

# Android智慧型手機程式設計

## Android介面設計

建國科技大學 資管系

饒瑞佺

2012/4 V1

2012/8 V2

2013/2 V3

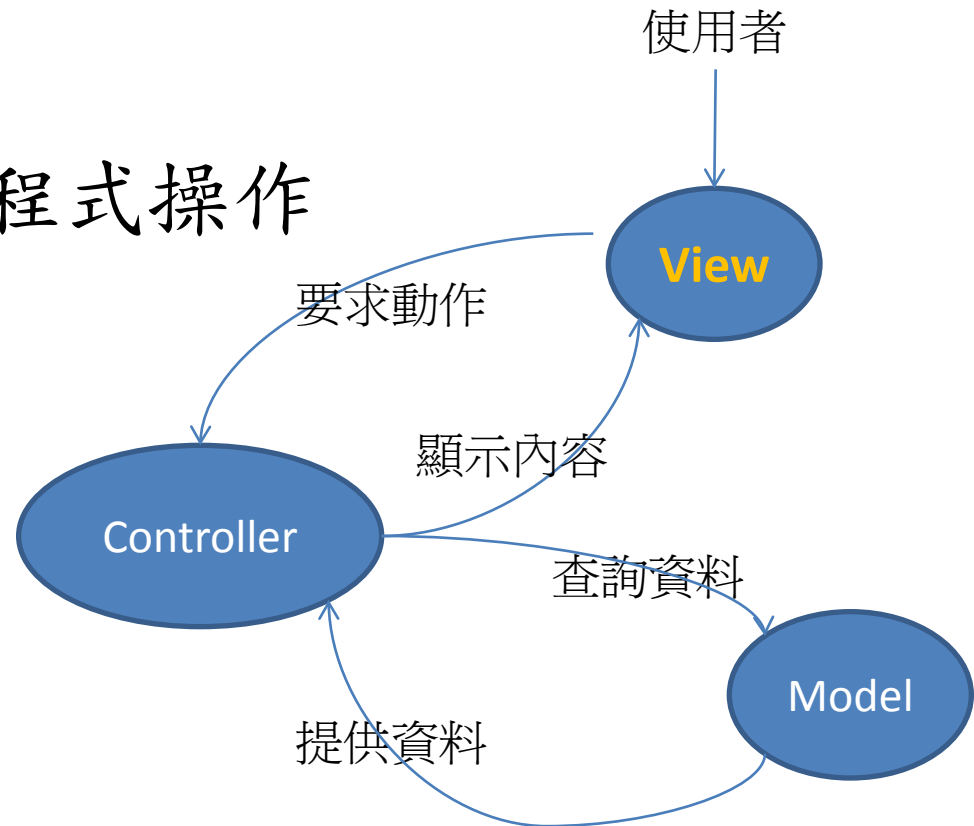
# Android介面設計

- 全部都透過XML檔案來決定
- res/layout內
- Activity內透過setContentView來設定
- 當然....需要R的幫忙

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

# MVC Model

- Model-View-Controller
- Model：儲存內容
- View：顯示介面
- Controller：控制程式操作



# Android視窗介面

- Views

- 按鈕元件
- 影像按鈕元件
- 單選選單元件
- 複選選單元件
- 對話視窗 ...

# Android視窗介面

- 透過main.xml佈局資源檔設計介面
  - － 裡面的android:id屬性設定UI名稱
  - － 是程式與介面的連結點
- 透過strings.xml文字資源檔設計介面文字
- 透過drawable圖檔資源檔設計介面圖形
- 透過anim.xml設計介面動畫

**XML**

# XML

- eXtensible-Markup-Language
- data of data
- 由tag構成，一定要有開始與結束tag
- 需有root tag
- Well-defined
- 開頭：  
    <?xml version="1.0" encoding="UTF-8" ?>
- 與HTML不同，並非HTML的進化

# XML

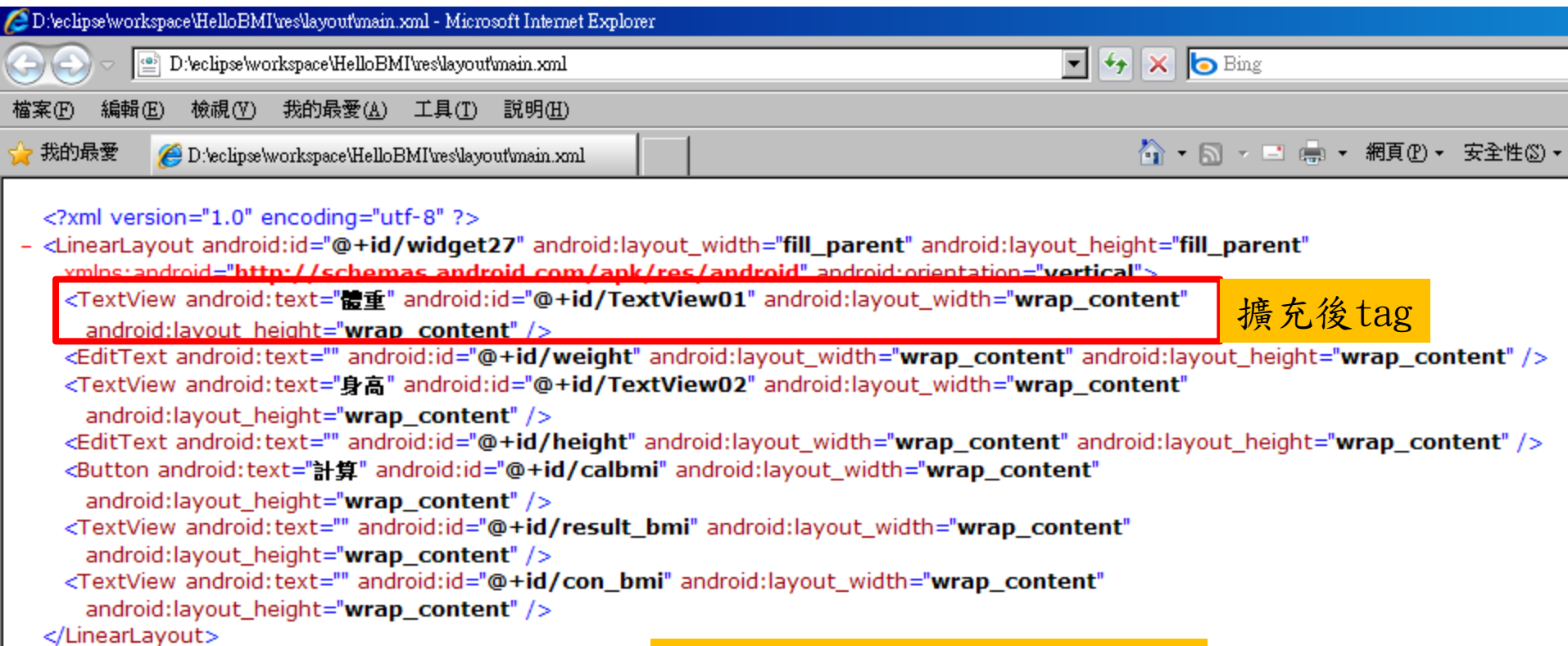
- 原始main.xml





# XML

- 擴充後main.xml



```
<?xml version="1.0" encoding="utf-8" ?>
- <LinearLayout android:id="@+id/widget27" android:layout_width="fill_parent" android:layout_height="fill_parent"
  xmlns:android="http://schemas.android.com/apk/res/android" android:orientation="vertical">
  <TextView android:text="體重" android:id="@+id/TextView01" android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
  <EditText android:text="" android:id="@+id/weight" android:layout_width="wrap_content" android:layout_height="wrap_content" />
  <TextView android:text="身高" android:id="@+id/TextView02" android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
  <EditText android:text="" android:id="@+id/height" android:layout_width="wrap_content" android:layout_height="wrap_content" />
  <Button android:text="計算" android:id="@+id/calbmi" android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
  <TextView android:text="" android:id="@+id/result_bmi" android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
  <TextView android:text="" android:id="@+id/con_bmi" android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>
```

擴充後tag

有幾個擴充後tag？

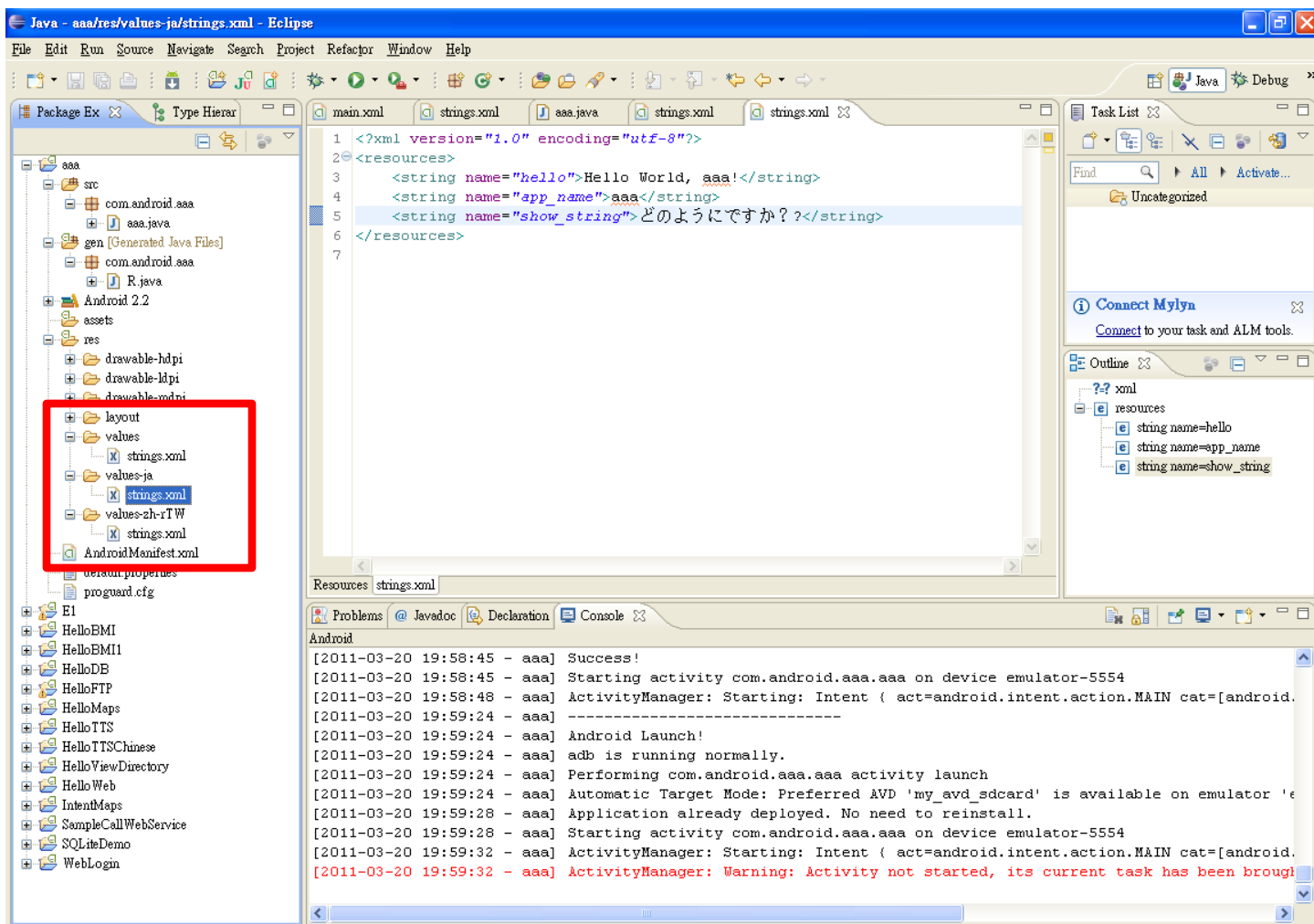
多國語言

# 多國語言

- main.xml不動
- 使用規定語法的strings.xml目錄
  - 繁體中文:values-zh-rTW
  - 簡體中文: values-zh-rCN
  - 日文:values-ja
  - 英式英文:values-en-rUK
  - 美式英文:values-en-rUS
- 只要Android上使用的語言更改，就會自動顯示對應的文字

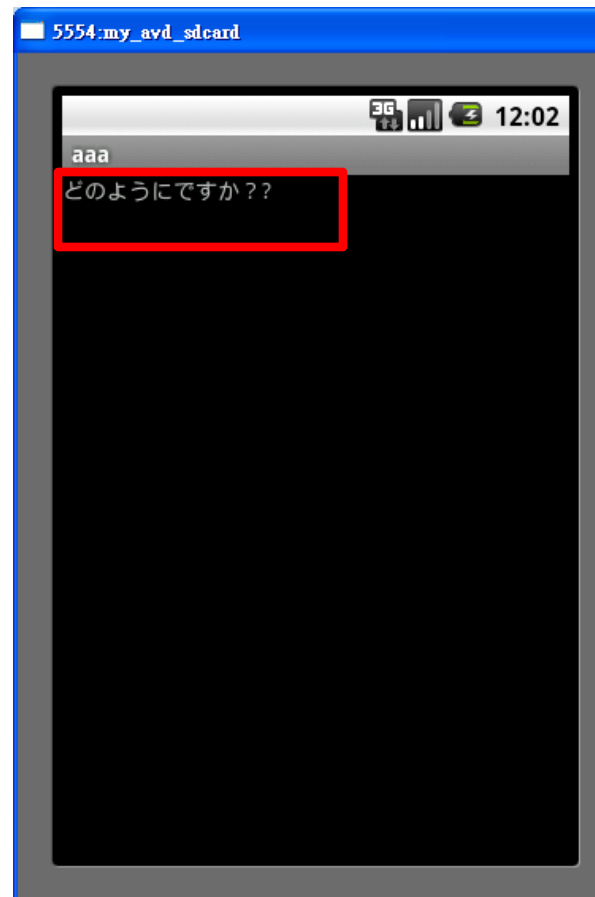
# 多國語言

- 依據規定建立各語言的目錄與strings.xml檔案
- 修改顯示文字



# 多國語言

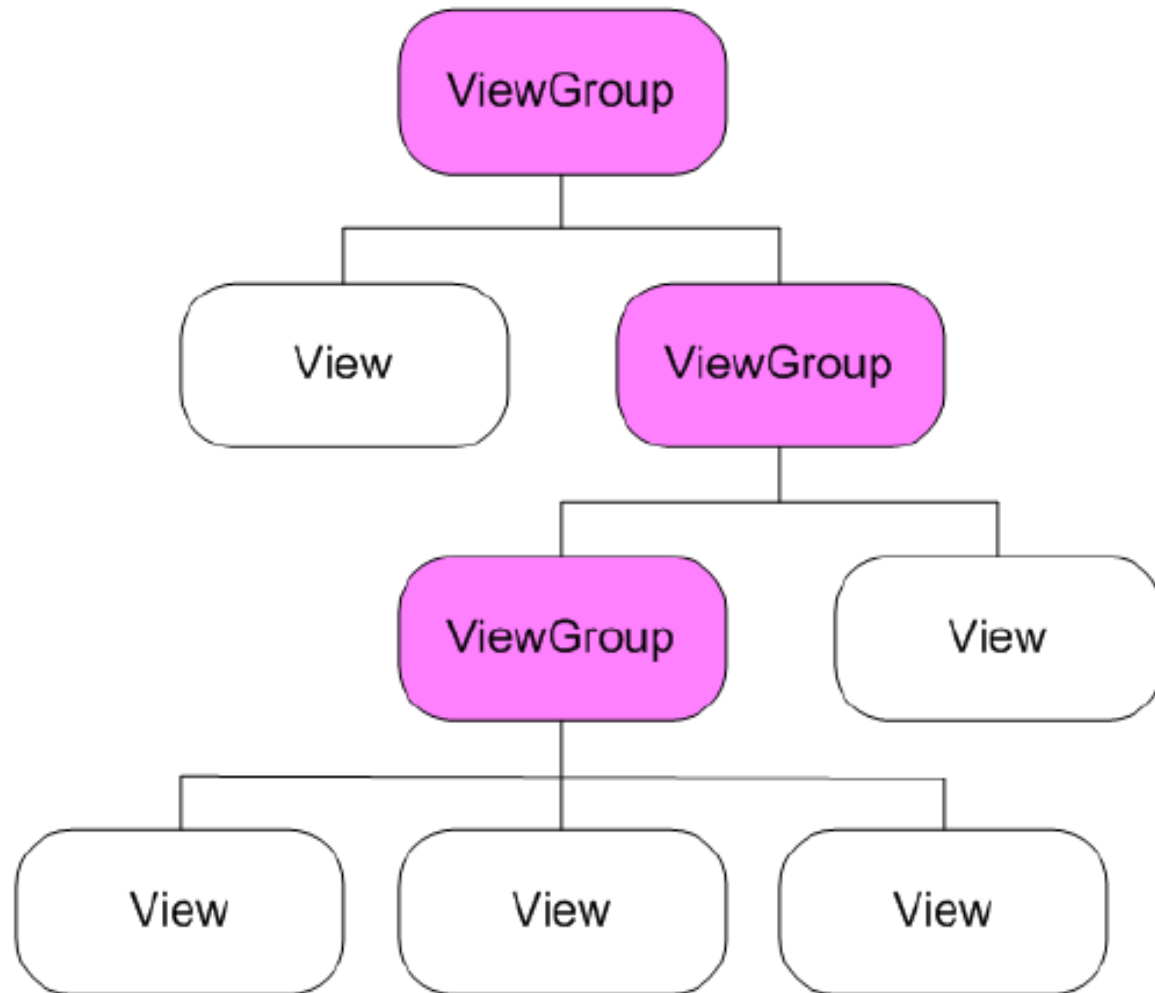
- 手機上切換語言
- 執行時顯示文字就會自動替換



介面佈局

Layout

# View Hierarchy



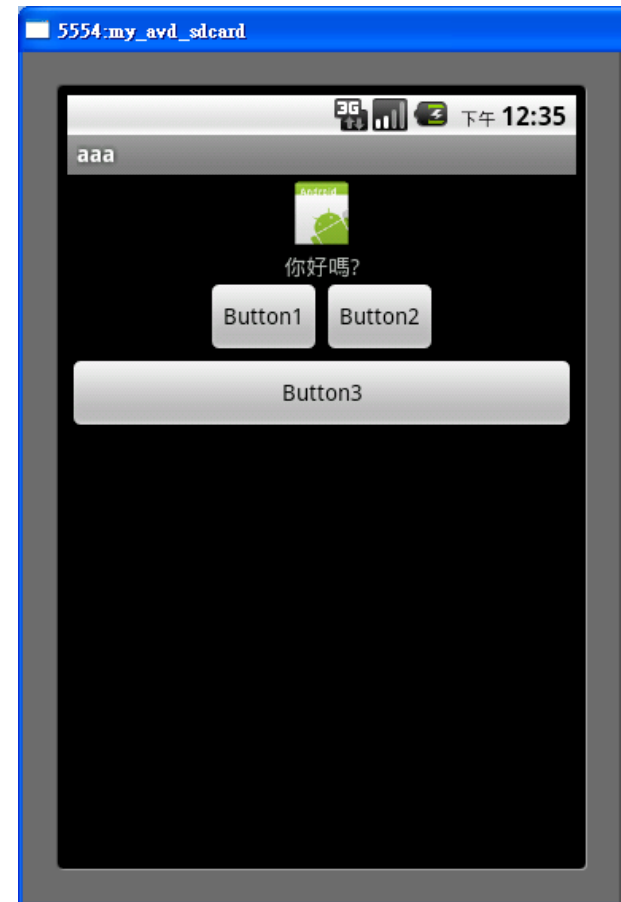
# Layout

- **LinearLayout**
  - 將View以水平或垂直方式做線性排列
- **RelativeLayout**
  - 將View依相對位置做排列
- **TableLayout**
  - 像建立表格般安排View/ViewGroup 的位置

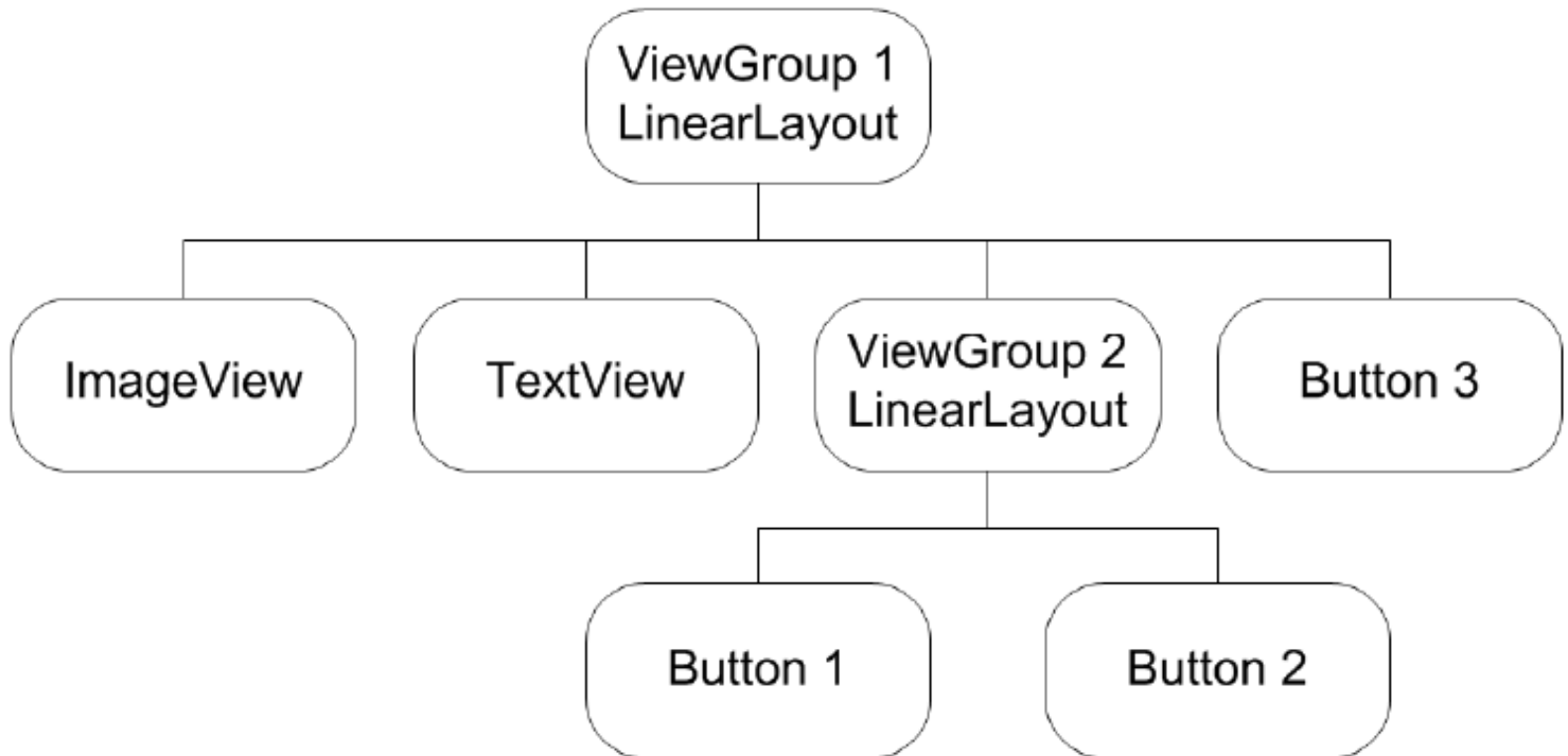


# LinearLayout

```
main.xml X R.java aaa.java
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5     android:orientation="vertical"
6 >
7     <ImageView
8         android:src="@drawable/icon"
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:layout_gravity="center_horizontal"
12    />
13    <TextView android:id="@id/android:empty"
14        android:layout_width="wrap_content"
15        android:layout_height="wrap_content"
16        android:text="@string/show_string"
17        android:layout_gravity="center_horizontal"
18    />
19    <LinearLayout
20        android:layout_width="fill_parent"
21        android:layout_height="wrap_content"
22        android:orientation="horizontal"
23        android:gravity="center">
24        <Button
25            android:layout_width="wrap_content"
26            android:layout_height="wrap_content"
27            android:text="Button1"
28        />
29        <Button
30            android:layout_width="wrap_content"
31            android:layout_height="wrap_content"
32            android:text="Button2"
33        />
34    </LinearLayout>
35    <Button
36        android:layout_width="fill_parent"
37        android:layout_height="wrap_content"
38        android:text="Button3"
39        android:layout_gravity="center_horizontal"
40    />
</LinearLayout>
```



# LinearLayout

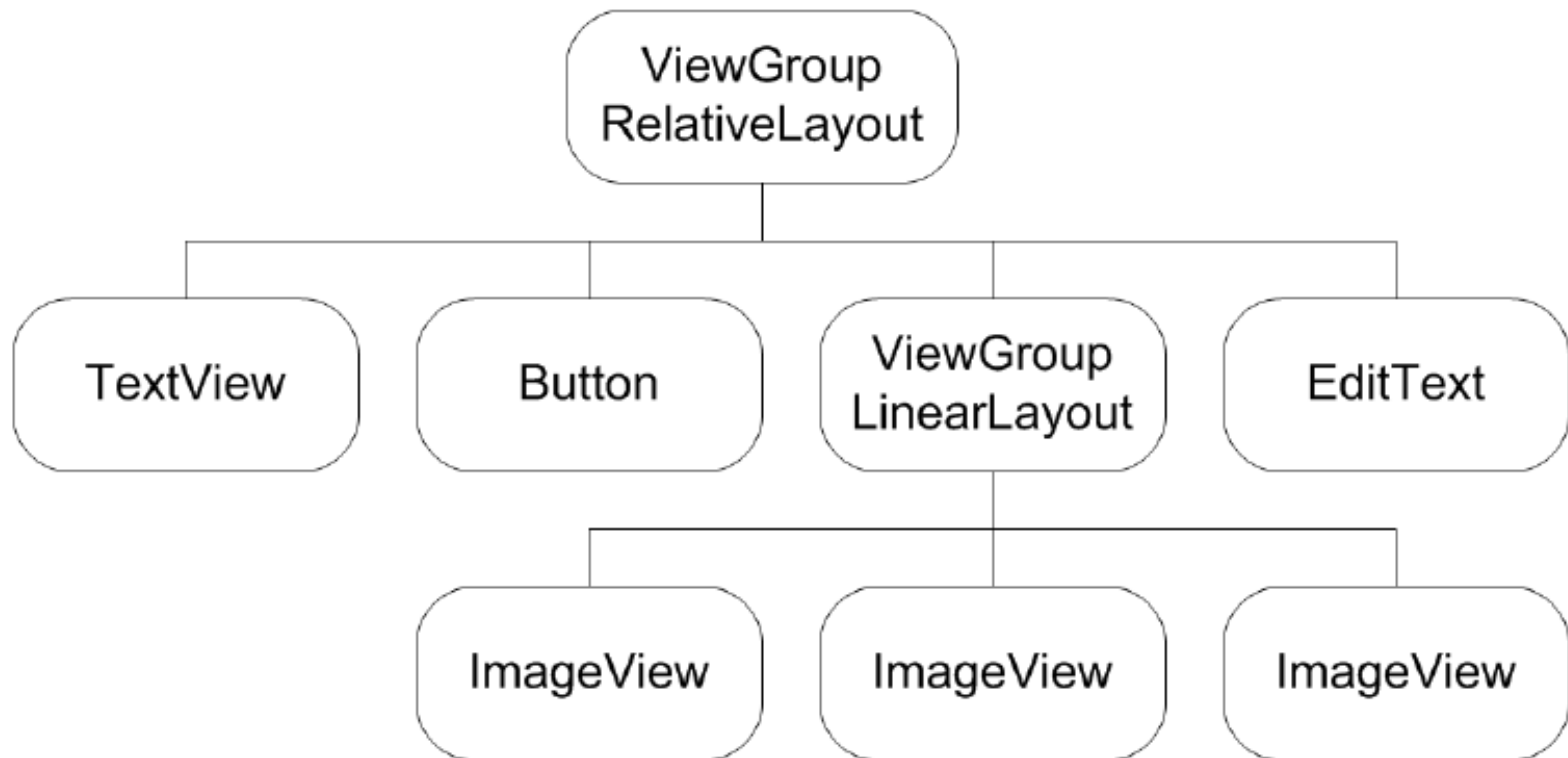


# RelativeLayout

```
*main.xml  R.java  aaa.java
1 <?xml version="1.0" encoding="UTF-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5 >
6     <TextView android:id="@+id/tv"
7         android:layout_width="fill_parent"
8         android:layout_height="wrap_content"
9         android:text="@string/show_string"/>
10
11     <Button android:id="@+id/btn"
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:layout_alignParentBottom="true"
15         android:text="Button1"/>
16
17     <LinearLayout android:layout_above="@id/btn"
18         android:layout_width="fill_parent"
19         android:layout_height="wrap_content"
20         android:orientation="horizontal">
21         <ImageView android:src="@drawable/icon"
22             android:layout_width="wrap_content"
23             android:layout_height="wrap_content"
24             android:layout_gravity="center_horizontal"/>
25         <ImageView android:src="@drawable/icon"
26             android:layout_width="wrap_content"
27             android:layout_height="wrap_content"
28             android:layout_gravity="center_horizontal"/>
29         <ImageView android:src="@drawable/icon"
30             android:layout_width="wrap_content"
31             android:layout_height="wrap_content"
32             android:layout_gravity="center_horizontal"/>
33     </LinearLayout>
34     <EditText android:layout_below="@id/tv"
35         android:layout_width="fill_parent"
36         android:layout_height="wrap_content"/>
37 </RelativeLayout>
```

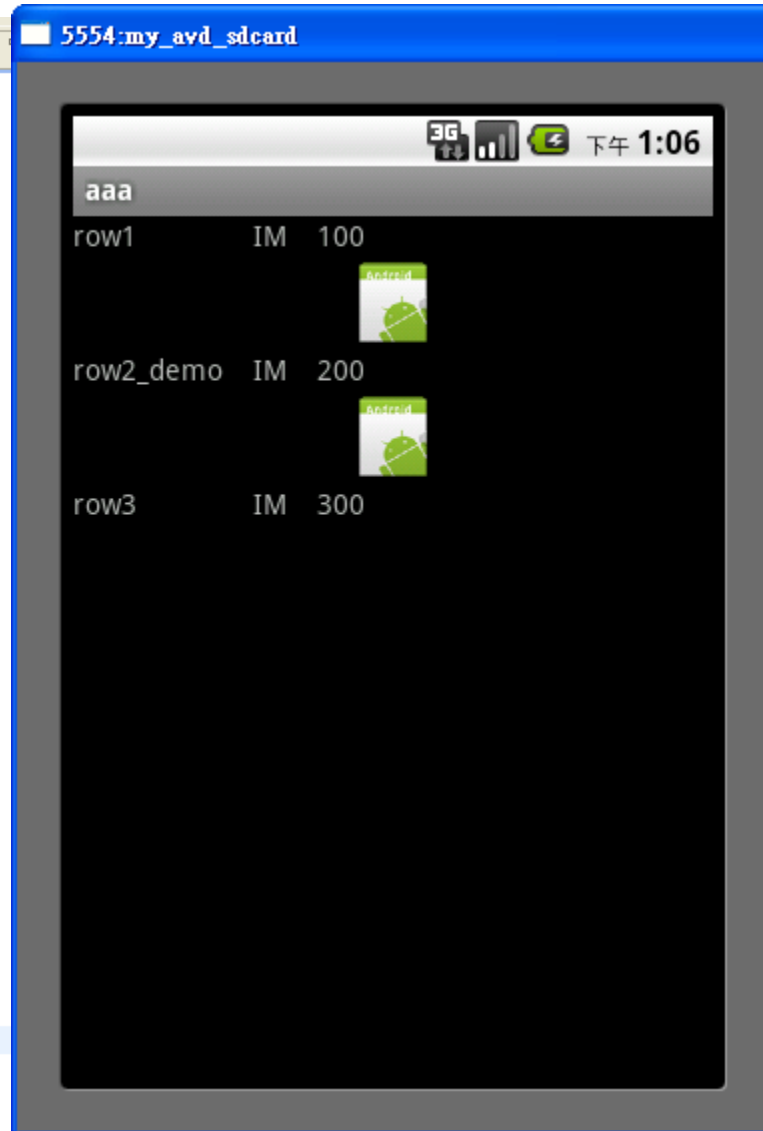


# RelativeLayout



# TableLayout

```
main.xml x R.java aaa.java
1 <?xml version="1.0" encoding="UTF-8"?>
2 <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5     >
6     <TableRow>
7         <TextView android:text="row1"/>
8         <TextView android:text="IM"/>
9         <TextView android:text="100"/>
10    </TableRow>
11    <ImageView android:src="@drawable/icon"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:layout_gravity="center_horizontal"/>
15    <TableRow>
16        <TextView android:text="row2_demo"
17            android:paddingRight="15px"/>
18        <TextView android:text="IM"
19            android:paddingRight="15px"/>
20        <TextView android:text="200"/>
21    </TableRow>
22    <ImageView android:src="@drawable/icon"
23        android:layout_width="wrap_content"
24        android:layout_height="wrap_content"
25        android:layout_gravity="center_horizontal"/>
26    <TableRow>
27        <TextView android:text="row3"
28            android:paddingRight="15px"/>
29        <TextView android:text="IM"
30            android:paddingRight="15px"/>
31        <TextView android:text="300"/>
32    </TableRow>
33 </TableLayout>
34
35
```

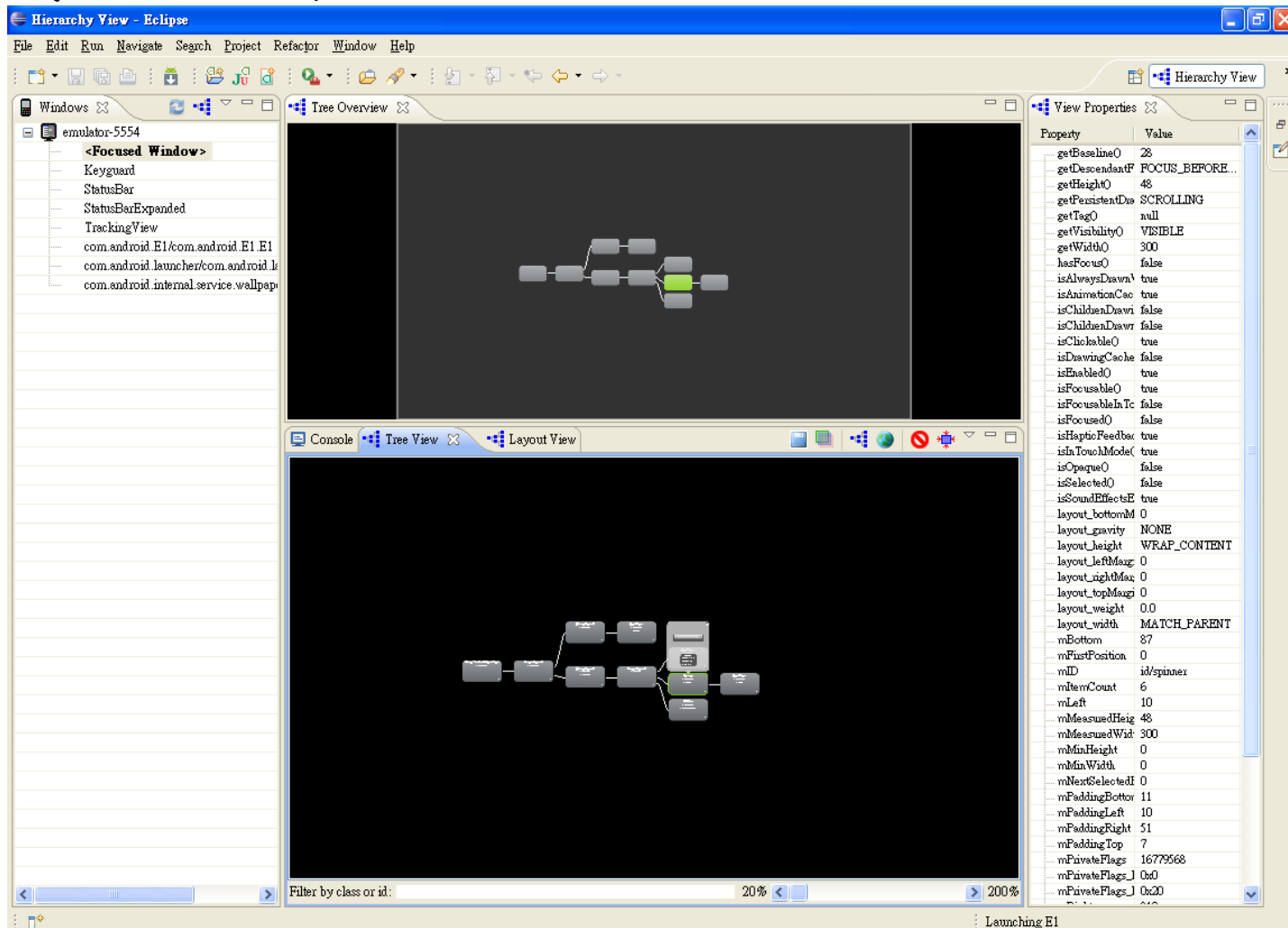


# 介面檢視工具

## Hierarchy View

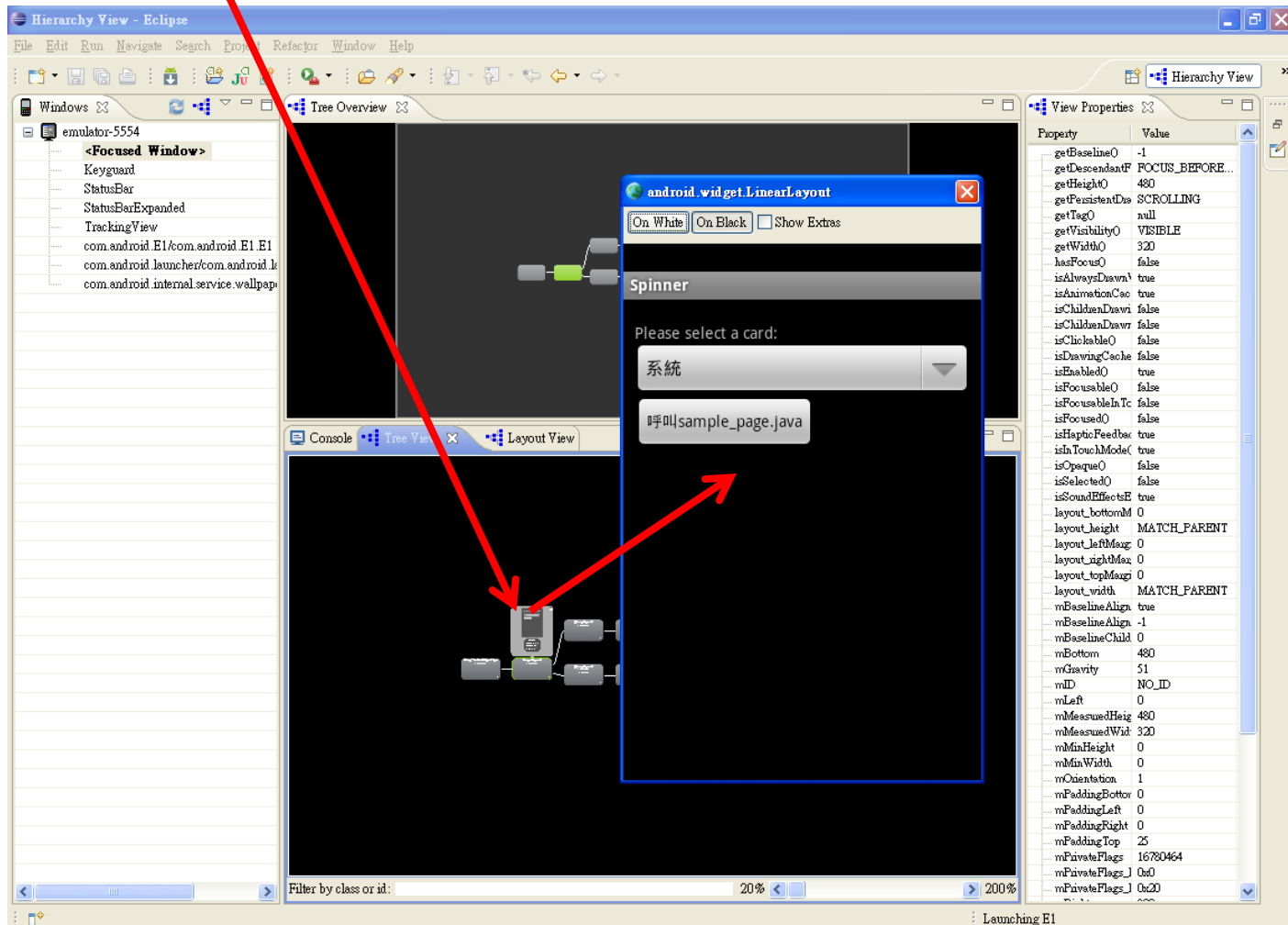
# Hierarchy View

- 要先啟動AVD SDK:hierarchyviewer



# Hierarchy View

- 點兩下可預覽





介面設計工具

DroidDraw

# DroidDraw

- 協助進行Android程式介面設計，拖拉放方式
- 協助產生XML碼，再貼回main.xml等檔案就可以
- 網路版
  - <http://droiddraw.org/>
- 單機版

# DroidDraw網路版

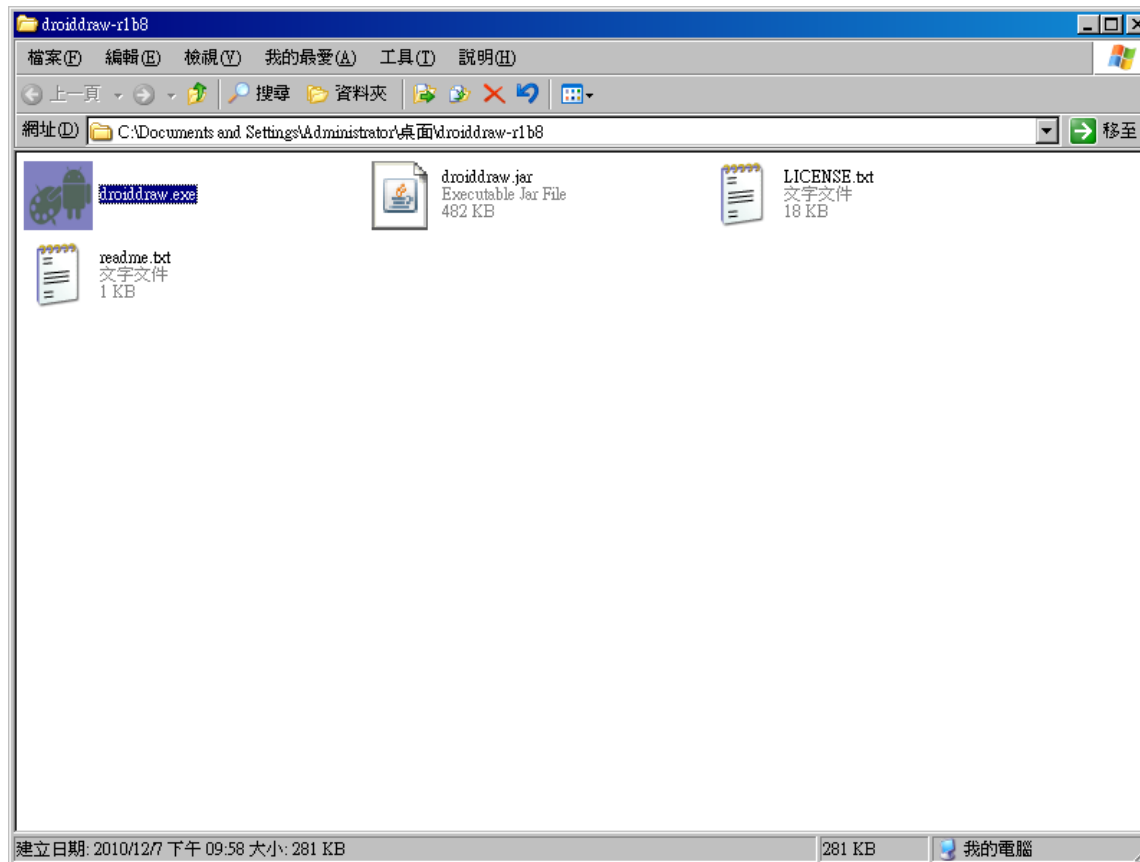


# DroidDraw 單機版



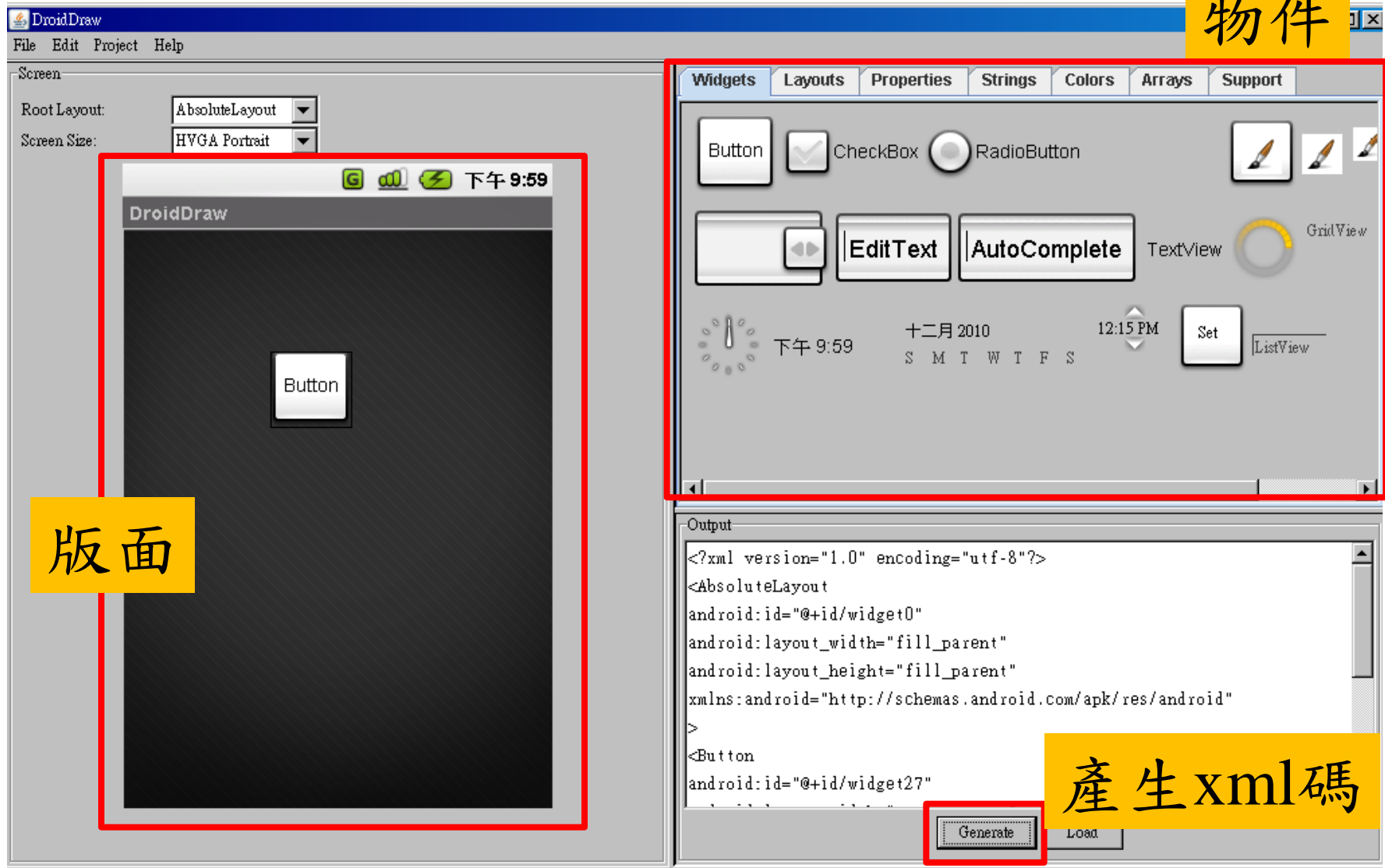
# DroidDraw 單機版

- 不需要安裝，直接執行droiddraw.exe



# DroidDraw 單機版

物件



版面

產生xml碼

# 其它 Views

<http://developer.android.com/guide/tutorials/views/index.html>

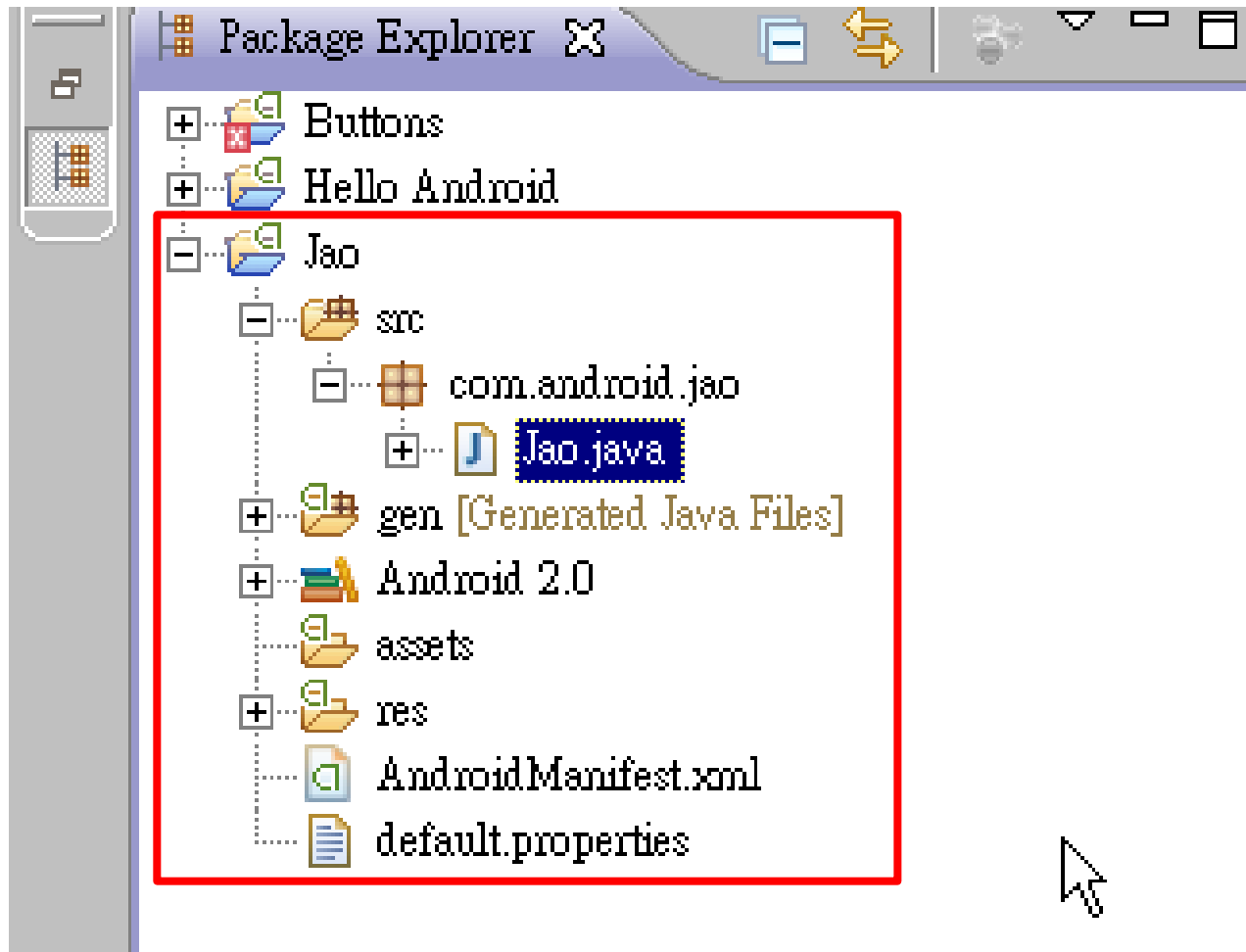
按鈕



# 按鈕View

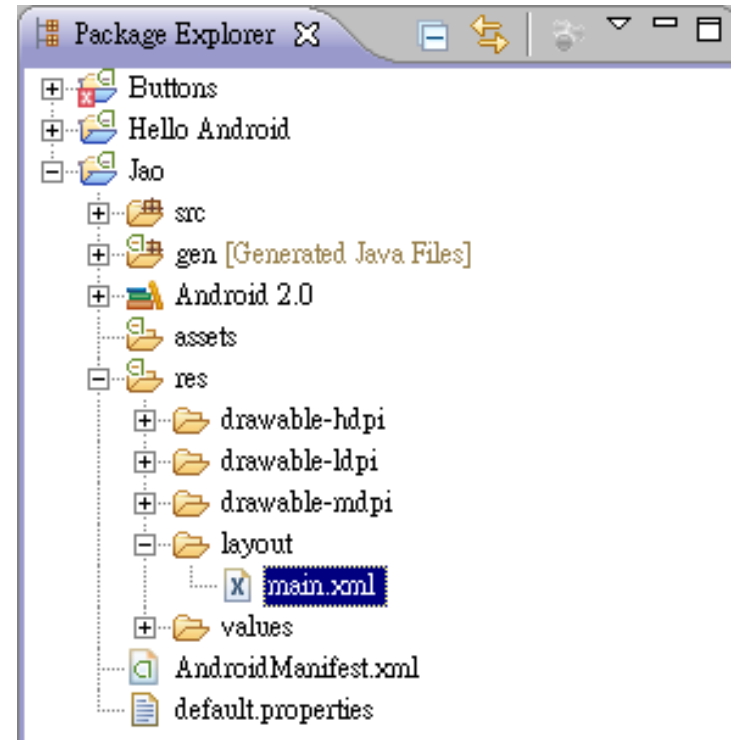
- 使用Eclipse建立新專案
  - Project Name=Jao
  - Application Name=MyButton
  - Package Name=com.android.jao
  - Create Activity= Jao

# Jao專案畫面



# 設計main.xml

- 介面佈局資源檔
- 設計介面的組成元件，例如按鈕...
- res\layout\main.xml



# 設計main.xml

## 原始main.xml

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

# 設計main.xml

1

加入scrollview讓介面可以上下捲動

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/a
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        >

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

    </LinearLayout>

</ScrollView>
```

大小寫有差！

# 設計main.xml

還有  
Relative layout  
Table layout

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/a
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"

        <Button android:id="@+id/button_large"
            android:text="@string/button_large"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="@string/hello"
            />
    </LinearLayout>
</ScrollView>
```

代表有錯誤！

程式與介面的連結  
android:id 屬性

加入button讓介面出現按鈕元件

2

# android:id

- 介面佈局資源檔main.xml與程式的連結
- 格式：**android:id="@+id/名稱"**
  - @：解讀成識別符號而不是一般內容
  - +：新增一個id
  - id/：識別符號被歸類在id類別下
- 自動在R.java中產生索引
- 取用方式：R.id.名稱

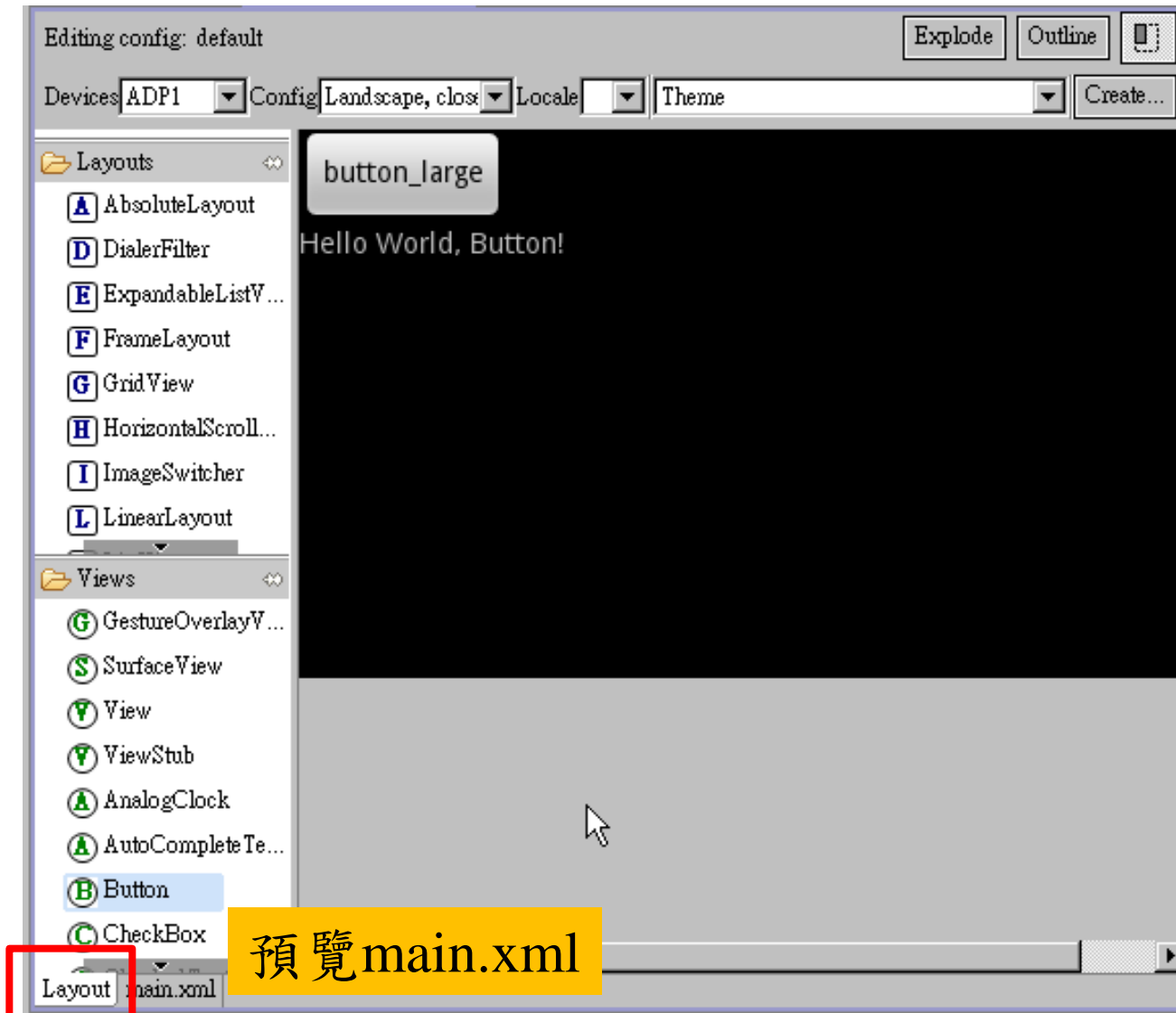
# 設計 strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, Button!</string>
    <string name="app_name">MyButton</string>
    <string name="button_large">button_large</string>
</resources>
```

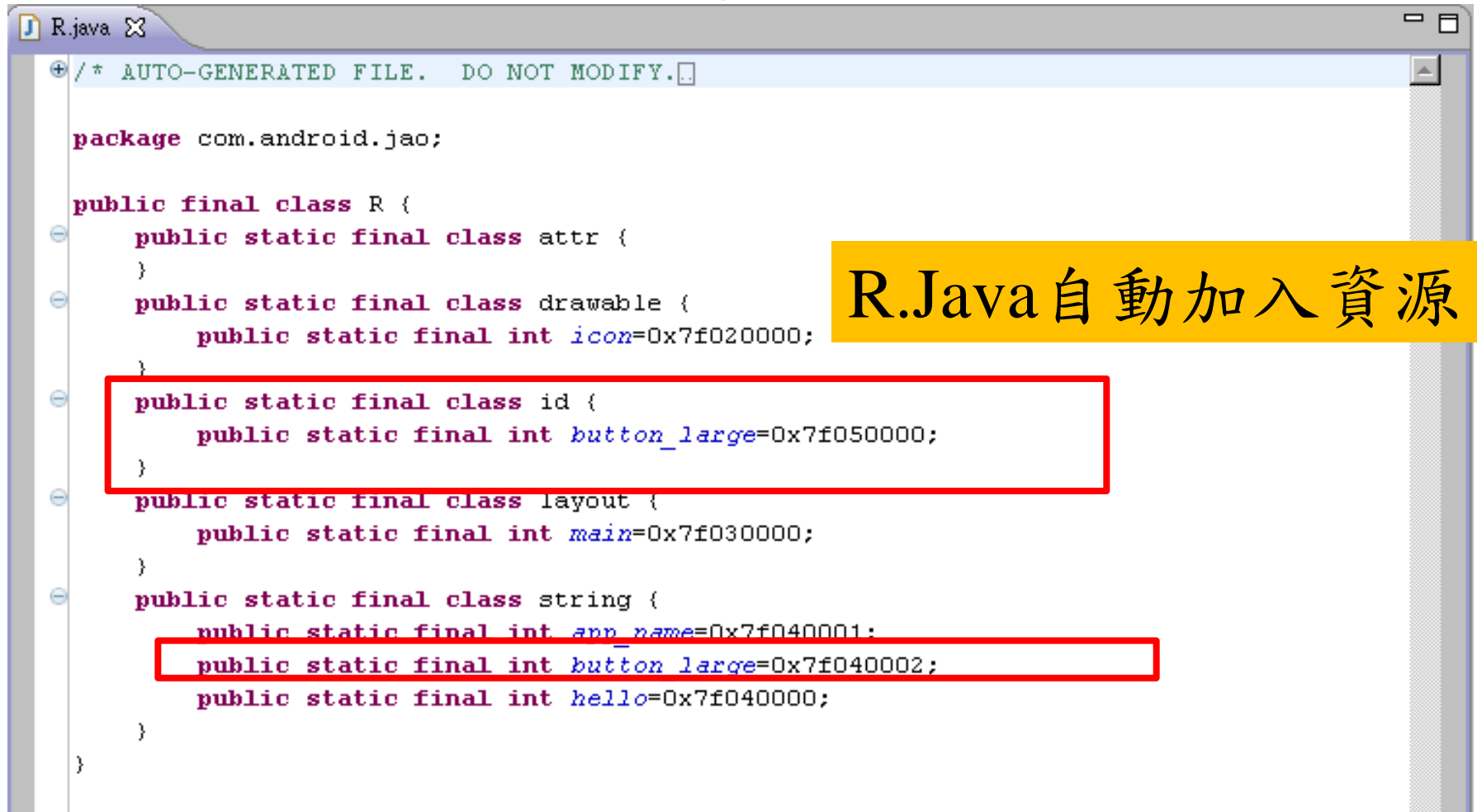
在 strings.xml 加入 button\_large 文字對應



# 預覽main.xml

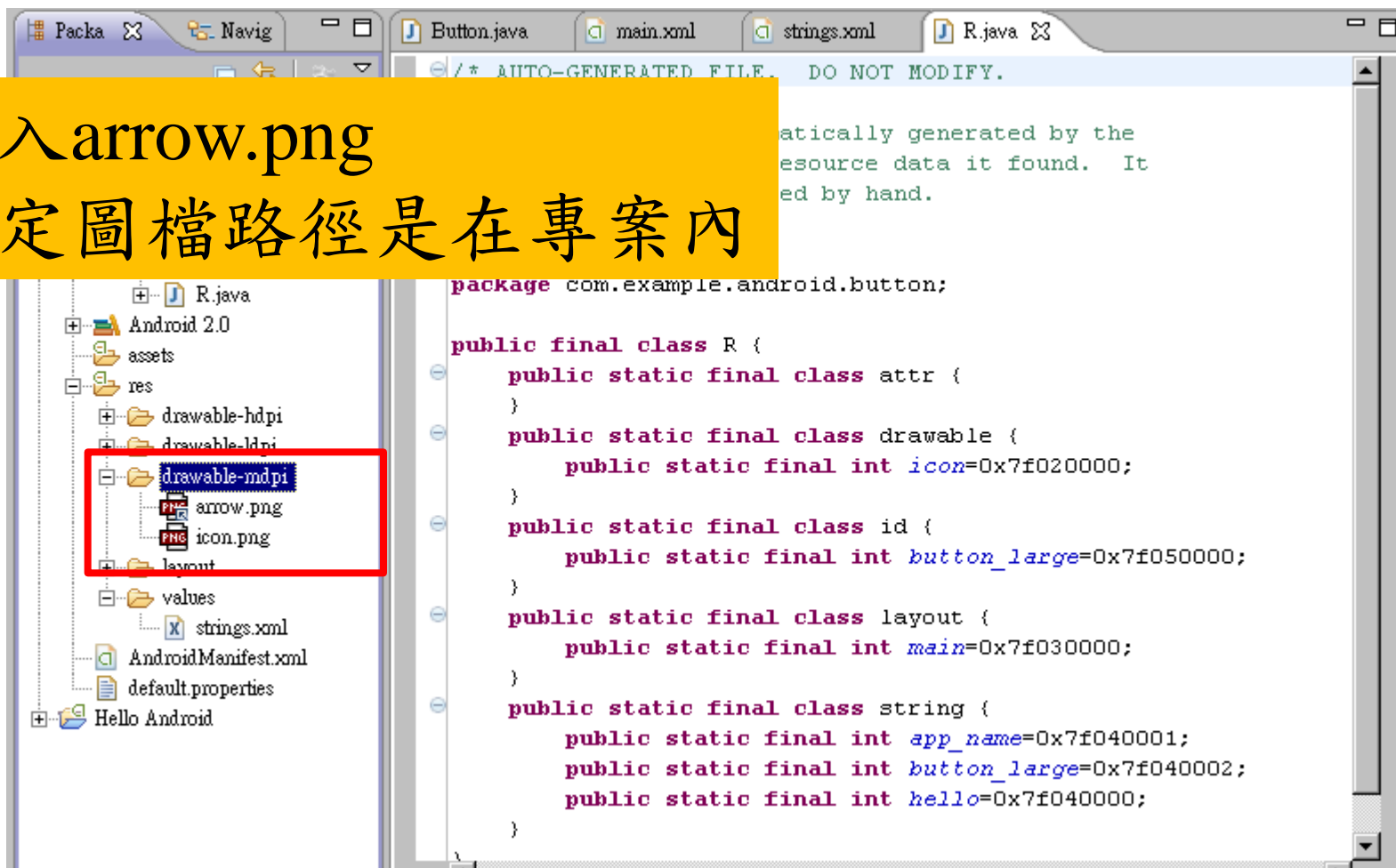


# R.java

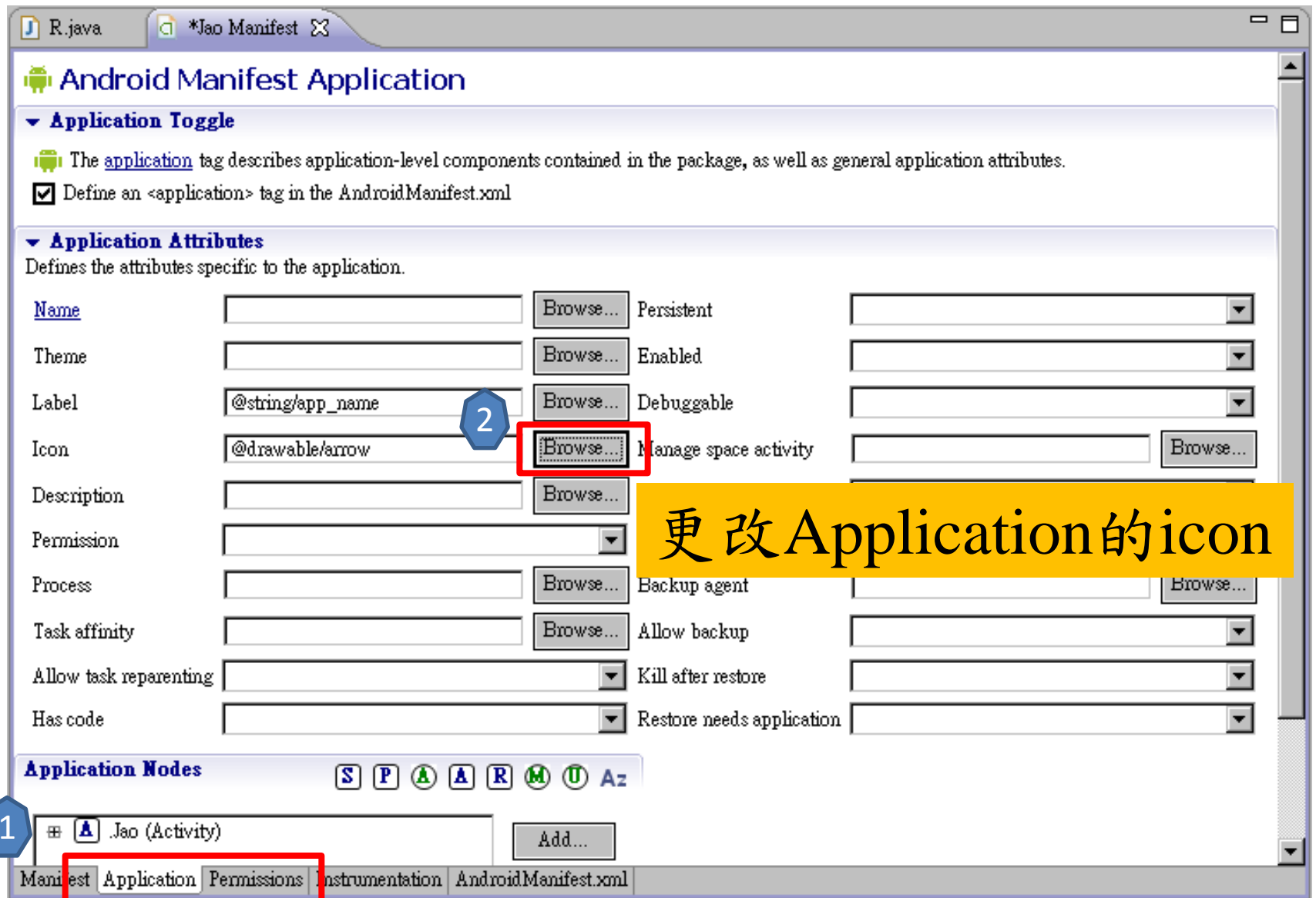


# 加入圖檔

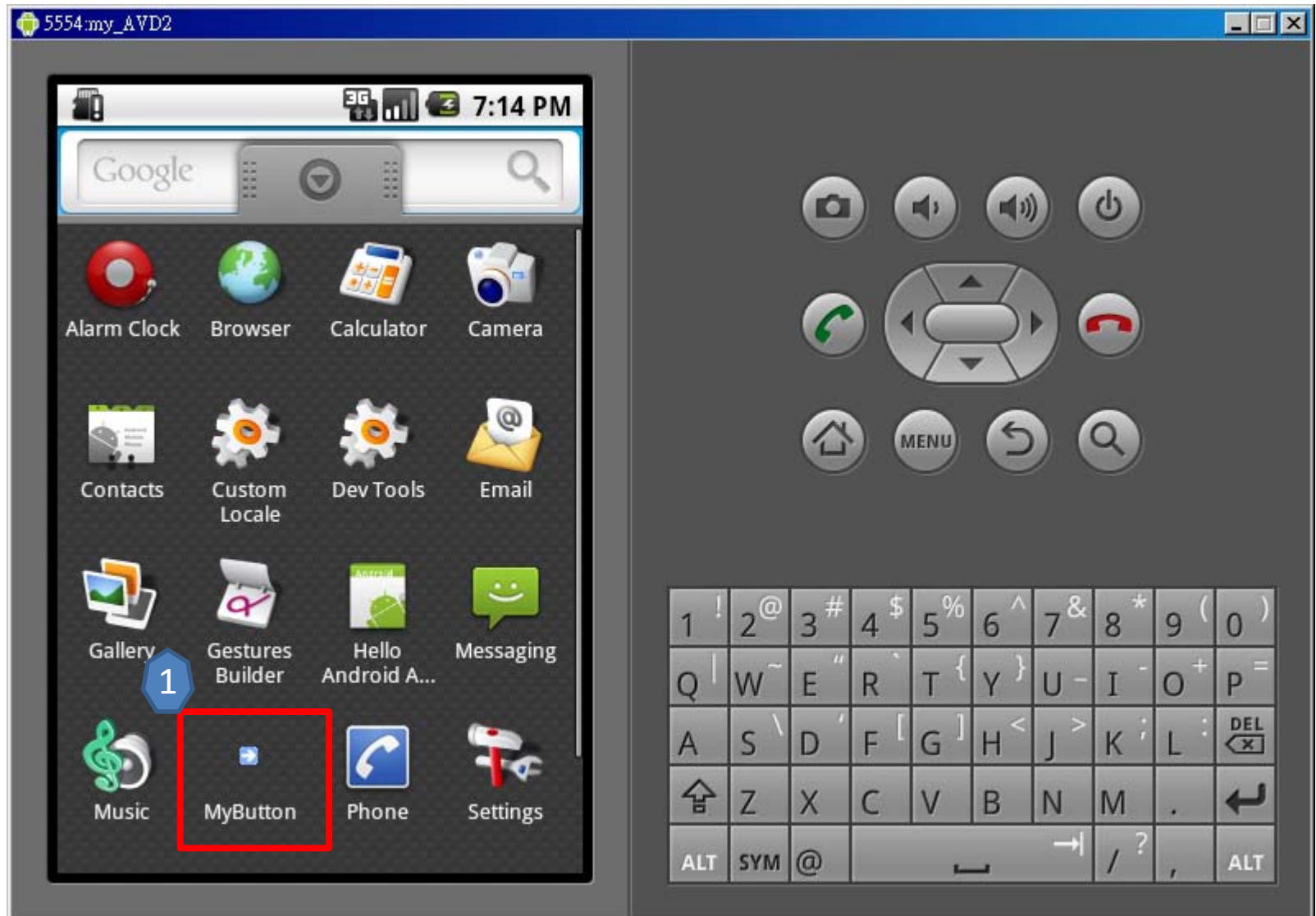
加入arrow.png  
確定圖檔路徑是在專案內



# AndroidManifest.xml



# 執行程式



# 執行程式



# 加入按鈕處理程式

```
package com.android.jao;
```

```
import android.app.Activity;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.*; // Button類別
```

```
public class Jao extends Activity {  
    /** Called when the activity is first created. */
```

```
    @Override
```

```
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);
```

main.xml中的android:id屬性

```
        Button clickbutton = (Button) findViewById(R.id.button_large); //加入Button物件
```

```
        clickbutton.setOnClickListener(new View.OnClickListener() { //onclick事件
```

```
            @Override
```

```
            public void onClick(View v) {  
                // TODO Auto-generated method stub  
                finish(); //結束程式
```

```
            }
```

```
        });
```

```
    }
```

```
}
```

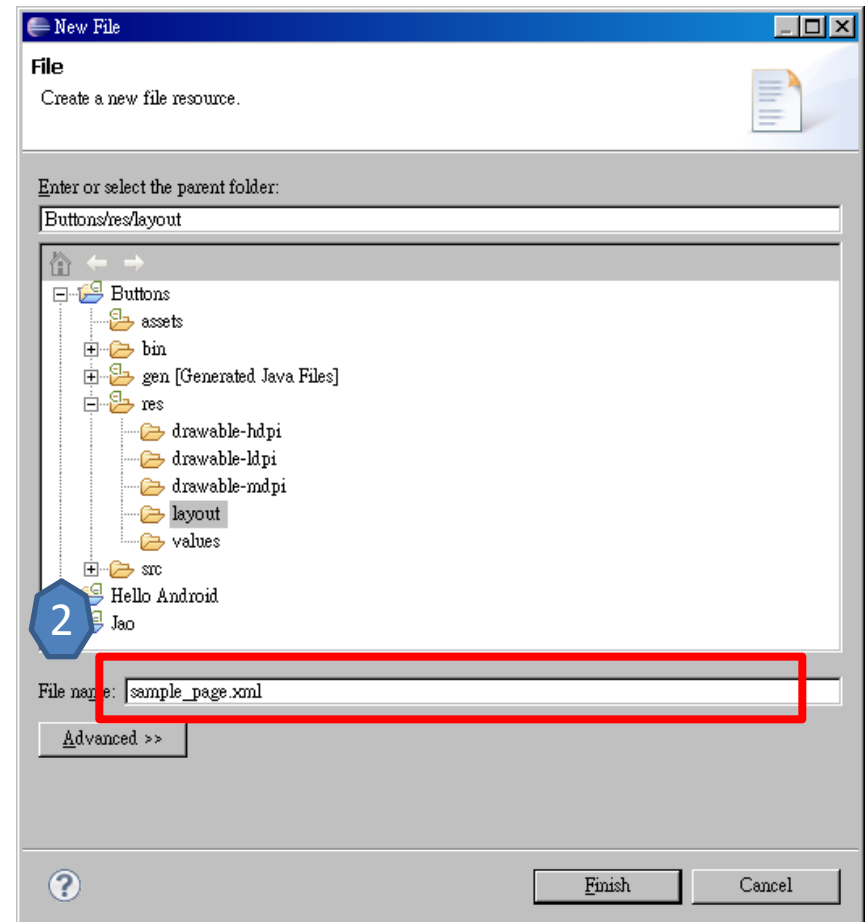
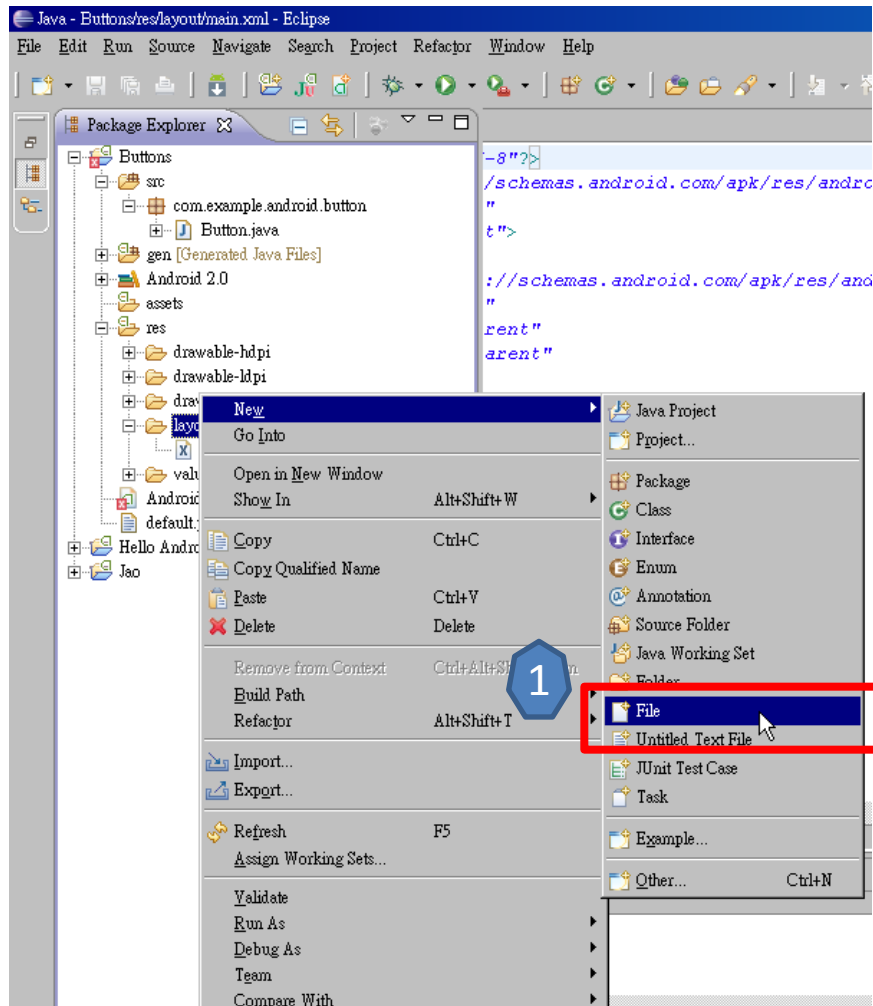
# 按鈕跳頁

- 建立對應的畫面sample\_page.xml  
–res/layout/sample\_page.xml
- 建立對應的程式sample\_page.java  
–src/com.example.android.button/  
sample\_page.java
- 在AndroidManifest.xml中加入  
Activity



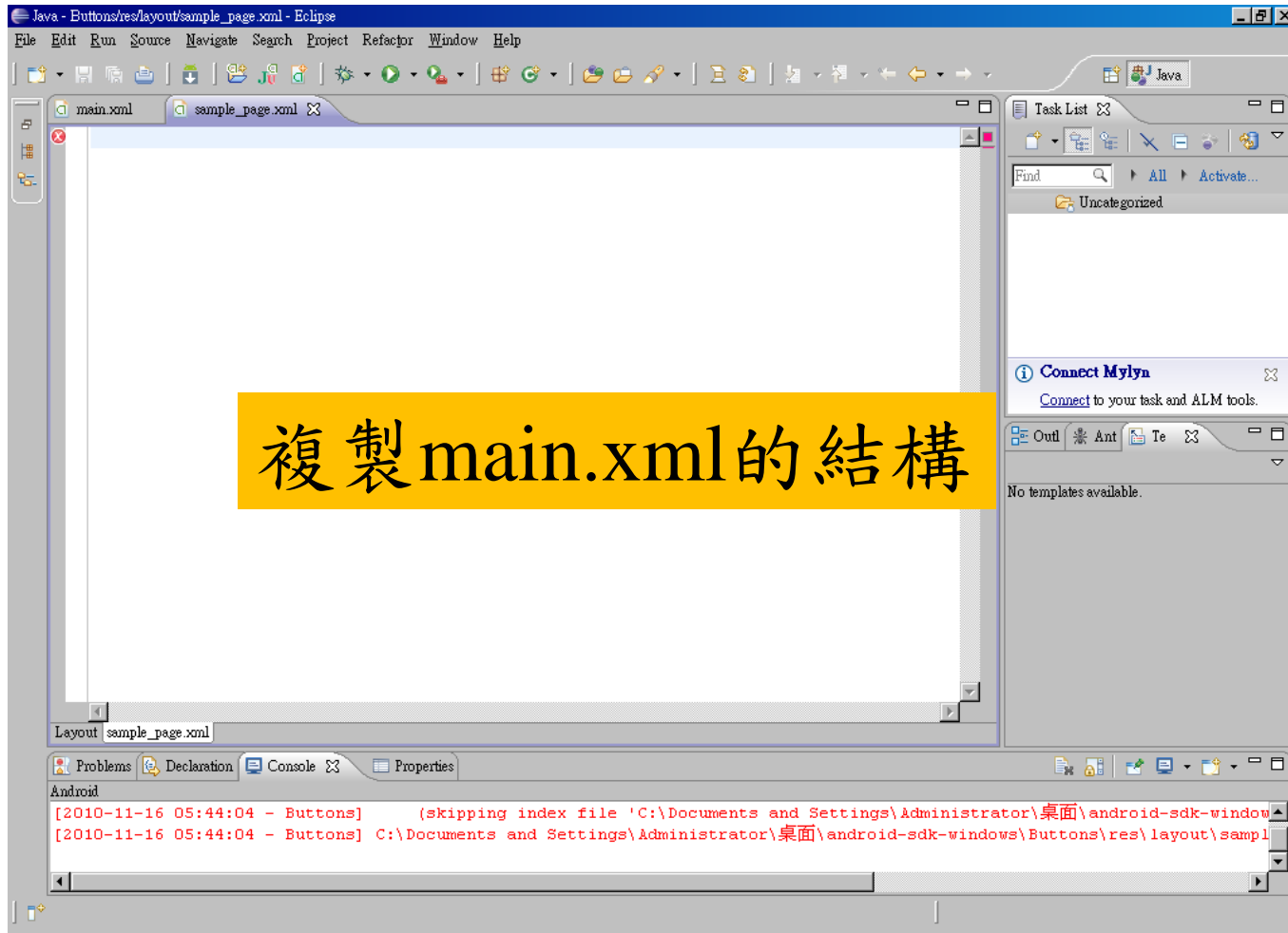
# 按鈕跳頁

- 建立對應的畫面sample\_page.xml



# 按鈕View跳頁

- 建立對應的畫面sample\_page.xml



# 按鈕跳頁

- 建立對應的畫面sample\_page.xml



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

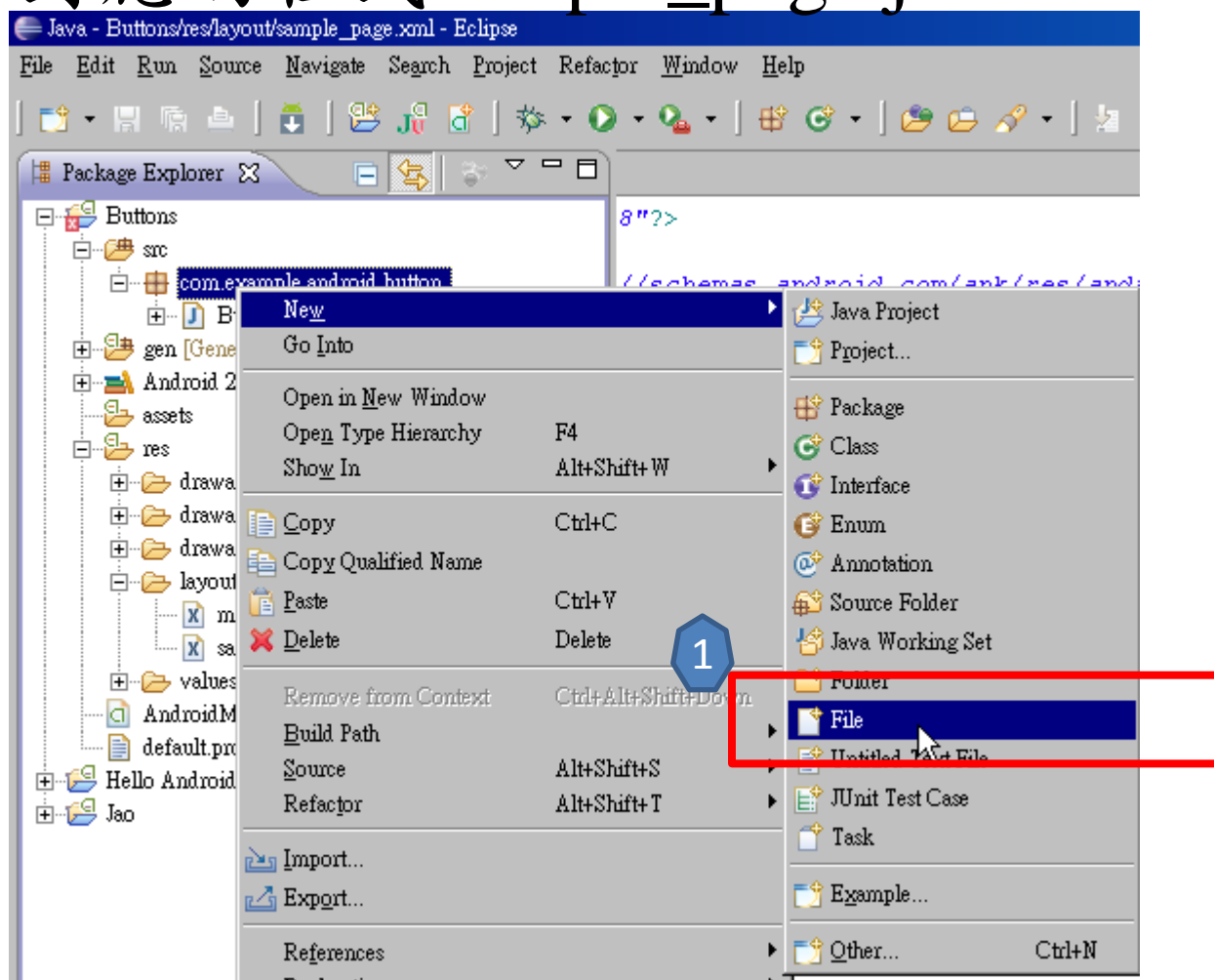
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >

    <Button android:id="@+id/button_sample"
        android:text="點我..點我.."
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```

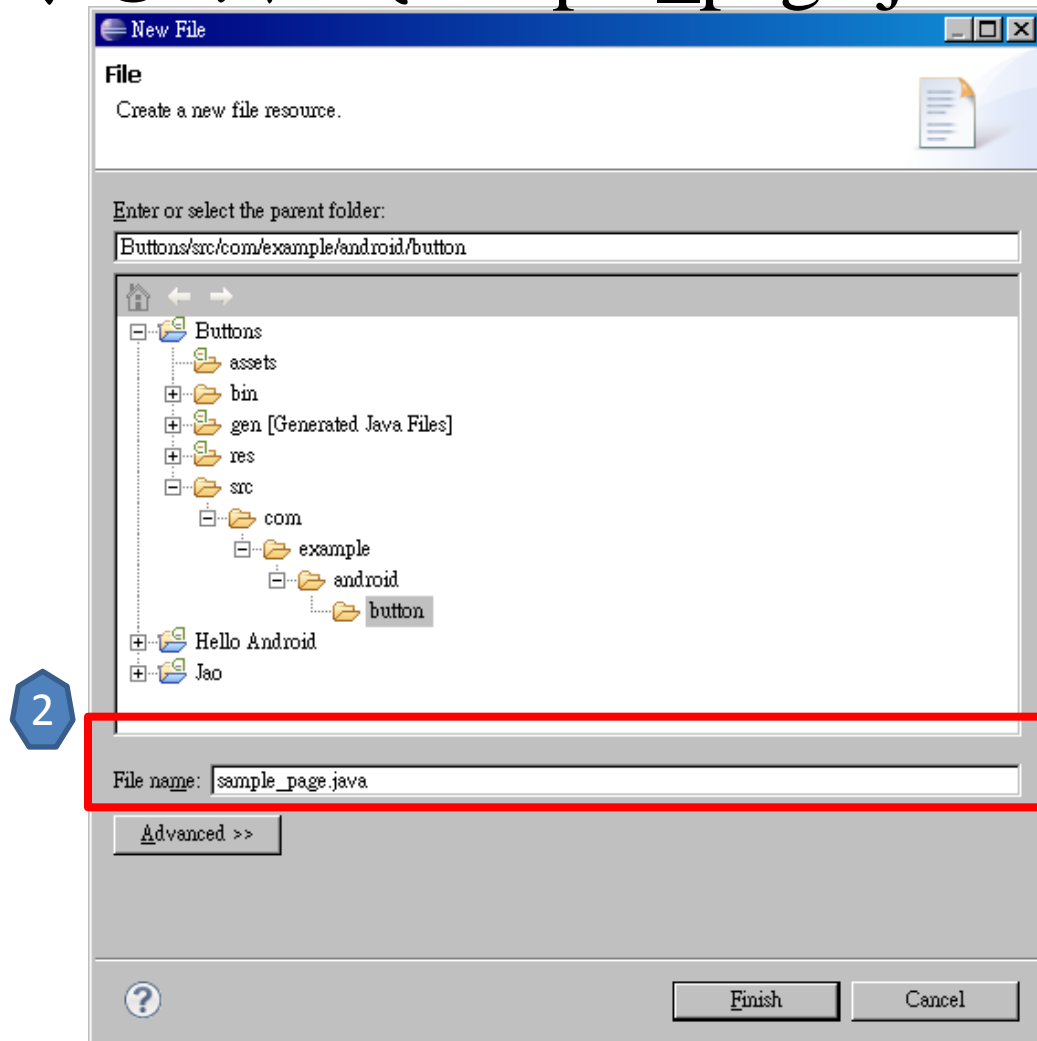
# 按鈕跳頁

- 建立對應的程式sample\_page.java



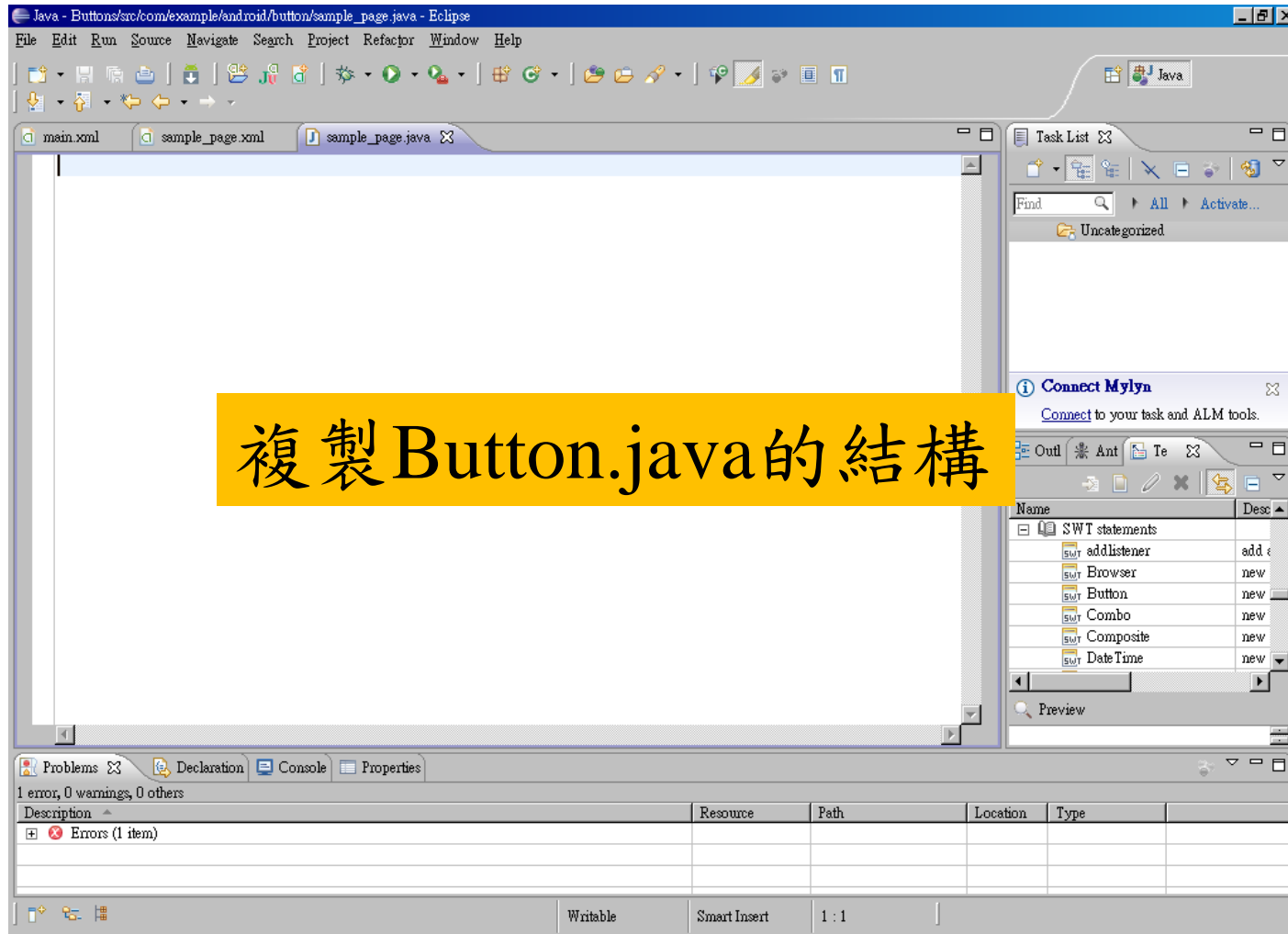
# 按鈕跳頁

- 建立對應的程式sample\_page.java



# 按鈕跳頁

- 建立對應的程式sample\_page.java



# 按鈕跳頁

- 建立對應的程式sample\_page.java

```
Java - Jao/src/com/android/jao/sample_page.java - Eclipse
File Edit Run Source Navigate Search Project Refactor Window Help

*sample_page.java Jao.java

1 import android.app.Activity;
import android.os.Bundle;
import android.widget.Button;

2 public class sample_page extends Activity {
    //宣告一個Button物件
    private Button button_sample;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.sample_page); //連結sample_page.xml畫面

3        //設定畫面的Title文字
        setTitle("這是第二的畫面");

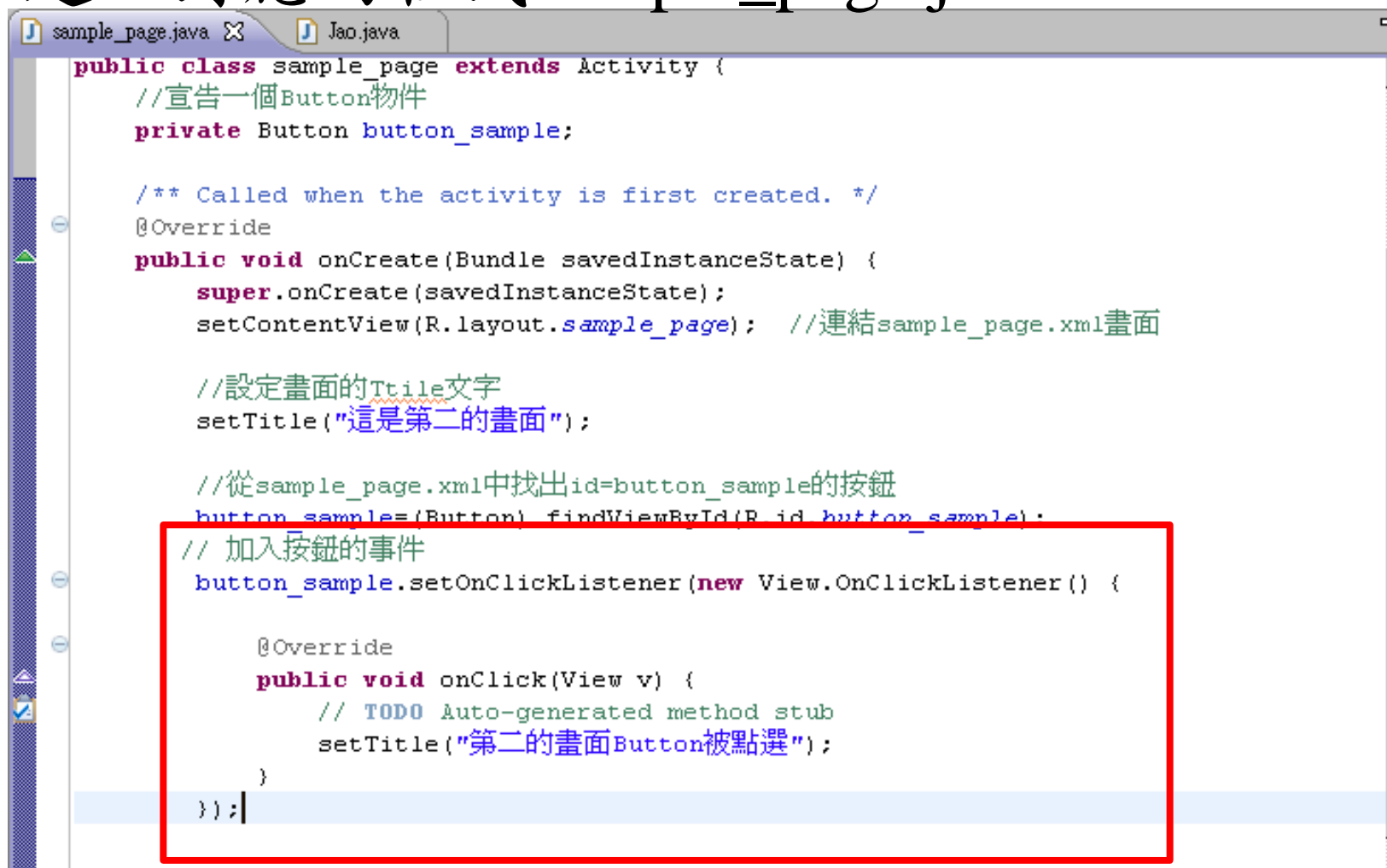
4        //從sample_page.xml中找出id=button_sample的按鈕
        button_sample = (Button) findViewById(R.id.button_sample);
        // 加入按鈕的事件
        button_sample.setOnClickListener(new View.OnClickListener() {
            public void onClick() {
                // TODO: Your code here
            }
        });
    }
}
```

View.OnClickListener() Anonymous Inner Typ  
View.OnCreateContextMenuListener() Anonym  
Interface definit  
clicked.

Summary

# 按鈕跳頁

- 建立對應的程式sample\_page.java



```
sample_page.java X Jao.java
public class sample_page extends Activity {
    //宣告一個Button物件
    private Button button_sample;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.sample_page); //連結sample_page.xml畫面

        //設定畫面的Title文字
        setTitle("這是第二的畫面");

        //從sample_page.xml中找出id=button_sample的按鈕
        button_sample=(Button) findViewById(R.id.button_sample);

        // 加入按鈕的事件
        button_sample.setOnClickListener(new View.OnClickListener() {


            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                setTitle("第二的畫面Button被點選");
            }

        });
    }
}
```



# 按鈕跳頁

- 加入從Jao.java呼叫sample\_page.java
- 先在main.xml加入一個按鈕



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

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        >

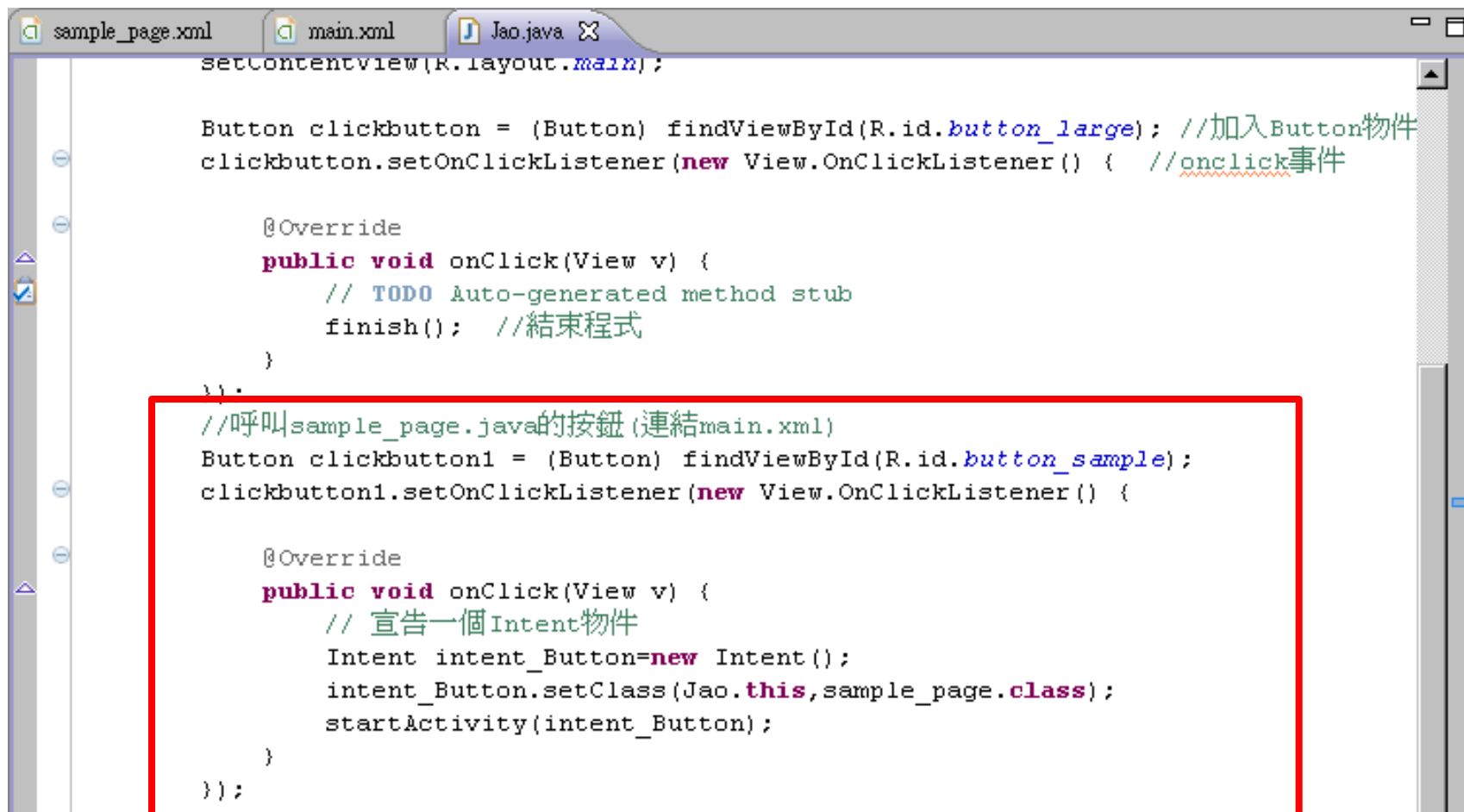
        <Button android:id="@+id/button_large"
            android:text="@string/button_large"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <Button android:id="@+id/button_sample"
            android:text="@string/button_sample"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

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

# 按鈕跳頁

- 在Jao.java加入呼叫按鈕與程式



```
sample_page.xml  main.xml  Jao.java  X3

setContentView(R.layout.main);

Button clickbutton = (Button) findViewById(R.id.button_large); //加入Button物件
clickbutton.setOnClickListener(new View.OnClickListener() { //onclick事件

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        finish(); //結束程式
    }
});

//呼叫sample_page.java的按鈕 (連結main.xml)
Button clickbutton1 = (Button) findViewById(R.id.button_sample);
clickbutton1.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // 宣告一個Intent物件
        Intent intent_Button=new Intent();
        intent_Button.setClass(Jao.this,sample_page.class);
        startActivity(intent_Button);
    }
});
```

# 按鈕跳頁

- 在AndroidManifest.xml中加入Activity



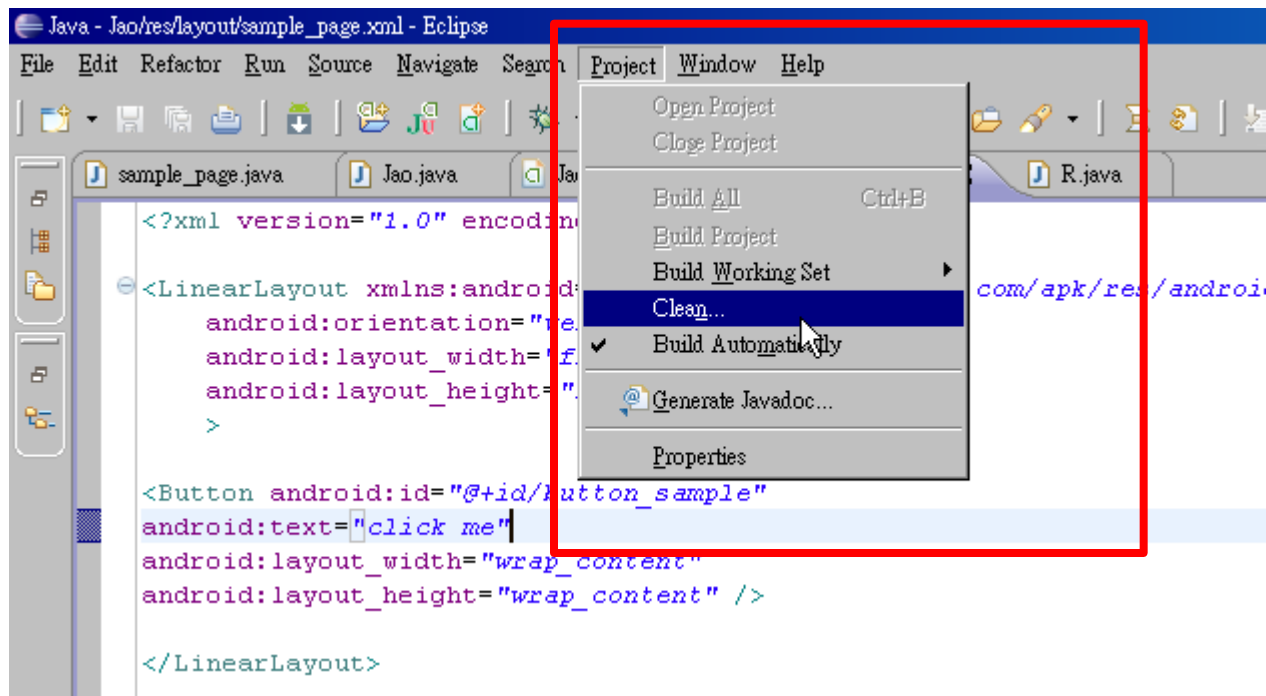
The screenshot shows an IDE window with four tabs: Jao.java, main.xml, sample\_page.java, and \*Jao Manifest. The \*Jao Manifest tab is active, displaying the contents of the AndroidManifest.xml file. The XML code is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.jao"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:label="@string/app_name" android:icon="@drawable/arrow">
        <activity android:name=".Jao"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".sample_page"></activity>
    </application>
    <uses-sdk android:minSdkVersion="5" />
</manifest>
```

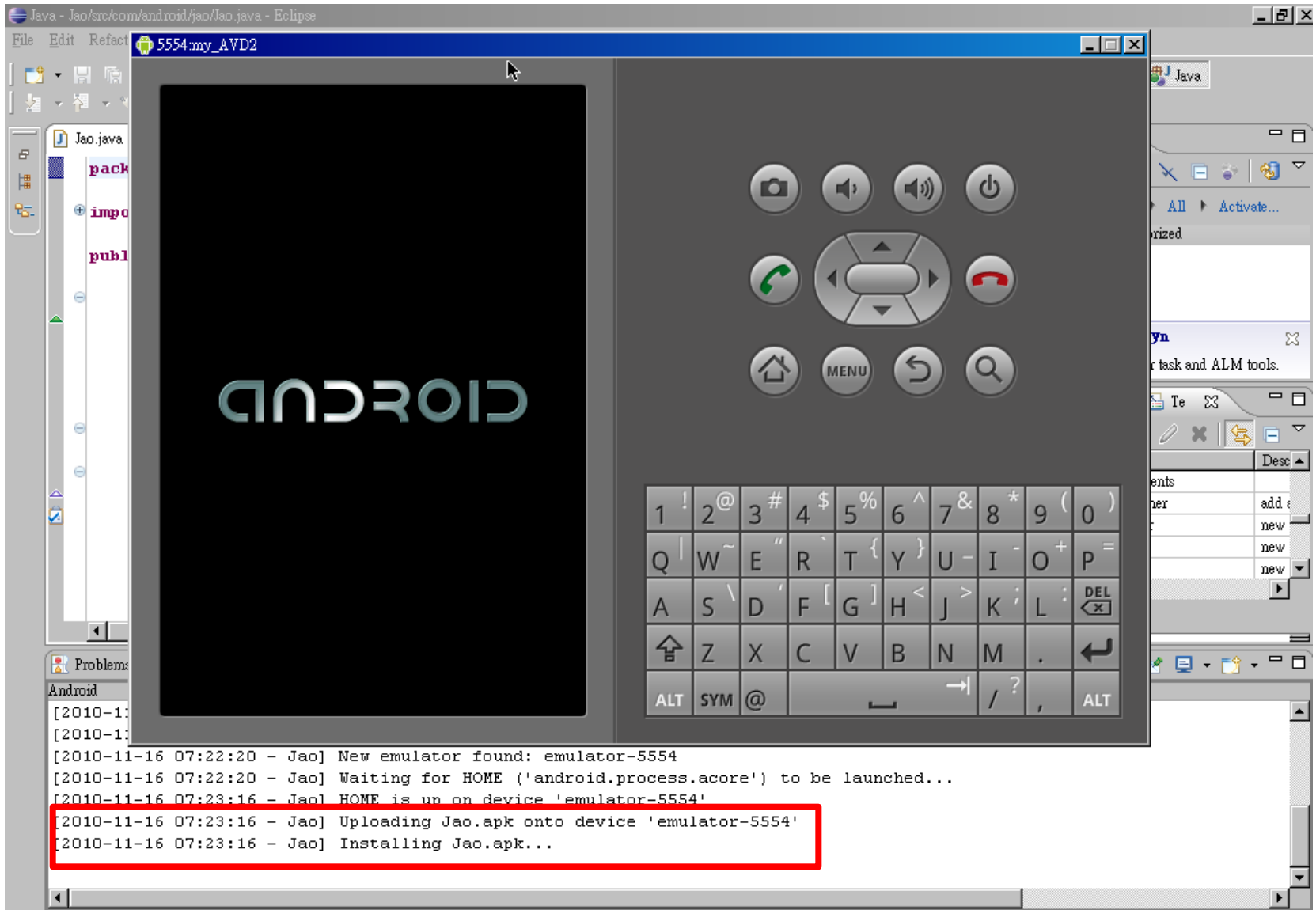
The line `<activity android:name=".sample_page"></activity>` is highlighted with a red rectangular box.

# 執行按鈕跳頁

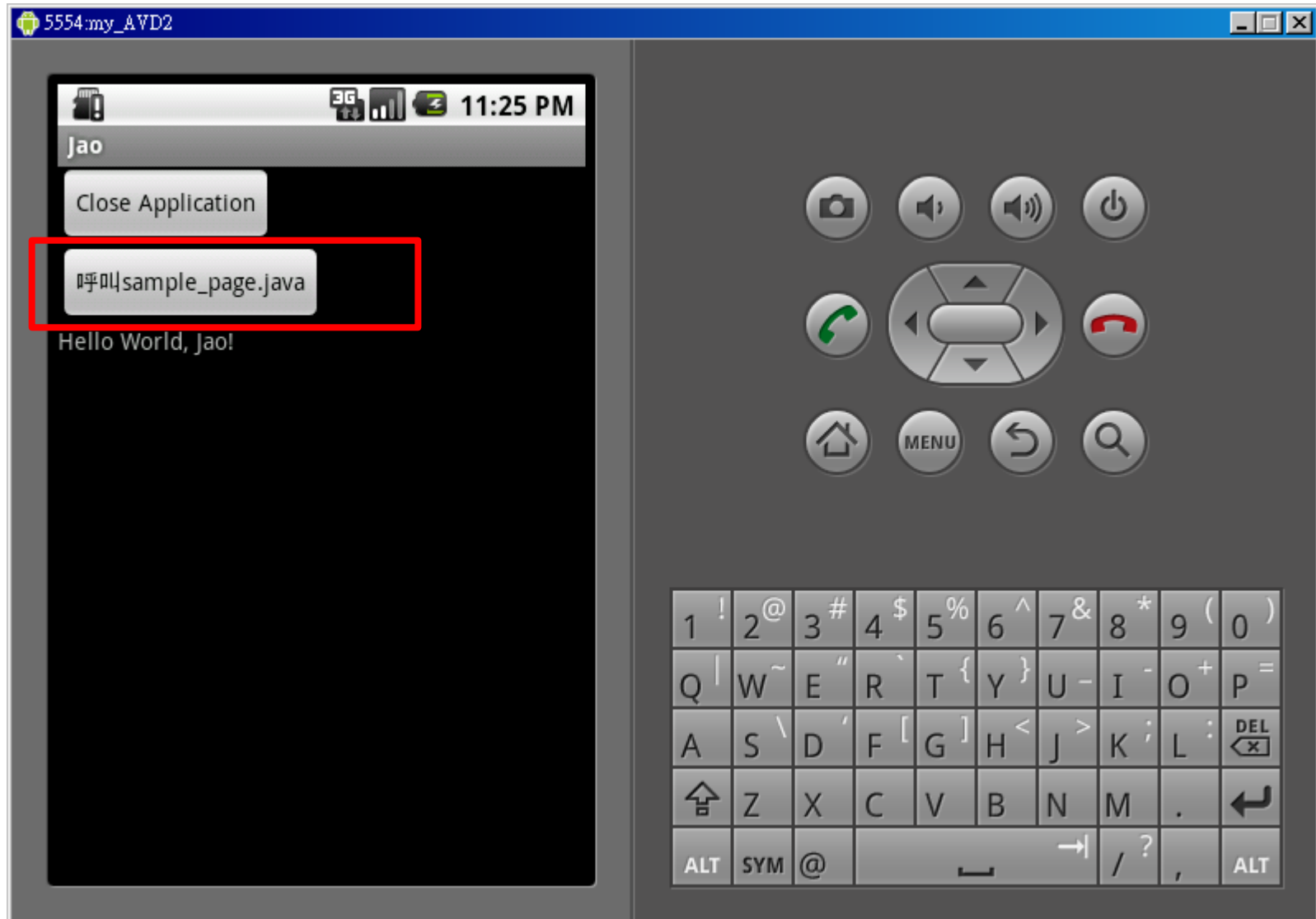
- 解決” Unparsed aapt error(s)! Check the console for output” 問題（如果有）



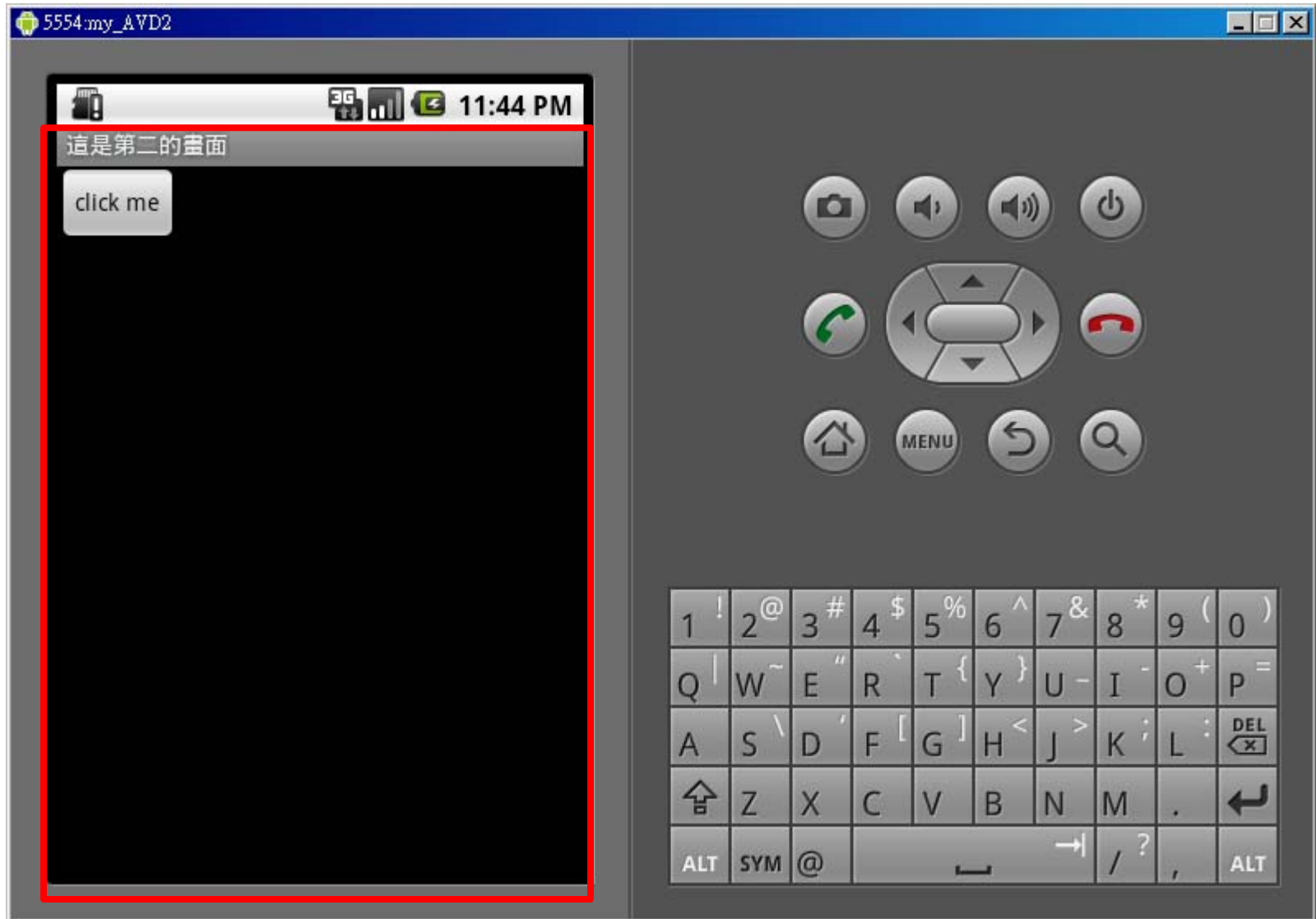
# 執行按鈕跳頁



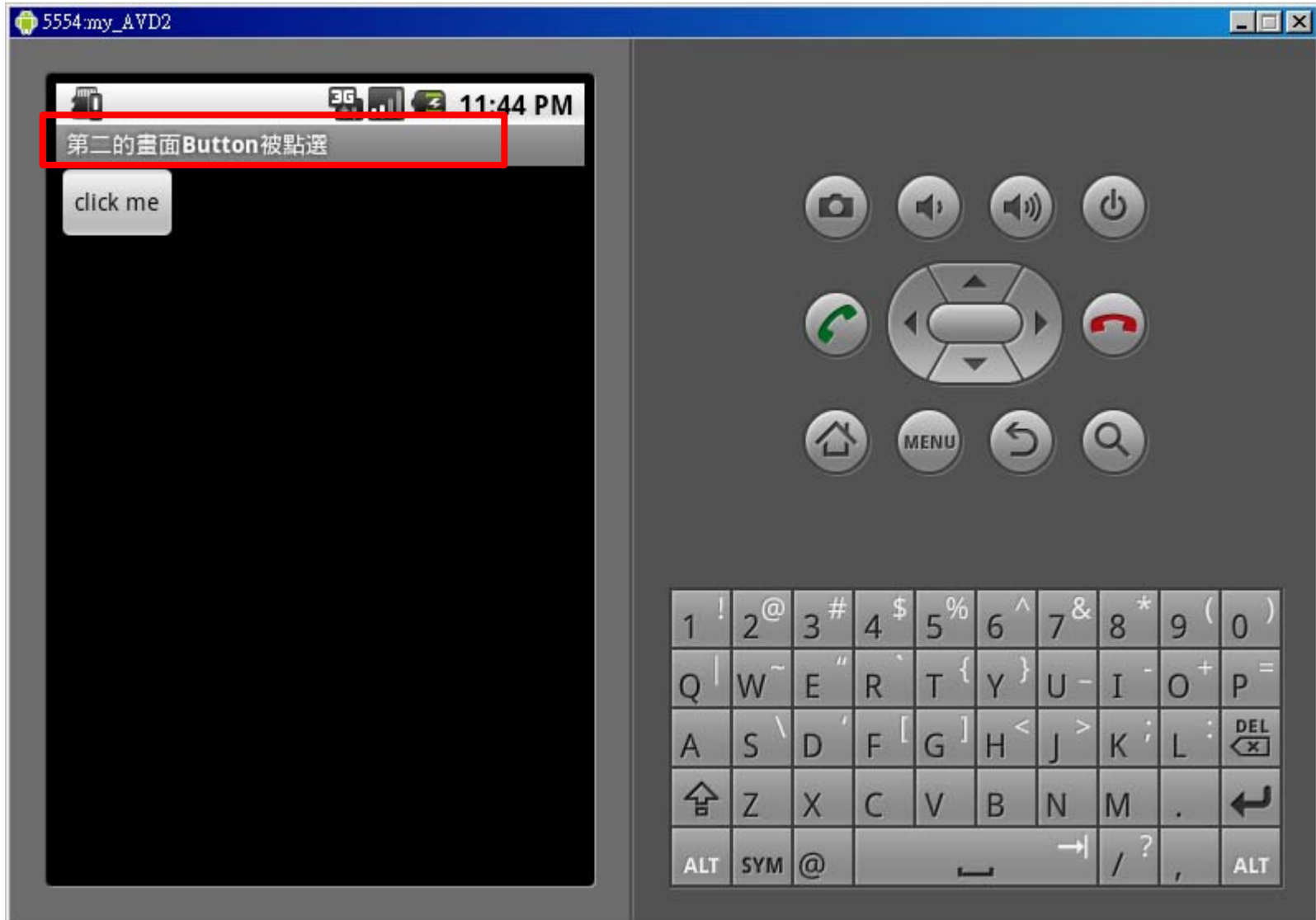
# 執行按鈕跳頁



# 執行按鈕跳頁



# 執行按鈕跳頁





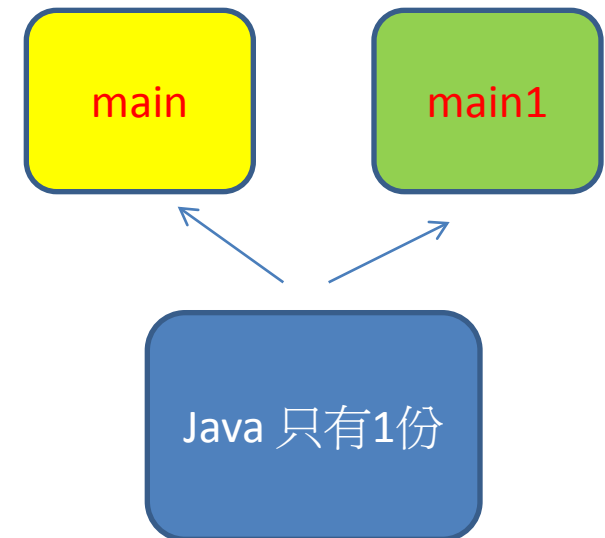
# 跳畫面但不換Activity

## 只換Layout

```
Button b1 = null;
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView( 1 );

    // 設定 Layout
    public void setContentView(final int layoutNum)
    {
        // 設定 Layout 與找出 Button 物件
        if( layoutNum == 1 )
        {
            setContentView(R.layout.main);
            b1 = (Button)findViewById( R.id.button1 );
        }
        else
        {
            setContentView(R.layout.main1);
            b1 = (Button)findViewById( R.id.button2 );
        }

        // 建立 Button onClick Trigger
        b1.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                setContentView( (layoutNum == 1) ? 2 : 1 );
            }
        });
    }
}
```



# BMI範例

# BMI計算範例

- **B**ody **M**ass **I**ndex = 體重(kg)/(身高\*身高)(m)
- 產生新專案HelloBMI
- 利用Droiddraw設計介面與產生xml介面檔
  - 修改main.xml
- 撰寫HelloBMI.java程式
  - 利用按鈕view計算BMI值
  - 利用兩個TextView呈現結果

# BMI UI

- **B**ody **M**ass **I**ndex = 體重(kg)/(身高\*身高)(m)
- 我們需要
  - 兩個顯示view(**TextView**) 來提示填入身高體重數字
  - 兩個輸入view(**EditText**)來填入身高體重數字
  - 需要一個按鈕view(**Button**)來開始計算
  - 需要兩個顯示view(**TextView**)來顯示計算結果

# 輸入View

<EditText

android:id="@+id/名稱"

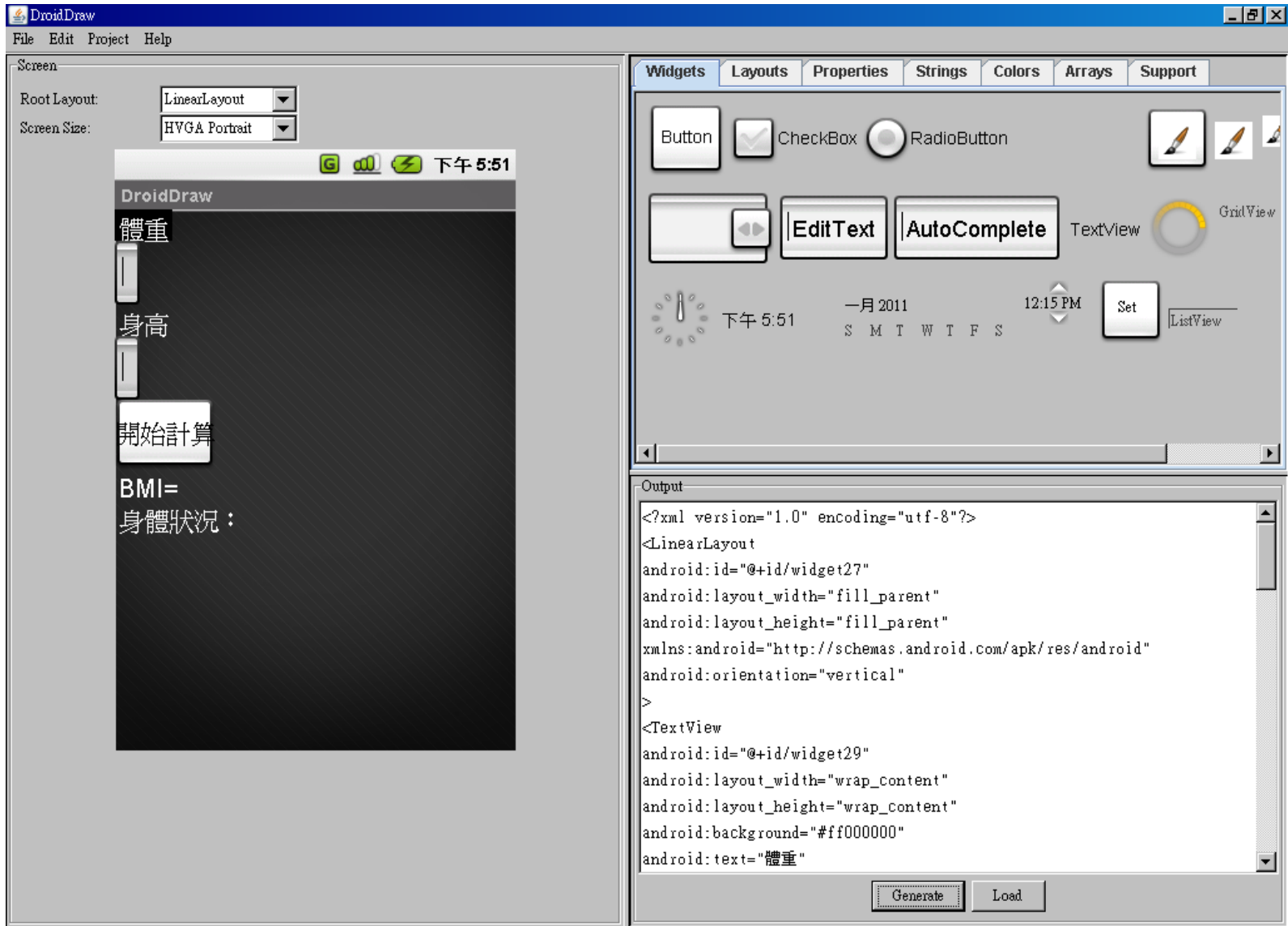
android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:numeric="true">

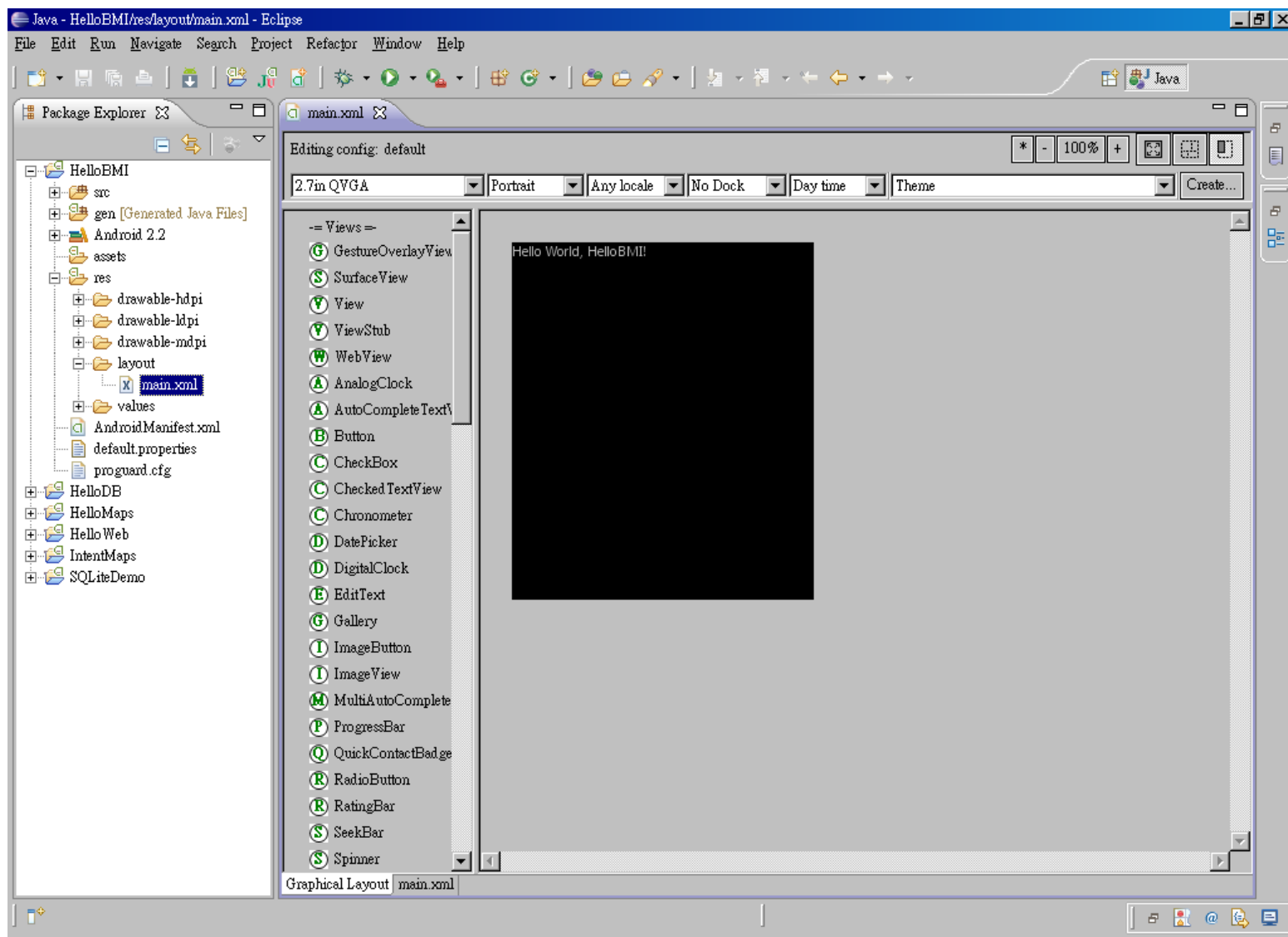
</EditText>

# BMI UI



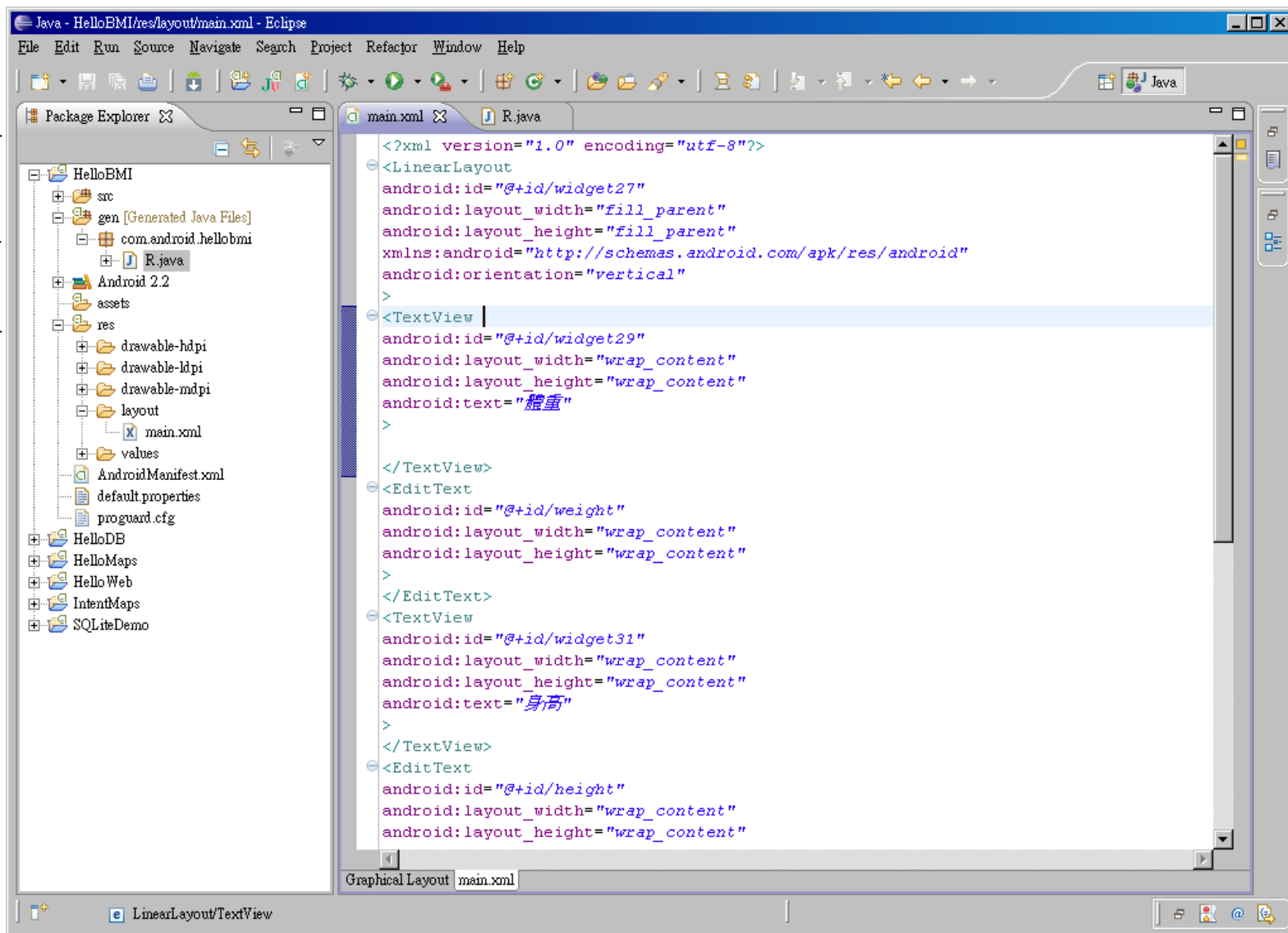
# 修改main.xml

未  
修  
改  
前



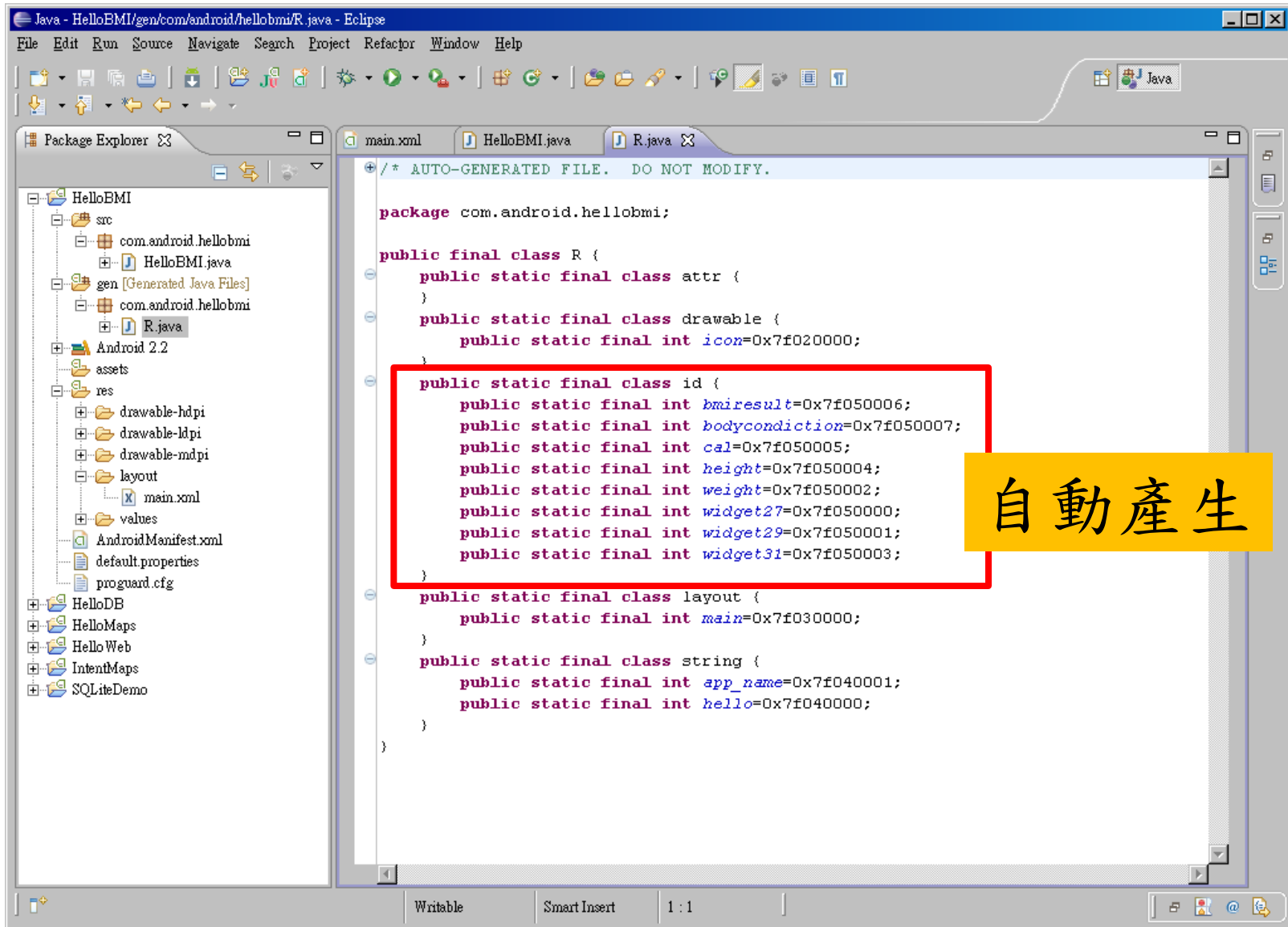
# 修改main.xml

修改後

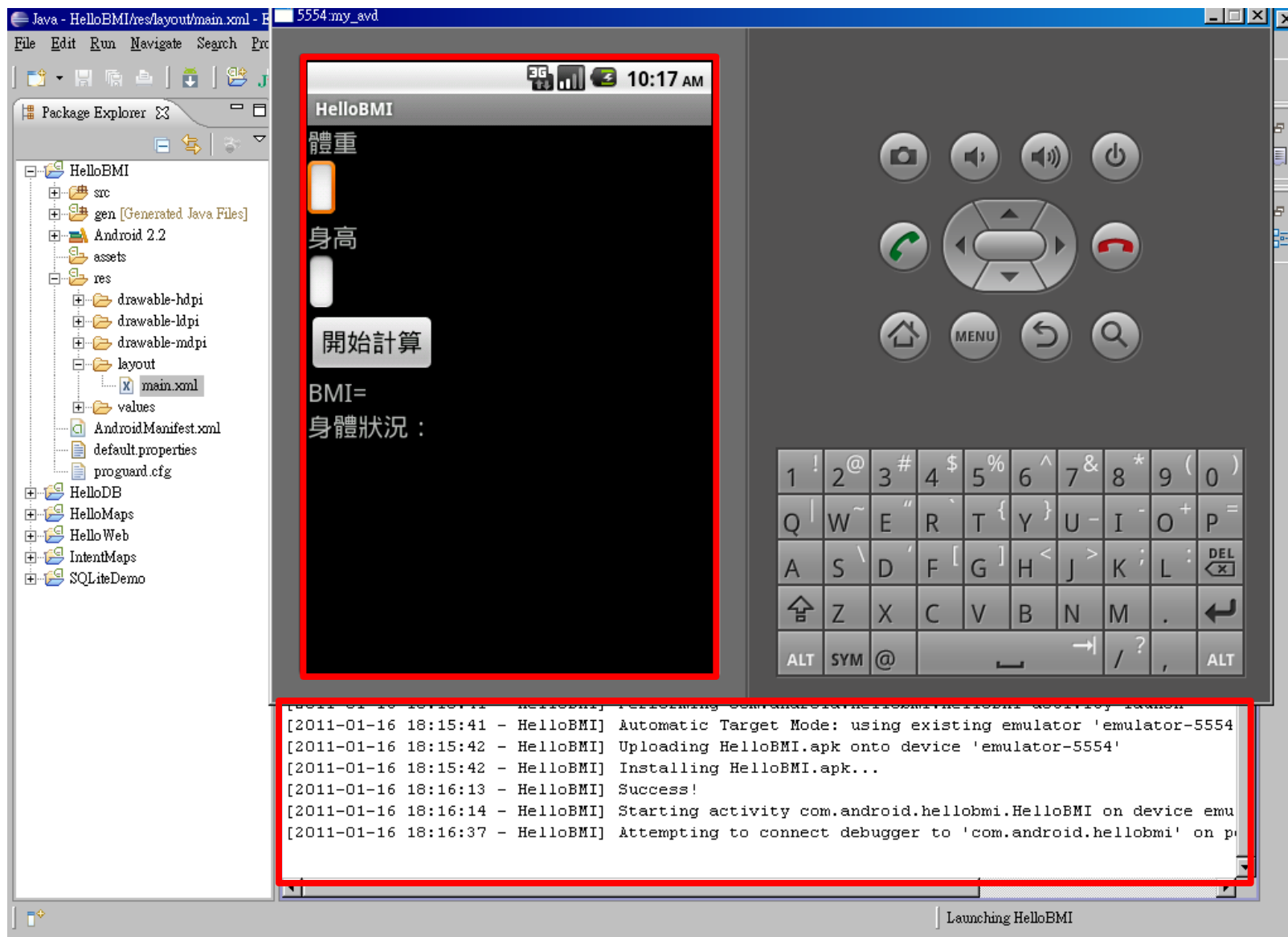




# R.java



# BMI UI執行結果



# BMI程式(I)

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

    Button button=(Button)findViewById(R.id.calbmi);
    button.setOnClickListener(calcbmi);
}
```

建立按鈕事件

# BMI程式(II)

```
public OnClickListener calcbmi=new OnClickListener(){  
    public void onClick(View v){
```

取得輸入值

```
        DecimalFormat nf=new DecimalFormat("0.00");  
        EditText fieldweight=(EditText)findViewById(R.id.weight);  
        EditText fieldheight=(EditText)findViewById(R.id.height);  
        double weight=Double.parseDouble(fieldweight.getText().toString());  
        double height=Double.parseDouble(fieldheight.getText().toString())/100;
```

```
        double bmi=weight/(height*height);
```

計算BMI

```
        TextView result=(TextView)findViewById(R.id.result_bmi);  
        result.setText("你的BMI值是"+nf.format(bmi));
```

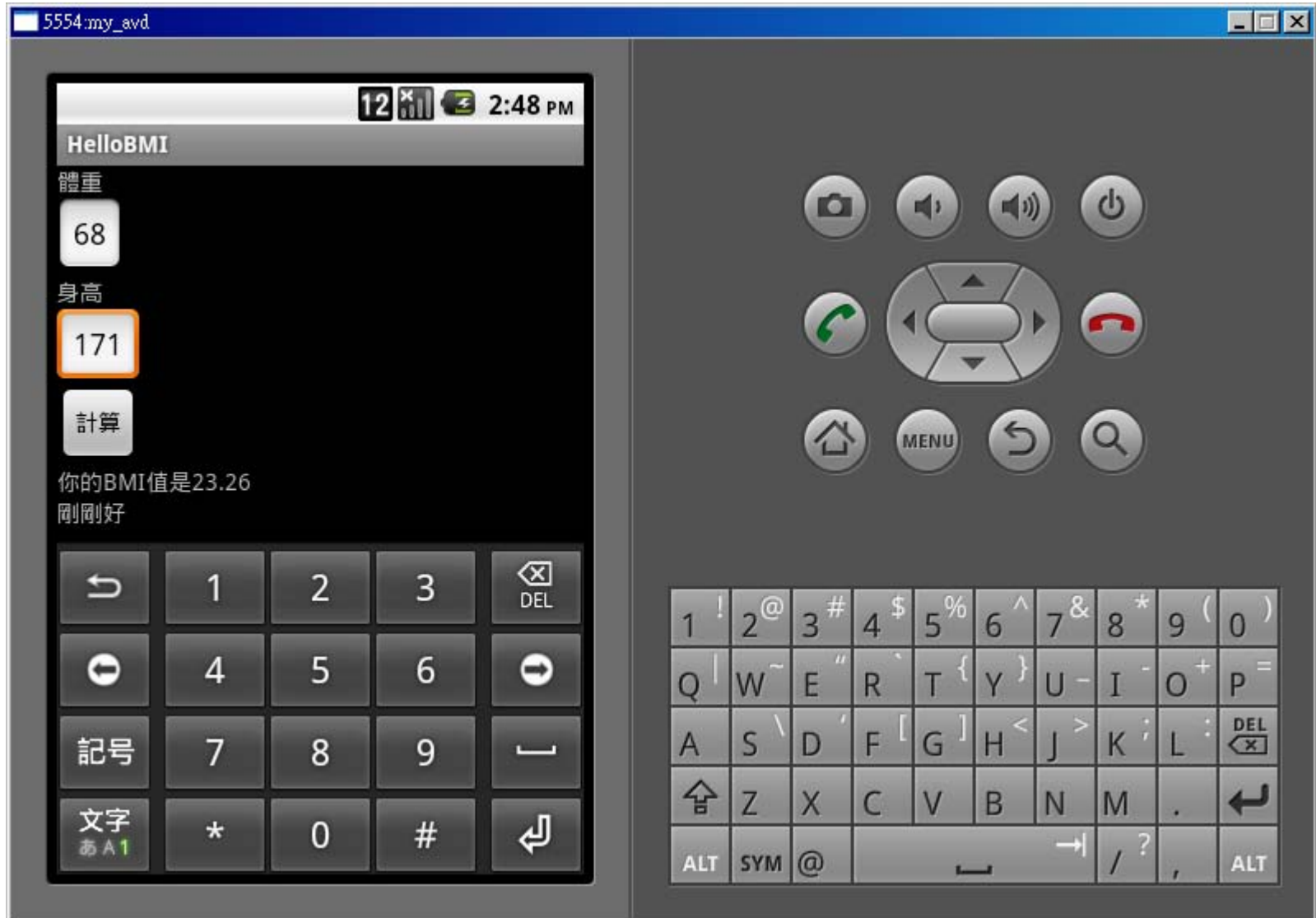
顯示BMI

```
        TextView suggest=(TextView)findViewById(R.id.con_bmi);  
        if(bmi>25){  
            suggest.setText("太胖了");  
        }else if(bmi<20){  
            suggest.setText("太瘦了");  
        }else{  
            suggest.setText("剛剛好");  
        }
```

顯示身體狀況

```
    }  
};
```

# BMI範例結果



# 練習

- 重構：將HelloBMI中的findViewById與setOnClickListener改成副程式
- 將字串改存入string.xml中，再在HelloBMI.java中顯示

# 練習解答

- 將HelloBMI中的findViewById與setOnClickListener改成副程式

```
public class HelloBMI extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        findViews();
        setListeners();
    }

    private Button button;

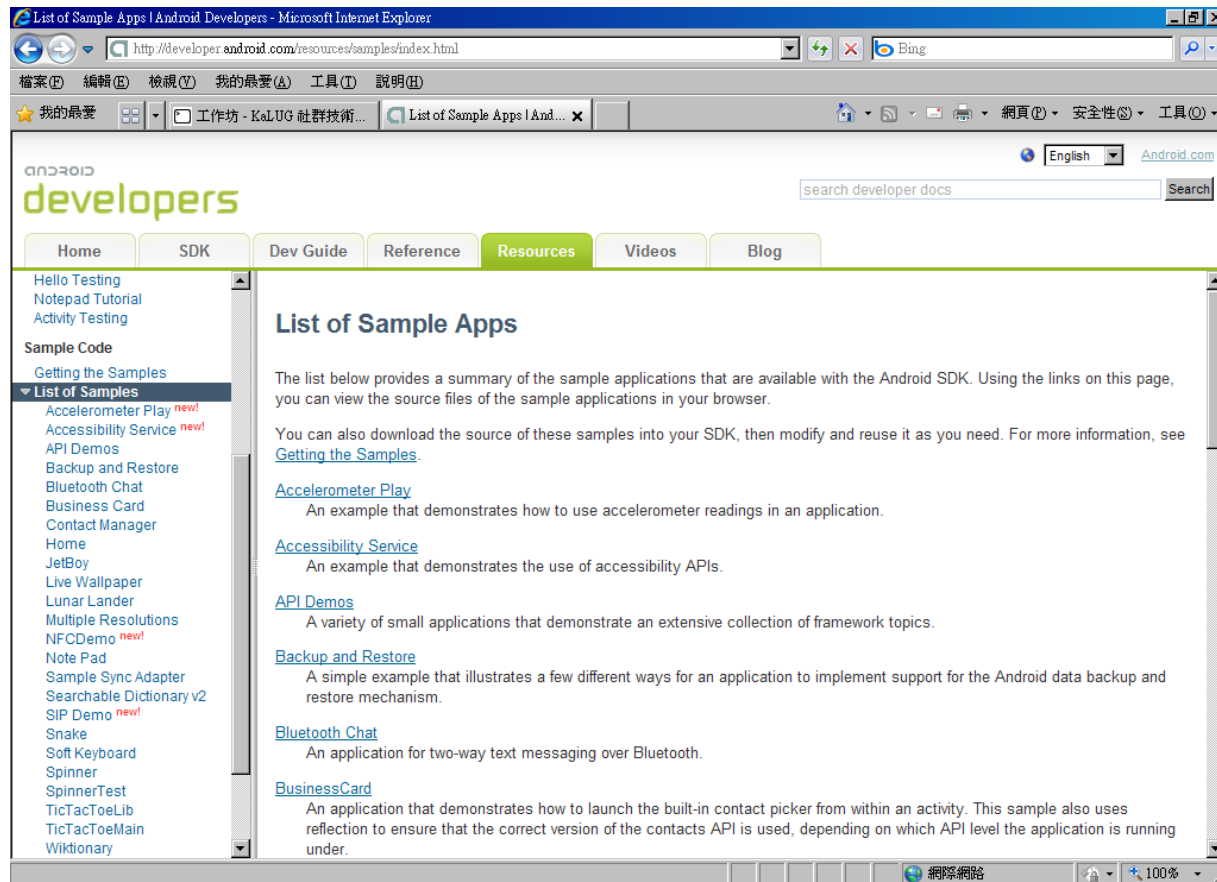
    private void findViews(){
        button=(Button)findViewById(R.id.calcbmi);
    }

    private void setListeners(){
        button.setOnClickListener(calcbmi);
    }

    public OnClickListener calcbmi=new OnClickListener(){
        public void onClick(View v){
            DecimalFormat nf=new DecimalFormat("0.00");
            EditText fieldweight=(EditText)findViewById(R.id.w
            EditText fieldheight=(EditText)findViewById(R.id.h
```

# More Samples

- <http://developer.android.com/resources/samples/index.html>





# 對話框(Dialog) I

- 結構：

```
AlertDialog.Builder builder=new AlertDialog.Builder(專案.this);  
builder.setTitle(對話框標題訊息);  
builder.setMessage(對話框內容);  
builder.show();
```

具備實體，會佔記憶體

# 對話框(Dialog) I



```
main.xml HelloBMI.java X
EditText fieldheight=(EditText)findViewById(R.id.height);
double weight=Double.parseDouble(fieldweight.getText().toString());
double height=Double.parseDouble(fieldheight.getText().toString())/100;

double bmi=weight/(height*height);

String s="";

TextView result=(TextView)findViewById(R.id.result_bmi);
result.setText("你的BMI值是"+nf.format(bmi));

TextView suggest=(TextView)findViewById(R.id.con_bmi);
if(bmi>25){
    suggest.setText("太胖了");
    s="太胖了";
} else if(bmi<20){
    suggest.setText("太瘦了");
    s="太瘦了";
} else{
    suggest.setText("剛剛好");
    s="剛剛好";
}
openDialog(s); // 開啟對話框
};

private void openDialog(String s){
    AlertDialog.Builder builder=new AlertDialog.Builder(HelloBMI.this);
    builder.setTitle("HelloBMI對話框範例");
    builder.setMessage(s);
    builder.show();
}
```

# 對話框(Dialog) I



# 對話框 (Dialog) II

- 結構：

```
new AlertDialog.Builder(專案.this)  
.setTitle(對話框標題訊息)  
.setMessage(對話框內容)  
.show();
```

匿名實體，不佔記憶體

# 對話框 (Dialog) II



```
main.xml HelloBMI.java X
EditText fieldheight=(EditText)findViewById(R.id.height);
double weight=Double.parseDouble(fieldweight.getText().toString());
double height=Double.parseDouble(fieldheight.getText().toString())/100;

double bmi=weight/(height*height);

String s="";

TextView result=(TextView)findViewById(R.id.result_bmi);
result.setText("你的BMI值是"+nf.format(bmi));

TextView suggest=(TextView)findViewById(R.id.con_bmi);
if(bmi>25){
    suggest.setText("太胖了");
    s="太胖了";
} else if(bmi<20){
    suggest.setText("太瘦了");
    s="太瘦了";
} else{
    suggest.setText("剛剛好");
    s="剛剛好";
}
openDialog(s); // 開啟對話框
};

private void openDialog(String s){
    new AlertDialog.Builder(HelloBMI.this)
        .setTitle("HelloBMI對話框範例")
        .setMessage(s)
        .show();
}
```

# 對話框 (Dialog) II



結果相同，但不佔記憶體

# 對話框(Dialog)按鈕

- 結構：

```
new AlertDialog.Builder(專案.this)
.setTitle(對話框標題訊息)
.setMessage(對話框內容)
.setPositiveButton("確認",new DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int which) {
        // TODO Auto-generated method stub
    }
})
.show();
```

匿名實體，不佔記憶體

# 對話框(Dialog)按鈕





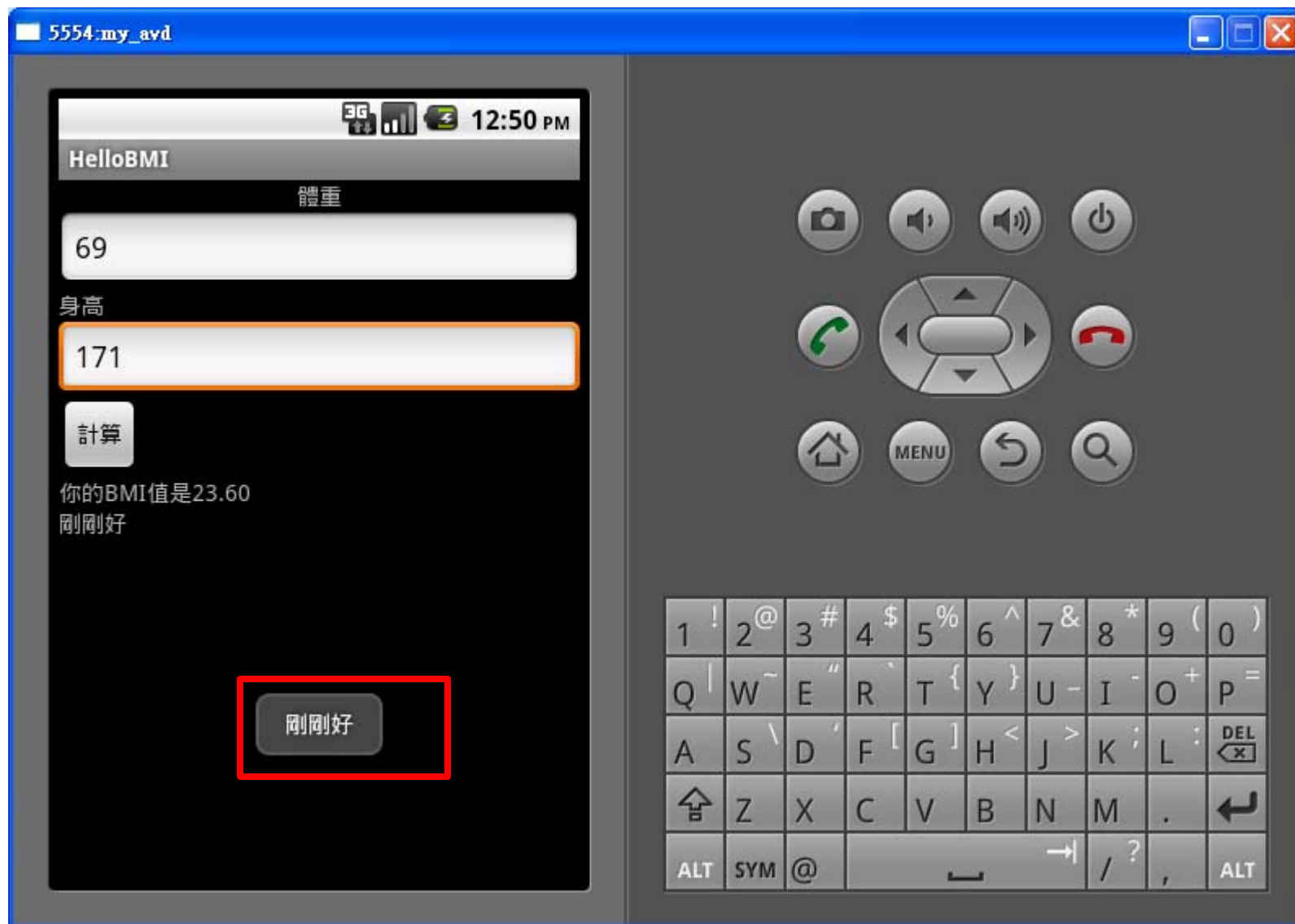
# Toast訊息框

- import android.widget.Toast;
- Toast.makeText(專案.this, 訊息, 顯示時間).show();

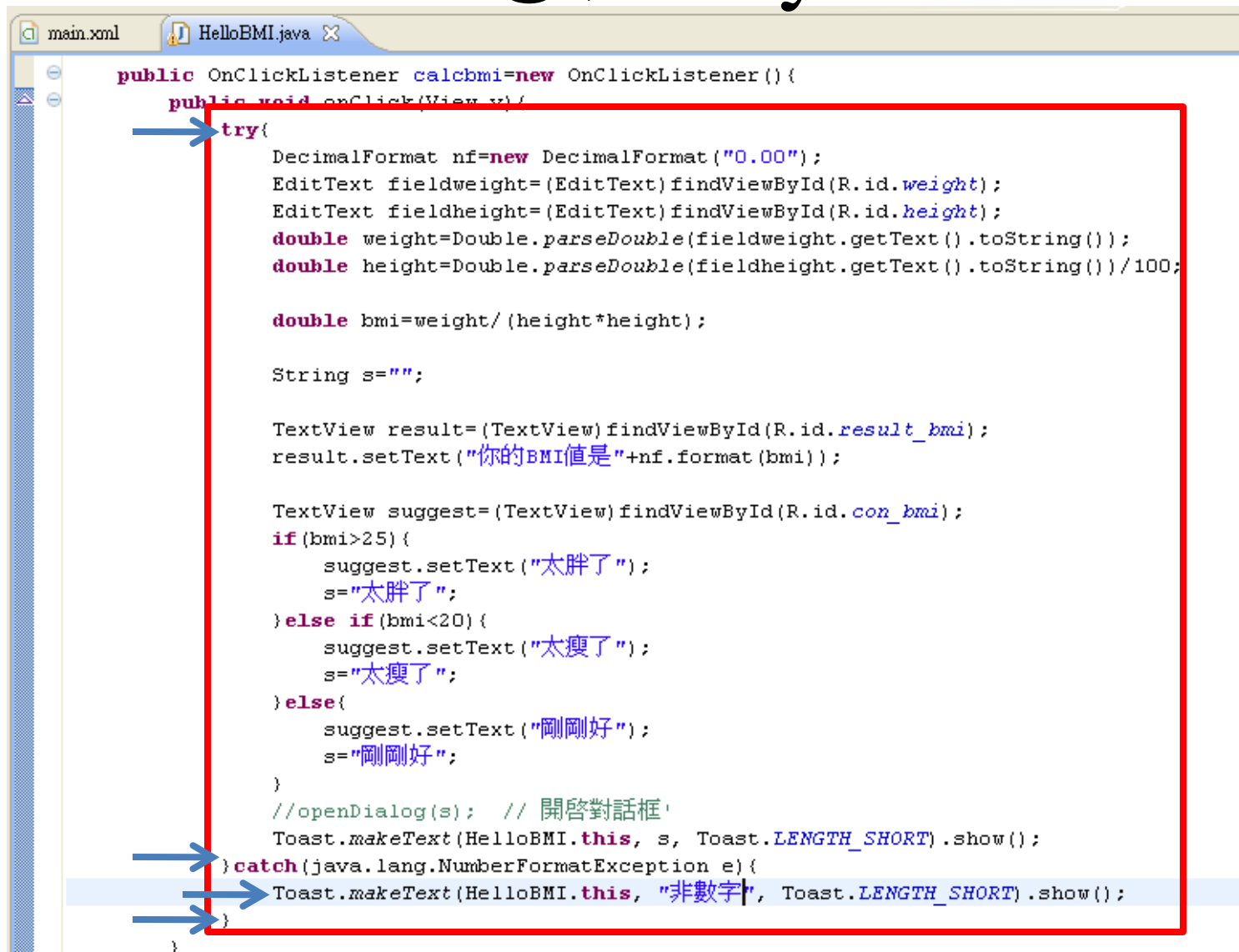
```
TextView suggest=(TextView)findViewById(R.id.con_bmi);
if(bmi>25){
    suggest.setText("太胖了");
    s="太胖了";
} else if(bmi<20){
    suggest.setText("太瘦了");
    s="太瘦了";
} else{
    suggest.setText("剛剛好");
    s="剛剛好";
}
//openDialog(s); // 開啟對話框
→ Toast.makeText(HelloBMI.this, s, Toast.LENGTH_SHORT).show();
};

private void openDialog(String s){
    new AlertDialog.Builder(HelloBMI.this)
        .setTitle("HelloBMI對話框範例")
        .setMessage(s)
        .setPositiveButton("確認",new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                // TODO Auto-generated method stub
            }
        })
}
```

# Toast訊息框



# Toast訊息框+try catch



```
main.xml HelloBMI.java X
public OnClickListener calcbmi=new OnClickListener(){
    public void onClick(View v){
        try{
            DecimalFormat nf=new DecimalFormat("0.00");
            EditText fieldweight=(EditText)findViewById(R.id.weight);
            EditText fieldheight=(EditText)findViewById(R.id.height);
            double weight=Double.parseDouble(fieldweight.getText().toString());
            double height=Double.parseDouble(fieldheight.getText().toString())/100;

            double bmi=weight/(height*height);

            String s="";

            TextView result=(TextView)findViewById(R.id.result_bmi);
            result.setText("你的BMI值是"+nf.format(bmi));

            TextView suggest=(TextView)findViewById(R.id.con_bmi);
            if(bmi>25){
                suggest.setText("太胖了");
                s="太胖了";
            }else if(bmi<20){
                suggest.setText("太瘦了");
                s="太瘦了";
            }else{
                suggest.setText("剛剛好");
                s="剛剛好";
            }
            //openDialog(s); // 開啟對話框
            Toast.makeText(HelloBMI.this, s, Toast.LENGTH_SHORT).show();
        }catch(java.lang.NumberFormatException e){
            Toast.makeText(HelloBMI.this, "非數字", Toast.LENGTH_SHORT).show();
        }
    }
}
```

# 練習

- 加入使用者輸入0的錯誤訊息顯示
- 使用 Try catch + Toast 顯示錯誤訊息

# 練習解答

```
try{
    DecimalFormat nf=new DecimalFormat("0.00");
    EditText fieldweight=(EditText)findViewById(R.id.weight);
    EditText fieldheight=(EditText)findViewById(R.id.height);
    double weight=Double.parseDouble(fieldweight.getText().toString());
    double height=Double.parseDouble(fieldheight.getText().toString())/100;
    if(weight!=0 && height!=0){
        double bmi=weight/(height*height);

        String s="";

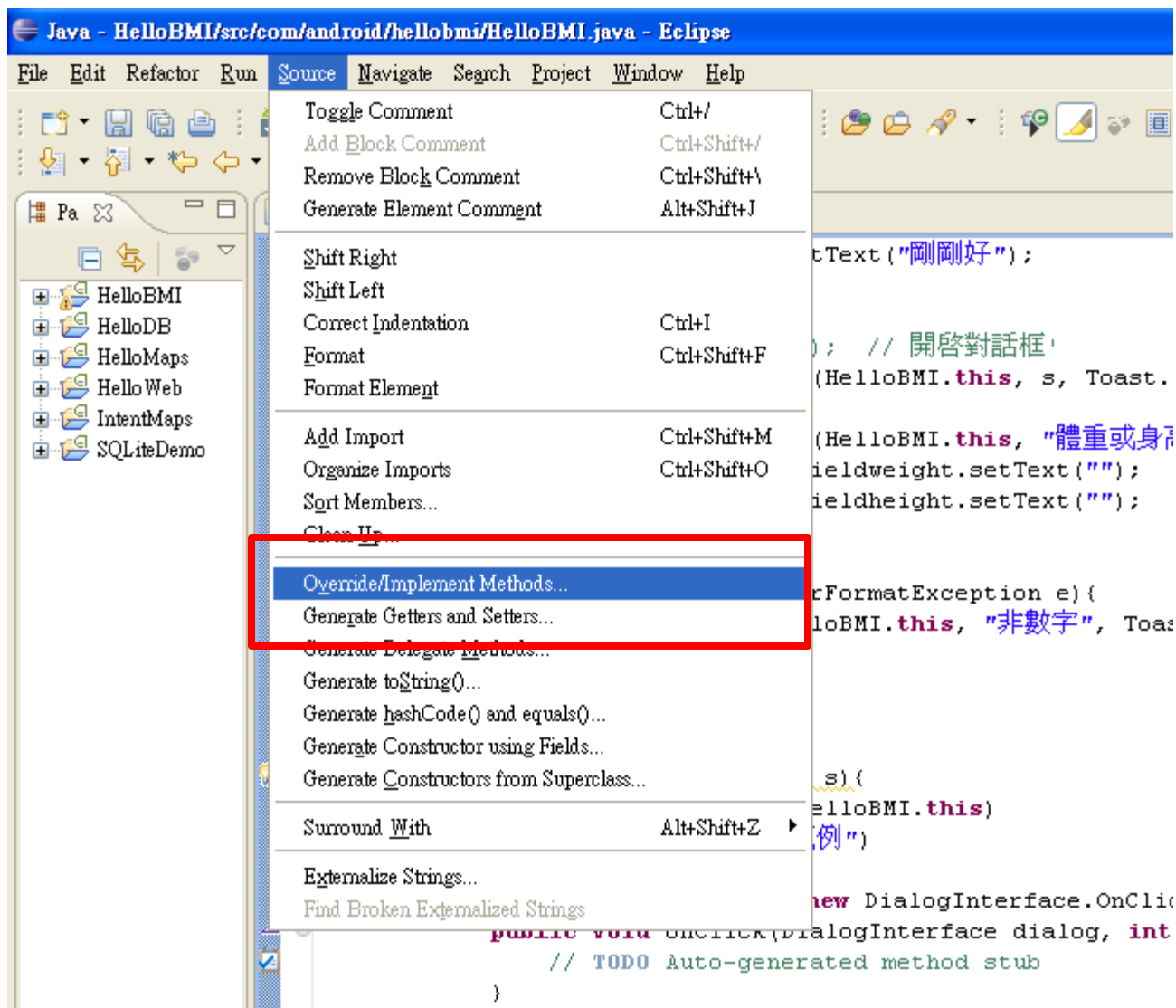
        TextView result=(TextView)findViewById(R.id.result_bmi);
        result.setText("你的BMI值是"+nf.format(bmi));

        TextView suggest=(TextView)findViewById(R.id.con_bmi);
        if(bmi>25){
            suggest.setText("太胖了");
            s="太胖了";
        }else if(bmi<20){
            suggest.setText("太瘦了");
            s="太瘦了";
        }else{
            suggest.setText("剛剛好");
            s="剛剛好";
        }
        //openDialog(s); // 開啟對話框
        Toast.makeText(HelloBMI.this, s, Toast.LENGTH_SHORT).show();
    }else{
        Toast.makeText(HelloBMI.this, "體重或身高不可以等於0", Toast.LENGTH_SHORT).show();
        if(weight==0) fieldweight.setText("");
        if(height==0) fieldheight.setText("");
    }
}
```

# 選單

- 使用手機硬體上的MENU鍵進行呼叫顯示
- 建立選單
  - onCreateOptionsMenu
- 處理選項動作
  - onOptionsItemSelected

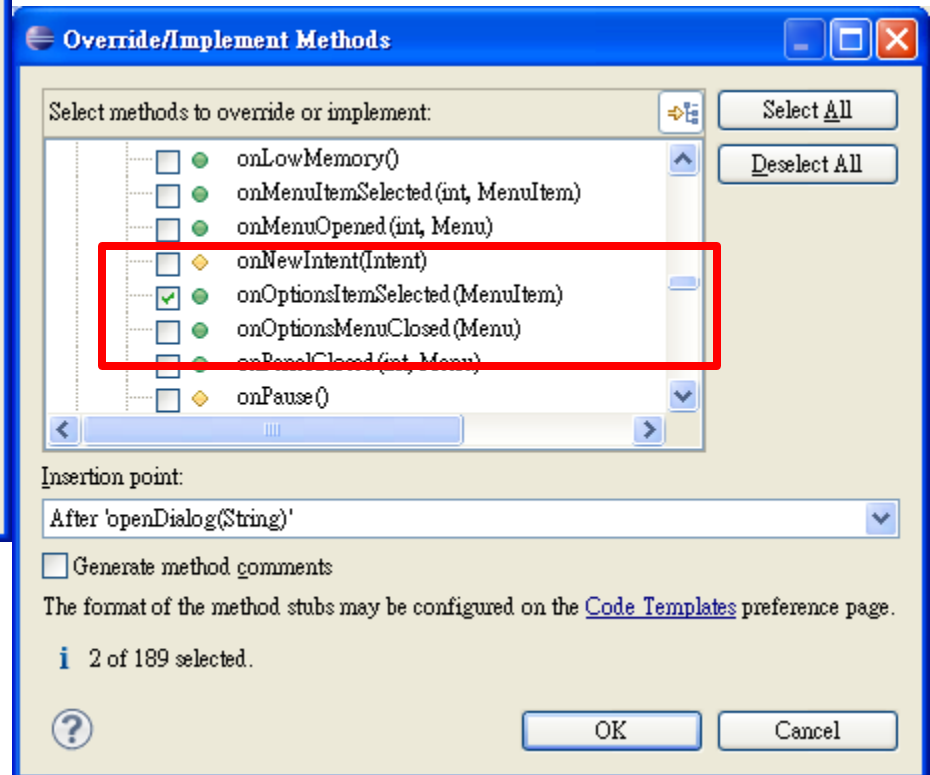
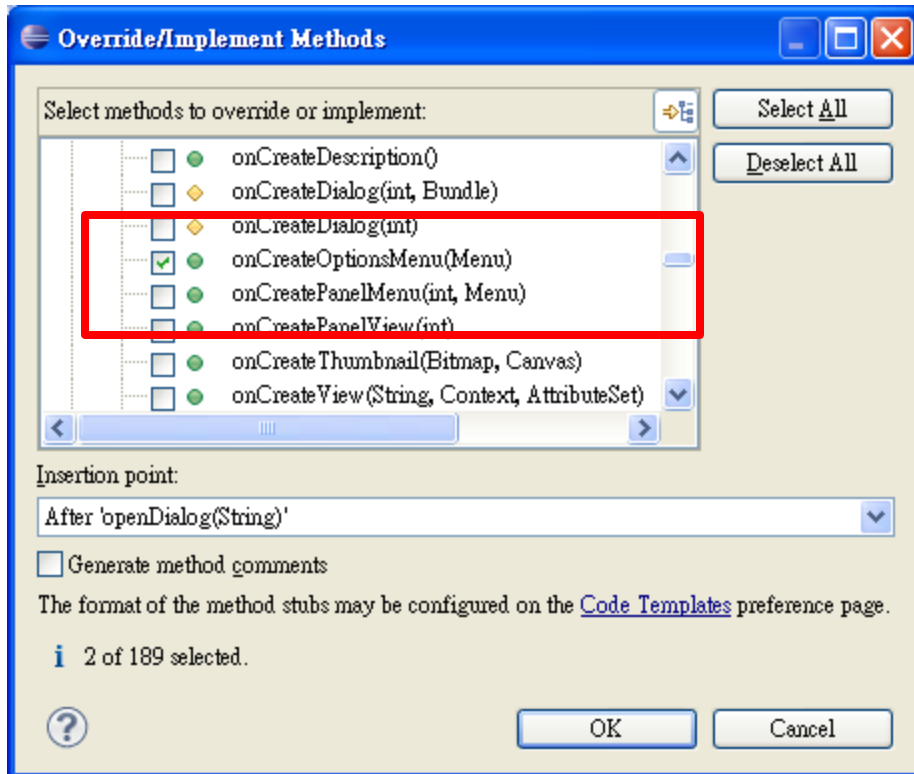
# 選單



# 選單

onOptionsItemSelected

onCreateOptionsMenu





# 選單基本結構

- 建立選單
  - onCreateOptionsMenu

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // TODO Auto-generated method stub
    return super.onCreateOptionsMenu(menu);
}
```

- 處理選項動作
  - onOptionsItemSelected

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // TODO Auto-generated method stub
    return super.onOptionsItemSelected(item);
}
```

# 加入選單選項

- 加入選項(群組id,i項目id,順序,名稱)
  - menu.add(0, 識別符號, 0, 顯示文字)

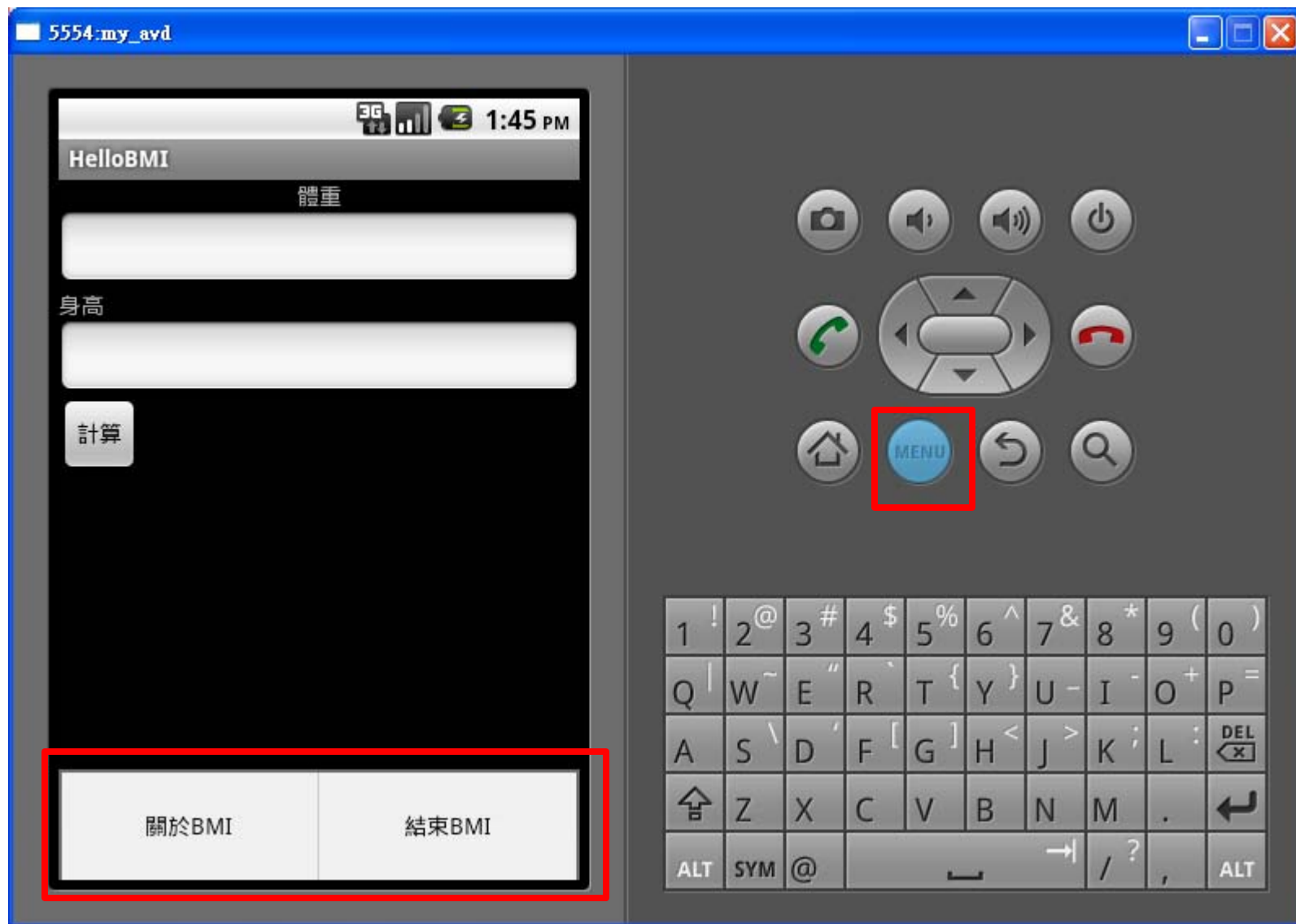
```
protected static final int MENU_ABOUT=Menu.FIRST;  
protected static final int MENU_QUIT=Menu.FIRST+1;
```

```
@Override
```

```
public boolean onCreateOptionsMenu(Menu menu) {  
    // TODO Auto-generated method stub  
    menu.add(0, MENU_ABOUT, 0, "關於BMI");  
    menu.add(0, MENU_QUIT, 0, "結束BMI");  
    return super.onCreateOptionsMenu(menu);  
}
```

```
menu.add(0, MENU_ABOUT, 0, "關於BMI").setIcon(android.R.drawable.ic_menu_help);  
menu.add(0, MENU_QUIT, 0, "結束  
BMI").setIcon(android.R.drawable.ic_menu_close_clear_cancel);
```

# 選單



# 處理選項動作

- 處理選項動作
  - onOptionsItemSelected

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // TODO Auto-generated method stub
    switch(item.getItemId()){
        case MENU_ABOUT:
            openDialog("這是選單範例"); // 開啓對話框
            break;
        case MENU_QUIT:
            finish(); //結束程式
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

```
protected static final int MENU_ABOUT=Menu.FIRST;  
protected static final int MENU_QUIT=Menu.FIRST+1;  
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    // TODO Auto-generated method stub  
    menu.add(0, MENU_ABOUT, 0, "關於BMI").setIcon(android.R.drawable.ic_menu_help);  
    menu.add(0, MENU_QUIT, 0, "結束BMI").setIcon(android.R.drawable.ic_menu_close_clear_cancel);  
    return super.onCreateOptionsMenu(menu);  
}
```

```
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    // TODO Auto-generated method stub  
    switch(item.getItemId()){  
        case MENU_ABOUT:  
  
        break;  
        case MENU_QUIT:  
  
        break;  
    }  
    return super.onOptionsItemSelected(item);  
}
```

# 選單



# 練習


- 在”結束”選項中加入詢問對話框
- 對話框中有”確定”與”取消”按鈕
- 按下”確定”就結束程式
- 按下”取消”就取消結束程式的動作

# 選單





# 練習解答



```
private void closeDialog(String s) {  
    new AlertDialog.Builder (HelloBMI.this)  
        .setTitle("HelloBMI範例結束確認")  
        .setMessage(s)  
        .setPositiveButton("確認", new DialogInterface.OnClickListener() {  
            public void onClick(DialogInterface dialog, int which) {  
                // TODO Auto-generated method stub  
                finish(); //結束程式  
            }  
        })  
        .setNegativeButton("取消", new DialogInterface.OnClickListener() {  
            public void onClick(DialogInterface dialog, int which) {  
                // TODO Auto-generated method stub  
            }  
        })  
        .show();  
}
```

```
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    // TODO Auto-generated method stub  
    switch (item.getItemId()) {  
        case MENU_ABOUT:  
            openDialog("這是選單範例"); // 開啟對話框  
            break;  
        case MENU_QUIT:  
            closeDialog("結束程式?");  
            break;  
    }  
    return super.onOptionsItemSelected(item);  
}
```

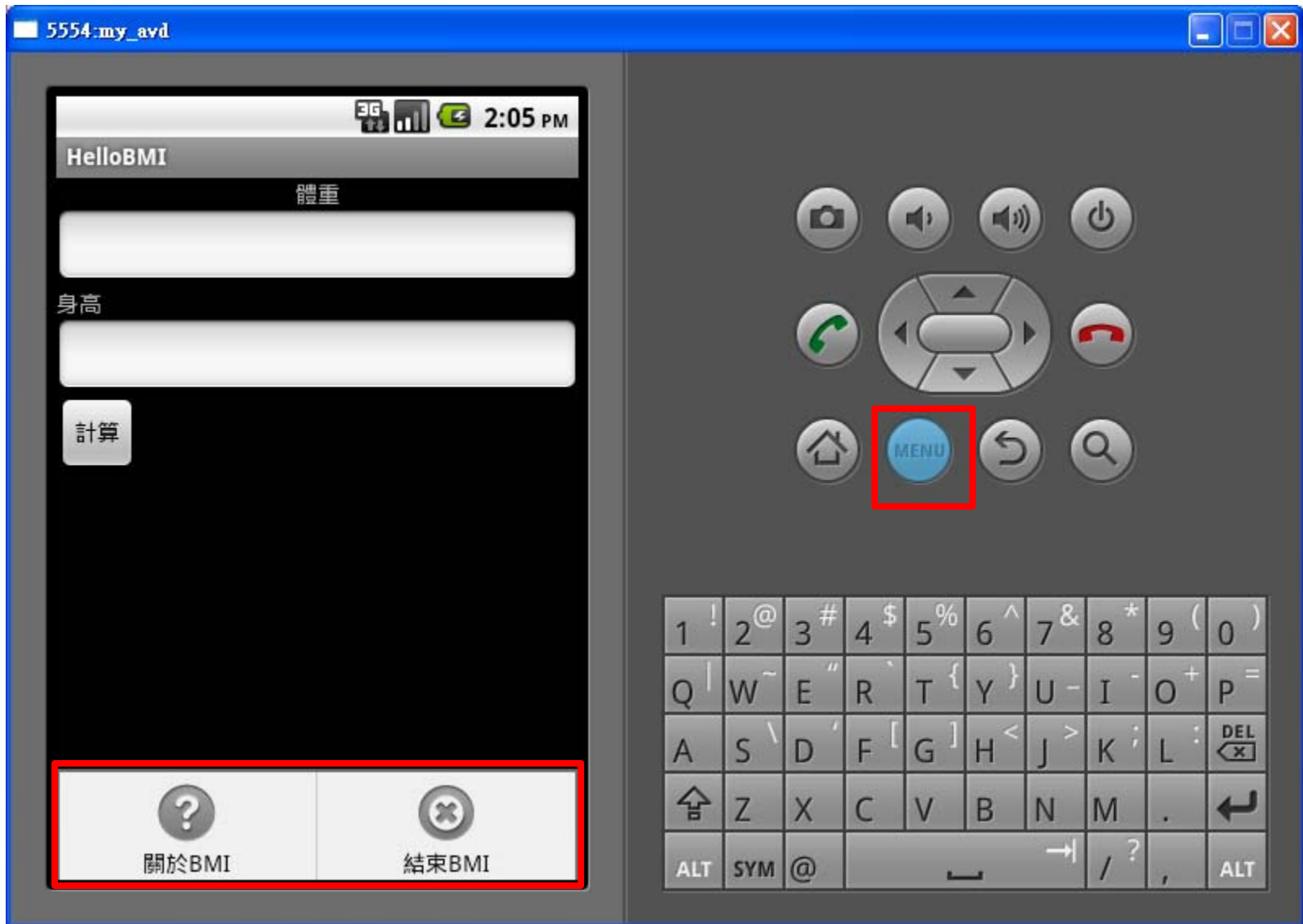
# 選單選項加入圖示

- 加入選項

- menu.add(0, 識別符號, 0, 顯示文字).setIcon();

```
protected static final int MENU_ABOUT=Menu.FIRST;
protected static final int MENU_QUIT=Menu.FIRST+1;
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // TODO Auto-generated method stub
    menu.add(0, MENU_ABOUT, 0, "關於BMI").setIcon(android.R.drawable.ic_menu_help);
    menu.add(0, MENU_QUIT, 0, "結束BMI").setIcon(android.R.drawable.ic_menu_close_clear_cancel);
    return super.onCreateOptionsMenu(menu);
}
```

# 選單選項加入圖示



# 控制螢幕方向

- 預設是可以隨著手機方向旋轉而旋轉
- 可以在AndroidManifest.xml 中設定不旋轉  
(很多Game APP就是這樣做)

— `<activity android:name="Main"  
    android:screenOrientation="portrait"></activity>`

landscape：橫向

portrait：直向

# 設計隨螢幕旋轉與大小無關的UI

- 多使用 DashboardLayout
- UI裡面也有include功能

請參考

<http://www.androidhive.info/2011/12/android-dashboard-design-tutorial/>

# 動態Layout

# result



# answer

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
  
    LinearLayout layMain = new LinearLayout(this); //建立LinearLayout  
    LinearLayout.LayoutParams forlayMain = new LinearLayout.LayoutParams(LayoutParams.FILL_PARENT, LayoutPa  
    layMain.setOrientation(LinearLayout.VERTICAL); //垂直排列  
    layMain.setLayoutParams(forlayMain);  
    layMain.setBackgroundResource(R.drawable.ic_launcher); //LinearLayout背景  
    //建立TextView物件  
    TextView tv1 = new TextView(this);  
    tv1.setText(getResources().getString(R.string.app_name));  
    tv1.setTextColor(Color.WHITE);  
    tv1.setTextSize(30);  
    tv1.setPadding(50, 30, 20, 0);  
    layMain.addView(tv1); //將TextView加入layMain  
  
    // 再建立一個LinearLayout  
    LinearLayout lin1= new LinearLayout(this);  
    LinearLayout.LayoutParams forlin1 = new LinearLayout.LayoutParams(LayoutParams.FILL_PARENT, LayoutPa  
    lin1.setOrientation(LinearLayout.HORIZONTAL); //水平排列  
    lin1.setLayoutParams(forlin1);  
    layMain.addView(lin1);  
    //建立按鈕  
    Button bt1=new Button(this);  
    bt1.setText("按鈕1");  
    lin1.addView(bt1); //將按鈕加入lin1  
    Button bt2=new Button(this);  
    bt2.setText("按鈕2");  
    lin1.addView(bt2);  
  
    setContentView(layMain); //設定最後的UI  
}
```