

Lesson 2

Android Workbenches: Android Studio & Eclipse

Victor Matos Cleveland State University

Portions of this page are reproduced from work created and <u>shared by Google</u> and used according to terms described in the <u>Creative Commons 3.0 Attribution License</u>.

CuuDuongThanCong.com

https://fb.com/tailieudientucntt

Android App's Anatomy

Android Applications (Just Apps)

- Android applications are usually created using the Java programming language ^[1]
- Apps must import various **Android Libraries** (such as android.jar, maps.jar, etc) to gain the functionality needed to work inside the Android OS.
- Android apps are made of multiple elements such as: user-defined classes, android jars, third-party libraries, XML files defining the UIs or views, multimedia resources, data assets such as disk files, external arrays and strings, databases, and finally a *Manifest* summarizing the 'anatomy' and permissions requested by the app.
- The various app components are given to the compiler to obtain a single signed and deployable **Android Package** (an **.apk** file).
- Like ".class" files in Java, ".apk" files are the **byte-code** version of the app that finally will be 'executed' by interpretation inside either a **Dalvik Virtual Machine** (DVM) or an Android-Runtime Engine (**ART**).

Android's Byte-Code Execution

Dalvik Virtual Machine vs. Android Runtime (ART)

The **Dalvik Virtual Machine** is a Just-in-Time (**JIT**) runtime environment (similar to the Oracle's Java Virtual Machine JVM) that interprets Android byte-code only when it's needed (however it will be phased out soon).

The newer **ART** (introduced as an option in Android 4.4 KitKat) is an anticipatory or Ahead-of-Time (AOT) environment that compiles code before it is actually needed. **cuu duong than cong . com**

ART promises:

- enhanced performance and battery efficiency,
- improved garbage collection,
- better debugging facilities,
- Improved diagnostic detail in exceptions and crash reports.

Quoting from

<u>https://source.android.com/devices/tech/dalvik/art.html</u> (Aug-27-2014) *Important:* Dalvik must remain the default runtime or you risk breaking your Android implementations and third-party applications.

You are a developer - Which is your SDK audience?

SDKs are named after types of desserts. Available versions at the time of writing are:

1.5	Cupcake,
	D

- 1.6 Donut,
- 2.1 Eclair,
- 2.2 Froyo,
- 2.3 Gingerbread,
- 3.x Honeycomb,
- 4.0 Ice Cream Sandwich
- 4.3 Jelly Bean
- 4.4 Kitkat
- 5.x Lollipop
- 6.X Marshmallow

Android SDK version	Current market share
4.4 (KitKat)	42.0 %
4.1-4.3 (Jelly Bean)	34.4 %
5.0-5.1 (Lollipop)	16.5 %
2.3 (Gingerbread)	3.5 %
4.0.x (ICS)	3.3 %
2.2 (Froyo)	0.2 %
3.0-3.2 (Honeycomb)	0.1 %
2.0-2.1 (Eclair)	0.0 %

Statistics accessed on Sept 3, 2015 from AppBrain at http://www.appbrain.com/stats/top-android-sdk-versions

Development Workbenches

Android apps are made out of many components. The use of an IDE is *strongly* suggested to assist the developer in creating an Android solution. There are various options including:

Tools for Constructing Android Apps

- **Eclipse+ADT.** The classic general purpose Eclipse IDE can be enhanced (with the ADT plugin) to provide a 'conventional' way to create and debug Android Apps. The associated SDK Manager allows you to reach the various API libraries needed by the apps.
- Android Studio is a new Android-only development environment based on IntelliJ IDEA. It is the 'preferred' IDE
- **Netbeans+Android.** Similar to Eclipse+ADT. Soon to be deprecated(?)





Eclipse + ADT + SDK

Typical Layout of the Eclipse-ADT IDE for Android Development



Note: The DDMS and Hierarchy View can be manually added by the user to Eclipse's tool bar

Android Studio



Setting up Android Studio

Downloading Android Studio IDE

Download IDE from: <u>https://developer.android.com/sdk/index.html</u> Run the executable, you are (*almost*) done!



Prepare your computer – Install SDK: Windows, Mac, Linux

We assume you have already installed the most recent Java JDK and Eclipse IDE in your computer

- Java JDK is available at: <u>http://www.oracle.com/technetwork/java/javase/downloads/index.html</u>
- Eclipse IDE for Java EE Developers is available at: <u>http://www.eclipse.org/downloads/</u>



The next instructions are given to:

(a) User wanting to add a newer SDK to their existing collection,

(b) First time users (who may or not be Eclipse users).

(a) Users Wanting to Update an Older Android Workbench



If you are currently using the Android SDK, you just need to *update* to the latest tools or platform using the already installed *Android SDK Manager*.

- 1. Click on the DSDK Manager icon.
- 2. You will see a form similar to the one on the right.
- Select the SDK packages and independent components you want to install (click 'Install' CUU du button and wait until they are setup in your machine...)

Android SDK Manager						
Packages Tools						
SDK Path: C:\Users\1002125\AppData\Local\Android\android-sd	k					
Destance						
Packages						
📫 Name	API	Rev.	Status			
🔺 🔲 🧰 Tools						
🔲 📌 Android SDK Tools		22.0.5	👼 Installed			
Android SDK Platform-tools		18.0.1	👼 Installed			
🔲 📌 Android SDK Build-tools		18.0.1	👼 Installed 📃			
🔲 差 Android SDK Build-tools		17	👼 Installed 🦷			
Android 4.3 (API 18)						
Documentation for Android SDK	18	1	🔯 Installed			
🔲 👘 SDK Platform	18	1	큕 Installed			
Samples for SDK	18	1	Tinstalled			
🔄 📑 ARM EABI v7a System Image	18	2	Tinstelled			
Intel x86 Atom System Image	18	1	Not installed			
Google APIs	18	2	N Installed			
Sources for Android SDK	18	1	🐼 Installed			
▶						
Android 4.1.2 (API 16)						
Android 4.0.3 (API 15)						
Android 4.0 (API14)						
Android 3.2 (API13)						
Android 3.1 (API 12)						
		1				
Show: Vpdates/New VInstalled Obsolete Select N	ew or <u>Up</u>	<u>dates</u>	Install 1 package			
Sort by: API level Repository	All		Delete packages			
			- EM			
Done loading packages.						

(b) First Time Android Users who have Eclipse already installed

- Obtain the appropriate (Windows, Max, Linux) Stand-alone SDK Tools for Windows from the page <u>http://developer.android.com/sdk/index.html</u> Execute the program, *remember the folder's name and location* in which the SDK is stored, you will have to supply this path to Eclipse.
- 2. Install the **ADT Plugin** for Eclipse (it must be already available in your machine)
 - 1. Start Eclipse, then select Help > Install New Software....
 - 2. Click Add button (top-right corner)
 - 3. In the next dialog-box enter "ADT Plugin" for the *Name* and the following URL for the *Location*: https://dl-ssl.google.com/android/eclipse/
 - 4. Click **OK**
 - 5. Select the checkbox next to **Developer Tools** and click **Next** > **Next**
 - 6. Accept the license agreements, then click Finish.
 - 7. After the installation end you need to restart Eclipse.
- 3. Add **Android platforms** and other components to your SDK (see previous option (a))

Configure the ADT Plugin

4. The next step is to inform your Eclipse+ADT workbench of the **android-sdk** directory's location (this is the path you saved on Step1)

- In Eclipse, select Window > Preferences... to open the Preferences panel (Mac OS X: Eclipse > Preferences).
- 2. Select Android from the left panel.
- To set the box SDK Location that appears in the main panel, click Browse... and locate your downloaded SDK directory (usually C:\Program Files (x86)\Android\android-sdk)
- 4. Click **Apply**, then **OK**.

Done! cuu duong than cong . com

Working with Virtual Devices (AVDs)



Ideally you should test your applications on a device (a physical phone or tablet). However, the SDK allows you to create realistic virtual devices on which your applications could be executed/debugged before they are deployed on actual hardware.

Ì	🛱 Android Virtual Device (AVD) Manager								
	Android Virtual Device Definitions UONG than cong. com								
	List of existing Android Virtual Devices located at C:\Users\1002125\.android\avd								
	AVD Name	Target Name	Platform	API L	CPU/ABI	Create			
	API 10Gingerbread 233	Android 2.3.3	2.3.3	10	Intel Atom (x86)	Start			
	API15-Icecream403	Android 4.0.3	4.0.3	15	Intel Atom (x86)	Edit			
	API16-JellyBean-412	Android 4.1.2 ng tha	4.1.2	16	Intel Atom (x86)	Repair			
	Api 19-Kitkat-442-ARM	Android 4.4.2	4.4.2	19	ARM (armeabi-v7a)	Delete			
	API 19-Kitkat 442-Intel	Google APIs (x86 System Image) (Google Inc.)	4.4.2	19	Intel Atom (x86)	Details			
	▲ Refresh								
	🖄 A repairable Android Virtual Device. 💥 An Android Virtual Device that failed to load. Click 'Details' to see the error.								
_	CuuDuongThanCong.com				https://fb.com/tailieudientucntt				

Creating a Virtual Device (AVD)

An AVD allows you to simulated devices and prototype your solution on a variety of SDKs. To create a virtual unit follow the next steps:

- Click on the AVD Manager > Create. The Create New AVD wizard appears requesting your input.
- 2. Type the name of the emulator, enter a value such as "API19-Kitkat-442-Intel" (see figure on the right)
- Select from the drop-downlist a Device (Nexus 4...) and CPU/ABI such as Intel Atom (x86)
- 4. Choose a target from the already installed SDKs (eg. "Android 4.4.2 API Level19").
- 5. Tick the *Keyboard* box to enable your PC's keyboard.
- 6. Choose a skin of your preference (...dynamic hard ...)
- 7. Set memory RAM to no more than 768 MB
- 8. Indicate how much internal storage the simulator will use (200MB).
- 9. Add a small SD card (9MB)
- 9. Click **OK** to create the **AVD**.

💮 Edit Android Vi	rtual Device (AVD)	×	
AVD Name:	API 19-Kitkat-442-Intel		
Device:	Nexus 4 (4.7", 768 × 1280: xhdpi)	•	
Target:	Android 4.4.2 - API Level 19	•	
CPU/ABI:	Intel Atom (x86)	•	
Keyboard:	Hardware keyboard present		
Skin:	Skin with dynamic hardware controls	•	
Front Camera:	None	•	
Back Camera:	None	•	
Memory Options:	RAM: 768 VM Heap: 64		
Internal Storage:	200 MiB	•	
SD Card:	Size: 9 MiB Brows	;e,,,	
Emulation Options:	Snapshot Use Host GPU		
Override the ex	isting AVD with the same name		
	OK Can	cel	





Creating a Virtual Device (AVD)

公)

T A S D F G H ∲ Z X C V B

QWER

36 2 1:39

Some examples:

5554:IceCream4x

Q Google

0

Camera



(::::)

5554:Honeycomb3x 3:01 Clack Calculato Q à Email Gallery Places 5554:Nexus-S iii 👔 💈 5:23 Android 5:23 Friday, March 9 🔁 Charging (50%) Tablet showing Honeycomb 3.x A (1) Gingerbread 2.3 running on a custom skin for Nexus-S. See page: http://heikobehrens.net/2011/03/15/android-skins/





Setting up Android Studio

Working with Virtual Devices (AVDs)

	Virtual Device Manager						<u></u> [
2	Your Virtual	Devices					
~	Android Stadio						
ype	Name	Resolution	API	Target	CPU/ABI	Size on Disk	Actions 🔻
	API 19 Kitkat 442	768 × 1280: xhdpi	19	Google APIs (x86 System Image)	x86	530 MB	► A -
	Generic_5in WVGA API	480 × 800: mdpi	22	Google APIs	x86_64	1 GB	▶ ∥ ▼
	Nexus 4 API 22	768 × 1280: xhdpi	22	Google APIs	x86_64	1 GB	► / -
	Nexus 5 API 22	1080 × 1920: xxhdpi	22	Google APIs	x86_64	1 GB	▶ <i>P</i> +
	Nexus 5 API 22	1080 × 1920: xxhdpi	22	Google APIs	x86_64	1 GB	► 2

The Android Studio process to create, edit, remove, and execute AVDs is similar to the strategy already discussed for Eclipse-ADT (only cosmetic differences on the GUI)

Example of an AVD Emulator wearing a HVGA Skin



AVD – Emulator wearing: Skin with dynamic hardw. controls

Numeric ID: 5554:17-IceCream 5554 ³⁶ 6:48 **Basic Controls** 5 (1) APPS WIDGETS Hardware Buttons 5 Q MENU -**API Demos** Calculator Calendar Browser Hardware Keyboard Use your physical keyboard to provide input -0 Clock **Dev Settings** Custom Camera Locale **Dev Tools** Downloads Email Gallery OQ. 9 Gestures Google Local Maps Builder Settings Messaging Music Navigation People CuuDuong Than Cong.com nttps://fb.com/tailieudientucntt

Controlling the AVD Operations

Keyboard	OS function
Escape	Back button
Home	Home button
F2, PageUp	Menu (Soft-Left) button
Shift-F2, PageDown	Start (Soft-Right) button
F3	Call/Dial button
F4	Hangup / EndCall button
F5	Search button
F7	Power button
Ctrl-F3, Ctrl-KEYPAD_5	Camera button
Ctrl-F5, KEYPAD_PLUS	Volume up button
Ctrl-F6, KEYPAD_MINUS	Volume down button
KEYPAD_5	DPad center
KEYPAD_4	DPad left
KEYPAD_6	DPad right
KEYPAD_8	DPad up
KEYPAD_2	DPad down
F8	toggle cell network on/off
F9	toggle code profiling
Alt-ENTER	toggle FullScreen mode
Ctrl-T	toggle trackball mode
Ctrl-F11, KEYPAD_7	switch to previous layout
Ctrl-F12, KEYPAD_9	switch to next layout

Controlling an Android Emulator through *your computer's* keyboard

Note: Keypad keys only work when *NumLock* is deactivated.



AVD – Emulator : Disk Images

Working with Emulator Disk Images

CuuDuongThanCong.com

- The Android simulator uses QEMU technology [Website: <u>www.qemu.org</u>]
- QEMU is an open source machine emulator which allows the operating system and programs made for one machine (e.g. an ARM CPU) run efficiently on a different machine (e.g. your Windows PC).



https://fb.com/tailieudientucntt

H

Upload/download Data, Music and Picture files to the Emulator's SDcard

Eclipse developers needs to add the DDMS perspective.

- Android-Studio uses the equivalent
 'Android Device Monitor' button.
- Change to the DDMS perspective. Make sure your AVD has started (You will see a layout similar to the figure on the lower right side)
- 4. Click on the File Explorer tab.
- 5. Expand the **mnt** (mounted devices) folder.
- 6. Expand the sdcard folder duong th
- 7. Open your Window's Explorer.
- Choose a file stored in your PC. Transfer a copy to the emulator by dragging and dropping it on top of the sdcard folder.



🐉 Java

Open Perspective

🔊 DDMS

🎋 Debug

Upload/download Data, Music and Picture files to the Emulator's SDcard



- -

Upload/download Data, Music and Picture files to the Emulator's SDcard

8. Return to the emulator. This time you may use native apps such as 'Music' and 'Gallery' to see your recently uploaded multimedia files. For instance...



Upload/download Data, Music and Picture files to the Emulator's SDcard

9. Pictures may be displayed by clicking the *Application Pad* and invoking the **Gallery** application



Locate your 'android-sdk' & AVD folder

After you complete your setup look for the following two subdirectories in your PC's file system

C:\Program Files (x86)\Android\android-sdk add-ons C:\Users\yourWindowsUserName\.android\avd build-tools extras API10Gingerbread233.avd platforms API15-Icecream403.avd platform-tools API16-JellyBean-412.avd samples Api19-Kitkat-442-ARM.avd sources API 19-Kitkat 442-Intel.avd system-images API 10Gingerbread 233 temp API15-Icecream403 tools API 16-JellyBean-412 AVD Manager Api19-Kitkat-442-ARM SDK Manager API 19-Kitkat 442-Intel SDK Readme uninstall

This folder contains your Android SDK, tools, and platforms CuuDuongThanCong.com This directory holds your Virtual Devices (AVDs)

Example 2.1 : HelloWorld App

We will use **Android Studio IDE** to create a bare bone app.

Click on the entry: 'Start new Android Studio Project'.

A wizard will guide you providing a sequence of menu driven selections.

The final product is the skeleton of your Android app.



Android apps are usually made of a rich collection of various type of components including Java code, multimedia resources, XML files, etc. The *New Android Studio Project* Wizard facilitates the assembly of those parts and organizes the components in various sub-directories.

- 1. Enter in the *Application Name* box: HelloApp
- Enter Company Domain: csu.matos

 (usually a dot-separated string consisting of company and programmer's name) uu duong than
- 3. Click Next



- Select Target Android Device. In this example Phone and Table is already checked. Other options are: Wear, TV, Auto, Glasses.
- Choose from drop-down list the Minimum SDK on which the app will han work. In this example we have selected: API22 Android 5.1 (Lollipod)
- 6. Click **Next**

Cancel the form factors your app will run on Different platforms may require separate SDKs Phone and Tablet Minimum SDK API 22: Android 5.1 (Lollpop) Coogle Play Store. Help me choose Minimum SDK API 21: Android 5.0 (Lollpop) Minimum SDK API 21: Android 5.0 (Lollpop) Minimum SDK API 21: Android 5.0 (Lollpop) Iminum SDK API 21: Android 5.0 (Lollpop) Minimum SDK API 21: Android 5.0 (Lollpop) Minimum SDK API 21: Android 5.0 (Lollpop) Iminum SDK Previous Iminum SDK	🙊 Create New Project		×
Select the form factors your app will run on Different platforms may require separate SDKs Phone and Tablet Minimum SDK API 22: Android 5.1 (Lollipop) Lower API levels target more devices, but nave rewer reatures available. By targeting API 22 and later, your app will run on < 1% of the devices that are active on the Google Play Store. Help me choose Wear Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK API 21: Android 5.0 (Lollipop) Galass (Not Installed) Minimum SDK Minimum SDK API 21: Android 5.0 (Lollipop) Minimum SDK API 21: Android 5.0 (Lollipop)	Reference to the second	Android Devices	
Sciect the form factors your app will run on Different platforms may require separate SDKs Phone and Tablet Minimum SDK API 22: Android 5.1 (Lollipop) Lower API ievels target more devices, but have rewer reatures available. By targeting API 22 and later, your app will run on < 1% of the devices that are active on the Google Play Store. Help me choose Wear Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK API 21: Android 5.0 (Lollipop) Intervention of the devices that are active on the conset of the device of the devices that are active on the conset of the device of the device of the device of the conset of the device of the			
Different platforms may require separate SDKs Phone and Tablet Minimum SDK API 22: Android 5.1 (Lollipop) Lower API levels target more devices, but nave fewer reacures available. By targeting API 22 and later, your app will run on < 1% of the devices that are active on the Google Play Store. Help me choose Wear Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK PI 21: Android 5.0 (Lollipop) TV Minimum SDK PI 21: Android 5.0 (Lollipop) TV Minimum SDK Previous Previous Previous Previous Previous Piret Cance Finish	Select the form fac	ctors your app will run on	
Phone and Tablet Minimum SDK API 22: Android 5.1 (Lollipop) Lower API levels target more devices, but have rewer reacures available. By targeting API 22: and later, your app will run on < 1% of the devices that are active on the Google Play Store. Help me choose Wear Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK Api 21: Android 5.0 (Lollipop) Imimum SDK Api 21: Android 5.0 (Lollipop) Imimum SDK Api 21: Android 5.0 (Lollipop) Imimum SDK Api 21: Android 5.0 (Lollipop) Imimum SDK Api 21: Android 5.0 (Lollipop) Imimum SDK Api 21: Android 5.0 (Lollipop) Imimum SDK Api 21: Android 5.0 (Lollipop) Imimum SDK Api 21: Android 5.0 (Lollipop) Imimum SDK Api 21: Android 5.0 (Lollipop) Imimum SDK Api 21: Android 5.0 (Lollipop) Imimum SDK Imimum SDK <td>Different platforms may requ</td> <td>ire separate SDKs</td> <td></td>	Different platforms may requ	ire separate SDKs	
 Phone and Tablet Minimum SDK API 22: Android 5.1 (Lollipop) Lower API levels target more devices, but have rewer reatures available. By targeting API 22 and later, your app will run on < 1% of the devices that are active on the Google Play Store. Help me choose Wear Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK API 21: Android 5.0 (Lollipop) Android Auto Glass (Not Installed) Minimum SDK Previous Mext Cancel 			
Minimum SDK API 22: Android 5.1 (Lollipop) Lower API levels target more devices, but have rewer reatures available. By targeting API 22 and later, your app will run on < 1% of the devices that are active on the Google Play Store. Help me choose Wear Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK API 21: Android 5.0 (Lollipop) Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK PI 21: Android 5.0 (Lollipop) TV Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK PI 21: Android 5.0 (Lollipop) TI Minimum SDK PI 21: Android 5.0 (Lollipop) TI Minimum SDK Firstalled) Minimum SDK Previous Mext Cancel Firsta	Phone and Tablet		٦
Lower API levels target more devices, but have fewer features available. By targeting API 22 and later, your app will run on < 1% of the devices that are active on the Google Play Store. Help me choose Wear Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK API 21: Android 5.0 (Lollipop) Android Auto Glass (Not Installed) Minimum SDK Previous Next Cancel		API 22: Android 5.1 (Lollipop)	I
Help me choose Wear Minimum SDK API 21: Android 5.0 (Lollipop) TV Minimum SDK API 21: Android 5.0 (Lollipop) Android Auto Glass (Not Installed) Minimum SDK Previous Next Cancel Finish		Lower AP1 levels target more devices, but have tewer reatures available. By targeting API 22 and later, your app will run on < 1% of the devices that are active on the Google Play Store.	,
Wear Minimum SDK API 21: Android 5.0 (Lollipop) Minimum SDK Apr 21: Android 5.0 (Lollipop) Minimum SDK Android Auto Glass (Not Installed) Minimum SDK Previous Minimum SDK Previous Cancel		Help me choose	
Minimum SDK API 21: Android 5.0 (Lollipop)	U Wear		
TV Minimum SDK API 21: Android 5.0 (Lollipop) Android Auto Glass (Not Installed) Minimum SDK Previous Next Cancel Finish	Minimum SDK	API 21: Android 5.0 (Lollipop)	
Minimum SDK API 21: Android 5.0 (Lollipop) Android Auto Glass (Not Installed) Minimum SDK Previous Previous Finish	□ TV		
Android Auto Glass (Not Installed) Minimum SDK Previous Previous Finish	Minimum SDK	API 21: Android 5.0 (Lollipop)	
Glass (Not Installed) Minimum SDK	Android Auto		
Minimum SDK	Glass (Not Installed)		
Previous Cancel Finish	Minimum SDK	▼	
Previous Cancel Finish			
Previous Cancel Finish			
		Previous Next Cancel Finish	

- Select the pre-defined app template to apply. In this example we choose: Blank Activity
- 8. Click Next



Example 2.1 : HelloWorld App

9. The wizard is ready to construct the solution. The text-boxes give you an opportunity to change any of the default names given to the main activity, the app's layout, its title, and menu. *Please do not change anything now*.

10. Click Finish

 You are done! (your next step is to try the app on the emulator – explained later in this lesson)



Example 2.1 : HelloWorld App

The app's GUI and the Palette (graphical toolbox) are shown. On the left pane, the Project Explorer shows the application's file structure.



- Java/ Holds your Main-Activity Java code. All other Java files for your application go here.
- res/ This folder stores application resources such as *drawable* files, UI *layout* files, *string* values, *menus*, multimedia, etc.
- manifests The Android Manifest for your project.



Example 2.1 : HelloWorld App – Java Code: MainActivity.java

```
package matos.csu.helloapp;
import ...
```

```
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.menu main, menu);
       return true;
   }
   @Override
   public boolean onOptionsItemSelected(MenuItem item) {
       // Handle action bar item clicks here. The action bar will
       // automatically handle clicks on the Home/Up button, so long
       // as you specify a parent activity in AndroidManifest.xml.
       int id = item.getItemId(); OUONE Than CONE
       //noinspection SimplifiableIfStatement
       if (id == R.id.action settings) {
           return true;
       }
       return super.onOptionsItemSelected(item);
```

}

}

Example 2.1 : HelloWorld App - Layout: activity_main.xml



Example2.1 (again...) : HelloWorld App

We will use **Eclipse + ADT** to create a bare bone app. All it is needed from the developer is to feed the New Android Application wizard with a few selections (no extra code will be added to the default app skeleton generated by the IDE+SDK).

The adjacent figures show the solution made by the wizard running on a *Jelly* Bean emulator and device.



- 1. Start Eclipse
- From menu choose File > New > Android Application Project
- Enter in the Application Name box: HelloWorldApp
- Enter Project name: HelloWorldAppProject u duong tl
- 5. Modify *Package Name* prefix to: csu.matos.helloworldappproject
- 6. For *Minimum Required SDK* choose: API 10: Android 2.3.3 (Gingerbread)
- 7. For *Target SDK* select the option: API 16:Android 4.1 (Jelly Bean)
- 8. Select for *Compile With* the option: API 16:Android 4.1 (Jelly Bean)
- 9. Click Next
- 10. Click Next

🔅 New Android Application
New Android Application Creates a new Android Application
Application Name: 1 HelloWorldApp
Project Name: HelloWorldAppProject
Package Name: 1 csu.matos.helloworldappproject
Minimum Required SDK: API 10: Android 2.3.3 (Gingerbread)
Target SDK: API 16: Android 4.1 (Jelly Bean)
Compile With: API 16: Android 4.1 (Jelly Bean)
Theme: 1 Holo Light with Dark Action Bar
Choose a target API to compile your code against, from your installed SDKs. This is typic: the first version that supports all the APIs you want to directly access without reflection.
an cong . com
Sack Next > Finish Cancel

Example : HelloWorld App

On the form **Configure Launcher Icon** do the following:

- 11. Foreground > Clipart > Choose
- 12. Select an icon from the set of available images > Close

14. Click Next



Example : HelloWorld App

The **Create Activity** form provides a number of basic templates from which your application could be constructed.

15. Select the Blank Activity template.
16. Click Next.



Example : HelloWorld App

The **Blank Activity** form provides a way to name the main Activity and Layout name.

17. Leave the default values shown in the form (Activity Name and Layout Name).
18. Click Finish.

At this point the wizard has completed all the steps required to make the app.

After a few seconds the Eclipse perspective shows the app's UI. The Java solution is shown in the PackageExplorer pane (see next pages)





app.

F

File Structure of a Typical Android App

- **src/** Includes your skeleton Activity Java file. All other Java files for your application go here.
- <<u>Android Version>/</u> (e.g., Android 4.1/) Includes the android.jar file that your application will build against.
- gen/ This contains the Java files generated by ADT, such as your R.java file
- assets/ This is empty. You can use it to store raw asset files.
- res/ This folder holds application resources such as *drawable* files, UI *layout* files, *string* values, etc.
- **bin/** The bytecode (.apk) version of your app is stored here
- AndroidManifest.xml The Android Manifest for your project.
- default.properties This file contains project settings, such as the build target.

Login into the Android OS shell

- Although it is *not* necessary, a developer may gain access to some of the innermost parts of the Android OS.
- For a UNIX-like experience you can log into the system by executing the emulator and issuing selected shell commands.

C:\windows\system	32\cmd.exe - adb sh	ell				
Microsoft Windov Copyright (c) 20	us [Version 6.: 109 Microsoft	1.7600] Corporatio	on. All ri	ights 1	reserved.	-
C:\Program Files # ls -1 ls -1	x86)\Androi	d∖android-	-sdk\platfo	erm-too	ols>adb shell	=
drwxrwxrwx root drwxrwxrwx root drwxrwxrwx root drwxrwxrwx root lrwxrwxrwx root lrwxrwxrwx root	root em cache root system root root root root	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	2012-03-10 2012-03-10 2012-03-10 2012-03-10 2012-03-10 2012-03-10 2012-03-10 2012-03-10 2012-03-10 202-03-10	00:01 10:33 00:01 00:01 00:01 00:01 00:01 00:01	<pre>config cache sdcard -> /mnt/sdcard acct mnt vendor -> /system/vendor d -> /system/vendor etc -> /system/etc vencotd ve</pre>	
-FW-F-F- FOOL -FW-F-F- FOOL drwxr-xr-x FOOL drwxr-xr-x FOOL drwxr-xr-x FOOL dr-xr-xr-x FOOL dr-xr-xr-x FOOL -FWXF-X FOOL -FWXF-X FOOL	root root root root root root root	3784 1 0 1 2 0 1 1 1 1 1 3805 1 1677 1	769-12-31 969-12-31 8011-02-03 969-12-31 969-12-31 969-12-31 969-12-31 969-12-31	19:00 19:00 18:01 19:00 19:00 19:00 19:00 19:00	ueventa.rc ueventa.goldfish.rc system sys sbin proc init.rc init.goldfish.rc	
-rwxr-x root -rw-r-r- root drwxrwx syste drwx root drwxr-xr-x root # df df	root root m system root root	94168 1 118 1 2 2 2	.969-12-31 .969-12-31 .012-03-09 .010-01-27 .012-03-10	19:00 19:00 23:02 19:59 00:02	init default.prop data root dev	
Filesystem /dev /mnt/asec /mnt/obb /system /data /cache /mnt/sdcard	Size 125M 125M 125M 64M 64M 1019M	Used Fr 32K 12 0K 12 96M 32M 3 1M 6 164M 85	№e B1ks1 25M 4096 25M 4096 25M 4096 0K 4096 11M 4096 11M 4096 52M 4096	ize		
/mnt/secure/asec # cd sdcard cd sdcard # ls -l ls -l drwxr-x <u>syste</u>	= 1019M ====================================	164M = 85	55M 2048 2012-03 <u>-09</u>	23:03	3 LOST.DIR	
drwxr-x syste rwxr-x syste drwxr-x syste rwxr-x syste rwxr-x syste	em sdcard_rw em sdcard_rw em sdcard_rw em sdcard_rw em sdcard_rw m sdcard_rw m	5239976 263230 314676	2012-03-10 2012-03-09 2012-03-09 2012-03-09 2012-03-09 2012-03-09	19:59 23:10 23:11 23:29 23:29	9 DCIM 9 Amarcord.mp3 1 Android 9 Bea-Strada-Volterra-12X17.jpg 9 Bea-Vic-Arno-Firenze.jpg • •	•



Login into the Android OS shell



STEPS

- Use the Eclipse AVD Manager to start one of your AVDs (say Gingerbread23)
- At the DOS command
 prompt level run the
 Android Debug Bridge
 (adb) application

adb shell

C:\windows\system32\cn	nd.exe - adb sh	ell					
Microsoft Windows []	levsion 6 1	26001	_				
Copyright (c) 2009	Microsoft (Corporati	ion.	A11 ri	ights :	reserved.	<u> </u>
C:\Program Files (x) # ls -1	86>\Android	l\andro id	l-sdk	\platfo	orm-to	ols≻adb shell	E
18 -1 dw-v woot	woot		2012	-03-10	00-01	config	
dwwxwwx sustem	cache		2012	-03-10	10:33	cache	
lruxruxrux root	root		2012	-03-10	00:01	$sdcard \rightarrow /mnt/sdcard$	
drwxr-xr-x root	root		2012	-03-10	00:01	acct	
drwxrwxr-x root	system		2012	-03-10	00:01	mnt	
lrwxrwxrwx root	root		2012	-03-10	00:01	vendor -> /system/vend	lor
lrwxrwxrwx root	root		2012	-03-10	00:01	d -> /sys/kernel/debug	y in the second s
lrwxrwxrwx root	root		2012	-03-10	00:01	etc -> /system/etc	
-rw-rr root	root	3764	1969	-12-31	19:00	ueventd.rc	
-rw-rr- root	root	0	1969	-12-31	19:00	ueventd.goldfish.rc	
drwxr-xr-x root	root		2011	-02-03	18:01	system	
drwxr-xr-x root	root		1969	-12-31	17:00	sys	
drwxr-x root	root		1393.	-12-31	17:00	SDIN	
ar-xr-xr-x root	root	10005	1303.	-12-31	10-00	proc	
-rwxr-x root	root	1600	1020.	-12-31	10-00	init coldfick no	
-rwxr-x root	root	94169	1060.	-12-31	10-00	init.goiurisn.rc	
-wu-ww woot	woot	118	1969.	-12-31	19:00	default nwon	
dwuywuyy sustem	sustem	110	2012	-03-09	23:02	data	
drwxrwx x system	Pont		2010	-01-27	19:59	root	
drwxr-xr-x root	root		2012	-03-10	00:02	deu	
# df	2000				00-04		
df							
Filesystem	Size	Used I	ree	B1ks i	ize		
∕dev	125M	32K 1	.25M	4096			
/mnt/asec	125M	ØK 1	.25M	4096			
/mnt/obb	125M	OK 1	.25M	4096			
/system	96M	96M	ØK	4096			
/data	64M	32M	31M	4096			
/cache	64M	11	62M	4076			
/mnt/sacara	10170	164M 0	155M	2048			
/mnt/secure/asec	10124	164m _ 8	155M	2048			
# CU SUCAPU							
tu sucaru # le _1							
drwxr-x sustem	sdcard ru		201	2-03-09	23:0	3 LOST_DIR	
drwxr-x system	sdcard rw		201	2-03-10	19:5	9 DCIM	
rwxr-x system	sdcard_rw	5239976	201	2-03-09	23:1	0 Amarcord.mp3	
drwxr-x system	sdcard_rw		201	2-03-09	23:1:	1 Android	
rwxr-x system	sdcard_rw	263230	3 201:	2-03-09	23:2	9 Bea-Strada-Volterra-1	.2X17.jpg
rwxr-x system	sdcard_rw	314676	201:	2-03-09	23:2	9 Bea-Vic-Arno-Firenze.	.jpg 🔻 🔻
•							▶

adb is a tool located in the directory: C:\Your-SDK-Folder\Android\android-sdk\platform-tools\

https://fb.com/tailieudientucntt

Login into the Android OS shell

If more than one emulator is running (or your phone is physically connected to the computer using the USB cable) you need to identify the target.

Follow the next steps:

1. Get a list of attached devices



2. Run the **adb** application as follows:

adb -s emulator-5554 shell

Remember, the adb tool is located at C:\Program Files (x86)\Android\android-sdk\platform-tools\

Login into the Android OS shell

Android accepts a number of Linux shell commands including the useful set below
ls
mkdir make a directory
rmdir remove directory
rm -r \ldots to delete folders with files
rm remove files
mvfiles
cat displaying short files
cd change current directory
pwd find out what directory you are in
df shows available disk space
chmod changes permissions on a file
date display date
exit terminate session

There is no copy (**cp**) command in Android, but you could use **cat** instead. For instance:

cat data/app/theInstalledApp.apk > cache/theInstalledApp.apk

Hacking: Moving an app from a Rooted Phone to the Emulator

If you want to transfer an app that is currently installed in your rooted developer's phone to the emulator, follow the next steps:

- Run command shell: > adb devices (find out your hardware's id, say HT096P800176)
- Pull the file from the device to your computer's file system. Enter the command adb -s HT096P800176 pull data/app/theInstalledApp.apk
 c:\theInstalledApp.apk
- 3. Disconnect your Android phone
- 4. Run an instance of the Emulator
- 5. Now install the app on the emulator using the command adb -s emulator-5554 install c:\theInstalledApp.apk adb -s emulator-5554 uninstall data/app/theInstalledApp.apk ← to uninstall

You should see a message indicating the size of the installed package, and finally: *Success*.

Simpler than Hacking: Install a File Manager for Android

Visit **Google Play Store** and choose a user-friendly file manager app from the various (usually very good) options available.

A file manager app allows you to easily administer the folders and files in the system's flash memory and SD card of your Android device (or emulator).



cuu duong than cong . com

A sample of *File-Management* apps seen at <u>https://play.google.com</u> on Aug-27th -2014

Using an alternate SD card & userData Image

Run Configurations		From the Eclinse menu create a		
Android Application				
Android Application	Name: 01-Hello-World Andred Target Common Deploymen: Target Common Manual Automatic Select a preferred Android Virtual Device for deployment: AVD Name Target Name Platform API Level CPU/ABI Gingerbread23 Google APIs (Google 2.3.3 10 ARM (arme Start)	new launch configuration: Run > Run Configurations > New icon		
Generic Server(Externa GlassFish Application GGWT Application GWT JUnit Test Jy GWT JUnit Test	Nexus-S Google APIs (Google 2.3.3 10 ARM (arme Honeycomb3x Google APIs (Google 3.2 13 ARM (arme	On the Target panel:		
 Grand Perview TrTP Preview Tron Python Run Iron Python unittest J2EE Preview Java Applet Java Application JU Junit JUInit Plug-in Test Jython run Jython run Jython unittest Mwe2 Launch Oracle Coherence Oracle WebLogic Script 	Refresh Manager Emulator launch parameters: Network Speed: Full Network Latency: None Object User Data Disable Boot Animation Additional Envolver Command Line Options -sdcard c:\My_Emulator_Data\myreallybigsdcard.img -datadir c:\My_Emulator_Data	 Select existing AVD (Gingerbread in this example) Enter additional Command Line Options (see caption below) Click on Apply > Run 		
Filter matched 39 of 168 items	Apply Revert Run Close			
Addit	ional Emulator Command Line Options: dcard c:\My_Emulator_Data\myreallybigsdcard.in	ng -datadir c:\My Emulator_Data		

2 - 51

Android Emulator – Simulate Texting

Sending Text Messages from your Window's PC to the Emulator

- 1. Start the emulator.
- 2. Open a new DOS command shell and type :

c:> adb devices

this way you get to know the emulator's numeric port id (usually **5554**, **5556**, and so on)

- Initiate a Telnet session with the sender at localhost, port 5556 identifies an active (receiving) Android emulator. Type the command:
 c:> telnet localhost 5554
- After receiving the telnet prompt, you can send a text message to the emulator on port 5554 (no quotes needed for the message) sms send <Sender's phone number> <text message>

Windows7 – temporarily install Telnet Client by using a command line

- 1. Click Start button, type cmd in the 'search programs and files' box, and then press ENTER.
- 2. Type the following command: **pkgmgr /iu:"TelnetClient"**

Android Emulator – Simulate Texting



Android Emulator – Simulate Phone Calls

Making a Phone Call from your PC to the Emulator

- 1. Start the emulator.
- 2. Open a new shell and type :

adb devices

to know the emulator's numeric port id (usually 5554, 5556, and so on)

cuu duong than cong . com

- Connect to the console using telnet command like:
 telnet localhost 5554 (5554 is the 'phone number' to be called)
- After receiving the telnet prompt you can place a call (voice) with the command
 gsm call <caller's phone number>

Android Emulator – Simulate Phone Calls





Using: Android Device Monitor

It is *much simpler* to test telephony operations (SMS/Voice) as well as GPS services using the controls included in the IDE (both AS and Eclipse)

- Telephony Status change the state of the phone's Voice and Data plans (home, roaming, searching, etc.), and simulate different kinds of network Speed and Latency (GPRS, EDGE, UTMS, etc.).
- 2. Telephony Actions perform simulated phone calls and SMS messages to the emulator.
- **3. Location Controls** send mock location data to the emulator so that you can perform location-aware operations requiring GPS assistance.
 - Manually send individual longitude/latitude coordinates to the device. Click Manual, select the coordinate format, fill in the fields and click Send.
 - Use a **GPX file** describing a route for playback to the device.
 - Use a **KML** file to place multiple *placemarker points* on a map

Using Eclipse's DDMS facility

DDMS Telephony Services DDMS - 01-Hello-World/src/csu/matos/HelloWorldActivity.java - Eclipse Send text-messages File Edit Source Refactor Navigate Search Project Run Window Help 😫 🞜 🖸 😭 🐉 Java 🔊 DDMS 🌣 Debua Make a phone call T P 28 🗐 🗐 🗄 🖡 🗸 🖓 🗸 🏷 🗸 🖒 🗸 (Å - -Devices 🖾 Emulator Control 🛑 File Ex 🛛 🖄 Threa 🛢 Heap 🛢 Alloca ÷ **Telephony Status** 0 3 3 × <u>8</u>. Speed: Full Name Size Voice: home 🗁 data • 6 Latency: None Data: home 🗁 mnt Name 🗁 asec **Telephony Actions** emulator-5554 🗁 obb Incoming number: 5551122 system process 🗁 sdcard Voice Beatriz Matos : Msg sent from Eclipse's com.android.inputmet Amarcord.mp3 5239976 com.android.phone SMS Android com.android.systemui Bailables.mp3 4948579 Message: Msg sent from Eclipse's Google Search com.android.launcher Bea-Strada-Volterra-12X17.jpg 263230 Emulator Control com.android.settings Bea-Vic-Arno-Firenze.jpg 314676 android.process.acore 3904513 Besame Mucho.mp3 com.google.process.ga Brazil Bahia.mp3 7372782 Send Hang Up com.android.deskclocl Cancin India.m4a 3077249 com.android.defconta Chrysanthemum.jpg 879394 Location Controls com.android.music Cinema Paradiso (Theme).mp3 6522671 com.android.quicksear 🕞 DCIM Manual GPX KML android.process.media Il cuore e' uno zingaro.mp3 3211768 Decimal com.android.mms P Io e te Maria.mp3 4538935 Sexagesimal com.google.android.ar 🕞 LOST.DIR com.android.email Longitude -122.084095 La Bambola.mp3 5671032 com.svox.pico Mack The Knife.mp3 4586372 Messagin 37.422006 Latitude 4594086 csu.matos Mi Tierra.mp3 Send 111 111 ъ nsole 🖾 D LogCat 🛃 📮 🕆 📑 😪 WLST Dialer Maps Contacts Browser -03-10 23:18:14 - 01-Hello-World] AD 10 10:10:14 A1 Halla Haaldl Andread Launch gn in to Google... . ∎∜ CuuDuongThanCong.com https://fb.com/tailieudientucntt

Lesson 2: **Setup: Android Workbench & Emulator Questions**?

Appendix 1 - Using a Hardware Device

Connecting your Physical Device to the Computer

- 1. Make sure the USB driver has been installed in your PC (click SDK Manager > Extras > check box [Google USB driver package] to install)
- 2. Use a mini-USB cable to connect the device to your computer.
- 3. Expand the Notification bar. Click on [USB connected] option.
- 4. Click on [*Turn on USB storage*] to mount the device.
- 5. Now you could now use the Eclipse-ADT-File Explorer and your Window's Explorer tool to pull/push/delete/rename files to the device.



Appendix 2 – Emulator-to-Emulator Interaction

- 1. Run two instances of the emulator (typical IDs are: 5554, 5556, ...)
- 2. Dial (or send SMS) from one of them (say 5554) to the other (5556)
- 3. Press the Green/Red call buttons to accept/terminate the call
- 4. Try sending SMS (use numbers 5554 and 5556)



Appendix 3 – Sync your Contacts

How to Transfer Your Google Contacts into the Emulator

- Go to your Gmail account using a web browser, click on Gmail > Contacts on the left sidebar.
- Select all the contacts you want on your emulator/phone. Then click on More > Export and select vCard format. Download the "contacs.vcf" file to your PC.
- 3. Push the contacs.vcf file from the PC to the emulator's **SD card**.
- 4. Open the emulator's **Contacts** app hit **Menu > Import**.
- 5. Choose the option *Import from SD card.*

Source visited on July 2009, link:

http://stackoverflow.com/questions/1114052/importing-gmail-contacts-on-android-emulator



Appendix 4

Shortcuts: Android-Studio IDE

Eclipse developers are used to typing

Ctrl + Shift + O

To Organize ALL imports.

To automatically accomplish the same effect, modify your Android Studio Workbench as indicated on the figure to the right.

File > Settings > Editor > General > Auto Import

Appearance & Behavior Keymap Iditor General Smart Keys Appearance Editor Tabs Code Folding Code Completion Auto Import Postfix Completion Code Style Inspections File and Code Templates File Encodings Live Templates File Types Copyright Images Intentions Language Injections Version Control Build, Execution, Deployment	2 Settings		Editor & Caparal & Auto Import	Paca
▶ Appearance & Behavior Keymap ▼ Editor ♥ Editor ♥ General Smart Keys Appearance Editor Tabs Code Folding Code Completion Auto Import Postfix Completion Console Folding Colors & Fonts Colors & Fonts Colors & Fonts File and Code Templates File Types Copyright Images Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment				Nese
Keymap Editor General Smart Keys Appearance Editor Tabs Code Folding Code Completion Autor Import Postfix Completion Console Folding Code Style Inspections File and Code Templates File Trypes Copyright Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment	Appearance & Behavior		XML	
Variation © General Smart Keys Appearance Editor Tabs Code Folding Code Completion Auto Import Postfix Completion Console Folding Code Style Postfix Completion Console Folding Code Style Pie Encodings Live Templates File Encodings Live Templates File Trypes Copyright Emmet Danges Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment	Keymap		Show import popula	
Smart Keys Appearance Editor Tabs Code Folding Code Completion Auto Import Postfix Completion Console Folding Colors & Fonts Code Style File and Code Templates File Encodings Live Templates File Types Copyright Emmet Images Intentions Language Injections Spelling Tobo Plugins Version Control Build, Execution, Deployment				
Smart Keys Appearance Editor Tabs Code Folding Code Completion Auto Import Postfix Completion Console Folding Colors & Fonts Code Style File and Code Templates File Encodings Live Templates File Types Copyright Images Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment	▼ General		lava	
Appearance Editor Tabs Code Folding Code Completion Auto Import Postfix Completion Console Folding Colors & Fonts Colors & Fonts Code Style Inspections File and Code Templates File Encodings Live Templates File Types Copyright Emmet Inages Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment	Smart Keys			
Code Folding Code Completion Auto Import Postfix Completion Console Folding Colors & Fonts Code Style Inspections File and Code Templates File Encodings Live Templates File Types Copyright Emmet Inages Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment	Appearance		Insert imports on paste: All	
Code Completion Auto Import Postfix Completion Console Folding Colors & Fonts Code Style Inspections File and Code Templates File Encodings Live Templates File Types Copyright Emmet Images Intentions Language Injections Speling TODO Plugins Version Control Build, Execution, Deployment	Editor Tabs		Show import popula	
Auto Import Postfix Completion Console Folding Colors & Fonts Code Style Inspections File and Code Templates File Encodings Live Templates File Types Copyright Emmet Images Intentions Cyc++ Spelling TODO Plugins Version Control Build, Execution, Deployment	Code Folding			
Postfix Completion Console Folding Colors & Fonts Code Style Inspections File and Code Templates File Encodings Live Templates File Types Copyright Images Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment	Code Completion		Optimize imports on the fly	
Console Folding Colors & Fonts Code Style Inspections File and Code Templates File Encodings Live Templates File Types Copyright Emmet Images Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment	Postfix Completion			
 Colors & Fonts Code Style Inspections File and Code Templates File Encodings Live Templates File Types Copyright Emmet Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment 	Console Folding		Add unambiguous imports on the fly	
 Code Style Inspections File and Code Templates File Encodings Live Templates File Types Copyright Emmet Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment 	Console Folding		Exclude from Import and Completion	
Code Style Inspections File and Code Templates File Encodings Live Templates File Types Copyright Emmet Images Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment OK Cancel Annly Hein	Code Style	B		
File and Code Templates File Encodings Live Templates File Types Copyright Emmet Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment	Inspections			-
File Encodings File Encodings Live Templates File Types Copyright Emmet Images Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment	File and Code Templates	Ē		_
Live Templates File Types Copyright Emmet Images Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment	File Encodings	G	No exclude patterns	
File Types Copyright Emmet Images Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment	Live Templates			
 Copyright Emmet Images Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment 	File Types			
Emmet ong that Images Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment	Copyright	G		
Images Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment		ha	n cong com	
Intentions Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment	Images	IIa	i cong . com	
 Language Injections Spelling TODO Plugins Version Control Build, Execution, Deployment 	Intentions			
Spelling TODO Plugins Version Control Build, Execution, Deployment OK Cancel Apply Help	Language Injections	G	C/C++	
TODO Plugins Version Control Build, Execution, Deployment OK Cancel Apply Help	Spelling	ē	Show import popup	
Plugins Version Control Build, Execution, Deployment OK Cancel Apply Help	TODO			
Version Control Build, Execution, Deployment OK Cancel Apply Help	Plugins			
Build, Execution, Deployment	Version Control			
OK Cancel Apply Help	Build, Execution, Deploym	ent		
			OK Cancel Apply H	Help

Appendix 4

Shortcuts: Android-Studio IDE

Operation	Android Studio Shortcut
Reformat code	CTRL + ALT + L
Optimize imports	CTRL + ALT + O
Code Completion cuu duong tha	CTRL + SPACE
Issue quick fix	ALT + ENTER
Surround code block	CTRL + ALT + T
Line Comment or Uncomment	CTRL + /
Block Comment or Uncomment	CTRL + SHIFT + /
Close Active Tab cuu duong tha	CTRL + F4
Build and run	SHIFT + F10
Build	CTRL + F9
All Options	Ctrl + Shift + A