	Installed
Samples for SDK	10	1	Installed
Intel x86 Atom System Image	10	2	Installed
LG Real3D Add-on	10	2	Not installed
Google APIs	10	2	Installed
Real3D	10	2	Installed
Android 2.2 (API 8)			
Android 2.1 (API 7)			
Android 1.6 (API 4)			
Android 1.5 (API 3)			
Extras			

At the bottom of the window, there are controls for filtering and sorting. The "Show:" section has checkboxes for "Updates/New" (checked), "Installed" (checked), and "Obsolete" (unchecked). There is a link "Select New or Updates". The "Sort by:" section has radio buttons for "API level" (selected) and "Repository". There is a link "Deselect All". Two buttons are present: "Install 6 packages..." and "Delete 6 packages...".

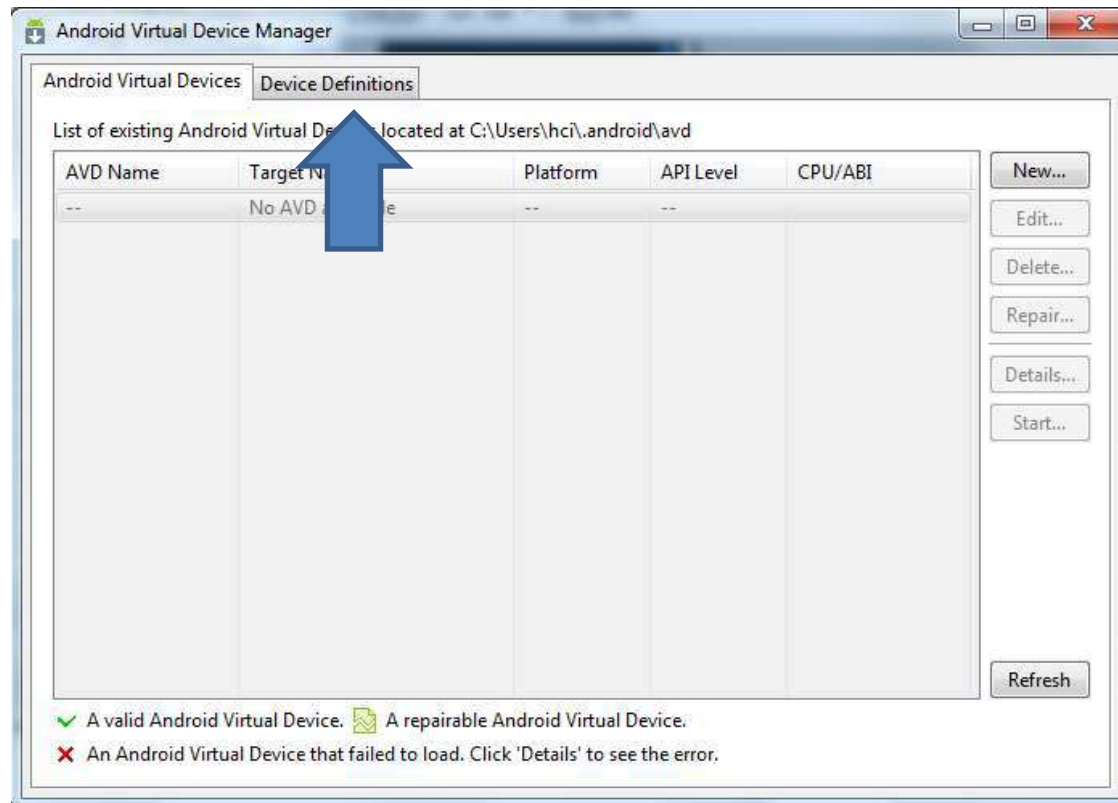
At the very bottom, there is a status bar that says "Done loading packages." and a small red icon.

Android Virtual Device Manager

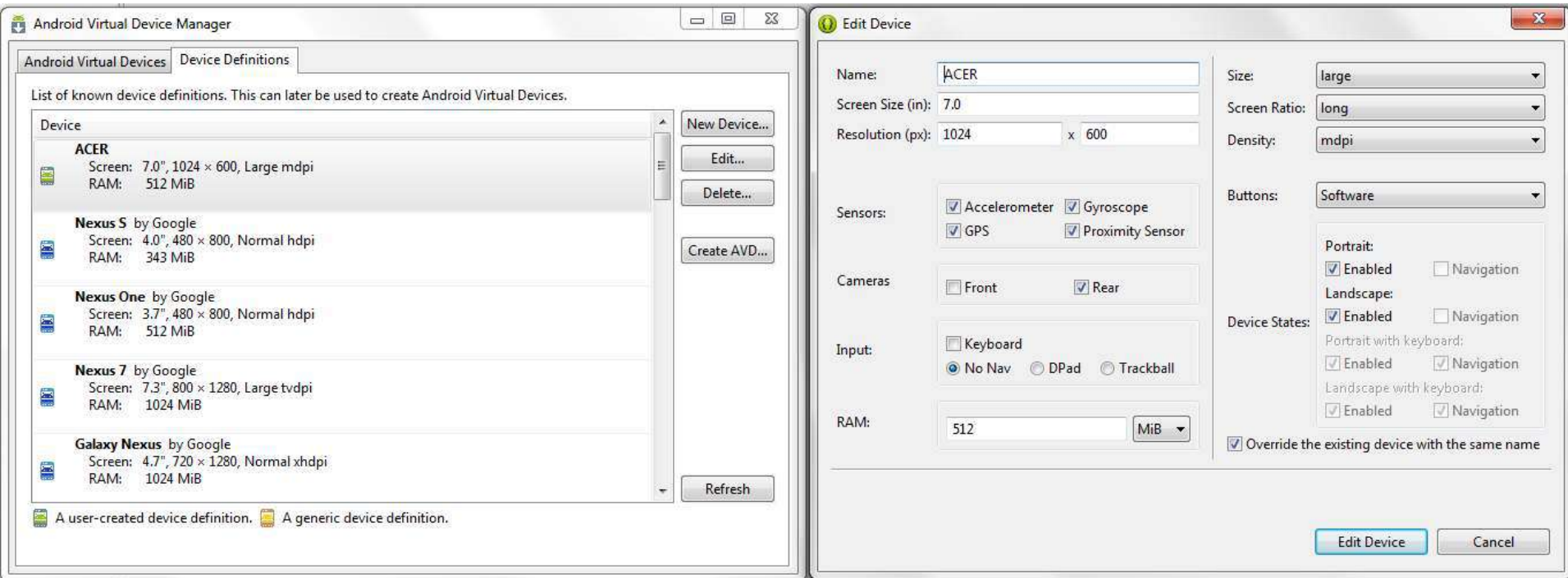
1



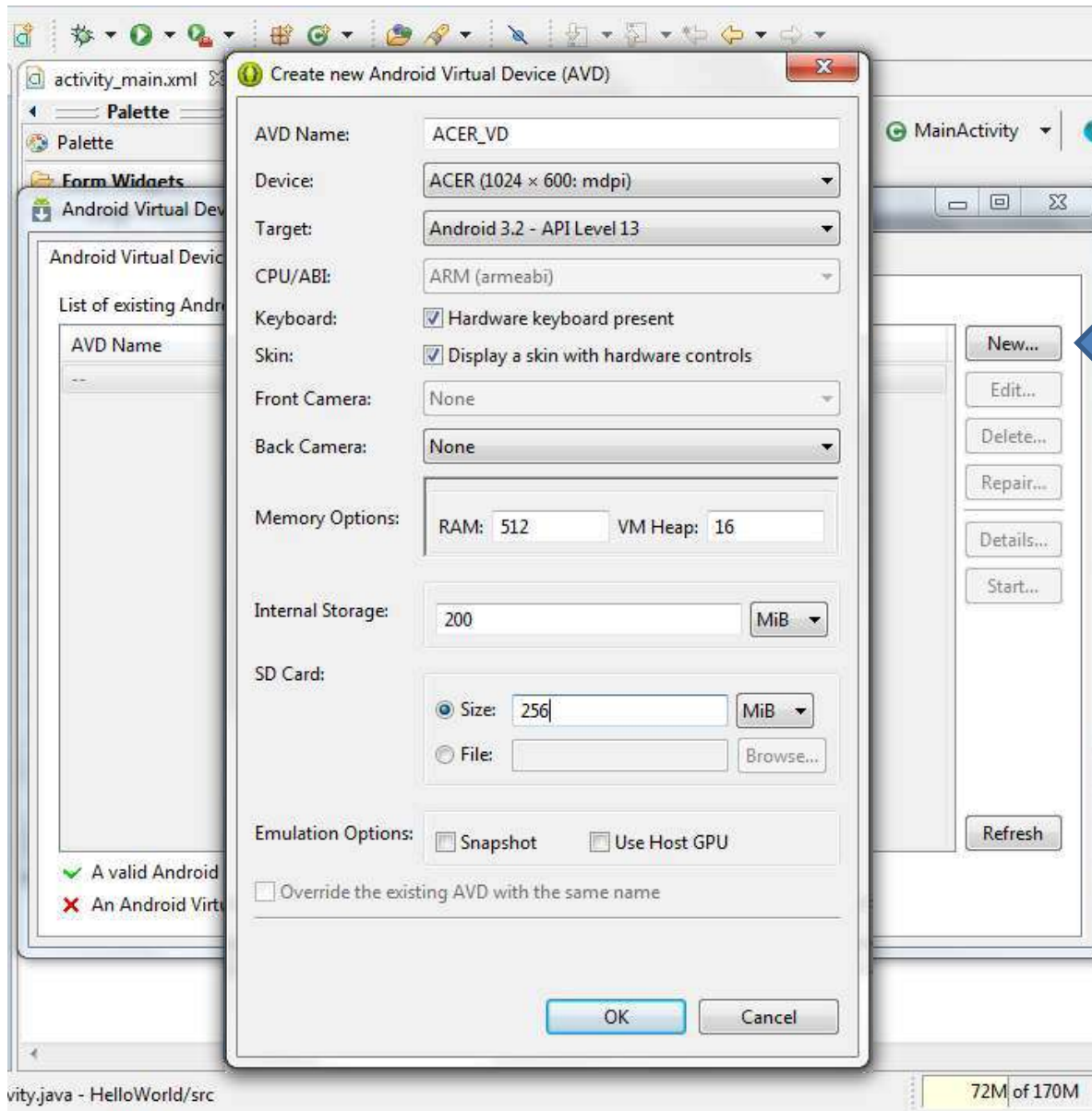
2



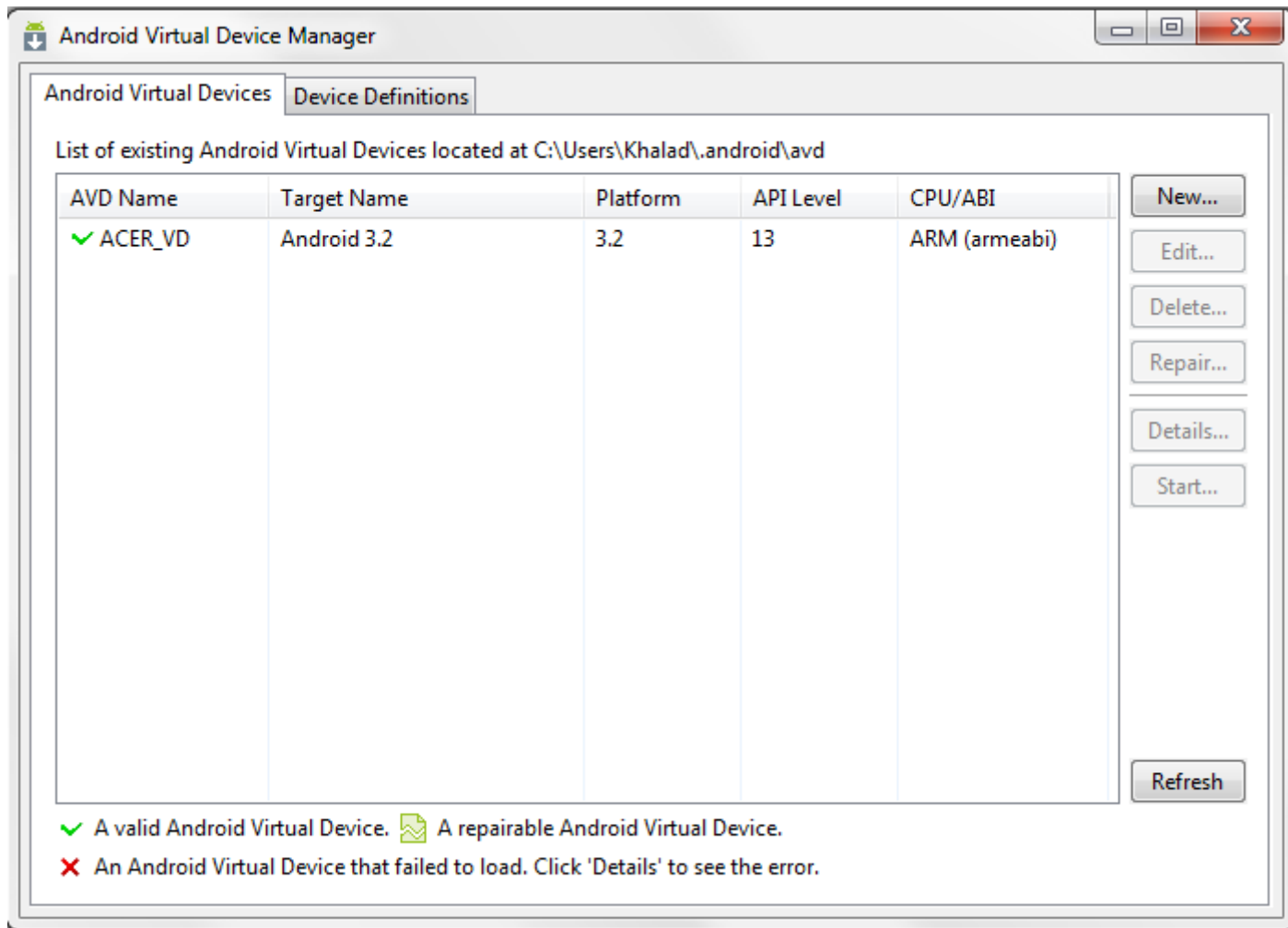
Android Virtual Device Manager



Android Virtual Device Manager



Android Virtual Device Manager



Enabling USB Debugging on device

Utilized to copy data between your device and computer.

For Android 3.2 or below:

Settings > Applications > Development > USB debugging

For Others:

Settings > Developer Options > USB debugging.



Installation Requirements

- ① Java Runtime Environment (JRE) ✓
- ② Java Development Kit (JDK) ✓
- ③ Android Developer Tools Bundle (ADT Bundle) ✓
 - Eclipse + ADT plugin
 - Android SDK Tools
 - Android Platform-tools
 - The latest Android platform
 - The latest Android system image for the emulator

File -> New -> Android Application Project

New Android Application
Creates a new Android Application

Application Name: HelloWorld

Project Name: HelloWorld

Package Name: com.khalad.helloworld

Minimum Required SDK: API 8: Android 2.2 (Froyo)

Target SDK: API 17: Android 4.2 (Jelly Bean)

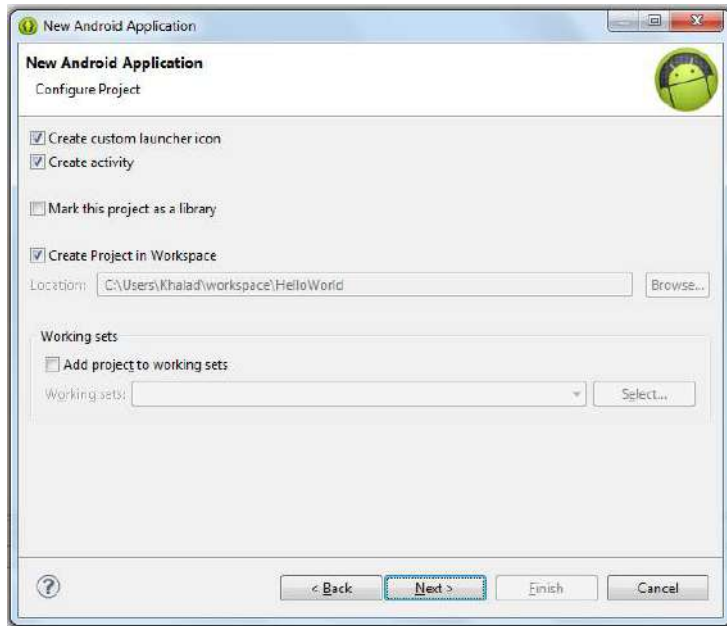
Compile With: API 17: Android 4.2 (Jelly Bean)

Theme: Holo Light with Dark Action Bar

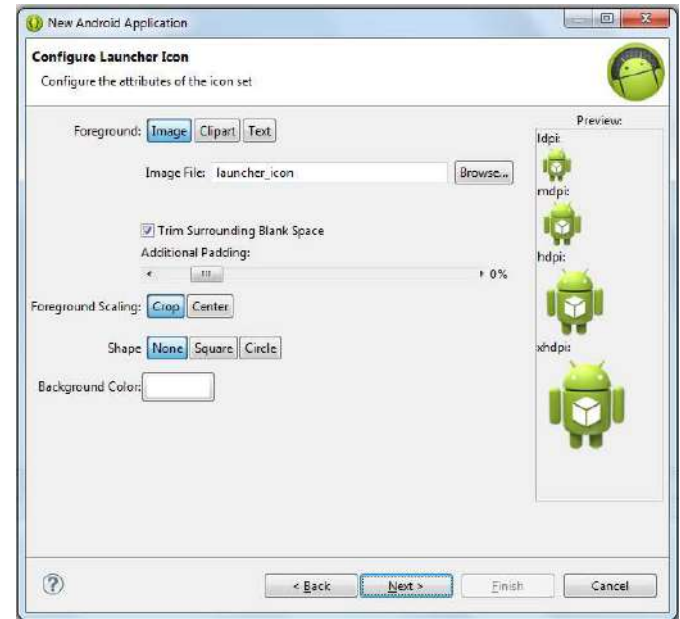
Choose the highest API level that the application is known to work with. This attribute informs the system that you have tested against the target version and the system should not enable any compatibility behaviors to maintain your app's forward-compatibility with the target version. The application is still able to run on older versions (down to minSdkVersion). Your application may look dated if you are not targeting the current

< Back **Next >** Finish Cancel

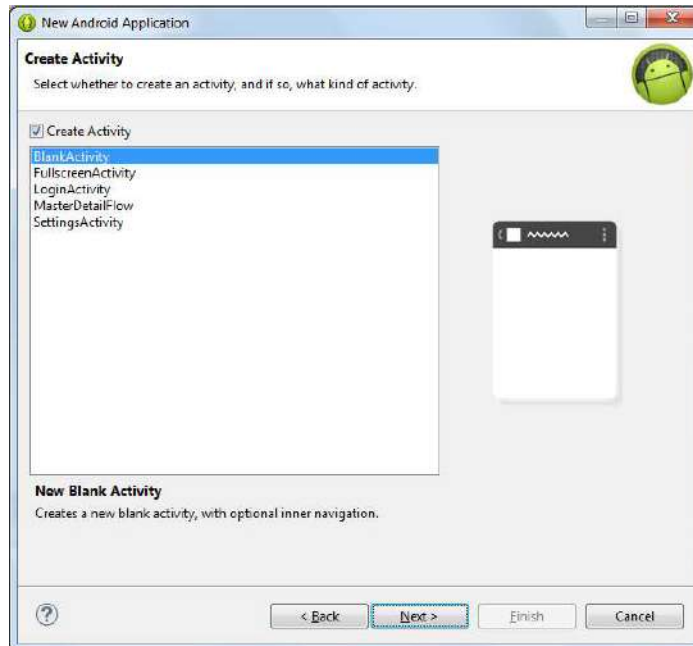
1



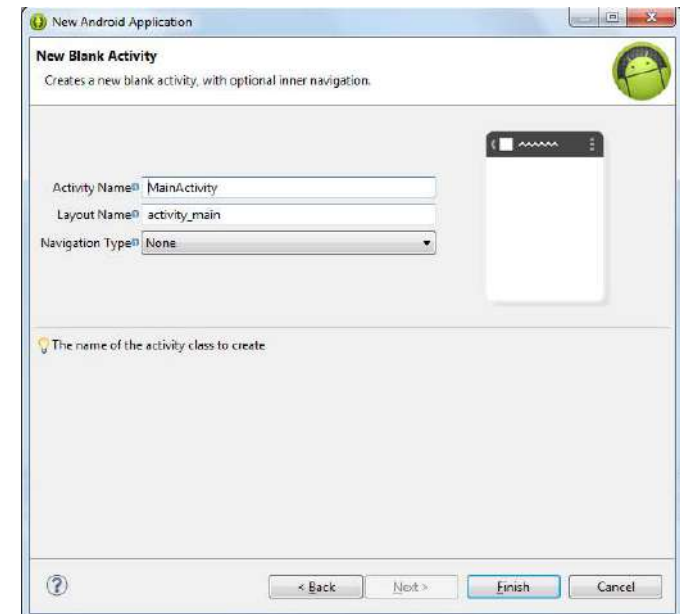
2

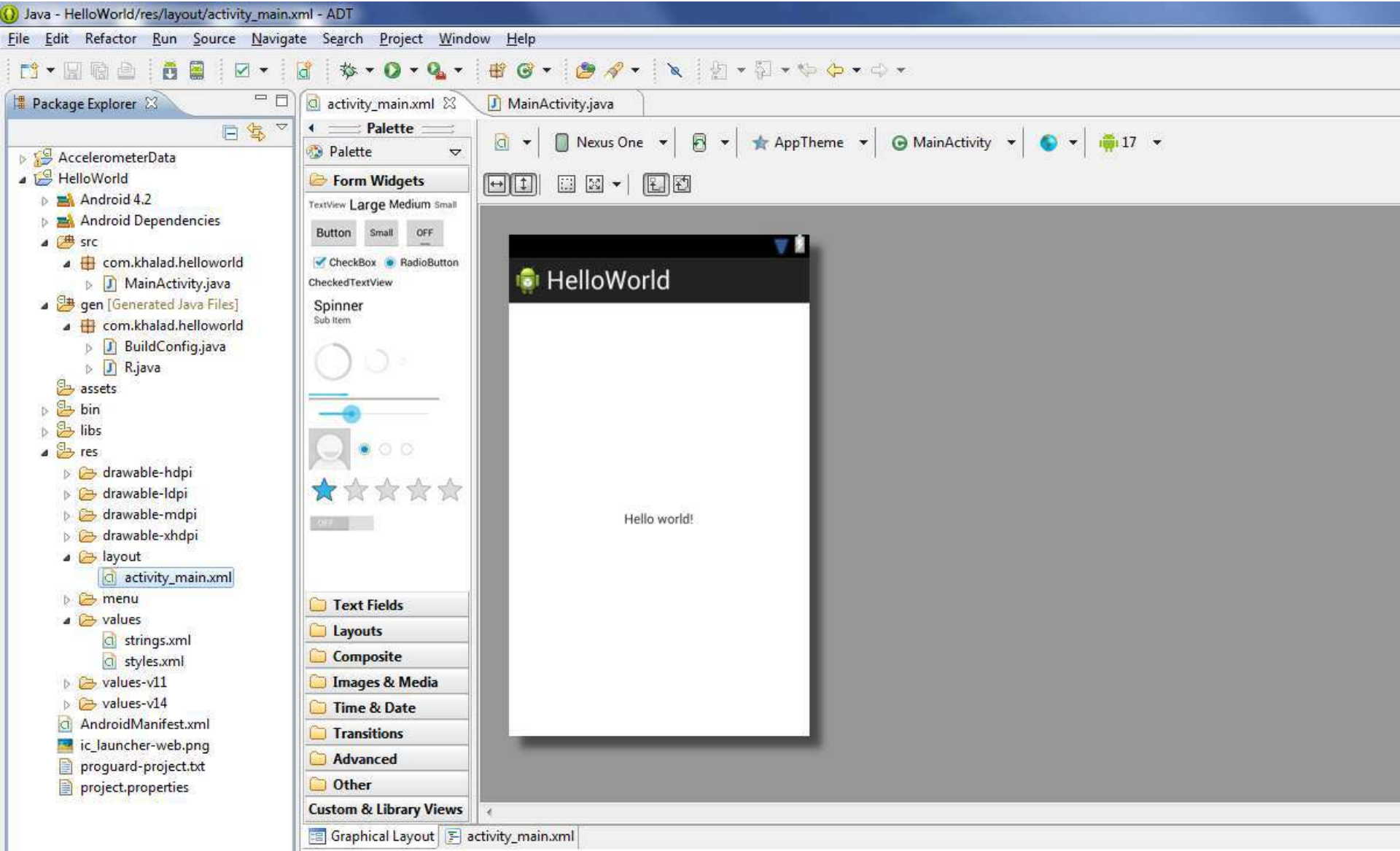


3



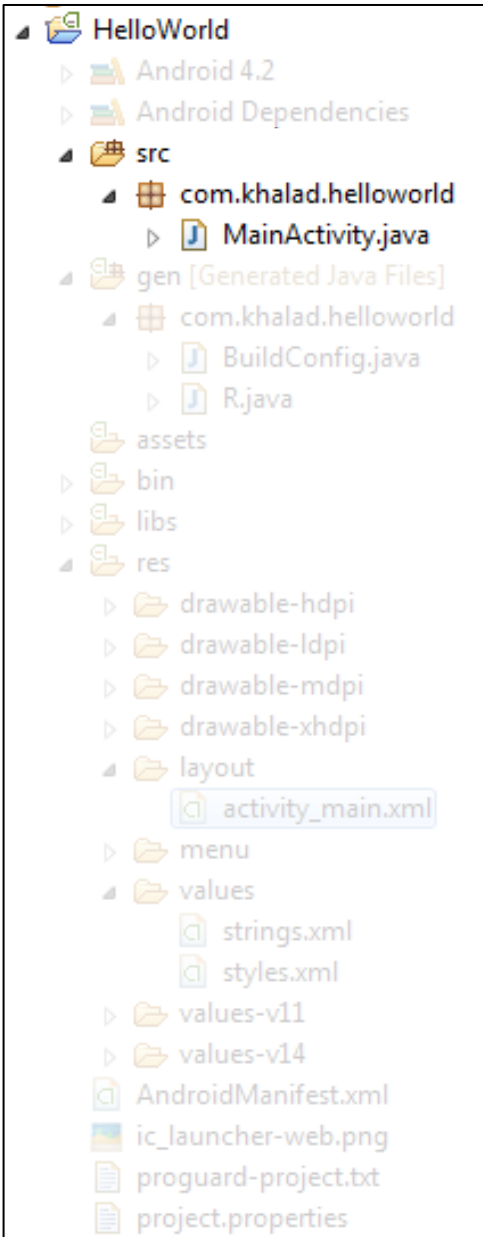
4





Folder Structure For Android Project

Src Folder (Source folder) – contains the java code of the application.



```
package com.khalad.helloworld;

import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

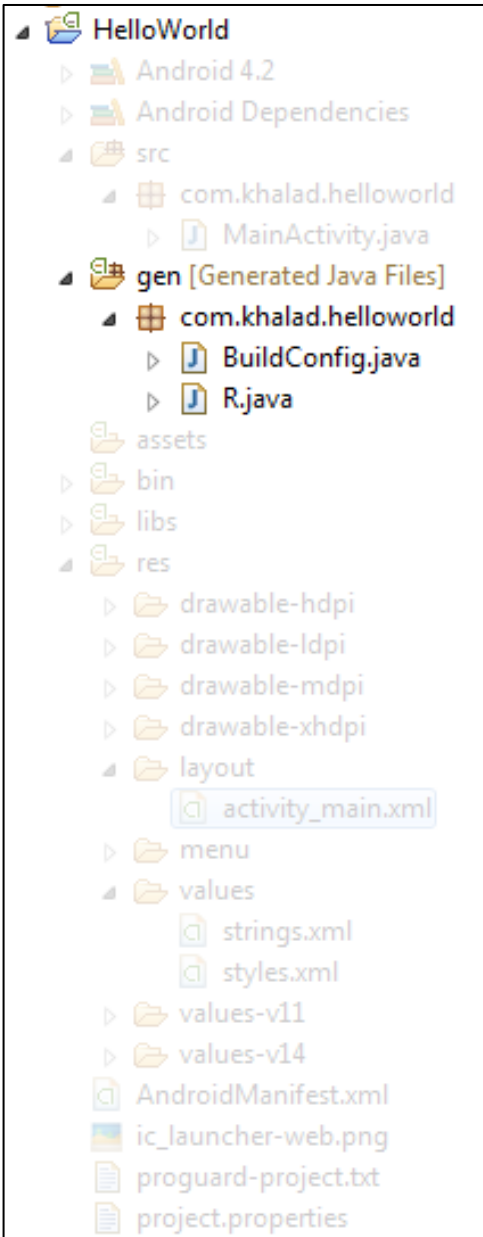
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

Folder Structure For Android Project

gen Folder: contains java files generated by ADT

These files have references to various resources placed in the application.

It contains a special class 'R' which contains all these references.



```
/* AUTO-GENERATED FILE. DO NOT MODIFY.

package com.khalad.helloworld;

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 id {
        public static final int menu_settings=0x7f070000;
    }
    public static final class layout {
        public static final int activity_main=0x7f030000;
    }
    public static final class menu {
        public static final int activity_main=0x7f060000;
    }
    public static final class string {
        public static final int app_name=0x7f040000;
        public static final int hello_world=0x7f040001;
        public static final int menu_settings=0x7f040002;
    }
}
```

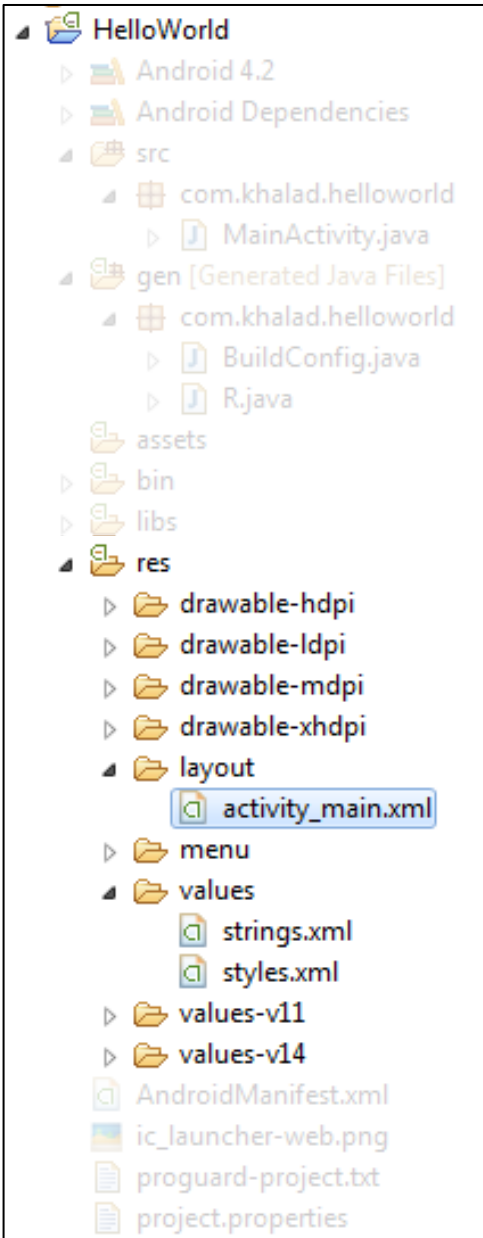

Folder Structure For Android Project

res Folder (Resource folder): Contains application resources, such as drawable files, layout files, and string values

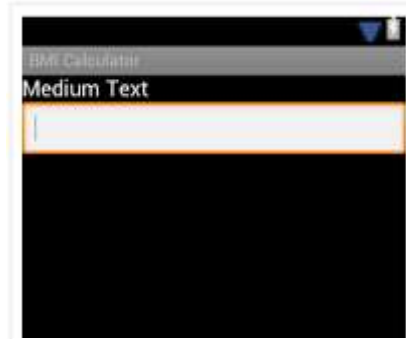
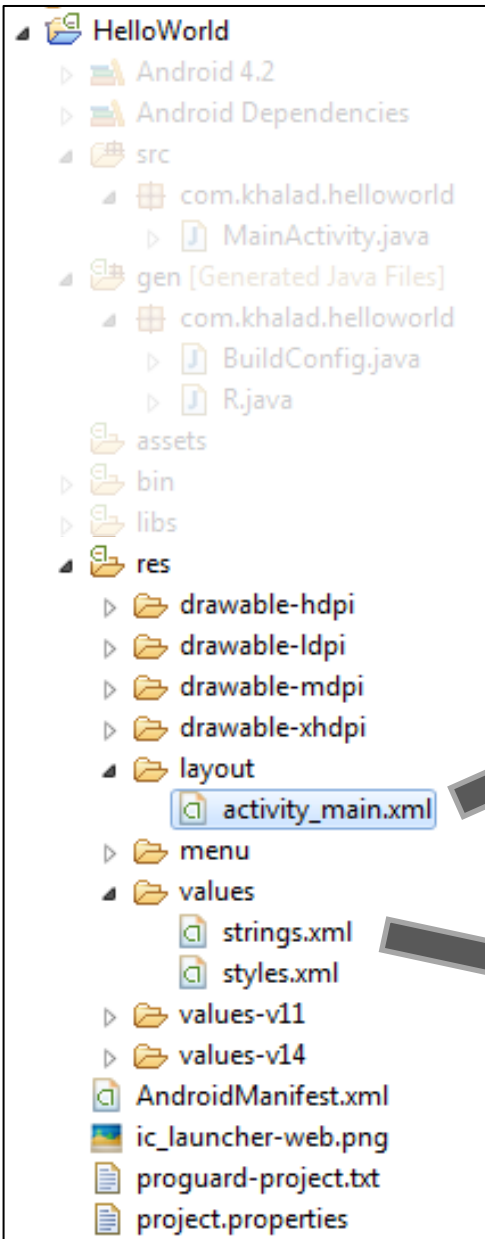
drawable - For bitmap files (PNG, JPEG, or GIF), image files, and XML files that describe Drawable shapes

layout - XML files that are compiled into screen layout

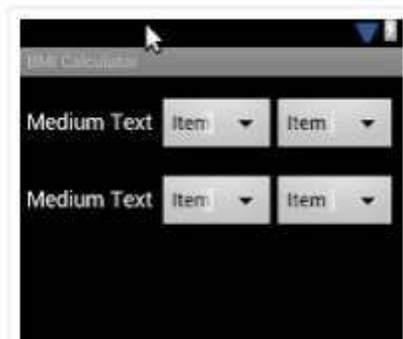
values - XML files that contain simple values, such as strings, integers, and colors



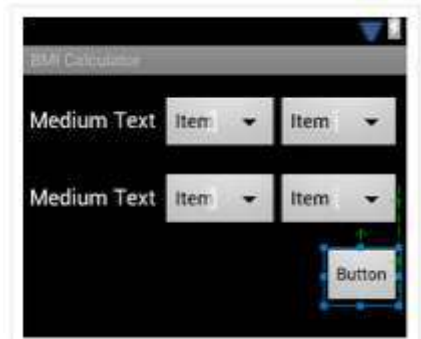
Common Layouts



(Vertical) Linear Layout



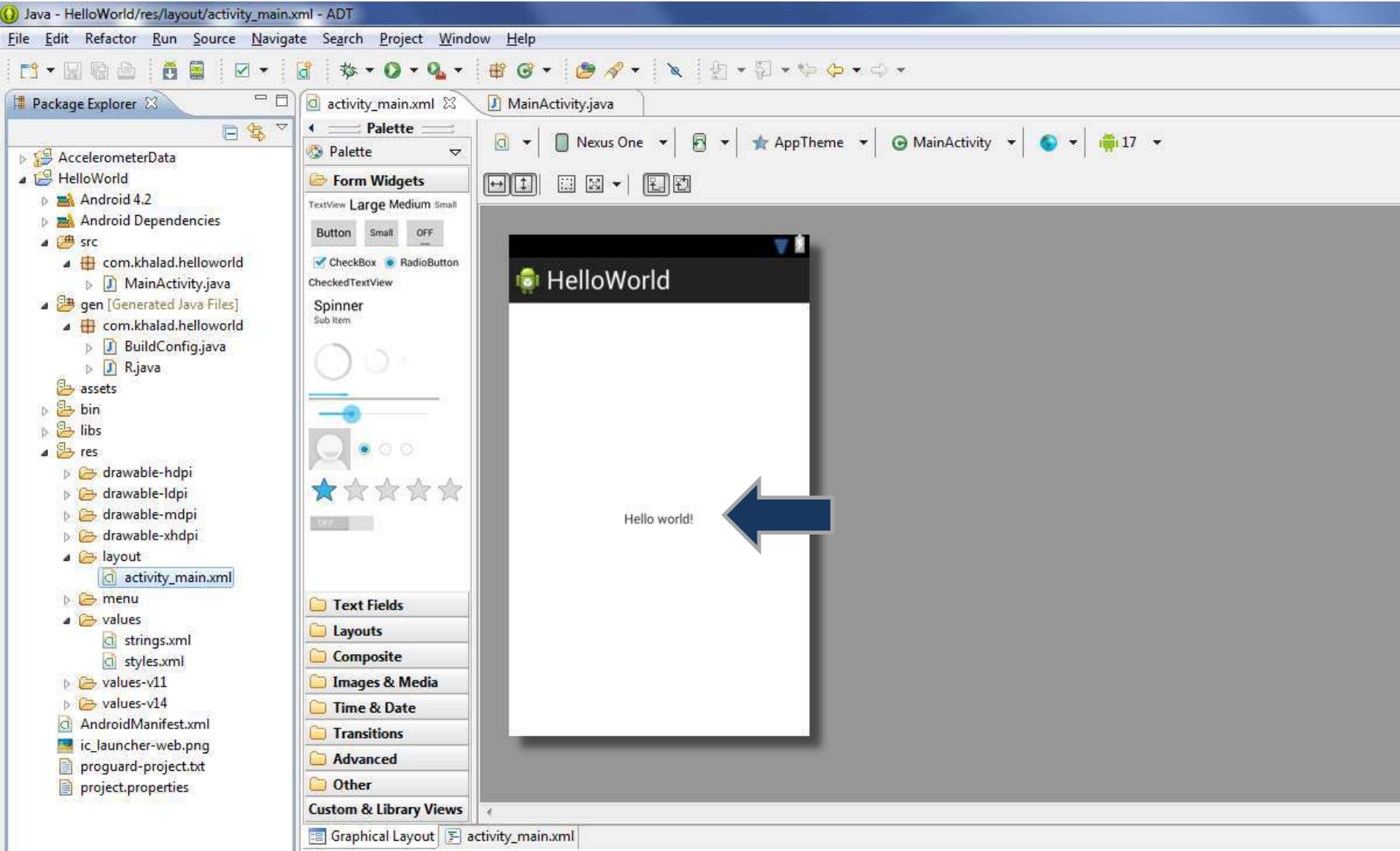
Example of TableLayout (Portrait View)



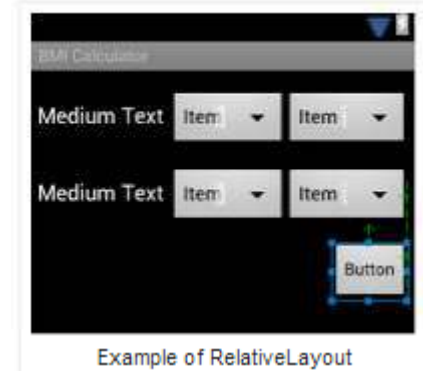
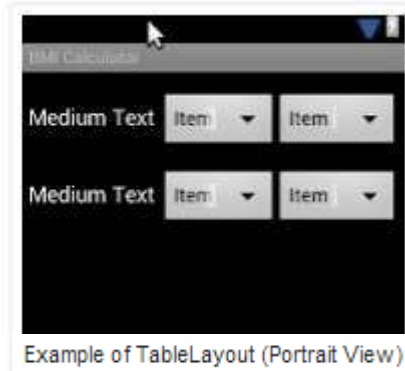
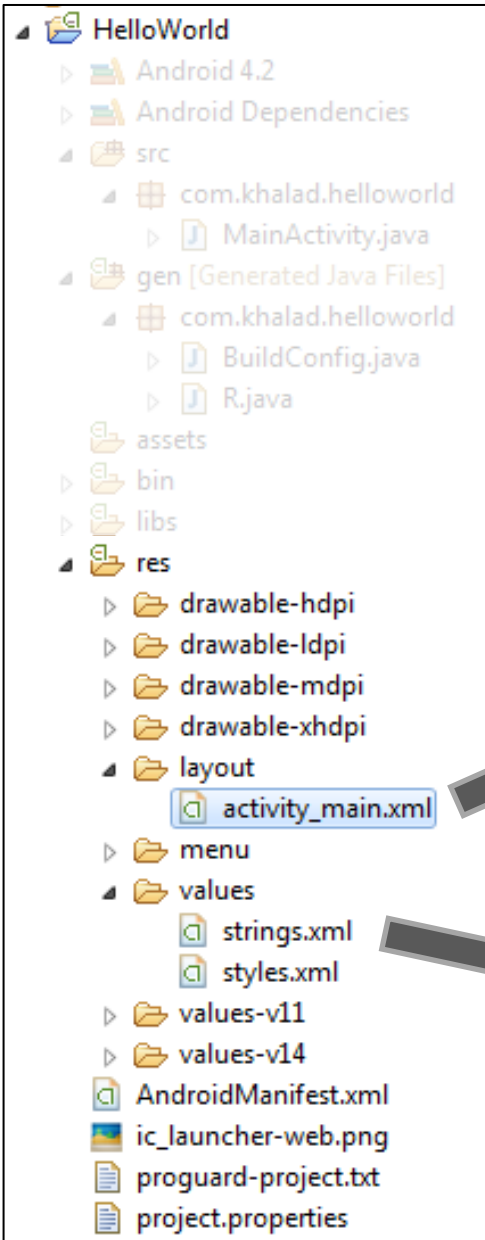
Example of RelativeLayout

```
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"  
    tools:context=".MainActivity" >  
  
    <TextView  
        android:id="@+id/textView1"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="@string/hello_world" />  
  
</RelativeLayout>
```

```
strings.xml ⌕  
  
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  
    <string name="app_name">HelloWorld</string>  
    <string name="action_settings">Settings</string>  
    <string name="hello_world">Hello world!</string>  
  
</resources>
```



Common Layouts



Layouts: <http://developer.android.com/guide/topics/ui/declaring-layout.html#CommonLayouts>

```
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"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
</RelativeLayout>
```

```
strings.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">HelloWorld</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>

</resources>
```

activity_main.xml file

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

```
    <EditText
        android:id="@+id/editText1" android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Username" >
    </EditText>
```

Text field

```
    <EditText
        android:id="@+id/editText2"
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword" />
```

Text field

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

Button

```
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        >
        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Gender"
            android:textAppearance="?android:attr/textAppearanceMedium" />
```

Text view

```
    <RadioButton
        android:id="@+id/radioButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Male" />
```

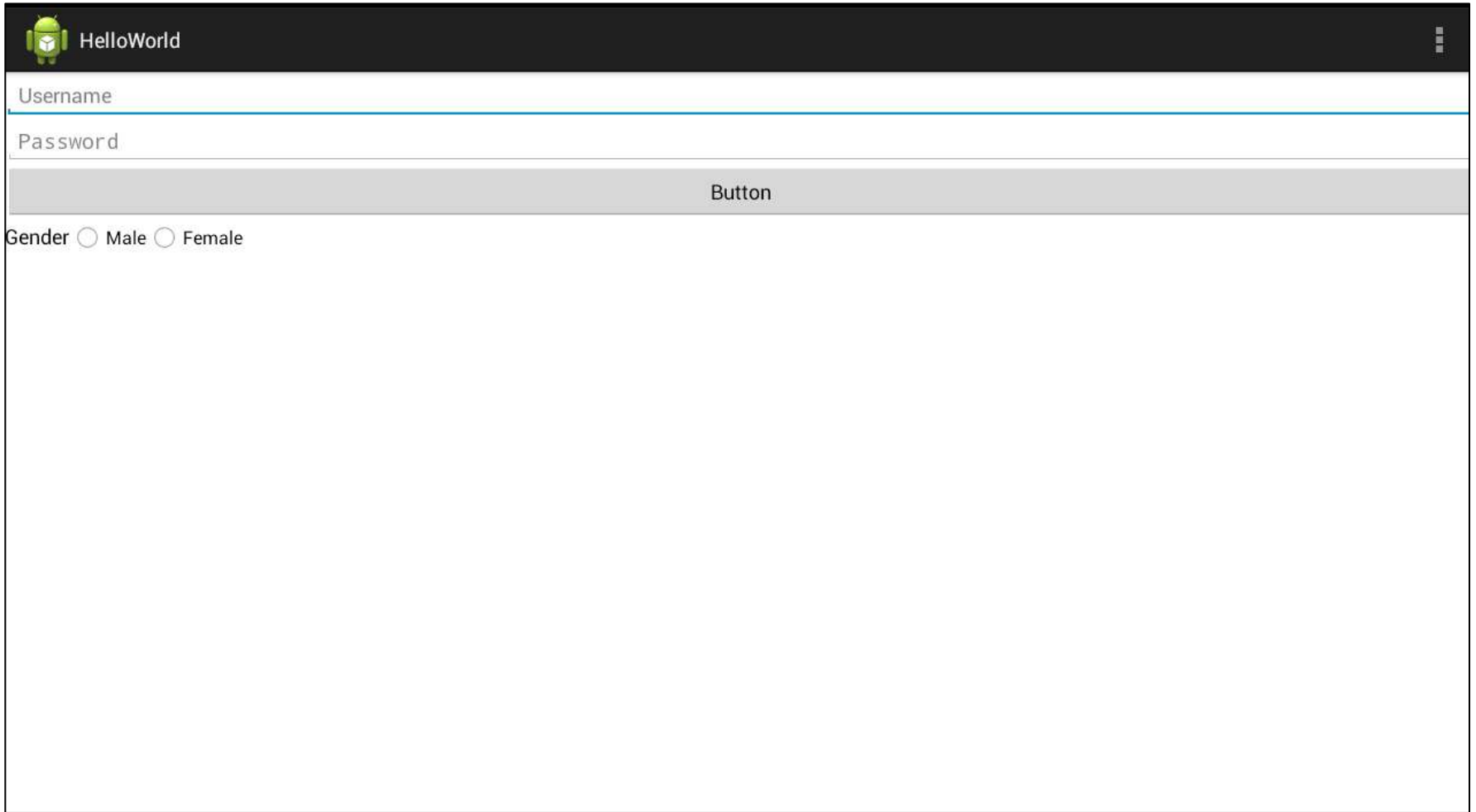
Radio Button

```
    <RadioButton
        android:id="@+id/radioButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Female" />
```

Radio Button

```
    </LinearLayout>
</LinearLayout>
```

Output

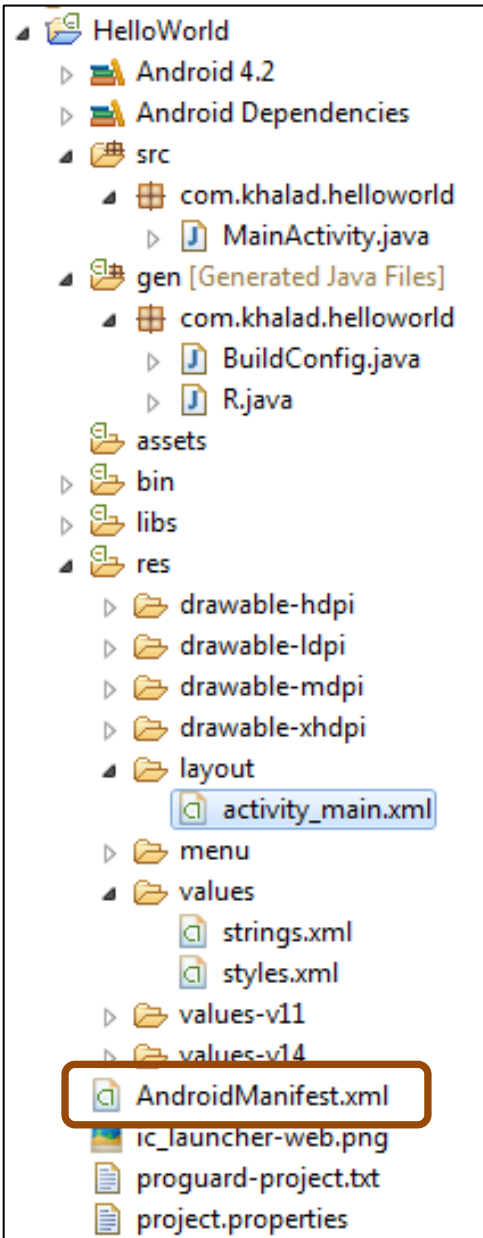


The screenshot displays an Android application interface with a black header bar. On the left of the header is a green Android robot icon, and next to it is the text "HelloWorld". On the right side of the header is a vertical ellipsis menu icon. Below the header, there are three input fields: "Username" (with a light blue border), "Password" (with a light gray border), and a wide gray button labeled "Button". At the bottom, there is a "Gender" label followed by two radio button options: "Male" and "Female".

Folder Structure For Android Project

Manifest file: describe the application

- Declare app's name, version, icon, permission, etc...
- Declare the application's components: activity, service ,receiver or provider



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

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

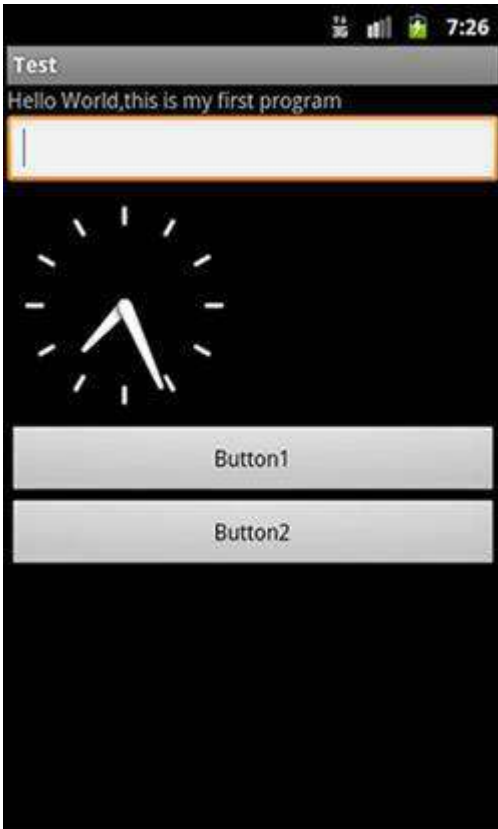
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.khalad.helloworld.MainActivity"
            android:label="@string/app_name"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

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

</manifest>
```


Core Components

Activity



- An *activity* is a user interface screen where visual elements (Views or widgets) can be placed
- In this example, there are five widgets (TextView, EditText, AnalogClock and two Buttons)
- An application might consist of just one activity or several

Core Components

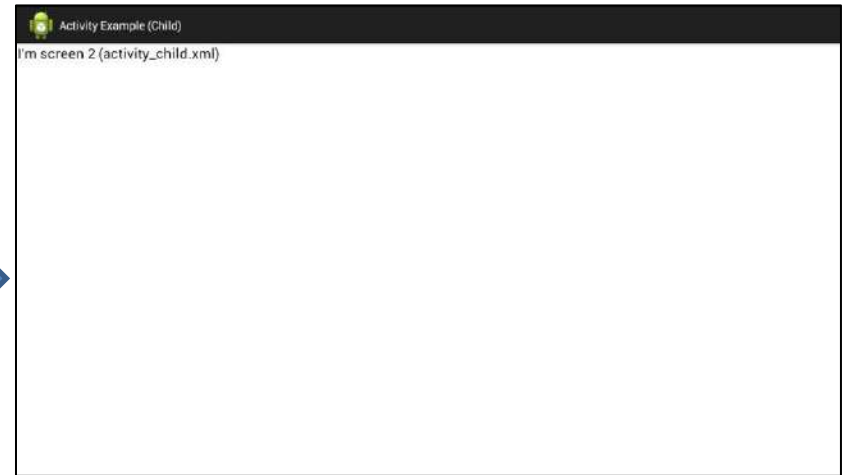
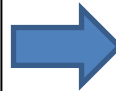
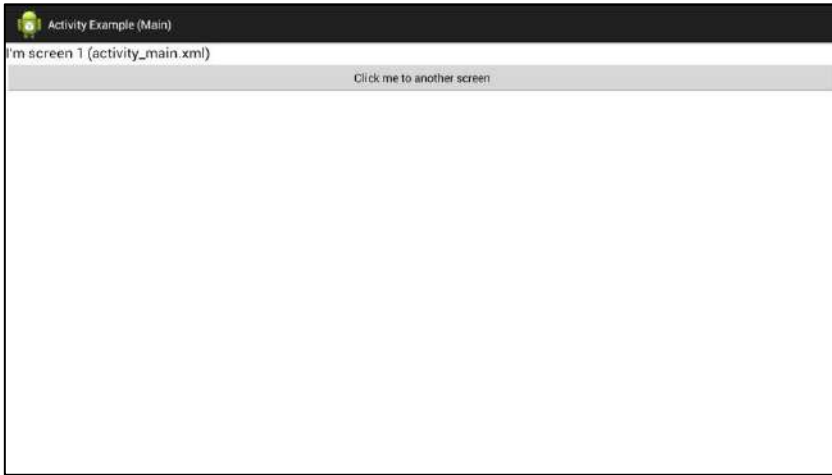
Intent, Service

- *Intent* is a mechanism to describe specific action, such a “Send an email”

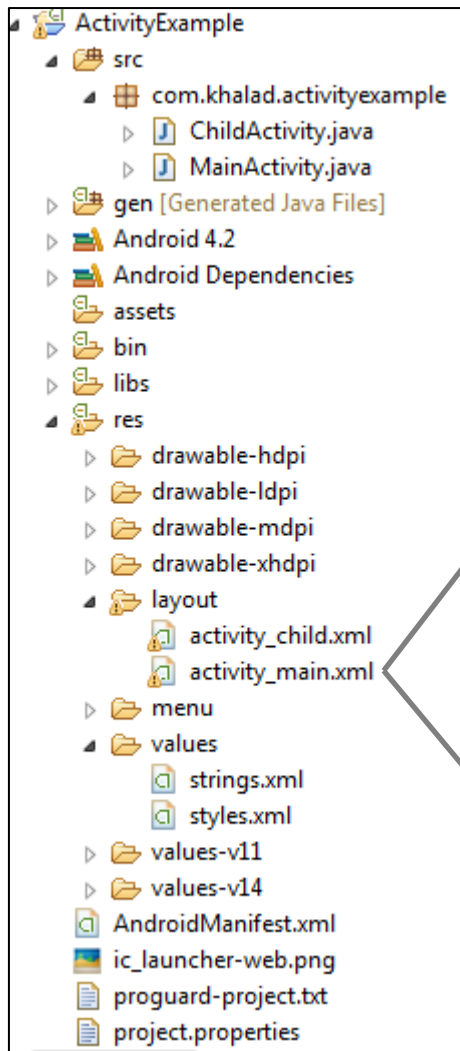


- A *service* is a task that runs in the background without the user’s direct interaction

Activity Example



Activity Example



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/linearlayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I&apos;m screen 1 (activity_main.xml)"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <Button
        android:id="@+id/button1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Click me to another screen" />

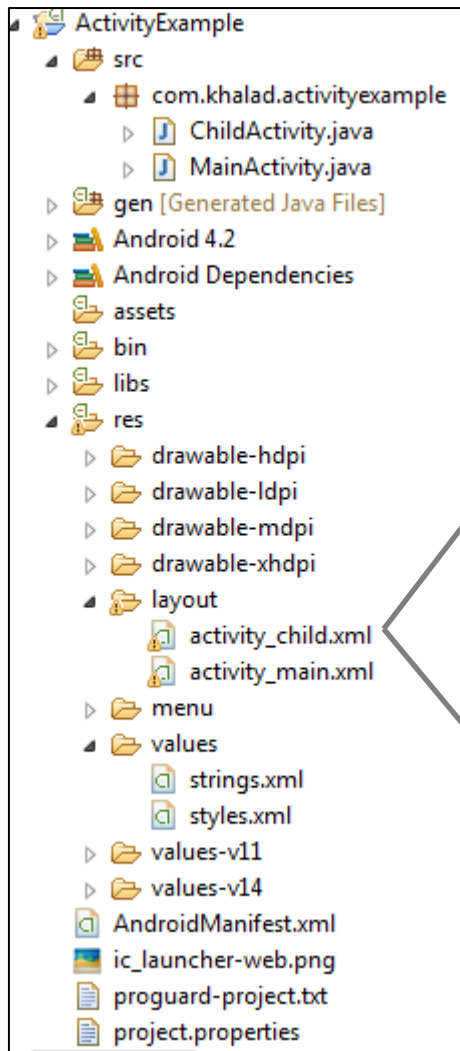
</LinearLayout>
```

Text View

Button



Activity Example



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/LinearLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

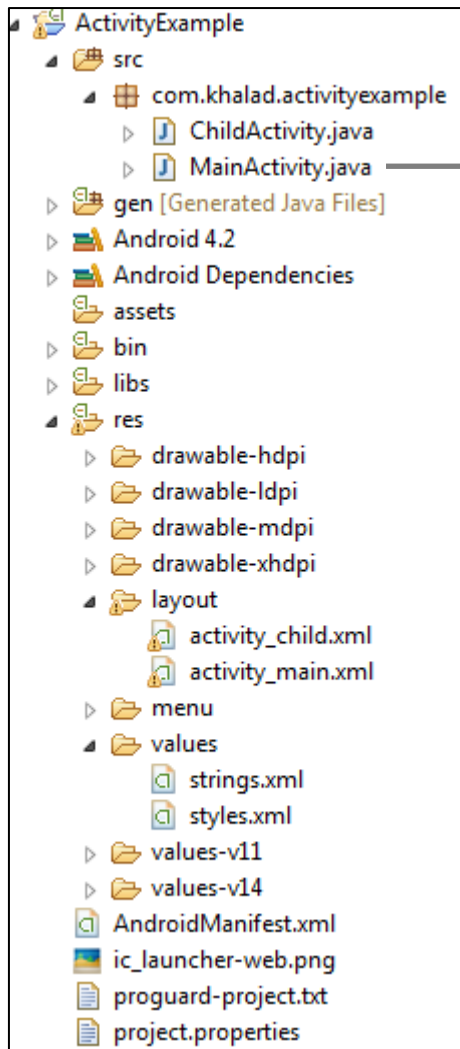
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I&apos;m screen 2 (activity_child.xml)"
        android:textAppearance="?android:attr/textAppearanceLarge" />

</LinearLayout>
```

Text View



Activity Example



```
package com.khalad.activityexample;

import android.app.Activity;

public class MainActivity extends Activity {

    Button button;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        addListenerOnButton();
    }

    public void addListenerOnButton() {

        final Context context = this;
        button = (Button) findViewById(R.id.button1);
        button.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {

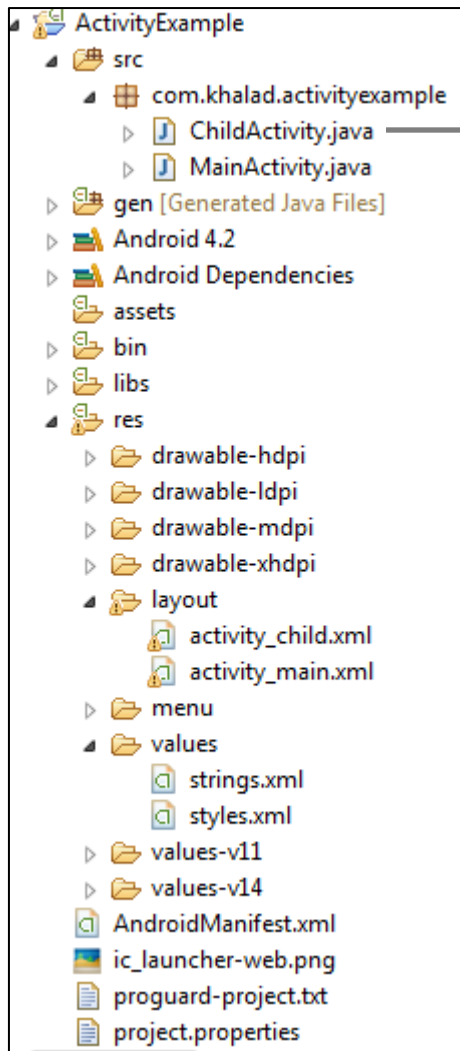
                Intent intent = new Intent(context, ChildActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

Inherit from the activity class

Set layout as describe in activity_main.xml

Find "button1" and set actions

Activity Example



```
package com.khalad.activityexample;

import android.app.Activity;

public class ChildActivity extends Activity {

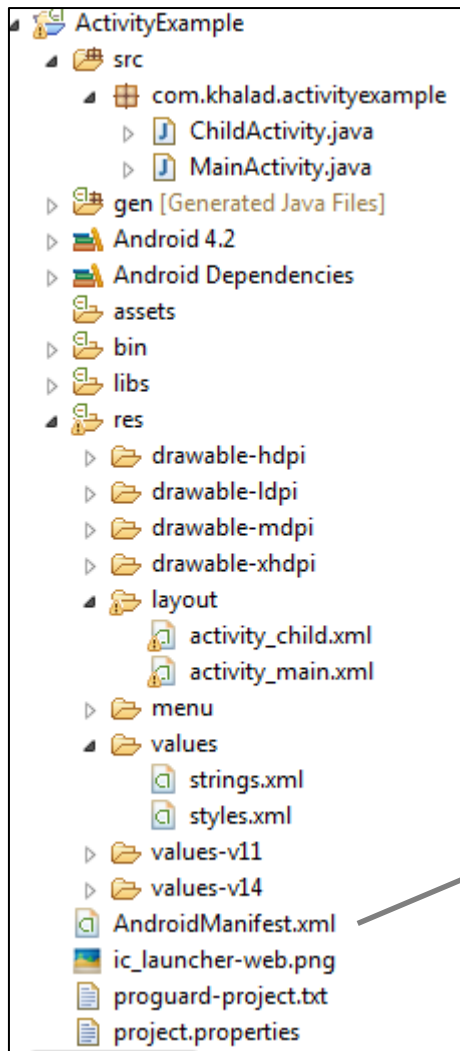
    Button button;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_child);
    }

}
```

Set layout as describe
in *activity_child.xml*

Activity Example



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

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name1"
        android:theme="@style/AppTheme" >

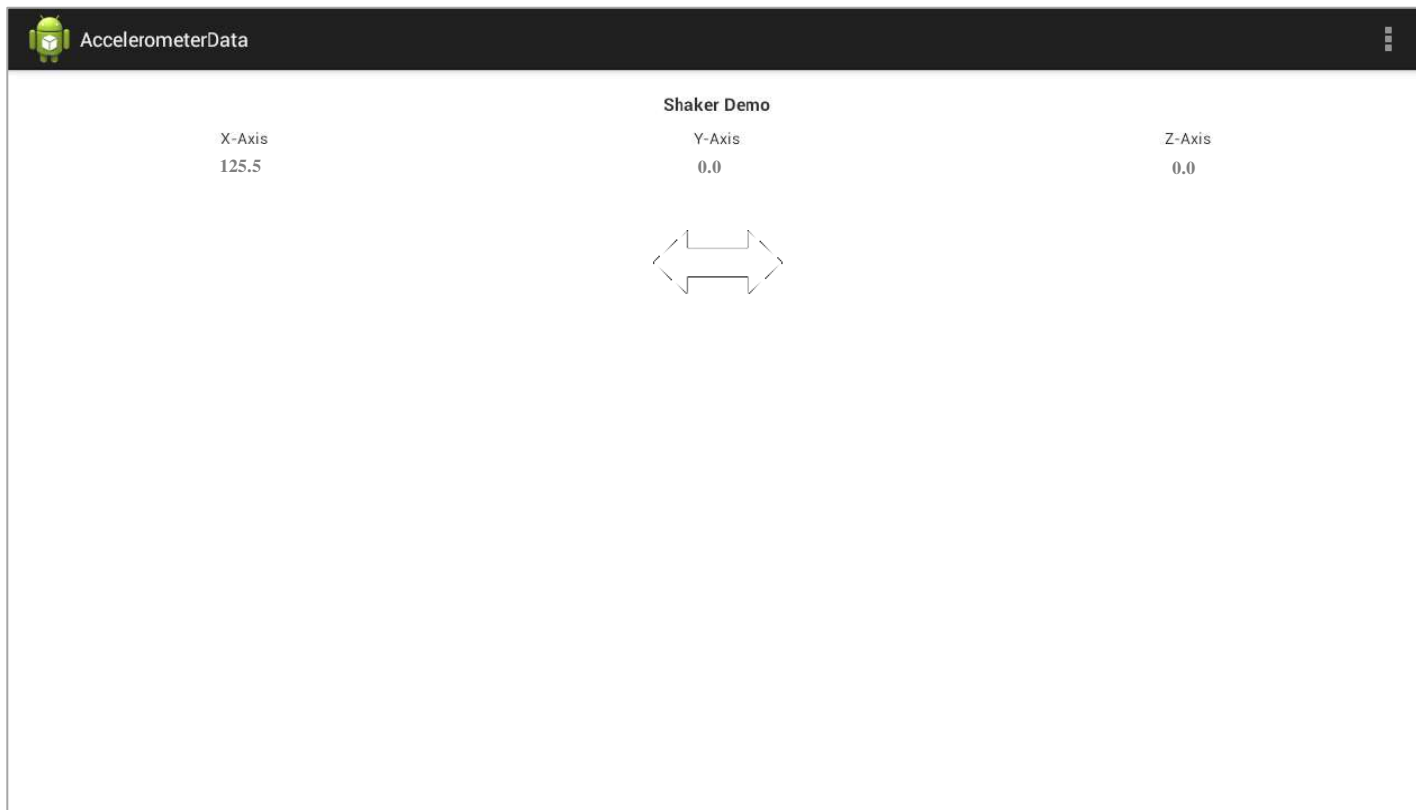
        <activity
            android:name="com.khalad.activityexample.MainActivity"
            android:label="@string/app_name1" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity
            android:label="@string/app_name2"
            android:name="com.khalad.activityexample.ChildActivity" >
        </activity>

    </application>
</manifest>
```

Accelerometer Example

An accelerometer is defined as an instrument for measuring the time rate of change of velocity with respect to magnitude or direction.



7 Text View, 1 Image View

Step 1-> Layout

```
<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:gravity="center"
  android:paddingTop="20dip"
  android:text="Shaker Demo"
  android:textSize="16sp"
  android:textStyle="bold" />
<TableLayout
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:paddingTop="10dip"
  android:stretchColumns="*" >
<TableRow>
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="X-Axis"
    android:textSize="14sp" />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Y-Axis"
    android:textSize="14sp" />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Z-Axis"
    android:textSize="14sp" />
</TableRow>
```

```
<TableRow>
  <TextView
    android:id="@+id/x_axis"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center" />
  <TextView
    android:id="@+id/y_axis"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center" />
  <TextView
    android:id="@+id/z_axis"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center" />
</TableRow>
</TableLayout>
<ImageView
  android:id="@+id/image"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_gravity="center"
  android:paddingTop="15dip"
  android:visibility="invisible" />
</LinearLayout>
```


Step 2 -> Java Main File

```
public class MainActivity extends Activity implements SensorEventListener {

    private float mLastX, mLastY, mLastZ;
    private boolean mInitialized;
    private SensorManager mSensorManager;
    private Sensor mAccelerometer;
    private final float NOISE = (float) 2.0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mInitialized = false;
        mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        mAccelerometer = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        mSensorManager.registerListener(this, mAccelerometer, SensorManager.SENSOR_DELAY_NORMAL);
    }
}
```

SensorManager → access the device's sensor

Get an instance of this class by calling getSystemService()

Step 2 -> Java Main File

```
protected void onResume() {  
    super.onResume();  
    mSensorManager.registerListener(this, mAccelerometer, SensorManager.SENSOR_DELAY_NORMAL);  
}  
  
protected void onPause() {  
    super.onPause();  
    mSensorManager.unregisterListener(this);  
}
```

onResume()

onPaues()

```

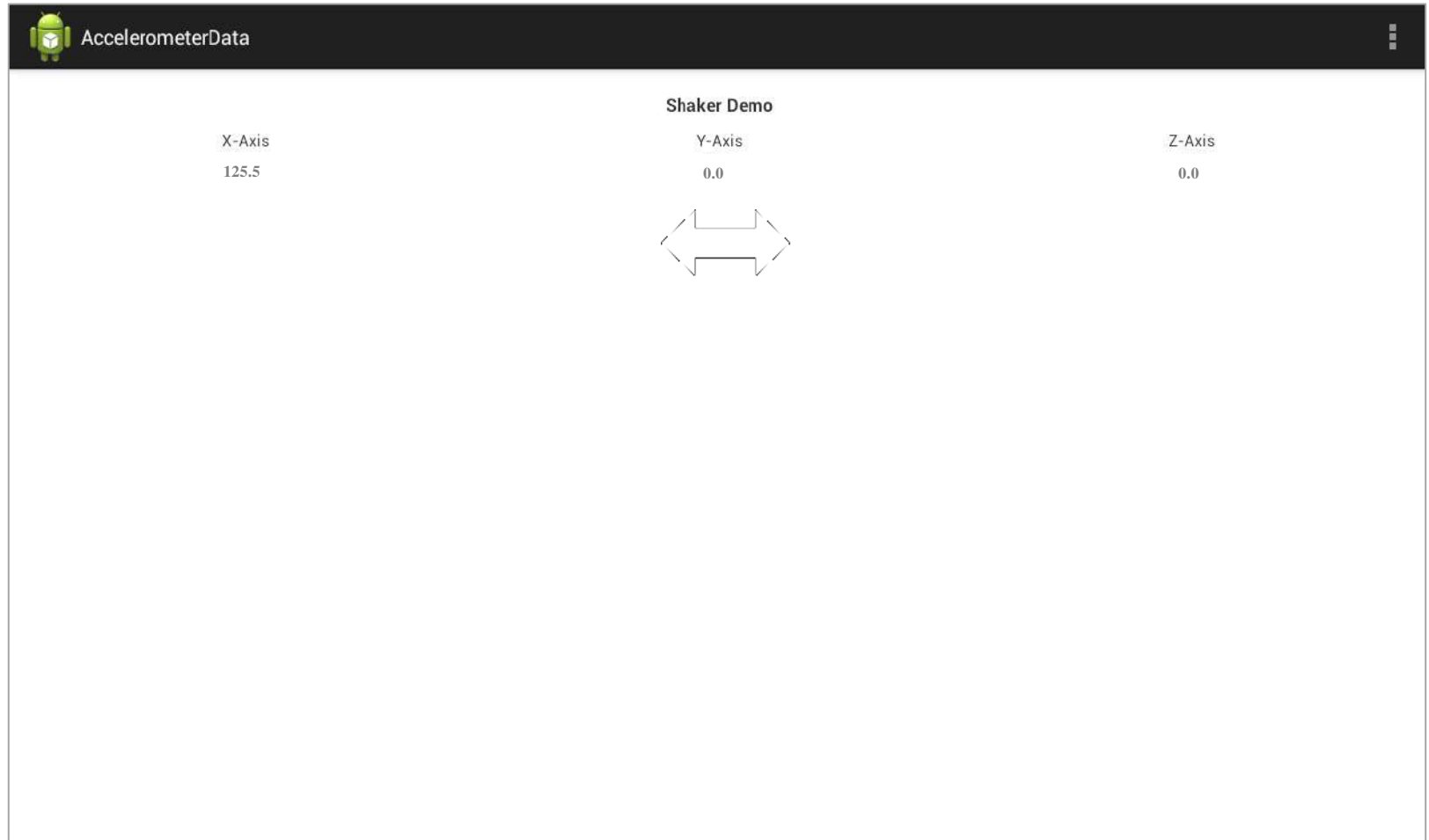
public void onSensorChanged(SensorEvent event)
{
    TextView tvX= (TextView)findViewById(R.id.x_axis);
    TextView tvY= (TextView)findViewById(R.id.y_axis);
    TextView tvZ= (TextView)findViewById(R.id.z_axis);
    ImageView iv = (ImageView)findViewById(R.id.image);
    float x = event.values[0];
    float y = event.values[1];
    float z = event.values[2];
} } Sensor values

    if (!mInitialized) {
        mLastX = x;
        mLastY = y;
        mLastZ = z;
        tvX.setText("0.0");
        tvY.setText("0.0");
        tvZ.setText("0.0");
        mInitialized = true;
    } else {
        float deltaX = Math.abs(mLastX - x);
        float deltaY = Math.abs(mLastY - y);
        float deltaZ = Math.abs(mLastZ - z);
        if (deltaX < NOISE) deltaX = (float)0.0;
        if (deltaY < NOISE) deltaY = (float)0.0;
        if (deltaZ < NOISE) deltaZ = (float)0.0;
        mLastX = x;
        mLastY = y;
        mLastZ = z;
        tvX.setText(Float.toString(deltaX));
        tvY.setText(Float.toString(deltaY));
        tvZ.setText(Float.toString(deltaZ));
    } } Difference between current and last values

    iv.setVisibility(View.VISIBLE);
    if (deltaX > deltaY) {
        iv.setImageResource(R.drawable.horizontal);
    } else if (deltaY > deltaX) {
        iv.setImageResource(R.drawable.vertical);
    } else {
        iv.setVisibility(View.INVISIBLE);
    }
} } Display in Text View
} } Display Image
}

```

Accelerometer Example



Permission

- Manifest update:
 - Add permission

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

```
  <uses-permission  
  android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

```
  <uses-sdk android:minSdkVersion="8" />  
  //...
```

Android Development Tutorial

Human-Computer Interaction (COMP 4020)
Winter 2014