

ผู้สอน: อ.สุธน แซ่ว่อง และคณาจารย์

วัตถุประสงค์

1. รู้จักแนวทางและการใช้เครื่องมือในการพัฒนาโปรแกรมบน Android Platform
2. รู้จักองค์ประกอบพื้นฐานของ Android
3. เข้าใจการใช้งาน XML ในการสร้าง User Interface
4. เข้าใจแนวคิด และประโยชน์ของการใช้งาน Intent

กำหนดส่งงาน และวิธีส่งงาน

วันที่ 6 ธันวาคม พ.ศ.2554 ก่อนเวลา 23.00 น. โดยส่งงานผ่าน LMS

สัดส่วนคะแนน

- Checkpoint ในระหว่างการทดลอง 30%
- การบ้านหลังการทดลอง 10%
- สอบ 60%

เอกสารอ้างอิง

- เอกสารที่มาพร้อมกับตัวติดตั้ง Android SDK (อยู่ในไดเรกทอรี docs)

งานหลังการทดลอง

นศ. เขียนโปรแกรมที่มีอย่างน้อย 1 Activity โดย Activity นี้จะต้องทำงานประสานกับองค์ประกอบอื่นของ Android 1 อย่าง (เช่น เรียก Activity ที่เขียนขึ้นเอง, เรียก Activity ที่มีอยู่แล้ว, ทำงานร่วมกับ Broadcast Intent, ทำงานร่วมกับ Service) ทั้งนี้นักศึกษาจะต้อง zip ไฟล์ต่อไปนี้ เป็นไฟล์เดียว แล้วส่งทาง LMS

- ไฟล์ซอร์สโค้ด (Eclipse Project ที่ทำการ clean แล้ว)
- ไฟล์รายงานในรูปแบบ PDF บอกรายละเอียดดังต่อไปนี้
 - วัตถุประสงค์ของโปรแกรม
 - องค์ความรู้จากแลบหรือภายนอก ที่ได้นำไปใช้ในโปรแกรม
 - ผลการทำงาน
- ไฟล์อื่นๆ หากโปรแกรม อ่านข้อมูลจากไฟล์

** โปรแกรมที่ส่ง ควรเป็นโปรแกรมง่ายๆ แค่เพียงใช้ความรู้ที่เรียนไป อย่างไรก็ตาม นักศึกษาสามารถใช้ความรู้นอกเหนือจากห้องเรียนได้ แต่จะต้องอธิบายเอาไว้ให้ชัดเจนในรายงาน



ANDROID



1 DEVELOPMENT ENVIRONMENTS

1.1 PREPARATIONS

ดาวน์โหลดไฟล์ 3 ไฟล์จาก <http://fivedots.coe.psu.ac.th/~suthon/android> ดังแสดงข้างล่าง

ไฟล์	คำอธิบายเพิ่มเติม
android-sdk_r11-windows-plat2-2.zip	Software Development Kit เป็นเครื่องมือหลัก ใน การพัฒนาโปรแกรมบน Android Platform ทั้งนี้ไฟล์นี้ได้ทำการรวม Android Platform 2.2 ไว้ (ใน SDK ตัวเดียว สามารถติดตั้ง Android Platform ได้หลายเวอร์ชัน) อย่างไรก็ตาม จำเป็นต้องมีการติดตั้ง Java SDK ไว้ร่วมด้วย
eclipse-java-indigo-win32.zip	Integrated Development Environment สำหรับพัฒนาโปรแกรม (ถูกสร้างขึ้น สำหรับพัฒนาโปรแกรมจาวา อย่างไรก็ตาม หากมีการติดตั้ง Plug-in (หรือ ปรับแต่ง) ก็สามารถนำไปพัฒนาโปรแกรมประเภทต่างๆ ได้
ADT-11.0.0.zip	Android Development Tools เป็น Plug-in สำหรับ Eclipse IDE เพื่อพัฒนา โปรแกรมบน Android Platform

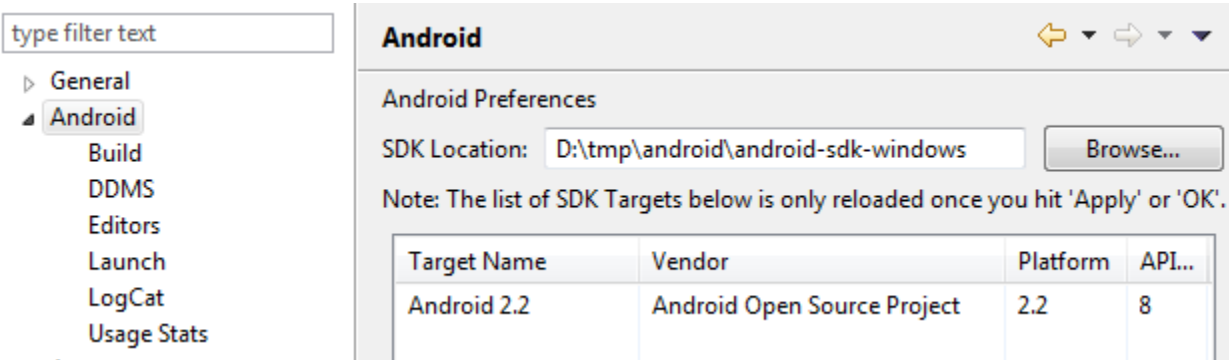
หมายเหตุ ให้แตกไฟล์เฉพาะ Eclipse และ Android SDK โดยให้แตกไฟล์โดยตรง (ไม่มีการสร้างไดเรกทอรีเพิ่มเติม) ทั้งนี้ตลอด เอกสาร จะใช้ \$LABDIR แทนไดเรกทอรีที่นักศึกษาได้แตกไฟล์ไว้

1.2 ECLIPSE & ADT & SDK

ในการทดลองตอนนี้ เราจะทำการติดตั้ง ADT ให้กับโปรแกรม Eclipse พร้อมทั้งระบุตำแหน่งของ SDK ให้ ADT รับรู้

1. เปิดโปรแกรม Eclipse (ที่ได้จากการแตกไฟล์ในตอน 1.1)
2. กำหนด Workspace ของ Eclipse เป็น \$LABDIR/workspace
3. เลือกเมนู Help -> Install New Software
4. กดปุ่ม Add (บริเวณมุมขวา) เพื่อเรียกไดอะล็อกสำหรับกำหนด Repository ของปลั๊กอินที่จะติดตั้งเพิ่มเติม
5. กดปุ่ม Archive เพื่อเรียกไดอะล็อกสำหรับเลือกไฟล์บนเครื่อง โดยให้เลือกไฟล์ ADT ที่ดาวน์โหลดไว้ในตอน 1.1
6. ทำเครื่องหมายถูกหน้า Developer Tools และนำเครื่องหมายถูกออกจากช่อง Contact all update sites ... (เพื่อความรวดเร็วในการติดตั้งสำหรับการลงปฏิบัติการ อย่างไรก็ตาม หากนักศึกษาติดตั้งที่เครื่องตนเอง สามารถเช็ค หน้าจอนี้ได้ด้วย เพื่อให้ Eclipse ดาวน์โหลดโปรแกรมใหม่ล่าสุด)
7. ตอบตกลงไปตามขั้นตอนต่างๆ จน Eclipse แจ้งว่า ต้องการ Restart ตัว Eclipse ให้เลือก Restart Now แล้วรอให้ Eclipse เริ่มทำงานอีกครั้ง

8. เลือกเมนู **Window -> Preferences** เพื่อเปิดไดอะล็อกสำหรับกำหนดค่าการทำงานของ Eclipse โดยให้เลือกหมวด **Android** จากด้านซ้ายมือ
9. กำหนด **SDK Location** ไปยังตำแหน่งของ SDK ที่ดาวน์โหลดไว้ตอน 1.1 แล้วกด **Apply**
10. ตรวจสอบความถูกต้อง โดย Eclipse จะแสดงให้เห็นว่า ตรวจพบ **Android Platform 2.2** ดังแสดงในรูปที่ 1



รูปที่ 1 ผลการติดตั้ง ADT และ SDK ให้กับ Eclipse

1.3 ANDROID VIRTUAL DEVICES

การพัฒนาโปรแกรมบน Android นอกจากทำการรันโปรแกรมบนอุปกรณ์ Android จริงแล้ว ตัว Android SDK ยังมีเครื่อง Android จำลอง (AVD: Android Virtual Devices) เพื่อให้ นักพัฒนามีความสะดวกในการพัฒนามากขึ้น ในการทดลองตอนนี้ เราจะทำการสร้าง AVD เพื่อใช้ในการทดสอบโปรแกรมที่เขียนขึ้น

1. เลือกเมนู **Window -> Android SDK and AVD Manager** เพื่อเรียกไดอะล็อกสำหรับการจัดการ AVD
2. กดปุ่ม **New** เพื่อเรียกไดอะล็อกสำหรับการสร้าง AVD
3. กำหนดค่าต่างๆ ดังแสดงในรูปที่ 2
4. เลือก AVD ที่สร้างไว้ แล้วกดปุ่ม **Start** เพื่อเปิด Virtual Devices ซึ่งจะใช้เวลาพอสมควรในการบูต เมื่อการบูตเสร็จสิ้น AVD จะเข้าสู่หน้า Home Screen ดังแสดงในรูปที่ 3

New Snapshot

Name: AVD

Target: Android 2.2 - API Level 8

ABI: ARM (armeabi)

SD Card:

☒ Size: 128 MiB

☐ File: Browse...

Snapshot:

☐ Enabled

Skin:

☒ Built-in: HVGA

☐ Resolution: x

Hardware:

Property	Value	
Abstracted LCD density	160	
Max VM application hea...	24	

New... Delete

รูปที่ 2 ข้อกำหนดในการสร้าง Android Virtual Device



รูปที่ 3 Android Virtual Device

2 FIRST PROGRAM

2.1 ECLIPSE PROJECT

ทำการสร้าง Android Project พร้อม Activity จำนวน 1 อัน แล้วทดลองรันโปรแกรม

1. เลือกเมนู **File -> New -> Project**
2. เลือกชนิดโปรเจ็คเป็น Android Project
3. กำหนดรายละเอียดของโปรเจ็คดังรูปที่ 4 แล้วกดปุ่ม **Finish**
4. โปรเจ็คใหม่ถูกสร้างขึ้น โดยจะมีชื่อโปรเจ็ค **First** ปรากฏขึ้นบริเวณด้านซ้ายมือบนของ Eclipse
5. ทำการคลิกขวาที่โปรเจ็ค **First** เลือก **Run as -> Android Application** แล้วสังเกตผลบน AVD
6. กดปุ่ม **Home** (ปุ่มรูปบ้านด้านขวามือของ AVD) เพื่อกลับสู่หน้า **Home Screen** แล้วกดปุ่ม **Application Drawer** บริเวณล่างกลาง เพื่อเรียกดูรายชื่อของโปรแกรมที่ถูกติดตั้ง จะพบว่าโปรแกรม **First** ได้ถูกติดตั้งเป็นที่เรียบร้อยแล้ว

Project name: first

Contents

- ☒ Create new project in workspace
- ☐ Create project from existing source
- ☒ Use default location

Location: D:/tmp/android/workspace/first Browse...

☐ Create project from existing sample

Samples: ApiDemos

Build Target

Target Name	Vendor	Platform	API ...
<input checked="" type="checkbox"/> Android 2.2	Android Open Source Project	2.2	8

Properties

Application name: First

Package name: coe.lab

☒ Create Activity: FirstActivity

Min SDK Version: 8

รูปที่ 4 รายละเอียดการตั้งค่าสำหรับโปรเจ็ค **First**

2.2 ANDROID FILES

ทำการค้นหาไดเรกทอรีและชื่อไฟล์ที่ทำหน้าที่ต่างๆ ดังต่อไปนี้

การทำงาน	ที่อยู่ของไฟล์
กำหนดข้อความที่แสดงใน <code>FirstActivity</code>	
กำหนดตำแหน่งและรูปแบบการจัดวางของข้อความ	
ได้คําจากของ <code>FirstActivity</code>	
ไฟล์ <code>R.java</code>	

2.3 RESOURCE FILE & UI

เปิดไฟล์ `R.java` เพื่อแก้ไขข้อความที่แสดงบน `TextView` ของ `FirstActivity` เป็น `Hello world`

```
<string name="hello">Hello World</string>
```

เปิดไฟล์ `R.java` เพื่อกำหนดสี โดยให้ชื่อเป็น `greeting`

```
<color name="greeting">#FF0000</color>
```

เปิดไฟล์ `layout/main.xml` เพื่อเพิ่มปุ่ม พร้อมระบุ Click Handler ชื่อ `doGreet`

```
<Button android:text="@string/click_me" android:textColor="@color/greeting"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:onClick="doGreet" />
```

เปิดไฟล์ `FirstActivity.java` เพื่อเพิ่มโค้ดของ Click Handler ทั้งนี้จะต้องมีการอิมพอร์ต `android.view.View` และ `android.widget.Toast`

```
public void doGreet(View v){
    Toast.makeText(this, "Hello world", Toast.LENGTH_LONG).show();
}
```

รันโปรแกรม สังเกตผล

2.4 ID & R FILE

เพื่อความเร็วในการทำงานของโปรแกรมเมื่อทำงานบนอุปกรณ์มือถือ ไฟล์ XML ทั้งหมดจะถูกแปลงเป็นไบนารี และอ้างอิงผ่าน ID เป็นเลขจำนวนเต็มขนาด 4 ไบต์ ทั้งนี้หมายเลข ID จะถูกสร้างขึ้นอัตโนมัติใน R.java

```
public final class R {
    public static final class attr {
    }
    public static final class color {
        public static final int greeting=0x7f050000;
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int click_me=0x7f040002;
    }
}
```

เปิดไฟล์ `layout/main.xml` กำหนด ID ให้กับ `TextView` ที่ใช้แสดงข้อความบน `FirstActivity` หลังการแก้ไขให้สังเกตการเปลี่ยนแปลงในไฟล์ `R.java`

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

เปิดไฟล์ `FirstActivity.java` เพื่อเพิ่มโค้ดด้านล่างไปในส่วนบนสุดของฟังก์ชัน `doGreet()` ของ `FirstActivity` ทั้งนี้จะต้องมีการอิมพอร์ต `android.widget.TextView` หลังการแก้ไข รันโปรแกรมแล้วสังเกตผล

```
TextView tv = (TextView)findViewById(R.id.greet_label);
tv.setTextColor(getResources().getColor(R.color.greeting));
```

CP-01

แสดงโปรแกรมที่แก้ไขตามการทดลองตอนที่ 2 เรียบร้อยแล้ว ทั้งนี้ให้กำหนดข้อความที่จะแสดงผ่าน `Toast` ไว้ใน

`strings.xml` แทนการเขียนลงไปในโค้ดจาวา

2.5 LOGGING & DDMS VIEW

เมื่อมีความจำเป็นต้องติดตามการทำงานของโปรแกรม วิธีที่ใช้ทั่วไปคือ การพิมพ์ข้อความเพื่อแสดงขั้นตอนการทำงาน แต่ในโปรแกรม Android ถูกออกแบบให้รวมวิธีการแสดงขั้นตอนการทำงานด้วยวิธีการล็อก (logging) โดยแบ่งการล็อกออกเป็นหลายระดับ เช่น **d** (debug), **i** (info), **e** (error), **wtf** (what the f_ck) โดยนักพัฒนาควรเลือกระดับการล็อกอย่างเหมาะสม นอกเหนือจากนั้น ในการล็อกแต่ละครั้ง นักพัฒนาสามารถระบุแท็ก (Tag) เพื่อช่วยในการกรองล็อกเพื่อง่ายต่อการอ่านอีกด้วย

เปิดไฟล์ `FirstActivity.java` เพื่อเพิ่มโค้ดด้านล่างไปในส่วนบนสุดของฟังก์ชัน `doGreet()` ของ `FirstActivity` ทั้งนี้จะต้องมีการอิมพอร์ต `android.util.Log` หลังการแก้ไข รันโปรแกรม

```
Log.i("LAB", "Hello world");
```

เปลี่ยน Perspective ของ Eclipse ไปที่ DDMS ทั้งนี้หากมุมมองด้านขวาของ Eclipse ยังไม่มี DDMS ให้เลือก สามารถเปิดใช้โดยการเลือกเมนู `Window->Open Perspective->Others` เพื่อเปิดไดอะล็อกแสดง Perspective ทั้งหมด

ใน DDMS Perspective บริเวณด้านล่างจะมีแท็บ `LogCat` ที่จะแสดงล็อกทั้งหมดที่เกิดขึ้น ทดลองใช้ตัวกรองล็อก

3 INTENTS

องค์ประกอบแกนหลักของโปรแกรมประยุกต์บน Android คือ Activity, Service และ Broadcast Receiver ซึ่งจะถูกเรียกใช้ให้ทำงานผ่านข้อความสื่อสาร ที่เรียกว่า Intent ซึ่งการทำงานร่วมกันนั้นอาจเกิดขึ้นบนโปรแกรมเดียวกัน หรือข้ามโปรแกรมกันก็ได้ ในความเป็นจริงแล้ว Intent เป็นเพียงโครงสร้างข้อมูลที่เก็บรายละเอียดเกี่ยวกับการดำเนินการที่ต้องการ หรือเป็นเพียงเหตุการณ์ หรือ ข้อมูลที่ต้องการประกาศ (broadcast) ออกไปให้องค์ประกอบอื่นๆ รับทราบ

****อ่านรายละเอียดเพิ่มเติมจาก Documents**

3.1 START ACTIVITY

เปิดไฟล์ `FirstActivity.java` เพื่อเพิ่มโค้ดด้านล่างไปในส่วนบนสุดของฟังก์ชัน `doGreet()` ของ `FirstActivity` ทั้งนี้จะต้องมีการอิมพอร์ต `android.content.Intent` และ `android.net.Uri` หลังการแก้ไข รันโปรแกรม

```
Uri uri = Uri.parse("http://www.google.com");
Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);
```


3.2 START ACTIVITY FOR RESULT

การทดลองในตอนนี้จะถูกใช้ทดสอบการใช้งาน Intent เพื่อประสานงานระหว่าง Activity โดยในกรณีศึกษานี้ Activity หลักคือ **FirstActivity** ที่เราพัฒนาขึ้น ส่วนอีก Activity จะเป็น Activity ของโปรแกรม Contacts ที่ใช้สำหรับการเลือก (pick) รายชื่อในโทรศัพท์มือถือ

ทดลองเพิ่มรายชื่อ โดยใช้โปรแกรม Contacts (นักศึกษาเรียกใช้โดยเปิด Application Drawer แล้วคลิกไอคอน Contacts)

เปิดไฟล์ **layout/main.xml** ทำการเพิ่มปุ่มอีกหนึ่งปุ่มเพื่อเรียกใช้ฟังก์ชัน **doPick()**

```
<Button android:text="@string/click_me" android:textColor="@color/greeting"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:onClick="doPick" />
```

เปิดไฟล์ **FirstActivity.java** เพื่อเพิ่มโค้ดด้านล่างลงไปในคลาส **FirstActivity**

```
public static final int PICK_CONTACT = 1;
public void doPick(View v){
    Intent intent = new Intent(Intent.ACTION_PICK,
        ContactsContract.Contacts.CONTENT_URI);
    startActivityForResult(intent, PICK_CONTACT);
}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    switch (requestCode) {
        case (PICK_CONTACT) :
            if (resultCode == Activity.RESULT_OK) {
                Uri contactData = data.getData();
                Cursor c = managedQuery(contactData, null, null, null, null);
                if (c.moveToFirst()) {
                    String name = c.getString(c.getColumnIndex(
                        ContactsContract.Contacts.DISPLAY_NAME));
                    TextView tv = (TextView)findViewById(R.id.greet_label);
                    tv.setText(name);
                }
            }
            break;
    }
}
```

เปิดไฟล์ **AndroidManifest.xml** เพื่อเพิ่มสิทธิ์ในการอ่านรายชื่อ โดยให้เพิ่มโค้ดข้างล่าง ภายใต้แท็ก **manifest** หลังจากนั้น รัน และสังเกตผล

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

CP-02

แสดงโปรแกรมตามการทดลองตอนที่ 2