### Special Interest Group on Innovative Software 2011-2012 Workshop on Android app development

#### Session 1: Application Fundamentals

#### Contents

1	Setup	o development environment and emulator	2
	1.1.	Download and install the ADT	2
	1.2.	Download the Android SDK	4
2	Steps	s for developing Android apps	7
3	Cont	rol the emulator	8
4	Unde	erstand fundamental concepts	11
	4.1.	Android architecture	11
	4.2.	Application components	11
	4.3.	Application resources	12
	4.4.	Define the app properties in Android	13
5	Creat	te your first application	14
	5.1	Run and debug apps in emulator	15
	5.2	Run and debug apps in actual device	17
6.	Creat	te UI element using XML	18
	6.1.	Understand the XML layout structure	19
	6.2.	Add text and image to the UI	19
7.	Hand	Ile user input and more interface design	26
	7.1.	Accept user input by adopting EditText and Button	26
	7.2.	Activity lifecycle	29
	7.3.	Use TableLayout for structural elements and data	30
	7.4.	Use ScrollView for large content	33
8.	Prepa	are to publish your app	35
	8.1.	Prepare the app icon	35
	8.2.	Change the app name and apply localization	36
	8.3.	Sign and export the APK	37
9.	Refer	rence and learning resources:	39

Prepared by Mr. Felix Tam, Committee Member of the Special Interest Group (SIG) on Innovative Software

2011-2012, Department of Computer Science, Hong Kong Baptist University.

All rights reserved. All content copyright and other rights reserved by its respective owners. Any content, trademark(s), or other material that may be found on this document remains the copyright of its respective owner(s). In no way does the Special Interest Group on Innovative Software claim ownership or responsibility for such items, and you should seek legal consent for any use of such materials from its owner.

## 1 Setup development environment and emulator

Tools that you will need for Android app development:

- 1. Eclipse IDE for Java Developers
- 2. Java SDK
- 3. ADT plugin for Eclipse
- 4. Android SDK

#### 1.1. Download and install the ADT

- Make sure you have the JDK installed on your computer. If no, please download it from <u>http://www.oracle.com/technetwork/java/javase/downloads/index.html</u> and select the version according to your operating system.
- Start Eclipse, go to Help > Install New Software, click on the Add in the top-right corner. (You may get Eclipse IDE for Java Developers at <a href="http://www.eclipse.org/downloads/">http://www.eclipse.org/downloads/</a>)
- In the Add Repository dialog, type ADT Plugin for the name and <u>https://dl-ssl.google.com/android/eclipse/</u> for the location respectively.

🌐 Add Re	pository	×
<u>N</u> ame:	ADT Plugin	L <u>o</u> cal
Location:	https://dl-ssl.google.com/android/eclipse/	Archive
?	ОК	Cancel

4. After a while you will see an option called **Developer Tools**. Select the checkbox next to those options and click **Next** to continue.

type filter text	
Name	Version
a 👿 💷 Developer Tools	
🔽 🖗 Android DDMS	15.0.1.v201111031820-219398
👿 🖗 Android Development Tools	15.0.1.v201111031820-219398
🖉 🖗 Android Hierarchy Viewer	15.0.1.v201111031820-219398
📝 🎼 Android Traceview	15.0.1.v201111031820-219398
Select All         Deselect All         4 items selected           Details	i
Show only the latest versions of available software	t are already installed
	installed?
Show only software applicable to target environment	
Contact all update sites during install to find required software	
(?) < <u>B</u> ack N	ext > Einish Cancel

5. Click **Next**, **Finish** and accept the terms of the license agreements in order to get the ADT plugin from the remote site. When you are asked for unsigned content, click **OK** to accept. You have to restart Eclipse after the installation is finished.

Installing Software
Installing Software
Fetching com.android.ide.eclipse.hierarchyvi//dl-ssl.google.com/android/eclipse/plugins/
Always run in background
Run in Background     Cancel

 After restarting Eclipse, you will be asked to configure an Android SDK as well. Click Cancel at this moment since we will download and configure the SDK manually.

### 1.2. Download the Android SDK

- Go to <u>http://developer.android.com/sdk/index.html</u> and download the latest SDK in zip format.
- 2. Unzip the zip file and you will get the folder **android-sdk-windows**. Put the SDK folder to your favorite location. In this tutorial, we will put it under the Eclipse root folder.
- 3. The SDK does not contain any Android platform binary initially. If you need to develop an app which bases on Android 4.0, you have to download the Android 4.0 platform binary and other related stuff. In this workshop, all of our tutorials and examples will base on Android 2.3.3. To download Android 2.3.3 platform binary, please navigate to your SDK folder and start SDK Manager.exe.
- 4. Download and install the packages according to the screenshots:

Android SDK Manager				
Packages Tools				
SDK Path: E:\Softwares\eclipse-java-indigo-SR1-win32\eclipse\and	droid-sd	k-windov	VS	
Packages				
			<b>C</b>	
· Name	API	Kev.	Status	
			<b>*</b> • • •	
Android SDK Tools		16	Installed	
Android SDK Platform-tools		9	Not installed	
▶ <b>N</b> ► Android 4.0 (API 14)				
Android 3.2 (API 13)				
Android 3.1 (API 12)				
Android 3.0 (API 11)				
▲	10	2	Not installed	
SDK Platform	10	2	Not installed	
Carala ABIa ha Carala las	10	1	Not installed	
Google APIs by Google Inc.	10	2	Not installed	
Dual Screen APIS by KYOCERA Corporation	10	1	Not installed	
RealsD by LGE	10	1	Not installed	
EDK 1.2 by Sony Ericsson Mobile Communications	10	1	↓ Not installea	
Android 2.2 (API 8)				
Android 2.1 (API7)				
Android I.0 (API4)				
Android 1.5 (AP15)				
Extras		-	Not installed	
Anarola Support package		3	Not installed	
Google Annob Aus Sak package		- 4	Not installed	
Google Analytics Sak package		2	Not installed	
Google Market Licencing pockage		1	Not installed	
Google USB Driver nackage		1	Not installed	
Google Webdriver package		2	Not installed	
		2		
Show: Vpdates/New VInstalled Obsolete Select New	w or Up	dates		Install 5 packages
Sort by:  API level  Repository Deselect A	<u>AII</u>			Delete packages
Done loading packages.				

5. Choose Accept All and click Install. This should take a few minutes.



- 6. When you are asked to restart ADB, choose Yes.
- 7. We can now create an Android 2.3.3 emulator for testing and debugging. Go to the SDK folder and start **AVD Manager.exe**.
- 8. In the Android Virtual Device Manager, click **New** to create a new Android emulator.
- 9. Please refer to the screenshot below for the options in creating the emulator.

Create new Android Virtual Device (AVD)										
Name:	Android_2	Android_2.3.3_GAPI								
Target:	Google Al	PIs (Google Inc.) - A	PI Level 10	•						
CPU/ABI:	ARM (arm	neabi)								
SD Card:										
	Size:	100		MiB 🔻						
	File:			Browse						
Snapshot:	<b>▼</b> Enable	✓ Enabled								
Skin:										
	Built-i	Built-in: Default (WVGA800)								
	Resolution: x									
Hardware:										
	Property	/	Value	New						
	Abstract	ted LCD density	240	Delete						
	Max VM	application hea	24	Derete						
	Device r	am size	256							

10. You can create emulators as many as you want. Please make sure you have downloaded the corresponding Android platform using the SDK Manager before making an emulator.

🗄 Android Virtual Dev	🖞 Android Virtual Device Manager											
Tools												
List of existing Android Virtual Devices located at C:\Users\felixtam\.android\avd												
AVD Name	AVD Name Target Name Platform API Level CPU/ABI New											
Android_2.3.3_G	Google APIs (Google Inc.)	2.3.3	10	ARM (armeabi)	Edit							
					Delete							
					Delete							
					Repair							
					Details							
					Start							
	Refresh											
✓ A valid Android Virt	tual Device. 😡 A repairable Andı	roid Virtual Device	2.									
🗙 An Android Virtual	Device that failed to load. Click 'D	etails' to see the e	error.									

We also need to specify the SDK directory in Eclipse. Inside Eclipse, select
 Window > Preferences > Android, click Browse to choose the location of your
 SDK directory.

Preferences						
type filter text	Android				↓ •	⇒ • •
General Android Ant Help Install/Update	Android Prefer SDK Location: Note: The list o	ences E:\Softwa of SDK Tar <u>c</u>	ares\eclipse-java-indigo-SR gets below is only reloaded	1-win32∖e once you	ec <u>B</u> ro hit 'Apply'	wse or 'OK'.
Java	Target Name	e	Vendor		Platform	API
Maven Mylyn Run/Debug Team Usage Data Collector Validation WindowBuilder XML	Android 2.3. Google APIs	3	Android Open Source Pro Google Inc.	ject	2.3.3 2.3.3	10 10

## 2 Steps for developing Android apps



Source: Google Inc., 461px x 738px, http://developer.android.com

## **3** Control the emulator

1. To start the emulator, go to the **Android Virtual Device Manager**. You can start the manager by double clicking the **AVD Manager.exe** in the SDK directory or the icon in Eclipse toolbar.



2. Choose the emulator, click **Start** and **Launch**.

Launch	Options	x					
Skin:	WVGA800 (48)	0x800)					
Densi	ty: High (240)						
Sc	ale display to real	size					
	Screen Size (in):	3					
	Monitor dpi:	96 ?					
	Scale:	default					
🗆 Wi	🔲 Wipe user data						
🔽 La	unch from snapsł	not					
V Save to snapshot							
	Launch	Cancel					

3. After the emulator has successfully booted, you will see the following the screen. (It may take several minutes to boot up the emulator for the first time.)

	ii ∎i 🔋 4:02	1									
Google	V										
	-								©		
				6					0		
•	• ••		20	2	15	5	6	78		0	
- In the second second		0	W	F	R	T	V	11	I	0	P
		A	S	0	E (	G	н	1	K	1	DIL
			z	X	c	V	В	N	M		4
		ALT	SYM	0			_	-	1		ALT

4. Here is a list of shortcut that you may use it during your development:

Emulated Device Key	Keyboard Key			
Home	НОМЕ			
Menu	F2 or Page-up button			
Back	ESC			
Call/dial button	F3			
Hangup/end call button	F4			
Search	F5			
Power button	F7			
Switch to previous layout orientation				
(for example, portrait, landscape)	KEYPAD_/, Ctrl-F11			
Switch to next layout orientation				
(for example, portrait, landscape)	KEYPAD_9, Ctrl-F12			
Toggle cell networking on/off	F8			
Toggle full screen mode	Alt-Enter			
Toggle trackball mode	F6			
Enter trackball mode temporarily				
(while key is pressed)	Delete			

Source: Google Inc., <u>http://developer.android.com</u>

#### Please refer to

http://developer.android.com/guide/developing/tools/emulator.html for more information.

 We can use DDMS (Dalvik Debug Monitor Server) to control and monitor emulators and actual devices. You will find DDMS in the tools/ directory of the SDK and we can also start it in Eclipse. To do this, click Window > Open Perspective > Other > DDMS.



6. By using DDMS, you will able to simulate phone calls, network status, GPS coordinate for the emulator. You can also browse the file inside the emulator and using Logcat. Logcat is very important towards your development as it provides you a view for collecting and viewing the system debug output.



7. For example, if your application is force closed, then you are required take a look at the logcat to see the exceptions thrown by the Android system

• < •				E (	DDMS 📲 Java	
location Tracker 👘 File Eq	ilorer 🆚	LogCat E3			- 0	
Sauch for matrices. Acco	ete lava e	name. Drefix with nick anni	tax or tests to light	under		
starter for managers were	pa zere n	Gener i terri unu hor, abbe	say of the to mile	Trees.		
L Time	PID	Application	Тад	Test		
D 12-05 16:21:0	75	system_process	SntpClient	request time failed: java.net.Socket	xception: Adc	
D 12-05 16:26:0	75	system_process	SntpClient	request time failed: java.net.Socket8	Exception: Add	
D 12-05 16:31:0	75	system_process	SatpClient	request time failed: java.net.Socket	Reeption: Adc	
I 12-05 16:31:1	289	com.google.proc	EventLogSe	Aggregate from 1323072102863 (log), 1	323072064780	
D 12-05 16:31:1	289	com.google.proc	dalvikvm	GC_CONCURRENT freed 579K, 55% free 30	БТК/6727К, ех	
0 12-05 16:36:0	75	system_process	ShtpClient	request time failed: java.net.Sockets	xception: Adc	
D 12-05 1613812	250	com.android.iad	dalvikvn	GC_EXTERNAL_ALLOC Freed /3K, 4/4 Free	JU92K/5031K,	
1 12-05 16:30:3	75	system_process	Activityna	Starting: intent   act-android.intent	-accion.main	
D 12-05 16:30:3	96	ayaten_process	ACLIVICYNG	Destant BECTN ! / austan /ann (Snava)	area ankt	
D 12-05 16-10-1	61.1		dalarikam	Devine: load line verifusor line	area.apr	
D 12-05 16+38+3	35		ingralld	Deving: FKD 1/susrem/ann/SpaceDat	te anizi (ener	
T 12-05 16:38:3	75	system process	ActivityMa	Displayed con.android.spare parts/.St	areParta: +1e	
1 12-05 16:38:3	75	system process	ACTIVITYMA	Starting: Intent   act-android.intent	ACTION. MAIN	
D 12-05 16:38:3	605	com, android, ana	AndroidRun	Shatting down VM		
¥ 12-05 16:00:0	605	con, android, spa	dalvikvn	threadidal: thread exiting with uncas	abt exception	
E 12-05 16:38:3	605	com.android.spa	AndroidRun	FATAL EXCEPTION: main		
E 12-05 16:38:3	605	com.android.spa	AndroidRun	android.content.ActivityNotFoundExcep	tion: Unable	
E 12-05 16:30:3	605	com.android.spa	AndroidRun	at android.app.instrumentation.check	tartActivityF	
E 12-05 16:30:3	605	com.android.spa	AndroidRun	at android.app.Instrumentation.execSt	artActivity(I	
E 12-05 16:38:3	605	com.android.spa	AndroidRun	at android.app.Activity.startActivity	ForResult (Act	
E 12-05 16:38:3	605	com.android.apa	AndroidRun	at android.app.Activity.startActivity	(Activity. hav	
£ 12-05 16:38:3	605	com.android.sps	AndroidRun	at android.preference.Preference.		
E 12-05 16:30:3	605	com.android.spa	AndroidRun	at android.preference.PreferenceS		
E 12-05 16:38:3	605	com.android.spa	AndroidRun	at android.widget.AdapterView.per		
E 12-05 16:38:3	605	com.android.spa	AndroidRun	at android.widget.ListView.perfor		rrv
E 12-05 16:38:3	605	com.android.spa	AndroidRun	at android.widget.AbsListViewSPer		1 I Y i
E 12-05 16:30:3	605	com.android.spa	AndroidRun	at android.os.Handler.handleCallb		
E 12-05 16:38:3	605	com.android.spa	AndroidRun	at android.os.Handler.dispatchMes		
E 12-05 16:38:3	605	com.android.spa	AndroidRun	at android.es.looper.loop(Looper.		
E 12-05 16:38:3	605	com.android.spa	AndroidRun	at android.app.ActivityThread.mai		
E 12-05 16:38:3	605	com.android.spa	AndroidRun	at java.lang.reflect.Nethod.invok	Tho and	nlication Snaro Darts
E 12-05 16:30:3	605	com.android.spa	AndroidRun	at java, lang.reflect.Hethod, invok	ine ap	plication spare raits
E 12-05 16:38:3	605	com.android.spa	AndroidRun	at com.android.internal.os.Zygote	duluae •	waps wessaging wusic
E 12-05 16:38:3	005	com.android.spa	AndroidRun	at com.android.internal.op.Zygote	Innoces	ss com android.
£ 12-05 1613013	605	com.android.apa	Androidkun	at dalvis.system.nativestart.main	(p) 0000	55 connarian ora:
H 12-05 1013013	75	system_process	ACCLVICYN8	Force finishing activity com.an	chara r	aarte) has stannad
W 12-03 18:30:3	13	system_process	ACCIVICYNS	activity pause timeout for mistor	spare i	Jaris) rias slopped
a we we avoid the	1.14	ateres_preses	and the second second	Accuracy descent contract for more		
					lineyne	ectedly. Please try again
				Android SDK Content Lo	ипслре	cically. I lease try again.
	_					
						A REPORT OF A R
						Force close

# 4 Understand fundamental concepts



## 4.1. Android architecture

Source: Google Inc., 713px x 512px, <u>http://developer.android.com</u>

For details, please refer to

http://developer.android.com/guide/basics/what-is-android.html

### 4.2. Application components

- 1. Activities
  - Each activity is a single screen with user interface
  - Every screen is implemented as a subclass of android.app.Activity



- 2. Services
  - Run in background
  - No user interface
  - Designed for performing long-running tasks
  - Every service is implemented as a subclass of android.app.Service



- 3. Content providers
  - A shared set of application data
  - E.g. Contacts, Calendar (Android 4.0)
- 4. Broadcast receivers
  - A component designed to response to global broadcast announcements
  - System broadcast announcements E.g. change of battery status/network status

#### 4.3. Application resources

- Provide other resources to your apps. E.g. Image, Video, Audio, Text etc...
- Resource are put under **res** directory of your Android project
- Every resource is identified by a resource ID

 Resources can have qualifier names for alternative E.g. layout-land, values-ja, drawable-hdpi etc



#### 4.4. Define the app properties in Android

We use a file called **Manifest.xml** to describe information about your app to the system and Android Market as well. Inside this file, you can:

- 1. Define the package name
- 2. Define version code and version name
- 3. Define app icon
- 4. Define the components (Activities/Services/Receivers etc.)
- 5. Define the permission(s) should be granted



#### You may refer to

http://developer.android.com/guide/topics/manifest/manifest-intro.html for details.

# **5** Create your first application

1. Start Eclipse, select File > New > Project and choose Android > Android Project.

New Project	_ <b>D</b> X
Select a wizard	
<u>W</u> izards:	
type filter text	
<ul> <li>▷ ➢ General</li> <li>▷ ➢ Android</li> <li>◎ ➢ Android Project</li> <li>◎ ➢ Android Sample Project</li> <li>J<sub>u</sub><sup>-</sup> Android Test Project</li> <li>▷ ➢ CVS</li> <li>▷ ➢ Java</li> <li>▷ ➢ Maven</li> <li>▷ ➢ Examples</li> </ul>	

2. Type Hello Android in Project Name.

😂 New Androi	d Project
Create Andro Select project	id Project name and type of project
Project Name:	Hello Android
Oreate new	project in workspace
Create proje	ect from existing source
Create proje	ect from existing sample
🔽 Use default	location
Location:	E:/Softwares/eclipse-java-indigo-SR1-win32/eclipse/workspace/
-Working sets	
📃 Add proj	ec <u>t</u> to working sets
W <u>o</u> rking set	s; Sglect

3. Select Google APIs (API level 10).

New Android Project			
Select Build Target Choose an SDK to select a	sample from		0
Build Target			
Target Name	Vendor	Platform	API Level
Android 2.3.3	Android Open Source Project	2.3.3	10
Google APIs	Google Inc.	2.3.3	10

Note:

- You will have more **Build Target** once you downloaded more emulator images.
- Google APIs includes both standard Android and Google libraries such as Google Maps. Thus, if your application requires the use of Google Maps, you must use Google APIs as the build target.
- 4. Type com.hkbu.helloandroid in Package Name, check the box Create Activity

and choose 10 (Google APIs (Google Inc.)) as the Minimum SDK.

New Android Proj	ect 📃 🗖 🗾 🔀
Application Info Configure the new	Android Project
Application Name:	Hello Android
Package Name:	com.hkbu.helloandroid
Create Activity:	HelloAndroidActivity
Minimum SDK:	10 (Google APIs (Google Inc.)) 🔹
Create a Test Pro	ject
Test Project Name:	Hello AndroidTest
Test Application:	Hello Android Test
Test Package:	com.hkbu.helloandroid.test

5. Since we have checked the option **"Create Activity"**, the ADT plugin will automatically generate a stub activity and related resources right after the project is created.



### 5.1 Run and debug apps in emulator

 We can try to test the app in the emulator by choosing "Run" and select "Android Application".



2. The ADT plugin will automatically launch the emulator, install and run the app once the emulator is successfully booted.



Note:

 If you have more than one emulator created or device connected and satisfied the requirements, you will have a dialog to choose which emulator should be used for testing.

😂 And	Iroid Device Choo	ser						×
Select a	a device compatib pose a running And	le with target Goo droid device	gle APIs (Googl	le Inc.).				
	Serial Number		AVD Name		Target		Debug	State
	39325E2D105/	400EC	N/A		A 2.3.6	Online		
	emulator-555	4	Android_2.3	.3_GAPI	🖌 Google	e APIs (	Yes	Online
🔘 Lau	nch a new Android	d Virtual Device						
	AVD Name	Target Name		Platform	API Level	CPU/A	BI	Details
		No AVD availa	ble					Start

### 5.2 Run and debug apps in actual device

1. Make sure your device driver is properly installed otherwise the ADB will not be able to detect the present of your device.



 On your device, go to Settings > Applications > Development and enable the USB debugging option. If you are using Android 4.0, the path will be System settings > Developer options instead.



- 3. Run the application from Eclipse using the same steps shown in 5.1.
- 4. Select your actual device in **Android Device Chooser** window. (Please refer to the note in 5.1)

## 6. Create UI element using XML

All UI elements are built using **View** and **ViewGroup** objects in an application. The View can be extended to a **TextView**, **ImageView**, **WebView** etc, and the ViewGroup serves as layout architecture which describes how the View to be organized in an activity. Apart from building UI



directly in the source code, we can also create the interface using XML format. By using the XML, we can simply our code and make debugging easier.

The XML layout file is saved in **res/layout**. By using the Hello Android example, you will find the initial UI is defined in main.xml. When you open the main.xml, Eclipse will show you a graphical UI editor for the interface preview.



However, the graphical UI editor is not sophisticated enough. The final appearance of the interface layout may differ from the preview in the editor. Therefore, the best practice is to create the UI elements by editing the XML code directly. To do this, you can click on the **XXX.xml** tab.

#### 6.1. Understand the XML layout structure

Using the main.xml as an example:

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

The LinearLayout is a ViewGroup which holds its children in a single column or a single row. The above LinearLayout has only one child – TextView. Android uses TextView to display text to users. For a complete list of ViewGroup and View, please refer to <a href="http://developer.android.com/reference/android/view/ViewGroup.html">http://developer.android.com/reference/android/view/ViewGroup.html</a> and <a href="http://developer.android.com/reference/android/view/View.html">http://developer.android.com/reference/android/view/ViewGroup.html</a> respectively.

Each kind of ViewGroup or View has its own attributes. The attributes usually start with **android:xxx** where the **xxx** is the property name of an attribute. Different ViewGroup and View share some common attributes but some of them are unique. For a complete set of attributes, please refer to the **Reference** section in developer.android.com.

To have a quick outlook on different layouts and UI elements, please go to <u>http://developer.android.com/resources/tutorials/views/index.html</u> for the samples. Now, we will see how to add an image just below the text in our main.xml.

#### 6.2. Add text and image to the UI

- Prepare an image. You may download this <u>http://developer.android.com/assets/images/home/ics-android.png</u> as an example.
- 2. Put the image under **res/drawable**. If this folder is not found, you can create it manually using the **Package Explorer** in Eclipse.

New Folder	
Folder Create a new folder resource.	
Enter or select the parent folder:	
Hello Android/res	
<ul> <li>☆ ↔</li> <li>↔</li> <li></li> <li>↔</li> <li></li> <li>↔</li> <li></li> <li></li></ul>	
Folder name: drawable	

3. Drag the image directly to the **drawable** folder through the **Package Explorer**. In the **File Operation**, choose **Copy files**. You also need to rename the file if the characters are not in (2.7) (0.9) (1) and (1)

characters are not in (a-z), (0-9),  $(_)$  and (.).

File Operation
Select how files should be imported into the project:
© Copy files
Create link locations relative to: PROJECT_LOC
Configure Drag and Drop Settings
OK Cancel

4. In the main.xml, add the following codes below the TextView.



- The **ImageView** is used to show an image in the interface.
- The android:src is used to specify the source of the image file. Here the path and the file name are @drawable/ics\_android. (File extension is omitted)
- The android:layout\_width and android:layout\_height="wrap\_content" is used to specify the ImageView size which is the same as the image itself.
   For the list of complete attributes and values of ImageView, please refer to <a href="http://developer.android.com/reference/android/widget/ImageView.html">http://developer.android.com/reference/android/widget/ImageView.html</a>.
- 5. You can press the Graphical Layout tab to preview the result.
- 6. Run your app in the emulator and see the actual result. The final appearance should look likes this:



- 7. The above example uses LinearLayout which is the most basic layout in Android. If we need to achieve the result below, you have to complete the following changes:
  - Create a new layout XML file and use the RelativeLayout
  - Resize the ImageView
  - Add one more ImageView and refer to the new image
  - Add more attributes to the TextView and ImageView



 Now we need to create a new layout file called main\_relative inside the res/layout. Right Click on the layout folder in Package Explorer and choose New > Other > Android > Android XML Layout File. Type main\_relative for File and select RelativeLayout in Root Element. Hit Finish when you are done.

New Android	Layout XML File
New Android Creates a new	Layout XML File Control Layout XML file.
Resource Type:	Layout v
Project:	Hello Android 🔹
File:	main_relative
Root Element:	
RadioButto	n
RadioGrou     RatingBar	0
RelativeLay	out
ScrollView	
SlidingDrav	ver
?	< <u>B</u> ack <u>N</u> ext > <u>Finish</u> Cancel

9. Add a **TextView** to the RelativeLayout by inserting following code:

<textview< th=""><th></th></textview<>	
android:layout_width=" <i>fill_parent</i> "	
android:layout_height="wrap_content"	
android:text="@string/hello"/>	

- Since elements in RelativeLayout are depending to each others. We have to add ID to each element as an identifier. To add an ID for the TextView, we can use android:id="@+id/MyText". (Now the ID of this TextView is MyText)
- Next, we can add two ImageViews for our two images. The second image can be downloaded via this link (Also put this image in res/drawable, replace the (-) with (\_) for the file name):

http://developer.android.com/assets/images/home/maps-large.png

12. Insert the following codes for the first ImageView:

```
<ImageView
android:id="@+id/ImageView1"
android:layout_width="138dp"
android:layout_height="102dp"
android:src="@drawable/ics_android"
android:layout_below="@id/MyText"/>
```

 We specify the width and height in android:layout\_weight and android:layout\_height. Please note that we should use dp for the unit of dimension. **dp** stands for **D**ensity-independent **P**ixel which is an abstract unit based on the physical density of the screen. This unit uses a 160 dpi screen as a reference. By using **dp**, we can ensure the same physical image dimension on every device with different screen sizes and resolutions. For details, please refer to:

http://developer.android.com/guide/practices/screens\_support.html

- The android:layout\_below:"@id/MyText" defines the first ImageView should be placed below MyText (TextView).
- Go to /src/com.hkbu.helloandroid/HelloAndroidActivity.java, change (R.layout.main) to (R.layout.main\_relative) for pointing to our new layout file.

5554:Android_2.3.3_GAPI								l	-	×
± ∎ <b>€</b> 8:17										
Hello Android										
Hello World, HelloAndroidActivity!										
The to work, Helio Aldroid Activity:								3) () ()		
		_@	<b>^</b> #	\$	_%	c^	- &	*		
	1	Z	3	4	5	6	/	8	9	0
	Q	W	E	R	Τ <sup>ι</sup>	Y	U -	Ι	0	Р
	A	s `	D	F	G	H <	<u> </u>	K	L	DEL
	숩	Z	X	С	V	В	N	M		÷
	ALT	SYM	0				→	1?	,	ALT

• Till now, the interface will look likes this:

- 13. To add the second image, we can copy the codes of the first ImageView and paste them as the last element of the RelativeLayout. However, you have to modify the following attributes:
  - Change the value of android:id to @+id/ImageView2
  - Change the value of android:src to @drawable/maps\_large
  - Add android:layout\_toRightOf="@id/ImageView1". This means that the second ImageView will be placed on the right-hand side of the first ImageView.
  - Till now, the interface will look likes this:



- 14. We can add margins for the TextView and ImageView to make them not to close to each others.
  - Add android:layout\_marginBottom="10dp" to the TextView
  - Add android:layout\_marginRight="10dp" to the first ImageView
  - Till now, interface will look likes this:

5554:Android_2.3.3_GAPI								-   I	1 ×
ti 👔 🦻 8:52									
Hello World, HelloAndroidActivity!									
Hartan Ver De Alexandre Les Alexandre De Ale							8 0 0		
	1 <sup>!</sup> 2	<sup>©</sup> 3 <sup>#</sup>	4 \$	5 <sup>%</sup>	6 ^	7&	8*	9(	0)
	Q W	~ E "	R	т {	γ }	U -	I	0+	Р
	A S	`D	F	G ]	Η <	ر ×	К	L	DEL
	순 Z	X	С	V	В	Ν	M		L.
	ALT SY	м @		-	_	→	1?	,	ALT

15. In the next step, we will align the two images to the center of the screen horizontally. Since RelativeLayout does not support centering more than one child element in the same height, we have to use LinearLayout to encapsulate the two ImageViews. Add the LinearLayout and move the ImageViews into it by using the following code:

- We moved the attribute android:layout\_below="@id/MyText" from the first ImageView to the LinearLayout to make sure our LinearLayout will be placed under the TextView. Also, remove the same attribute for the second ImageView.
- You also need to remove android:layout\_toRightOf="@id/ImageView1" from the second ImageView since it is only for RelativeLayout before.
- The default direction for aligning elements of LinearLayout is vertical, we have to use android:orientation="horizontal" for horizontal alignment.
   android:gravity="center" is used to center ImageViews in the screen.
- The final result:



## 7. Handle user input and more interface design

In this section we will learn how to handle simple user input and manipulate the UI elements in Java code. We will add a **TextField** and a **Button** under our images.



#### 7.1. Accept user input by adopting EditText and Button

- 1. Add an ID to the LinearLayout (E.g. android:id="@+id/MyLinearLayout")
- 2. Add an EditText (For inputting text) below the LinearLayout

```
<EditText android:id="@+id/MyTextField"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:layout_below="@+id/MyLinearLayout"/>
```

3. Add a Button at the bottom of the screen.

#### <Button

```
android:id="@+id/MyButton"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:text="Click Me!" />
```

- 4. When users click on this button, the app should perform following actions:
  - The text in the TextView will be change to those users typed in the EditText
  - The images in the two ImageViews will be swapped.

 From now on, we will write the Java code to listen the action when users click on the button. Add the following code above the "onCreate()" method in HelloAndroidActivity.java.

Button myBtn;
<pre>TextView myTextView;</pre>
<pre>EditText myEditText;</pre>
<pre>ImageView myImgView1;</pre>
<pre>ImageView myImgView2;</pre>

Here we create four null references for our Button, TextView and two ImageViews respectively. Import any necessary package suggested by Eclipse.

 In the onCreate() method, add the following code below the setContentView() method:

```
myBtn = (Button)findViewById(R.id.MyButton);
myTextView = (TextView)findViewById(R.id.MyText);
myEditText = (EditText)findViewById(R.id.MyTextField);
myImgView1 = (ImageView)findViewById(R.id.ImageView1);
myImgView2 = (ImageView)findViewById(R.id.ImageView2);
```

- We use the **findViewById()** method to get the reference to the UI created using XML format.
- The R class will be automatically generated by AAPT (Android Asset Packaging Tool) and updated once you added or updated a resource. We will always use this class to handle all the elements located in the res/ directory. For details, please refer to

http://developer.android.com/guide/topics/resources/accessing-resources. html

7. Add the following code below the onCreate() method to create an anonymous implementation of OnClickListener:

```
private OnClickListener myBtnListener = new OnClickListener() {
    @Override
    public void onClick(View arg0) {
    }
};
```

We may use Eclipse's quick-fix feature to fix most of errors. Note that for onClickListener you should import android.**view.View**.OnClickListener

8. Inside the onCreate() method, add the following code to register the myBtnListener to myBtn:

myBtn.setOnClickListener(myBtnListener);

9. Add the following code to the onClick function:

myTextView.setText(myEditText.getText());

- We use the getText() and setText() method to grab the text from the EditText and then set it to the TextView.
- 10. Then add the following code to swap the images for our ImageViews:

myImgView1.setImageResource(R.drawable.maps\_large);

myImgView2.setImageResource(R.drawable.ics\_android);

- We use the **R** class again to refer to our images. The **R** class has different subclasses for each type of resources. For image, it is **R.drawable**.
- 11. That's it! Try to run it on the emulator or on your device. For a list of R subclasses, please refer to

http://developer.android.com/guide/topics/resources/available-resources.html



## 7.2. Activity lifecycle

In our previous example, we create references and register event listener inside the onCreate() method. It is because the onCreate() method will be called once the activity is started.



Source: Google Inc., 545px x 711px, <u>http://developer.android.com</u> Things you have to know: Entire lifetime: Happens between onCreate() and onDestory Visible lifetime: Happens between onStart() and onStop() Foreground lifetime: Happens between onResume() and onPause() For details, please refer to <u>http://developer.android.com/reference/android/app/Activity.html</u>

### 7.3. Use TableLayout for structural elements and data

Sometimes if you like to position your UI elements or displaying a set of data in rows and columns, using TableLayout will be much more convenient than using LinearLayout or RelativeLayout. Suppose we have the following set of data need to be shown:

Chinese (Traditional):	Chinese - Hong Kong SAR (zh_HK)
Arabic:	Israel (ar_IL)
Bulgarian:	Bulgaria (bg_BG)
Catalan:	Spain (ca_ES)
Czech:	Czech Republic (cs_CZ)
Danish:	Denmark(da_DK)

 Start a new project in Eclipse using the steps written in section 5. For the Project Name and Package Name, we will use Table Layout Sample and com.hkbu.tablelayoutsample respectively.

New Android Proj	ect	
Application Info Configure the new	Android Project	<b>(</b>
Application Name:	Table Layout Sample	
Package Name:	com.hkbu.tablelayoutsample	
Create Activity:	TableLayoutSampleActivity	
Minimum SDK:	10 🗸	

2. Edit the main.xml, change the LinearLayout to TableLayout. Remove the **android:orientation="vertical"** attribute and the default TextView.



 To add rows to the TableLayout, we can use <TableRow></TableRow> to enclose our UI elements. For example, we can use the following code to construct the first row of text.

```
<TableRow>
    <TextView
    android:layout_width="fill_parent"
```

```
android:layout_height="wrap_content"
android:text="Chinese (Traditional):" />
<TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Chinese - Hong Kong SAR (zh_HK)" />
</TableRow>
```

\*The number of elements in **<TableRow></TableRow>** equals to the number of columns in a row. We do not need to explicitly specify the column.

- 4. Please use the same way for inserting the rest of data.
- 5. Till now, the interface will look likes this:



6. As you can see, if the text is too long in a cell, it will not be wrapped automatically. Therefore, we have to add an additional attribute android:shrinkColumns to <TableLayout>. This attribute will shrink a column based on the index provided. You can shrink more than one column by separated index using comma. Please note that the column is a zero-based index. In our example, add the following attribute to <TableLayout>

android:shrinkColumns="1"

After adding this attribute, the result will be like this:



7. If we need a cell to span over columns, all you need is to add an attribute called android:layout\_span to the elements in <TableRow>. For example, if we need to add a TextView as a header to the table, you can add the following code to be the first element in <TableLayout>.

<tablerow></tablerow>			
<textvie< th=""><th>W</th><th></th></textvie<>	W		
android:layout_width=" <i>fill_parent"</i>			
andro	oid:layout_height="wrap_	content"	
andro	<pre>pid:text="System Locales</pre>	n.	
andro	oid:layout_span="2"		
andro	<pre>pid:gravity="center" /&gt;</pre>		
5554:Android_2.3.3_GAPI		_ • • <u>• × •</u>	
	랿 💵 💈 5:23		
Table Layout S	ample		
Chinese (Traditio	System Locales nal):Chinese - Hong Kong SAR (zh_HK)		
Arabic:	Israel (ar_IL)		
Dulardar	$D_{1} = \frac{1}{2} \left( b = DC \right)$		
Bulgarian: Catalan:	Bulgaria (bg_BG) Spain (ca ES)		
Bulgarian: Catalan: Czech: Danish:	Bulgaria (bg_BG) Spain (ca_ES) Czech Republic (cs_CZ) Denmark(da_DK)		

- To add space between each row, we can add some padding to the <TableRow> For example: android:padding="5dp"
- To add space between each column, we can add some margin to the elements (TextViews in this case). For example: android:layout\_marginLeft="5dp"
- 10. Till now, the interface will look like this:

5554:Android_2.3.3_GAPI		_	
	랿 💵 🦻 5:20		
Table Layout Samp	le		
Sys	tem Locales		
Chinese (Traditional)	: Chinese - Hong Kong SAR (zh_HK)		
Arabic:	Israel (ar_IL)		6
Bulgarian:	Bulgaria (bg_BG)		
Catalan:	Spain (ca_ES)	C	•
Czech:	Czech Republic (cs_CZ)		
Danish:	Denmark(da_DK)		

### 7.4. Use ScrollView for large content

When using the TableLayout for showing a large volume of data, the TableLayout may not be shown completely, as it is larger than the screen size.

5554	4:Android_2.3.3_GAPI		
		👬 💵 🛃 5:04	04
Т	able Layout Sampl	e	
	Syst	tem Locales	
C	Chinese (Traditional):	Chinese - Hong Kong SAR (zh_HK)	
A	rabic:	Israel (ar_IL)	
В	Bulgarian:	Bulgaria (bg_BG)	
C	atalan:	Spain (ca_ES)	
C	Zech:	Czech Republic (cs_CZ)	
D	)anish:	Denmark(da_DK)	
Ģ	German:	Austria (de_AT)	
Ģ	German:	Switzerland (de_CH)	$\frac{1 \cdot 2^{\textcircled{0}} 3^{\#} 4^{\$} 5^{\textcircled{0}} 6^{\land} 7^{\textcircled{0}} 8^{\$} 9^{(0)}}{1 \cdot 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - $
Ģ	German:	Germany (de_DE)	
Ģ	German:	Liechtenstein (de_LI)	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$
Ģ	Greek:	Greece (el_GR)	
E	nglish:	Australia (en_AU)	Some data cannot be shown since
E	nglish:	Canada (en_CA)	the screen size is not large enough
E	nglish:	Britain (en_GB)	
E	nglish	Ireland (en_IE)	

- To solve this problem, we can use ScrollView is to enable scrolling for the TableLayout. The ScrollView extends the FrameLayout and it will enable the scroll functionality of its child. Please note that the ScrollView can only contain one child. If you want to scroll more than one UI elements at the same time, you have to encapsulate UI elements first.
- 2. Add ScrollView to enclose your <TableLayout>:

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent">
<TableLayout>... ...</TableLayout>
</ScrollView>
```

\*Note: You have to insert

xmlns:android="http://schemas.android.com/apk/res/android" for the outermost

element.

3. Now you can view more data by scrolling the table up and down.



# 8. Prepare to publish your app

Before you publish the app to Android Market or distribute it to someone else, you need to design an icon for your app instead of using the default one. Also, you may need to change the app name according to your preference later on.

### 8.1. Prepare the app icon

For the app icon, we have to prepare at least three different sizes for different screen resolutions. Namely: ldpi, mdpi, hdpi. (There are also xhdpi and tvdpi)

- Please prepare an app icon and resize it to 36x36px for ldpi, 48x48 for mdpi and 72x72 for hdpi. For the app icon design guideline. Please refer to <u>http://developer.android.com/guide/practices/ui\_guidelines/icon\_design\_lau</u> <u>ncher.html</u>. You can also download some free icons with different sizes at www.iconarchive.com
- 2. Put the icon with the same file name (E.g. icon.png) into drawable-hdpi (72x72), drawable-mdpi (48x48) and drawable-ldpi (36x36) respectively.
- 3. Open the AndroidManifest.xml, change

android:icon="@drawable/ic\_launcher" to android:icon="@drawable/icon". You can also delete the default launcher icon.



4. The new icon will be shown in the launcher after you deploy your app into your emulator or device.



### 8.2. Change the app name and apply localization

When you look at the AndroidManifest.xml file, you can see an attribute android:label="@string/app\_name" is already added to the <application> tag. The @string/app\_name refers to a value of an elements call app\_name located in value/strings.xml. By storing the string in the values/strings.xml, we can easily localize our apps by using different qualifier name (language code this time). Notes: zh-rTW for Traditional Chinese, zh-rCN for simplified Chinese

- 1. Go to values/strings.xml, change the app name to HKBU.
- 2. Create a folder called values-zh-rTW in res directory.
- 3. Create a strings.xml file inside the values-zh-rTW and insert the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="app_name">香港浸會大學</string>
</resources>
```

- 4. Run your app in the emulator or device, you can see the app name is changed.
- 5. Now, change the locale setting in the emulator or device by going in Settings > Language and keyboard > Select language > 中文 (繁體). You should see the name of the app will be automatically changed to Chinese.



\*Note: If the launcher cannot display the Chinese app name, try to force close the launcher in Settings > Applications > Manage applications > All > Launcher > Force stop

### 8.3. Sign and export the APK

For every Android app, you have to sign it before uploading it to Android Market or distributing it to other users. The certificate used by ADT for deploying your app in the emulator or device is a debug key only. When exporting the Android app in Eclipse, you can create your own private key if you have not created it before.

 In Eclipse, highlight your project and select File > Export > Android > Export Android Application.

Export	
Select	
Select an export destination:	
type filter text	
<ul> <li>▷ ➢ General</li> <li>▷ ➢ Android</li> <li>○ ➢ Install</li> <li>▷ ➢ Run/Debug</li> <li>▷ ➢ Tasks</li> <li>▷ ➢ Team</li> <li>▷ ➢ XML</li> </ul>	
? < Back Next > Einish	Cancel

- 2. Click **Next** to continue if the selected project is correct.
- 3. Select **Create new keystore**, choose the location for saving the new keystore and enter the password.

👙 Export A	ndroid Application			
Keystore s	selection			
<ul> <li>Use exis</li> <li>Create r</li> </ul>	<ul> <li>Use existing keystore</li> <li>Create new keystore</li> </ul>			
Location:	E:\Softwares\eclipse-java-indigo-SR1-win32\eclipse\wor			
Password:	•••••			
Confirm:	••••••			
?	< Back Next > Einish Cancel			

4. Enter your personal information in **Key Creation** step. Below is a sample screenshot:

Export Android Application			
Key Creation	<b>(</b>		
Alias:	androidworkshop		
Password:	••••••		
Confirm:	•••••		
Validity (years):	100		
First and Last Name:	Felix Tam		
Organizational Unit:	Department of Computer Science		
Organization:	Hong Kong Baptist University		
City or Locality:	Hong Kong		
State or Province:	China		
Country Code (XX):			
< Back			

5. Choose your location for saving the published APK file.

😂 Export Android Ap	plication		
Destination and key/certificate checks			
Destination APK file: Certificate expires in 1	a-indigo-SR1-win32\Table Layout Sample. .00 years.	.apk Browse	
? < <u>B</u> ac	k <u>N</u> ext > <u>Finish</u>	Cancel	

6. Now you are ready to distribute your APK file to Android Market or install it directly in any devices.

# 9. Reference and learning resources:

- Official Android developer website (especially the Resources section): <u>http://developer.android.com/index.html</u>
- Many common questions raised by other developers previously: <u>http://stackoverflow.com/</u>
- Many Android tutorials with complete source code: <u>http://www.anddev.org</u>
- Create your app interface online
   <u>http://www.droiddraw.org/</u>
- 3<sup>rd</sup> party UI development framework: https://github.com/cyrilmottier/GreenDroid
- Different ways for implementing listeners in Android app <u>http://tseng-blog.nge-web.net/blog/2009/02/14/implementing-listeners-in-you</u> <u>r-android-java-application/</u>
- Sams teach yourself Android application development in 24 hours Online access from HKBU library: <u>http://www.hkbu.edu.hk/lib-cgi/ejour/safari.pl?t=9780132786904</u>)
- Special Interest Group on Innovative Software homepage: <u>http://www.comp.hkbu.edu.hk/~sigis/</u>

- End -