


Android เบื้องต้น

- ▶ การสร้าง Emulator
 - ▶ เริ่มต้นสร้างโปรเจค
 - ▶ ออกแบบหน้าจอ
 - ▶ เขียนโค้ดเพื่อกำหนดการทำงานของ App
 - ▶ ทดสอบการทำงานของ App ผ่าน Emulator
- 

การสร้าง Emulator

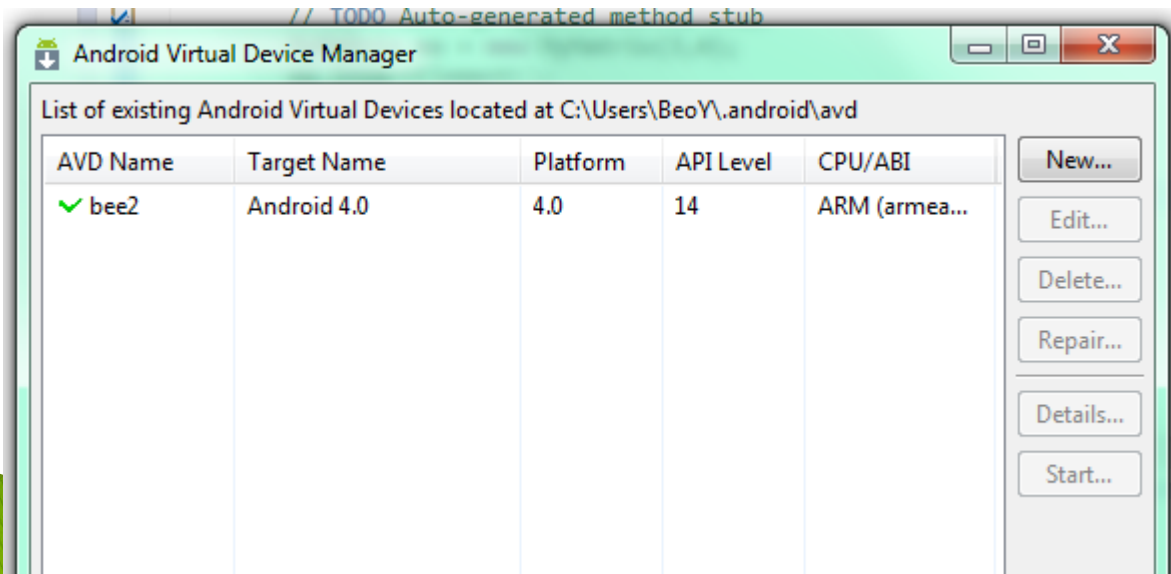
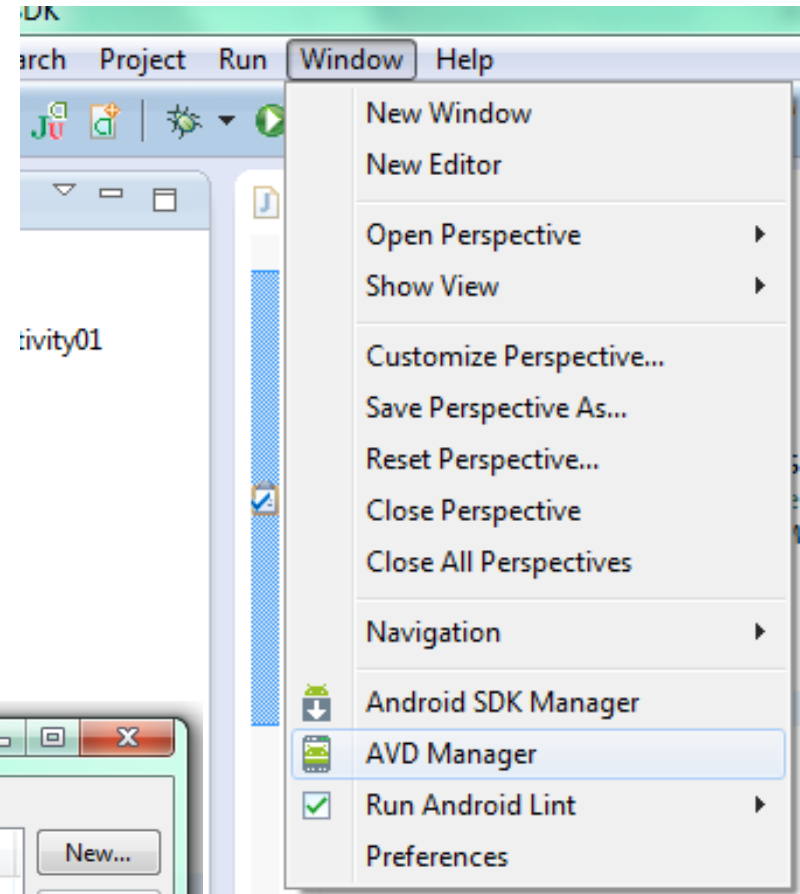
- ▶ การพัฒนา Application จำเป็นต้องทำการสร้าง Emulator เพื่อใช้ในการจำลองเป็นโทรศัพท์มือถือ สำหรับผู้ไม่มีโทรศัพท์ ที่มีระบบปฏิบัติการ Android
- ▶ การจำลองอุปกรณ์โทรศัพท์มือถือบนระบบ Android ขึ้นมาเรียกว่า ADV Android Virtual Device
- ▶ โดยปรกติ หากเราทำการพัฒนา Application บน Emulator สำเร็จแล้ว ก็สามารถนำไปใช้งานได้จริงเช่นกัน

ข้อดีของการสร้าง Emulator

- ▶ ผู้ใช้ไม่จำเป็นต้องซื้อโทรศัพท์จริงๆ
- ▶ สามารถใช้ Feature ที่ระบบ Android มีให้ได้ครบเช่น Google map
- ▶ สามารถพัฒนา App ได้เร็วกว่า เพราะว่าจะต้องมีการตรวจสอบ และแก้ไขข้อผิดพลาดก่อนที่จะนำไปติดตั้งจริง

ขั้นตอนการสร้าง

1. เปิดโปรแกรม Eclipse
2. คลิกที่ window เลือก AVD Manager
3. ทำการเลือก New เพื่อสร้าง AVD ขึ้นมา



ขั้นตอนการสร้าง

- ▶ New ทำการกรอกชื่อ
- ▶ Tager ทำการเลือกรุ่นหรือ version ของ Android ที่จะพัฒนา
- ▶ Size กรอกขนาดความจุของหน่วยความจำ
- ▶ Built-in ทำการเลือกขนาดสำหรับการแสดงผล

ผล

เมื่อกำหนดข้อมูลเรียบร้อยแล้วก็ทำการคลิกที่ Create AVD

Create new Android Virtual Device (AVD)

Name:

Target:

CPU/ABI:

SD Card:

Size: MiB

File: Browse...

Snapshot:

Enabled

Skin:

Built-in:

Resolution: x

Hardware:

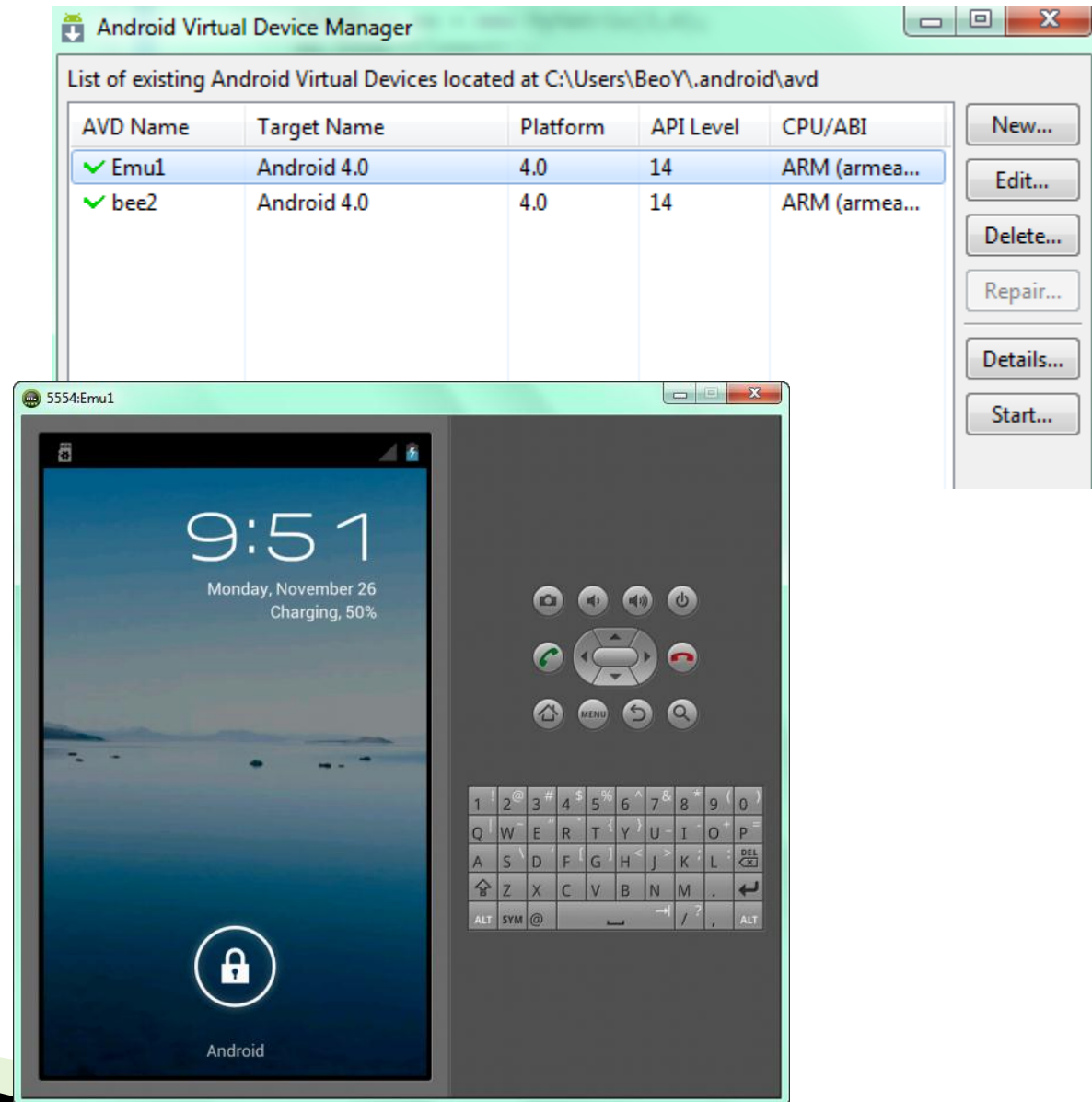
Property	Value	

Override the existing AVD with the same name

Create AVD Cancel

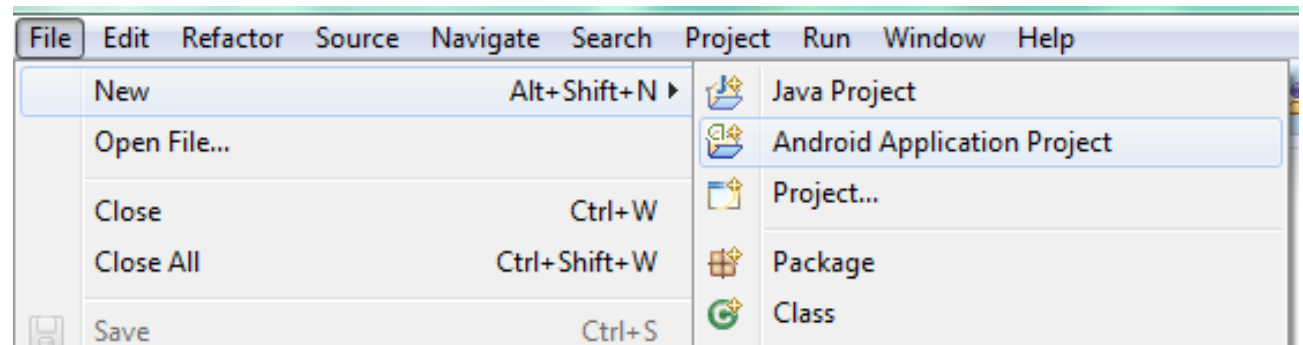
ขั้นตอนการสร้าง

ทำการเลือก Emu ที่เราสร้าง
จากนั้นกด Start และ Launch
เราก็จะได้ Android สำหรับ
ใช้พัฒนา Application



เริ่มต้นสร้างโปรเจค

ทำการคลิกที่ File-> New -> Android Application Project



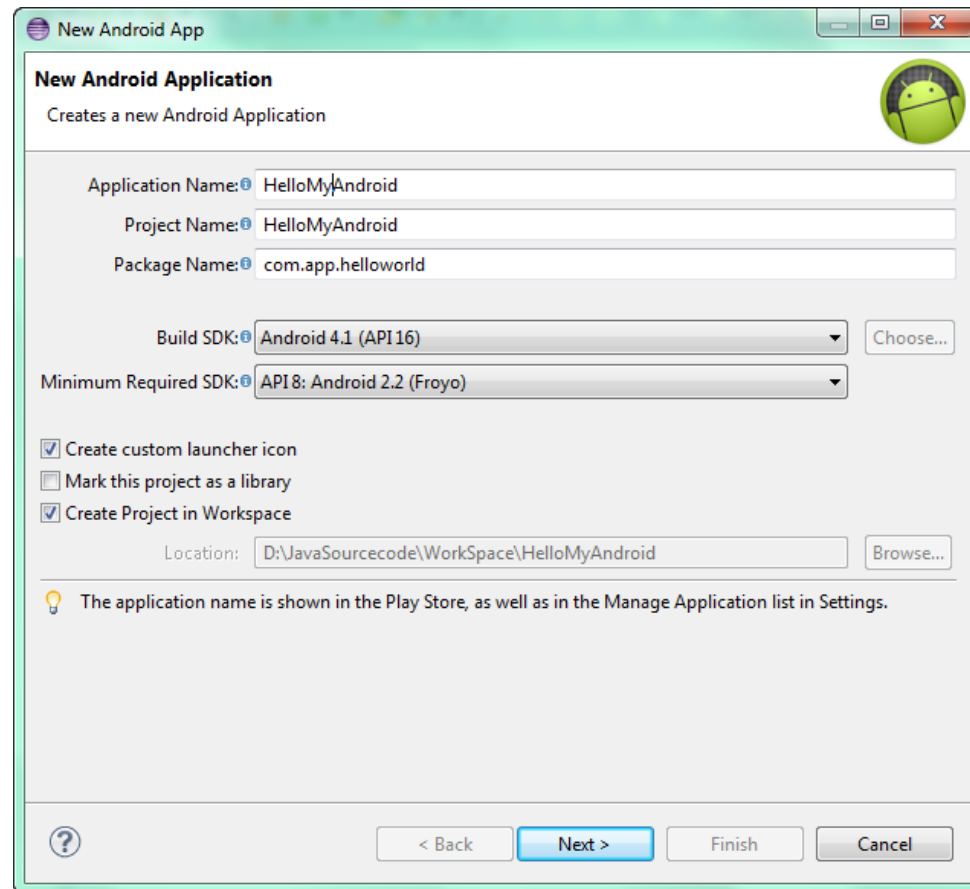
เริ่มต้นสร้างโปรเจค

- ▶ ทำการตั้งชื่อ Application Name
- ▶ ทำการตั้งชื่อ Project Name
- ▶ ทำการตั้งชื่อ Package Name

ชื่อ package จะอยู่ในรูป xx.xx.xx

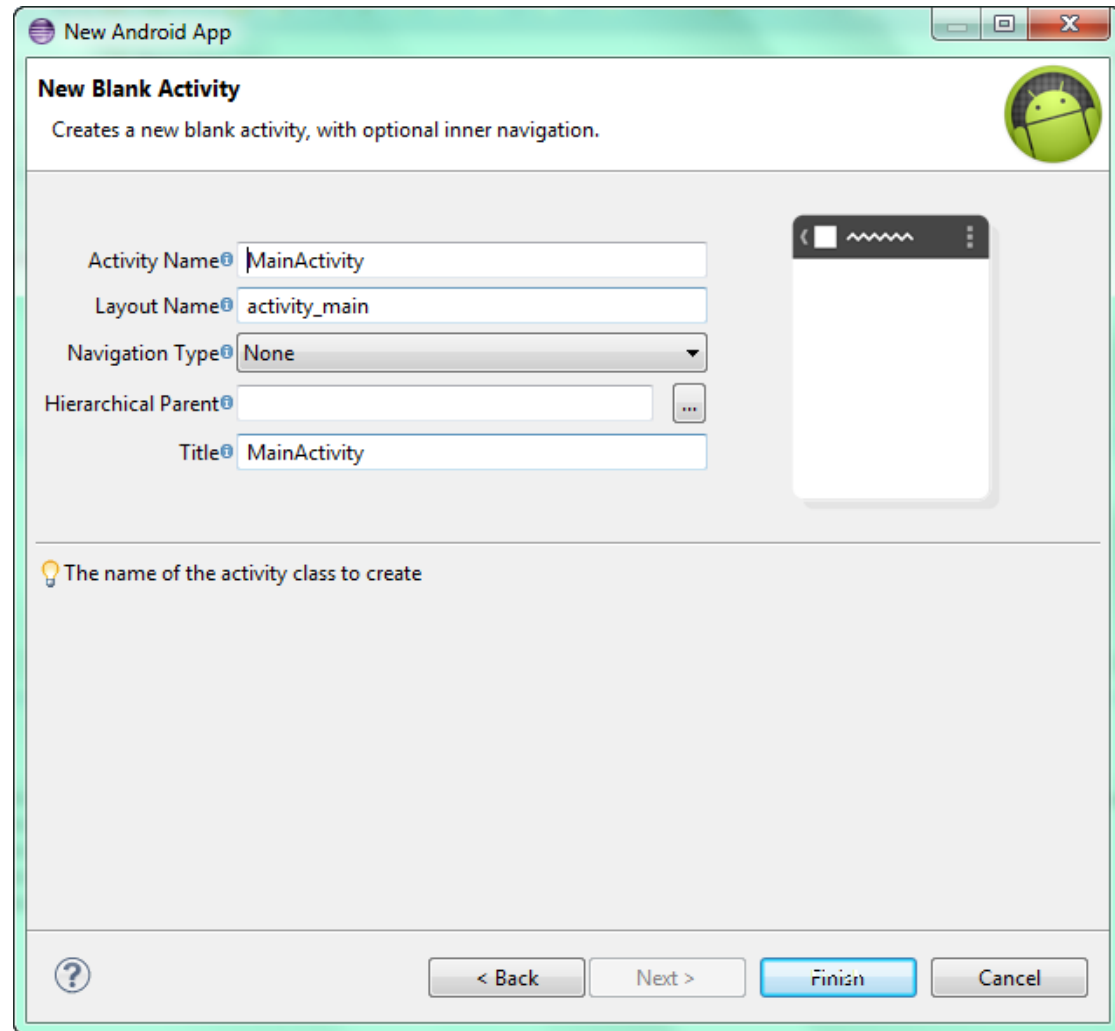
ในที่นี้ขอให้ตั้งเป็น com.app.helloworld

จากนั้นกด Next



เริ่มต้นสร้างโปรเจค

หน้านี้ ใช้สำหรับกำหนด MainActivity เพื่อกำหนดให้ Class ใดเป็น Class แรกที่ทำงาน เมื่อตั้งชื่อแล้วกด Finish



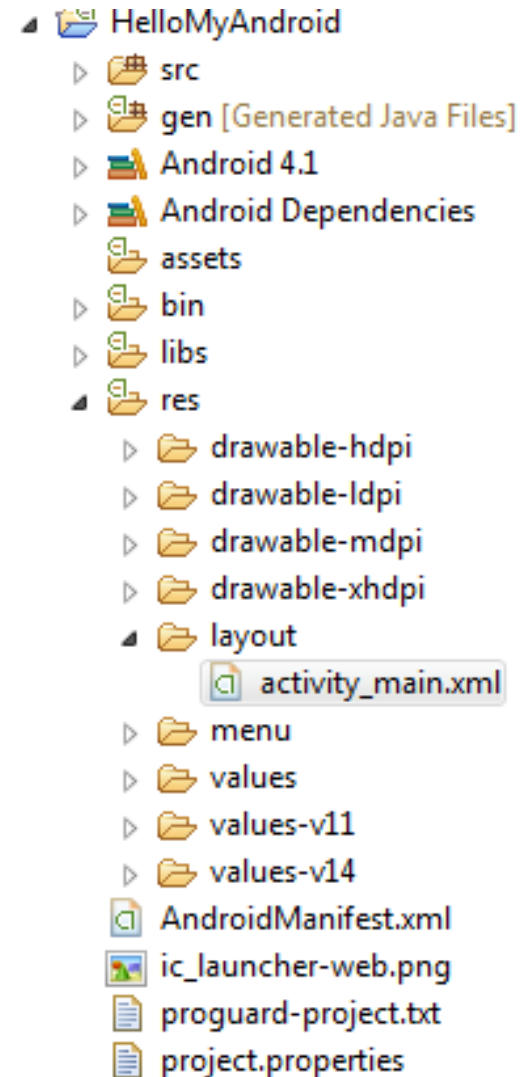
เริ่มต้นสร้างโปรเจค

- ▶ จะได้ไฟล์ ที่ ระบบสร้างให้ดังรูป

จะมี Activity_Main.xml

ใช้สำหรับออกแบบหน้าจอว่าจะให้มีลักษณะใด

ให้ทดลองทำการรัน Application ที่เราสร้างขึ้นมา

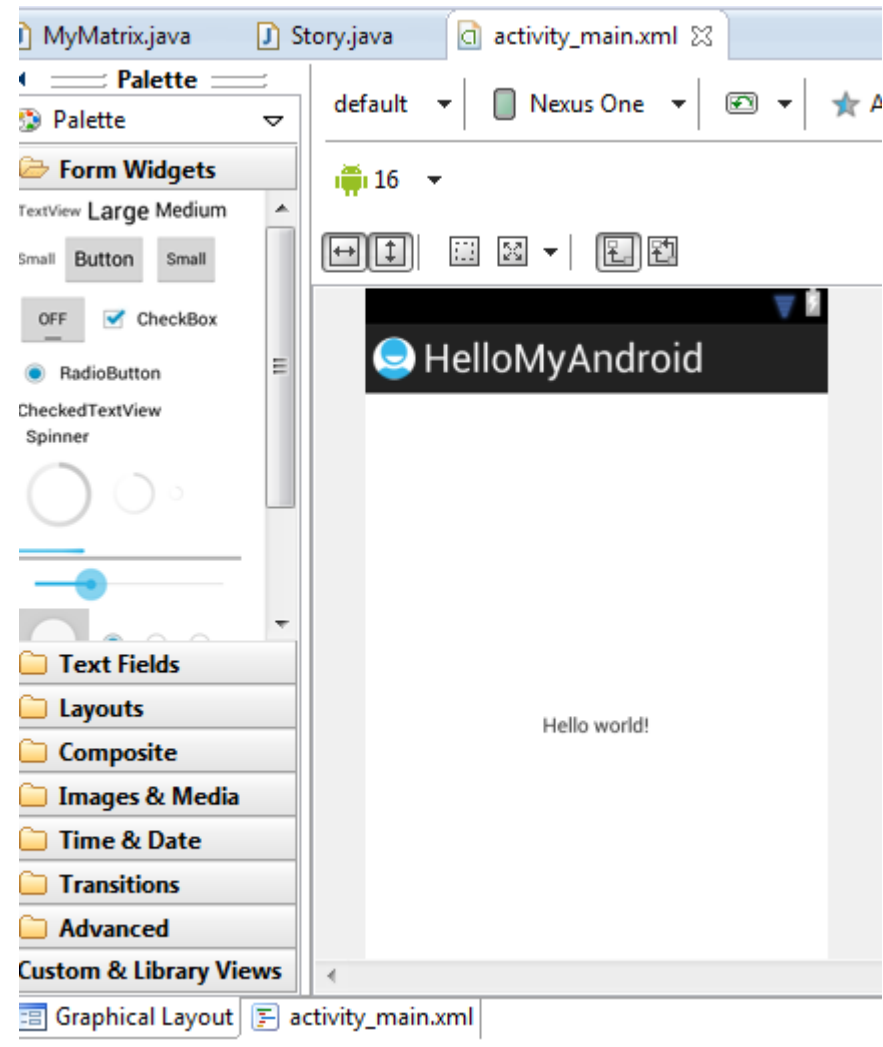


ออกแบบหน้าจอ

- ▶ การออกแบบหน้าจอจะเป็นการแก้ไขโค้ดของ XML ซึ่งอยู่ใน Folder Layout แต่และ App สามารถสร้าง หน้าจอได้หลายๆ หน้าขึ้นอยู่กับผู้พัฒนา
- ▶ แต่สำหรับผู้ที่ใช้ Eclipse ผู้พัฒนาสามารถลาก และวาง เครื่องมือที่ต้องการลงในหน้าจอได้เลย
- ▶ Eclipse จะทำการแก้ไข XML ให้ตามที่คุณใช้กำหนด

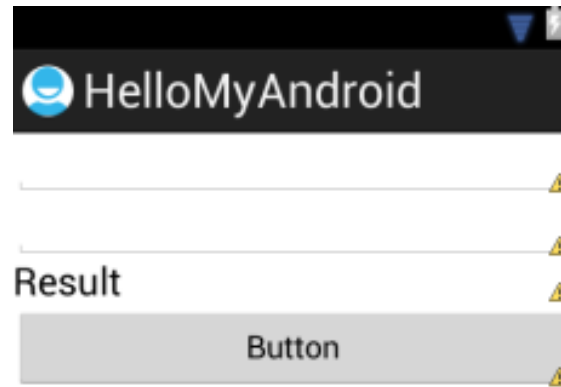
ออกแบบหน้าจอ

- ▶ จะเห็นว่า หน้าจอของ Android มี 2 mode คือ Graphical Layout และ ส่วนที่เป็น XML File โดยที่เมื่อเราทำการแก้ไขใน Graphic XML ก็จะถูกแก้ไขไปด้วย ดังนั้นผู้พัฒนาไม่จำเป็นต้องมีความรู้เกี่ยวกับ XML มากนัก

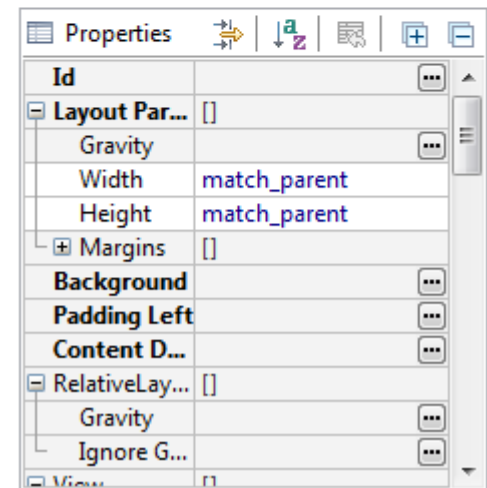
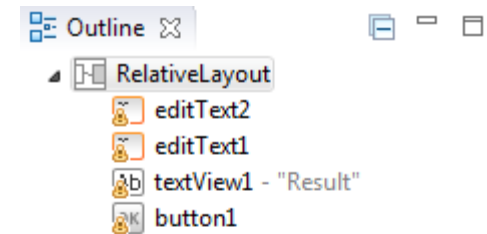


ออกแบบหน้าจอ

- ▶ ให้นักศึกษาลองสร้างหน้าจอดังรูป



- Outline เป็นกล่องเครื่องมือช่วยในการสร้าง ลบ จัดลำดับก่อนหลังของ Element ต่าง
- Properties เป็นกล่องเครื่องมือเพื่อกำหนดคุณสมบัติต่างๆ ของ Element



► คู่มือในส่วนของ XML

จะสังเกตเห็นว่า EditText มีส่วนของ

`android:id="@+id/editText2"`

เราสามารถตั้งชื่อ edittext แต่ละอันได้โดย

แก้ไขชื่อ ID เหล่านี้ ชื่อ ID เหล่านี้จะถูก

อ้างอิงในการเขียนโปรแกรมด้วย

```
android:id="@+id/LinearLayout1"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
android:orientation="vertical" >
```

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

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

```
<TextView
```


เขียนโค้ดและกำหนดการทำงาน

- ▶ ให้นักศึกษาทำการเปิดไฟล์ MainActivity.java ขึ้นมา ไฟล์นี้จะถูกรันเป็นไฟล์แรกขณะที่ App เริ่มทำงาน
- ▶ เมื่อเปิดไฟล์ขึ้นมาเราจะได้ดังภาพ

```
package com.app.helloworld;

import android.os.Bundle;

public class MainActivity extends Activity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

เขียนโค้ดและกำหนดการทำงาน

- ▶ ทำการแก้ไขโค้ดโดยแก้ไขโค้ดดังรูป

```
public class MainActivity extends Activity {  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        Button buttt = (Button) findViewById(R.id.button1);  
  
        buttt.setOnClickListener(new OnClickListener() {  
  
            public void onClick(View v) {  
                EditText name = (EditText) findViewById(R.id.editText1);  
                Toast.makeText(MainActivity.this, name.getText().toString(), Toast.LENGTH_SHORT);  
            }  
        });  
    }  
}
```

เขียนโค้ดและกำหนดการทำงาน

ทำการรันโปรแกรม และ พิมพ์ข้อความลงในช่องแรก ละกวดที่ปุ่ม
สังเกตผลการรันโปรแกรม

แบบฝึกหัด

ให้นักศึกษาทำการแก้ไขโค้ดโปรแกรม โดยกำหนด พิมพ์ตัวเลขลงใน 2 ช่องแรก จากนั้นให้ทำการแสดงผลการ + กันของ ทั้ง 2 ตัวแรก และ แสดงผลลัพธ์ ลงในช่อง Result เมื่อทำการคลิกที่ปุ่ม Button

หาก 2 ช่องแรก ไม่ใช่ตัวเลขให้ แสดงข้อความเตือนและให้ทำการกรอกข้อมูลใหม่