



Android Lab1

Lab 1 Overview

1. Setup android develop environment(Android Studio)
2. Setup android phone emulator(Check Point-1)
3. Layout design(Check Point-2)
4. Use the google map API(Check Point-3)

Notice

- If you see **CHECK POINT** text, please call the TA to check your progress!

在Windows安裝Android開發環境

Android Studio

- Android Studio is the official IDE for Android application development, based on IntelliJ IDEA. On top of the capabilities you expect from IntelliJ.



在Windows安裝Android開發環境

- 下載Java Development Kit (JDK)
 - <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- ① 先點選step1後依照作業系統平台選取適合檔案下載，下載後安裝於電腦上。

Java SE Development Kit 8u161

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

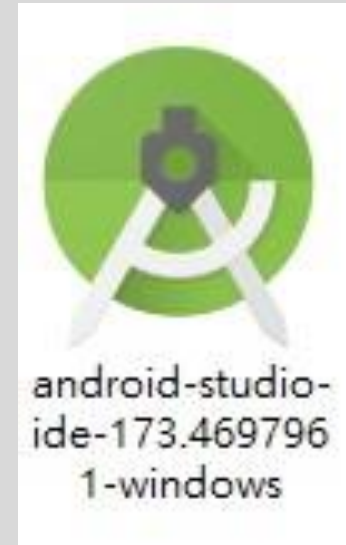
☒ Accept License Agreement ☐ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.92 MB	jdk-8u161-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.88 MB	jdk-8u161-linux-arm64-vfp-hflt.tar.gz
Linux x86	168.96 MB	jdk-8u161-linux-i586.rpm
Linux x86	183.76 MB	jdk-8u161-linux-i586.tar.gz
Linux x64	166.09 MB	jdk-8u161-linux-x64.rpm
Linux x64	180.97 MB	jdk-8u161-linux-x64.tar.gz
macOS	247.12 MB	jdk-8u161-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.99 MB	jdk-8u161-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.29 MB	jdk-8u161-solaris-sparcv9.tar.gz
Solaris x64	140.57 MB	jdk-8u161-solaris-x64.tar.Z
Solaris x64	97.02 MB	jdk-8u161-solaris-x64.tar.gz
Windows x86	198.54 MB	jdk-8u161-windows-i586.exe
Windows x64	206.51 MB	jdk-8u161-windows-x64.exe

選擇OS

在Windows安裝Android開發環境

- Download the latest Android Studio:
<https://developer.android.com/studio/index.html>
- Open and install **android-studio-bundle**
- Next , next , next
- Android studio will automatically download the latest version of Android SDK



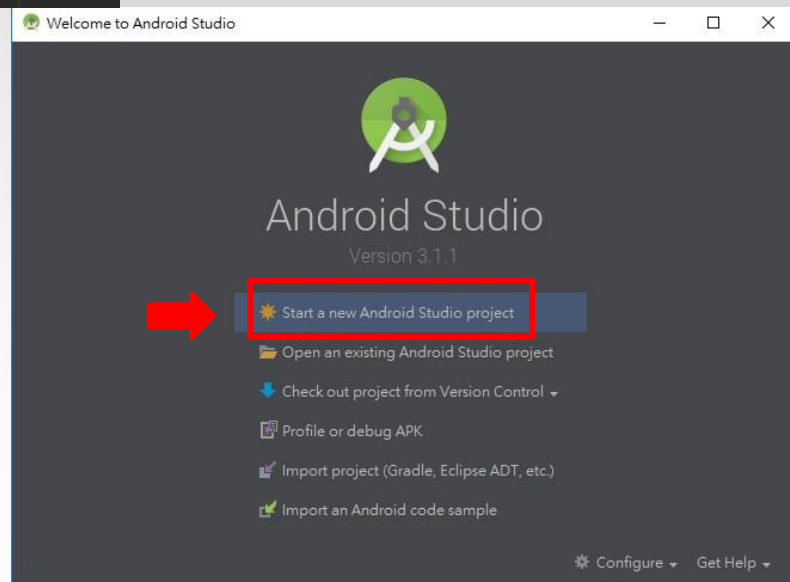
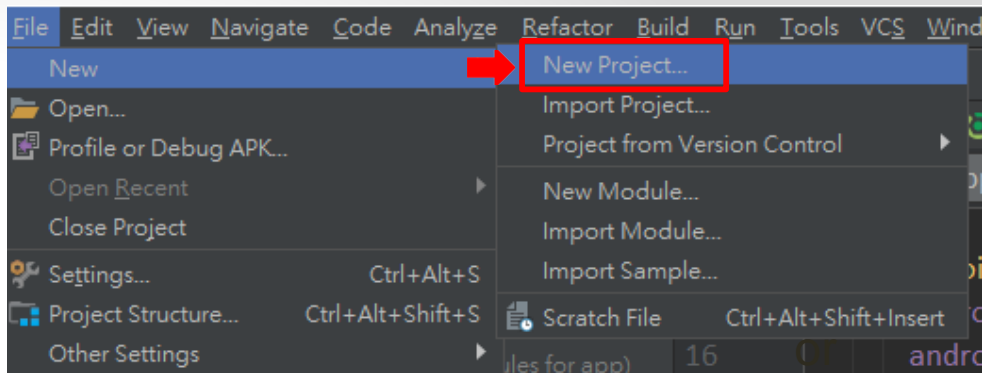
Android SDK

- A software development kit that enables developers to create applications for the Android platform.
- Ex:Google API

開啟新的Android專案

Android的新專案

- 如何新增一個專案？ 選擇 “File” → “New” → “New project”



Choose your project

Phone and Tablet

Wear OS

TV

Android Auto

Android Things

Add No Activity



Basic Activity



Empty Activity



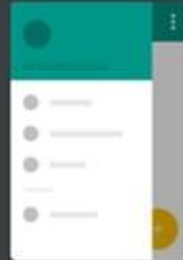
Bottom Navigation Activity



Fullscreen Activity



Master/Detail Flow



Navigation Drawer Activity



Google Maps Activity

Empty Activity

Creates a new empty activity

Previous

Next


Cancel

Finish

Create New Project

×

Configure your project



Empty Activity

Creates a new empty activity

Name

Package name

Save location

Language

Java

Minimum API level

API 21: Android 5.0 (Lollipop)

! Your app will run on approximately 85.0% of devices.

[Help me choose](#)

☐ This project will support instant apps

☐ Use androidx.* artifacts

! The application name for most apps begins with an uppercase letter

Previous

Next

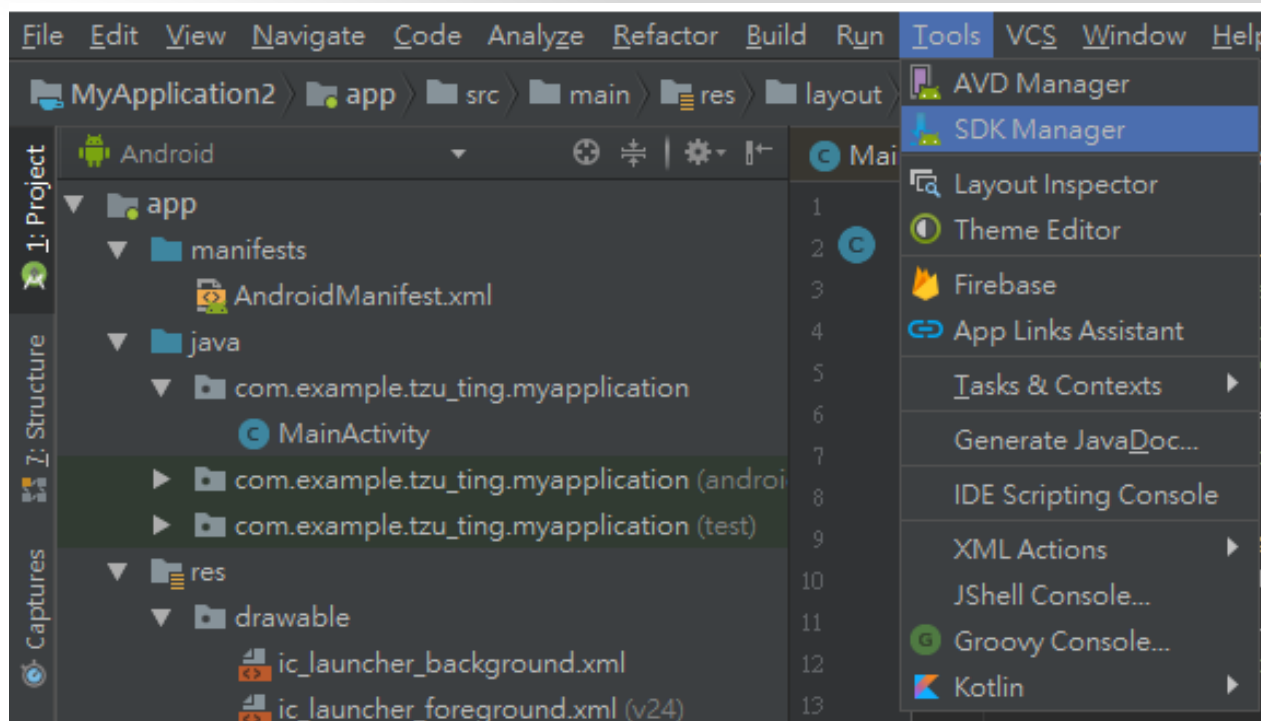
Cancel

Finish

版本越低，app兼容性越高，
盡量選21以下的版本

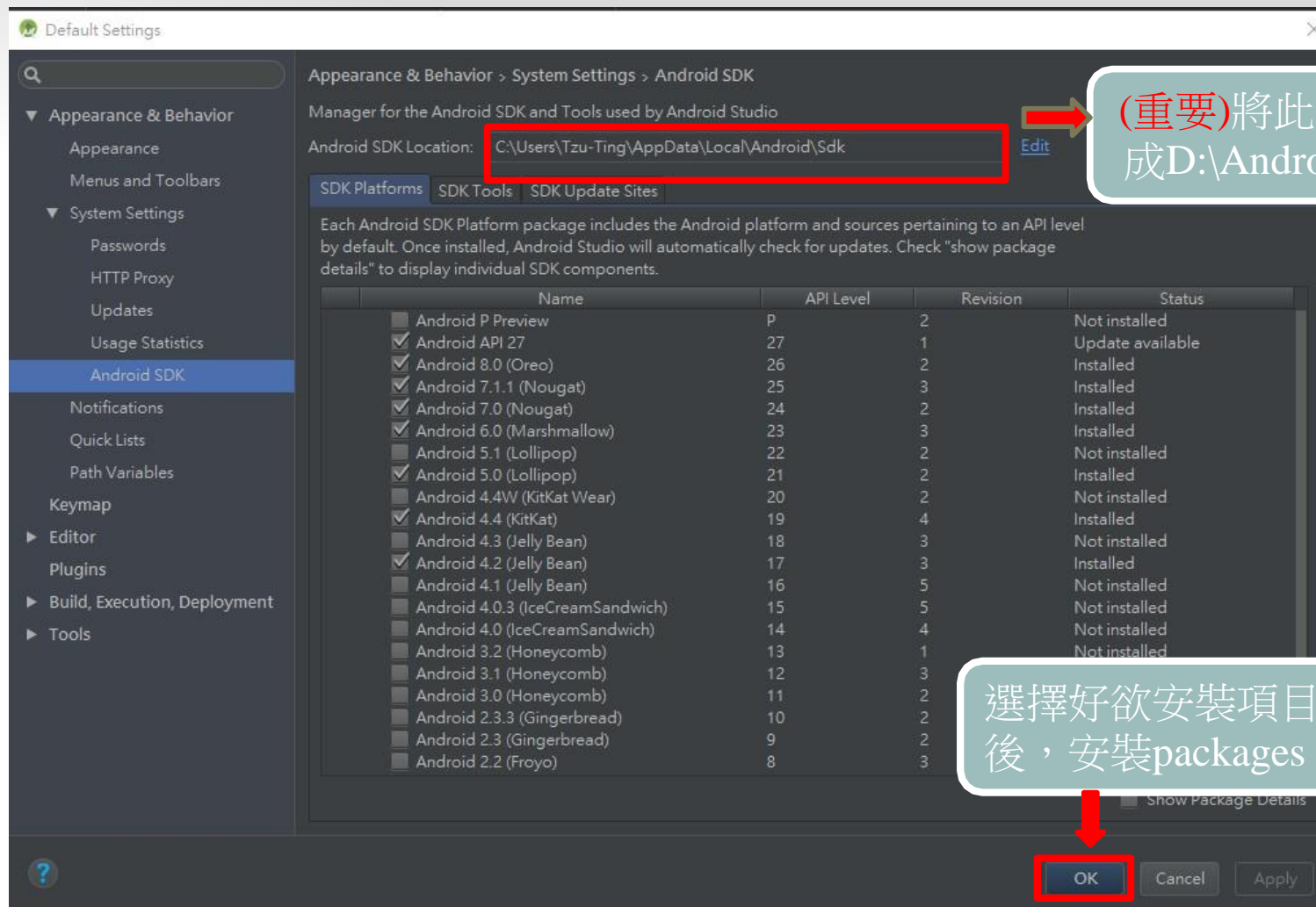
check and install android SDK

- Tools -> Android -> SDK Manager

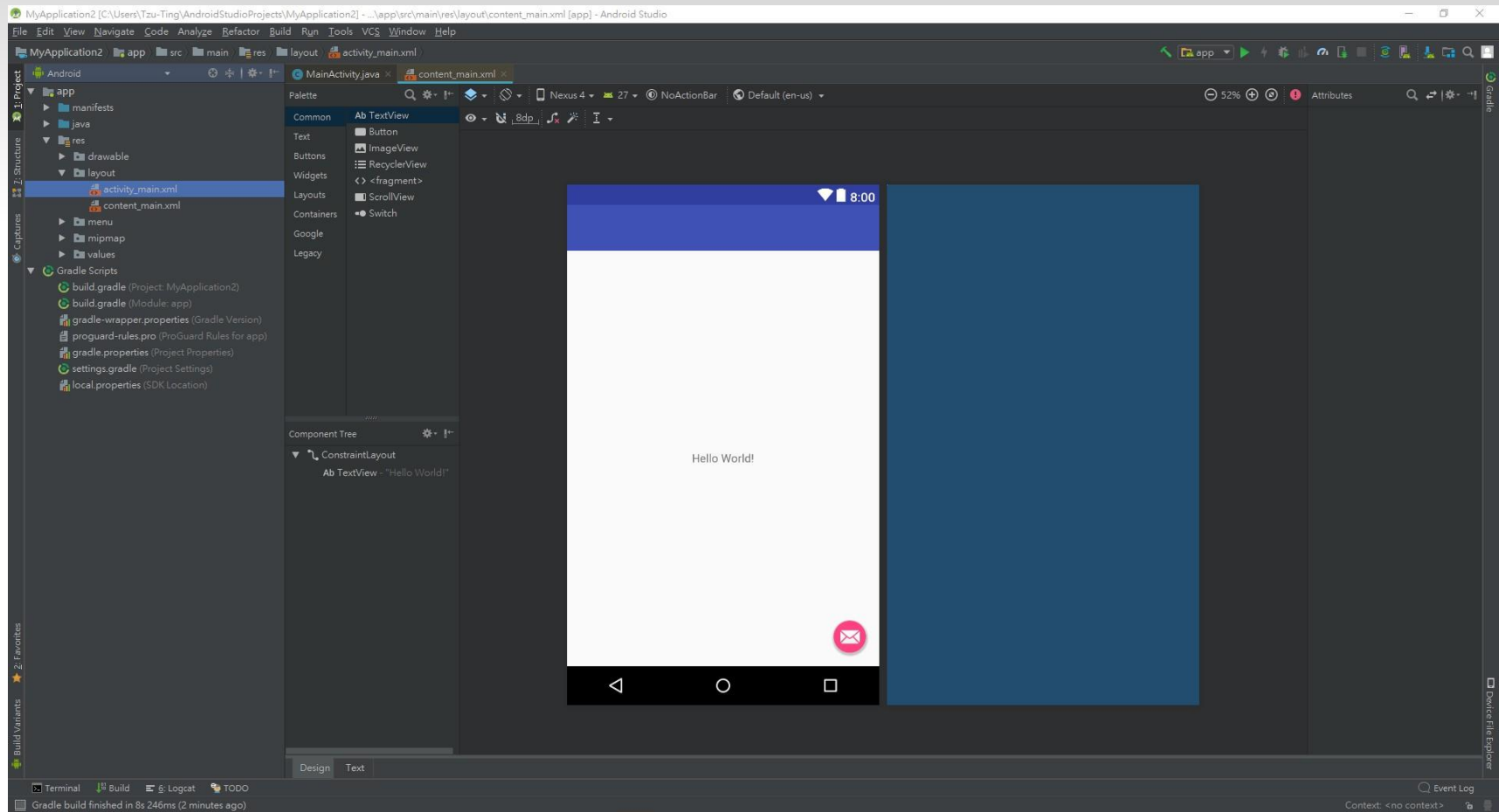


在Windows安裝Android開發環境

- 依自己想開發版本安裝android SDK並勾選下方選項即可。



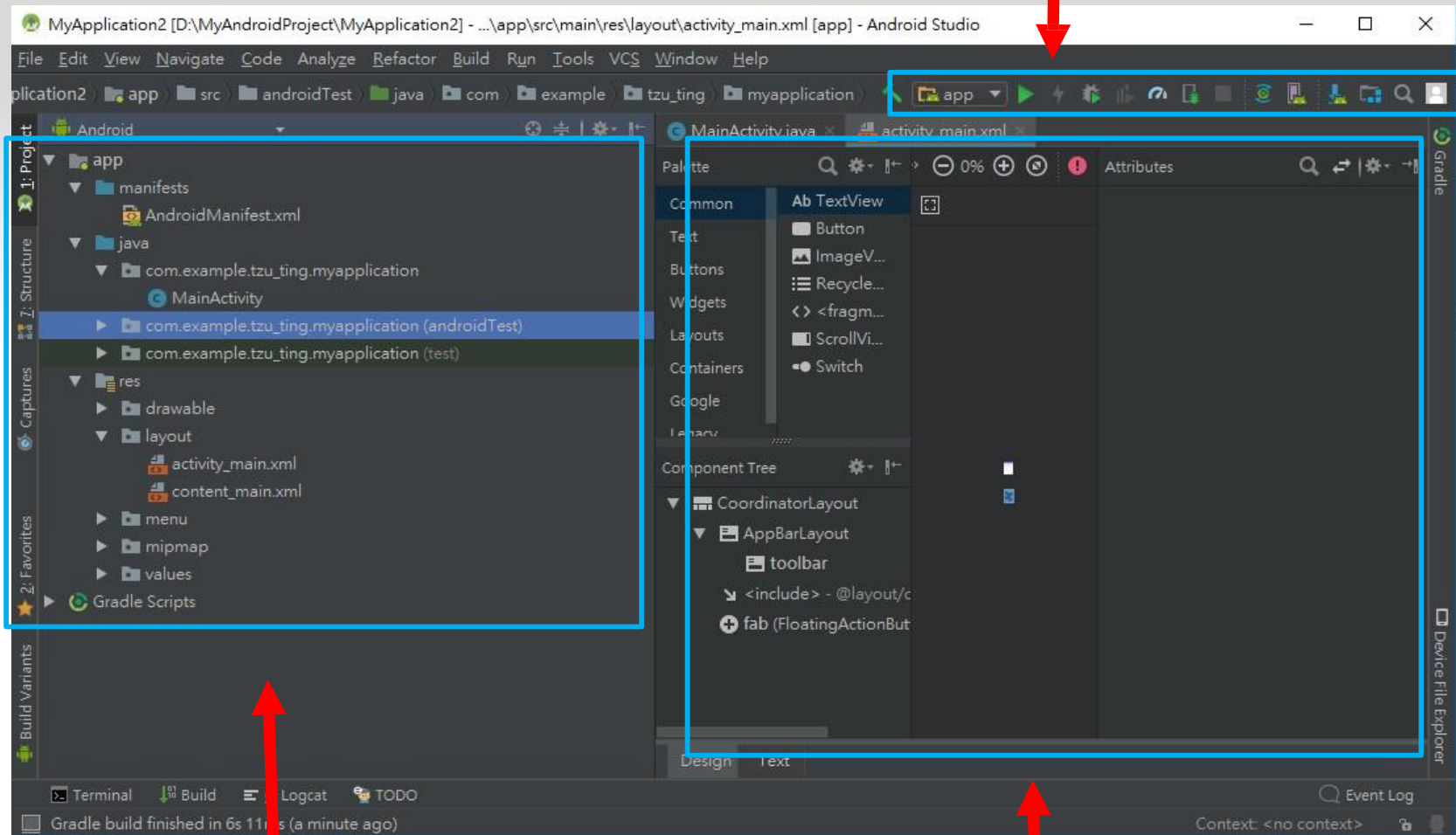
Android的新專案



如果要複製、移動或刪除專案，直接對project的資料夾動作即可。

- 記得要先close project!!

快捷工具箱



專案瀏覽面板
(可看出架構)

編輯視窗
(寫code或拉物件的地方)

新增Android Virtual Device (模擬器)

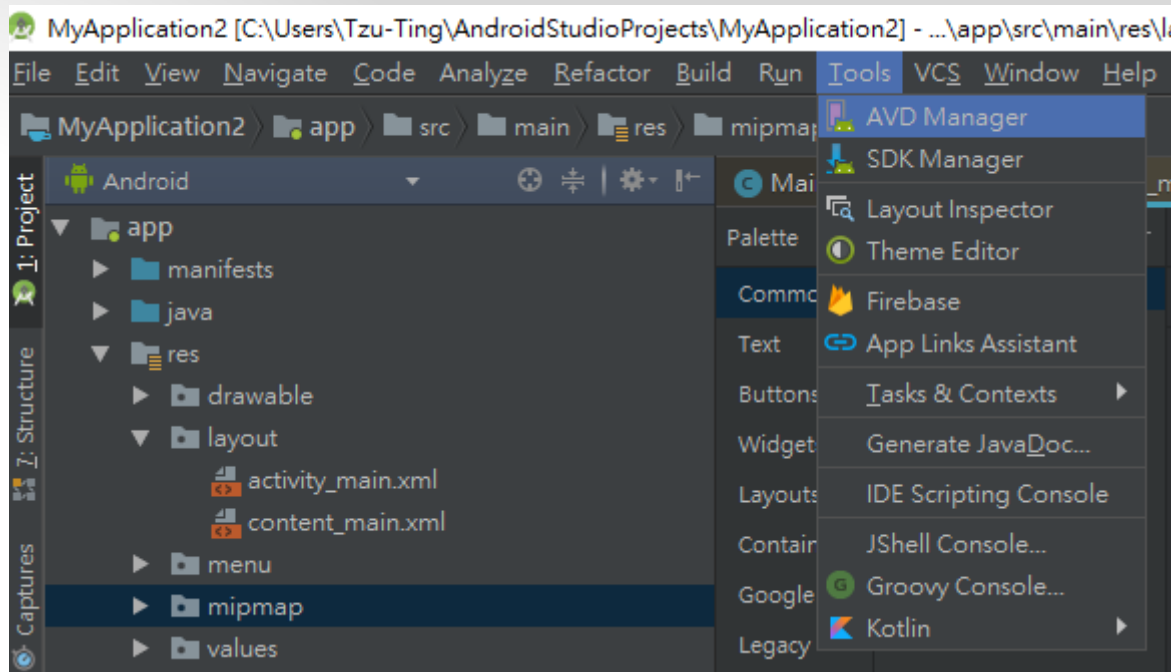
Android AVD

The Android SDK includes a mobile device emulator — a virtual mobile device that runs on your computer. The emulator lets you develop and test Android applications without using a physical device.

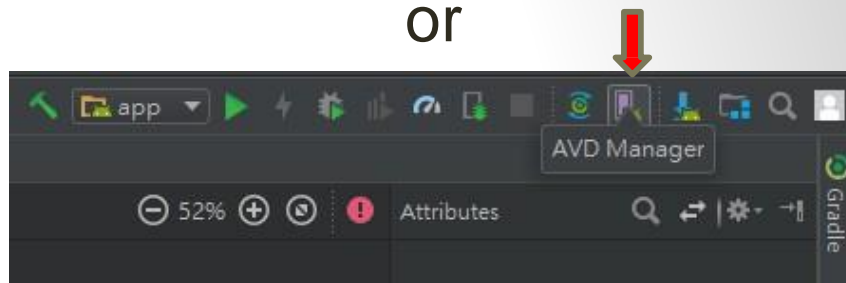


新增Android Virtual Device(AVD)

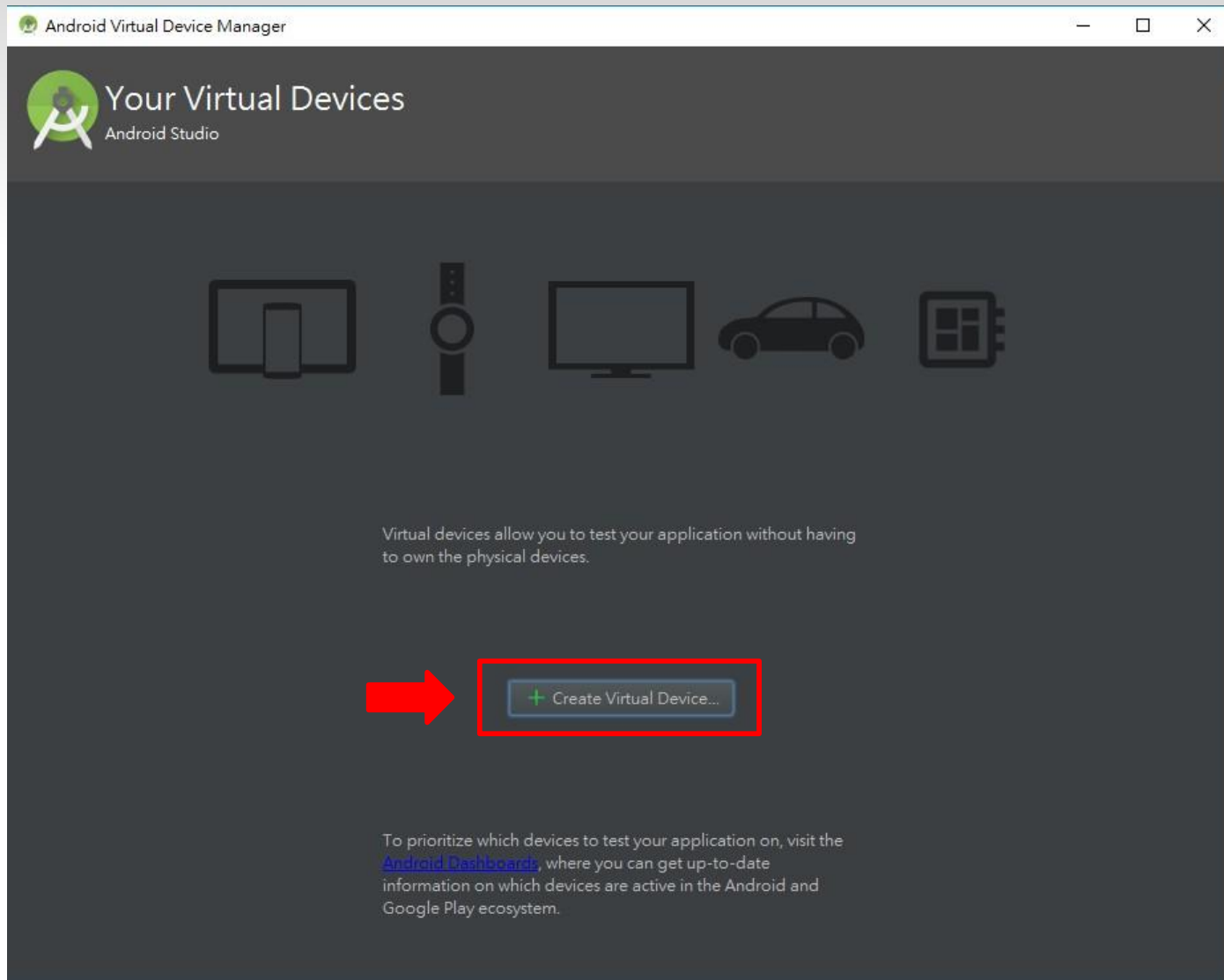
- 在AVD管理器裡開啟模擬器



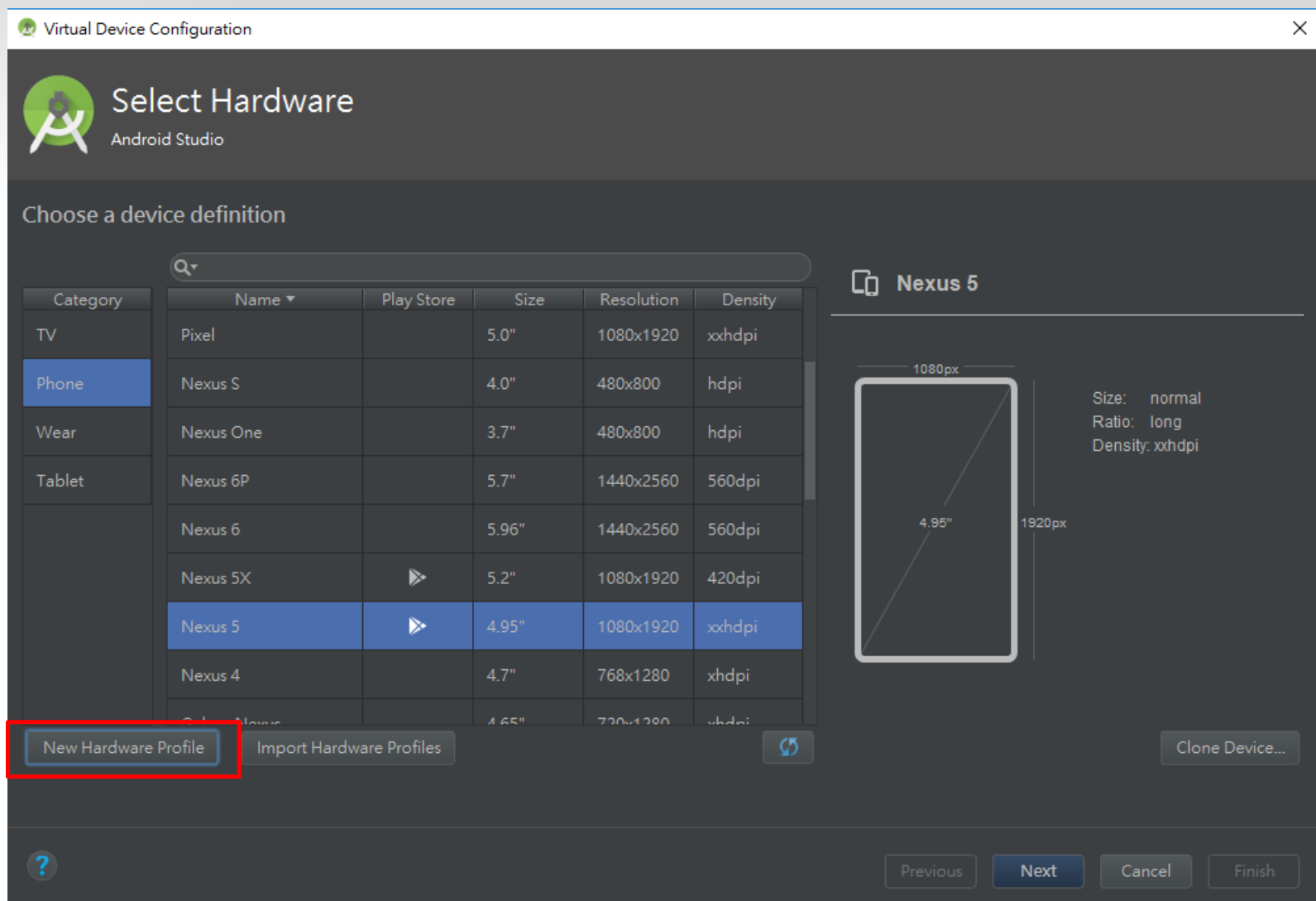
or



新增Android模擬器



新增Android模擬器



新增Android模擬器

Hardware Profile Configuration

Configure Hardware Profile
Android Studio

Configure this hardware profile

iAndroid

Phone/Tablet

Screen size: 5.0 inch

Resolution: 1080 x 1920 px

☐ Round

RAM: 2048 MB

☐ Has Hardware Buttons (Back/Home/Menu)

☐ Has Hardware Keyboard

Navigation Style: None

☒ Portrait

☒ Landscape

Size: large
Ratio: long
Density: xxhdpi

1080px

5.0"

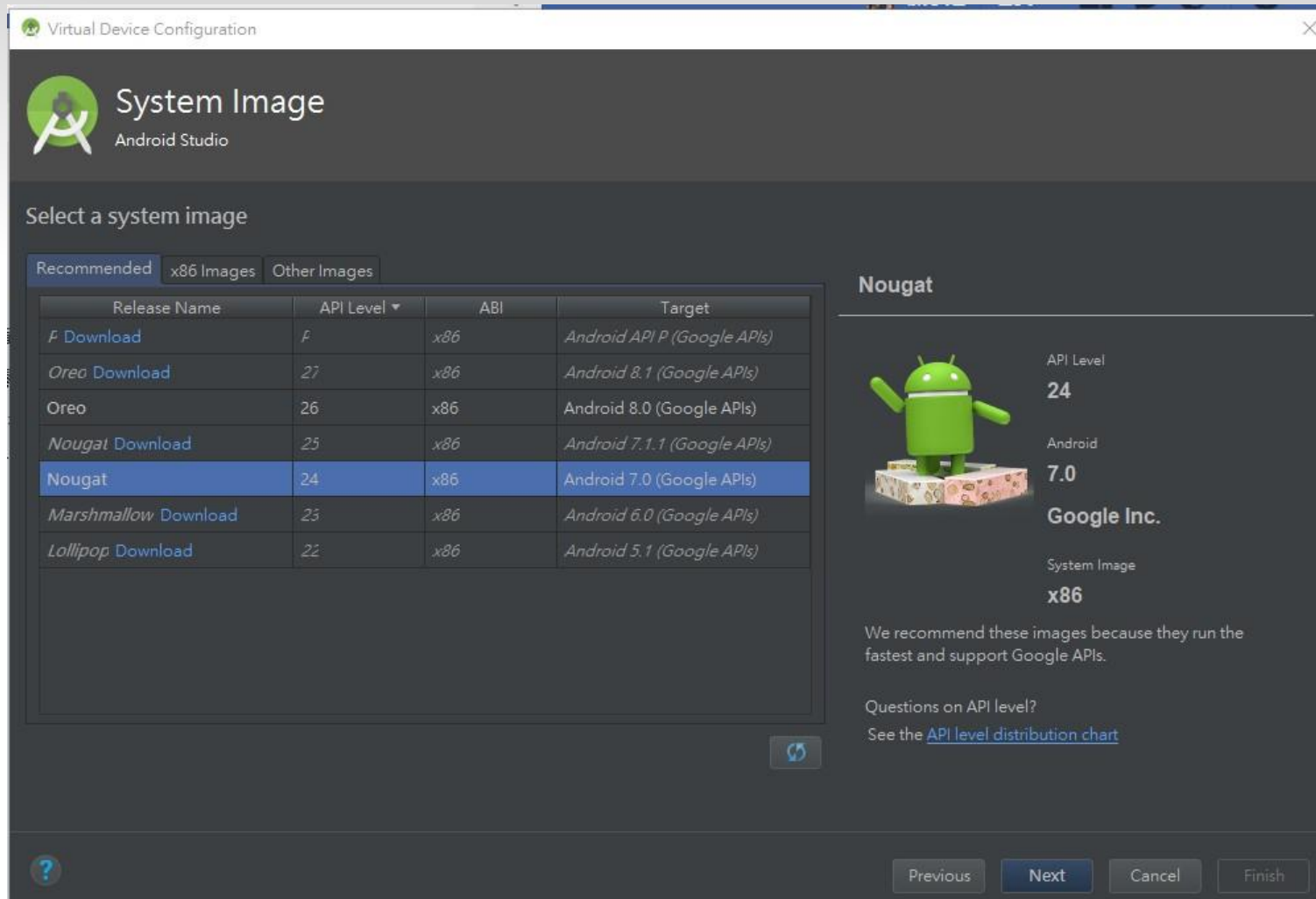
1920px

Name of the Device Profile

Previous Next Cancel Finish

填寫建立AVD
template 所需資訊


啟動Android模擬器



Choose the android version for your AVD template

建立AVD

Virtual Device Configuration





Android Virtual Device (AVD)


Android Studio

Verify Configuration


AVD Name

 iAndroid	5.0 1080x1920 xxhdpi	<button>Change...</button>
 Lollipop	Android 5.1 x86	<button>Change...</button>

Startup orientation



Portrait



Landscape

Emulated Performance


Graphics:

Device Frame ☐ Enable Device Frame

Show Advanced Settings

AVD Name

The name of this AVD.



Previous

Next

Cancel

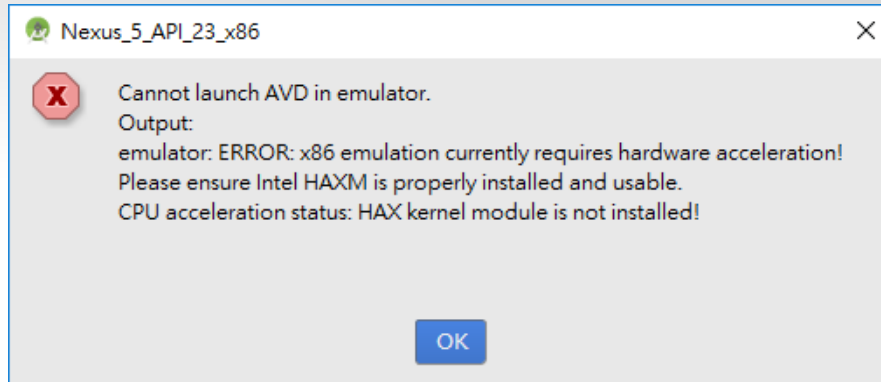
Finish

啟動Android模擬器

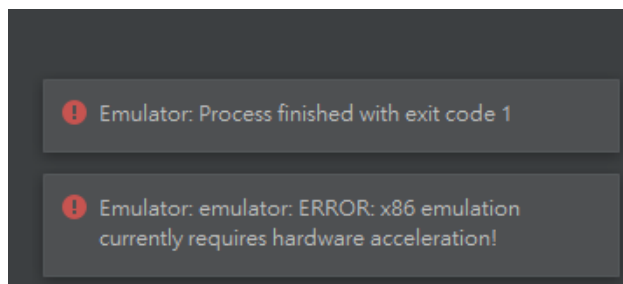
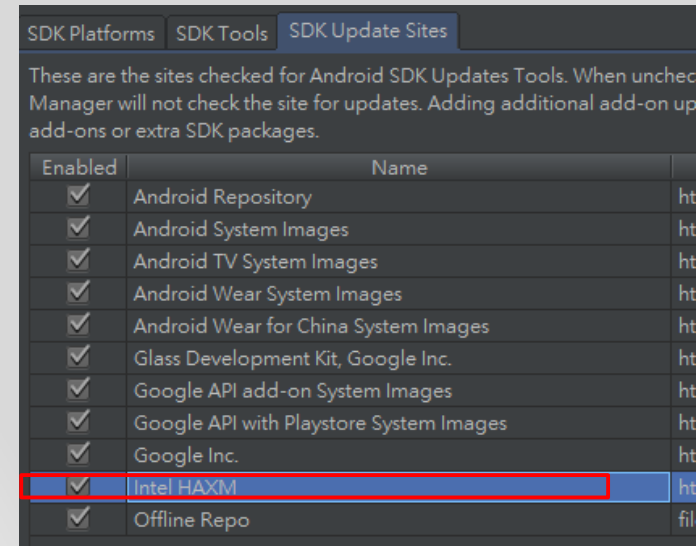


啟動Android模擬器

- 若無法啟動模擬器，並跳出以下訊息，以下為解決方法



- 1. Tools->SDK Manger->SDK Update Sites
- 2. 再次啟動模擬器



- 參照
- <https://www.itread01.com/p/91848.html>

Dalvik Debug Monitor Service

What is DDMS

Android提供一個全名為Dalvik Debug Monitor Server (DDMS)的工具，提供以下調試監控裝置的功能

Debug :

1. File explorer: support adb push/pull files
2. 列出目前在跑的progress
3. Screen capture(截圖)
4. logcat
5. device state dump
6. radio state dump

環境模擬 :

1. 撥電話到emulator
2. 發簡訊到emulator
3. 提供GPS位置到emulator

如何使用DDMS

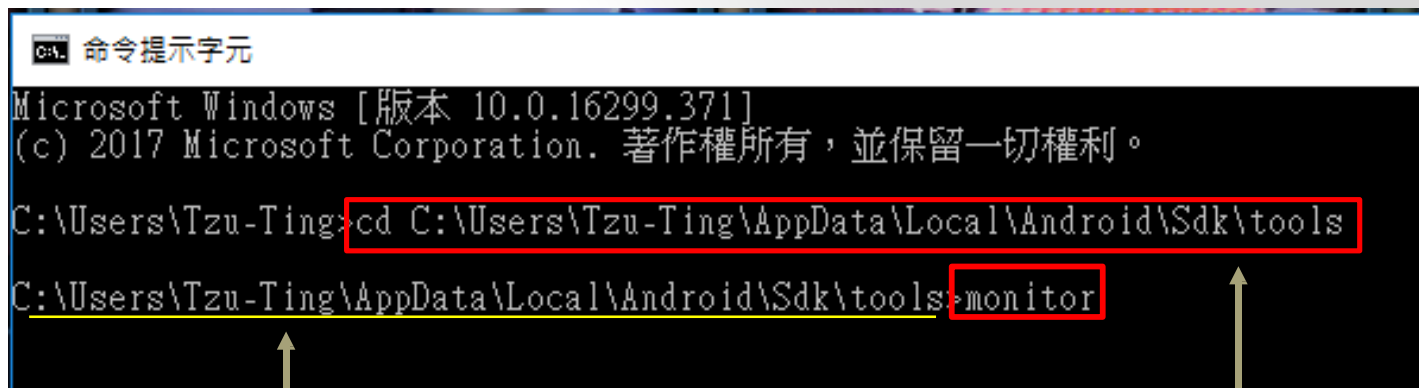
- 1.左下角開始=>輸入cmd=>開啟命令提示字元

- 2.進入剛剛安裝SDK元件的目錄:

將這行輸入並按Enter => `cd D:\Android\Sdk\tools`

(可到Android Studio的Tools/SDK Manger再次確認你安裝SDK的目錄)

- 進入目錄後輸入`monitor`並按Enter



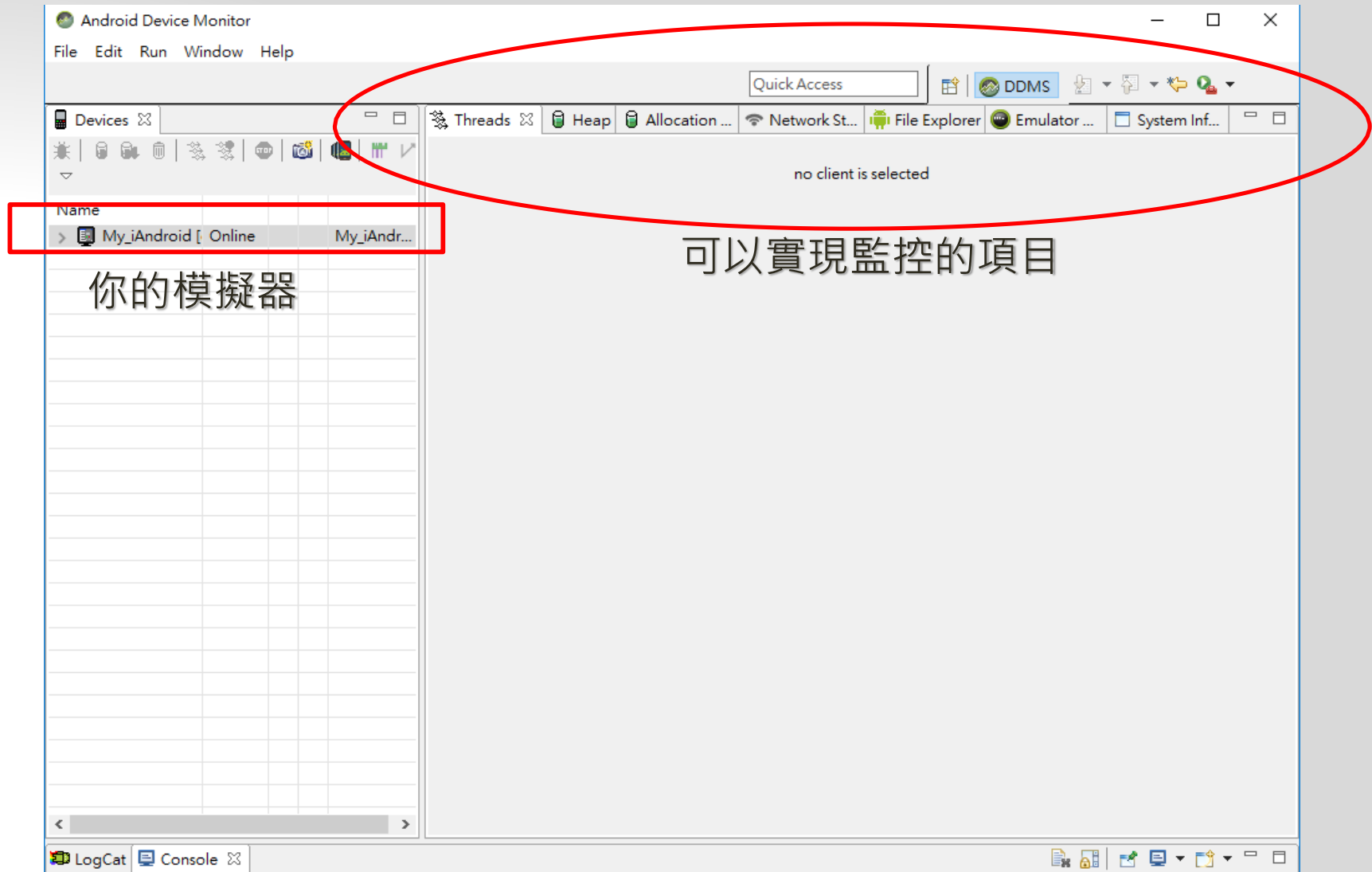
```
命令提示字元
Microsoft Windows [版本 10.0.16299.371]
(c) 2017 Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\Tzu-Ting>cd C:\Users\Tzu-Ting\AppData\Local\Android\Sdk\tools
C:\Users\Tzu-Ting\AppData\Local\Android\Sdk\tools>monitor
```

The screenshot shows a Windows Command Prompt window. The title bar is '命令提示字元'. The text inside shows the user navigating to the Android SDK tools directory and then running the 'monitor' command. The path 'C:\Users\Tzu-Ting\AppData\Local\Android\Sdk\tools' is highlighted with a red box in the input line, and the word 'monitor' is also highlighted with a red box. Two yellow arrows point from the text below to the path and the command.

可看到成功進入路徑

參考路徑，不要完全照圖片打

Dalvik Debug Monitor Service



Dalvik Debug Monitor Service

The screenshot shows the Android Studio interface with the Dalvik Debug Monitor Service (DDMS) tab selected. The 'Emulator Co...' tab is highlighted with a red box. The 'Telephony Actions' section is also highlighted with a red box, showing options for 'Voice' and 'SMS'. A green callout box states: 'You can simulate a call or SMS to your device'. The 'LogCat' tab is visible at the bottom, showing a list of log messages.

Telephony Status

Voice: home Speed: Full

Data: home Latency: None

Telephony Actions

Incoming number: 0800p91000

☒ Voice

☐ SMS

Message:

Call **Hang Up**

Location Controls

Manual GPX KML

☒ Decimal

☐ Sexagesimal

Longitude -122.084095

Latitude 37.422006

LogCat

Saved Filters + -

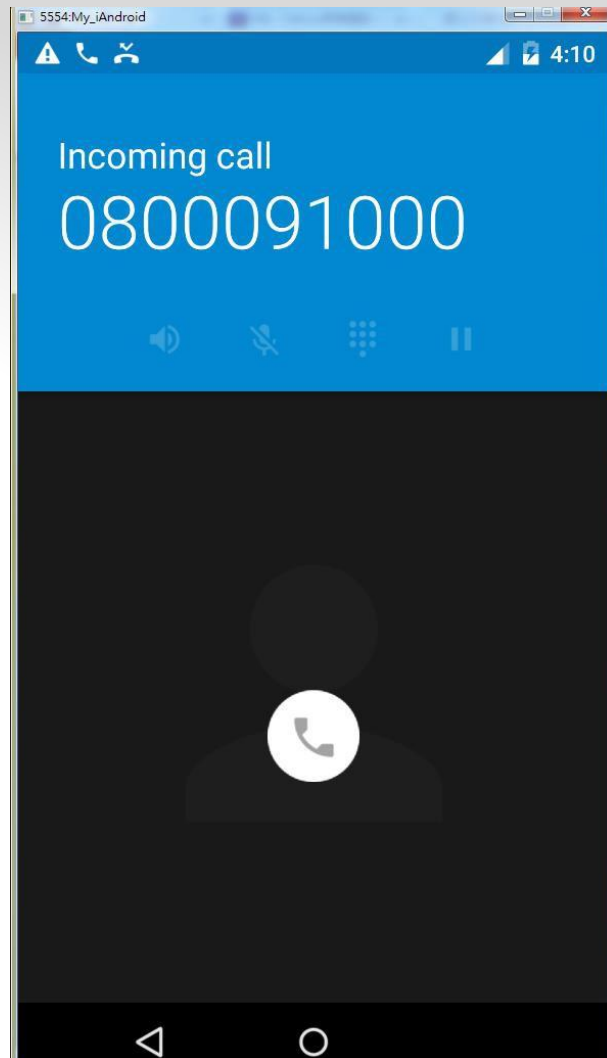
All messages (no filters)

Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope. verbose

L...	Time	PID	TID	Application	Tag	Text
						(241KB) LOS objects, 29% free, 9MB/13MB, paused
W	03-09 03:57:43.773	1901	2767	com.google.proce...	Uploader	no longer exists, so no auth token.
W	03-09 03:57:44.290	1901	2767	com.google.proce...	Uploader	No account for auth token provided
W	03-09 03:57:44.611	1901	2767	com.google.proce...	Uploader	No account for auth token provided

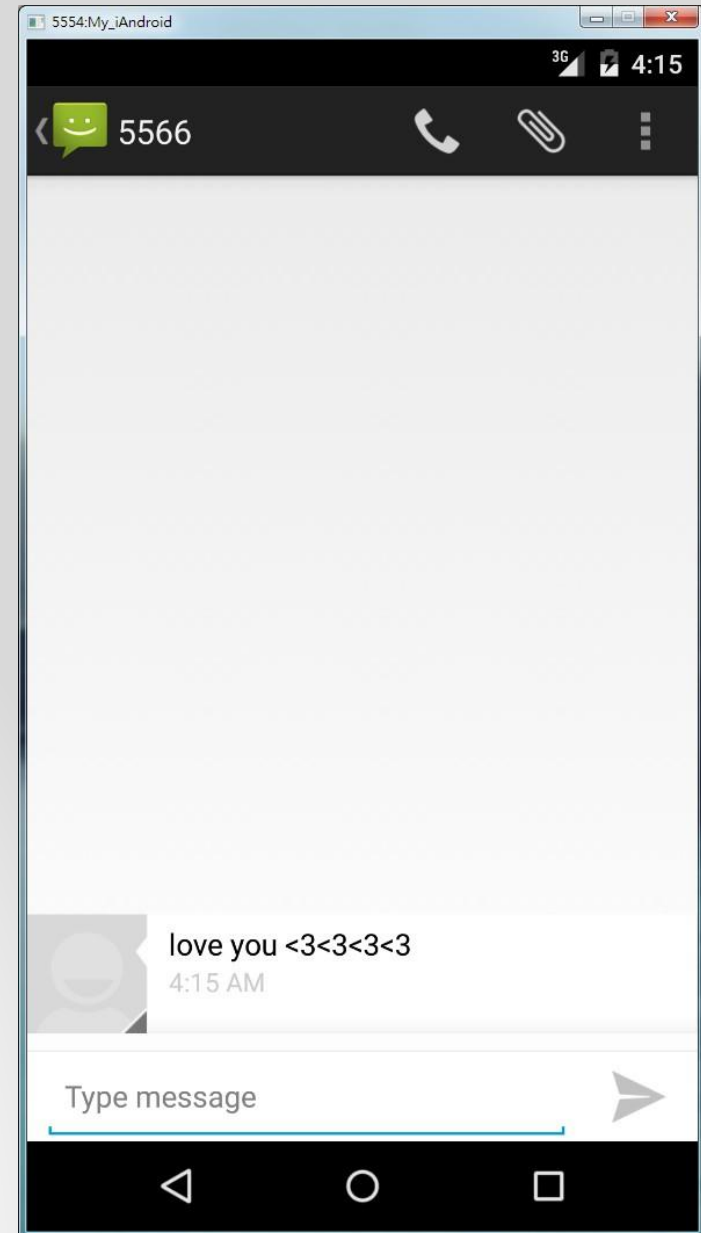
125M of 491M

Example - Simulated Call of DDMS



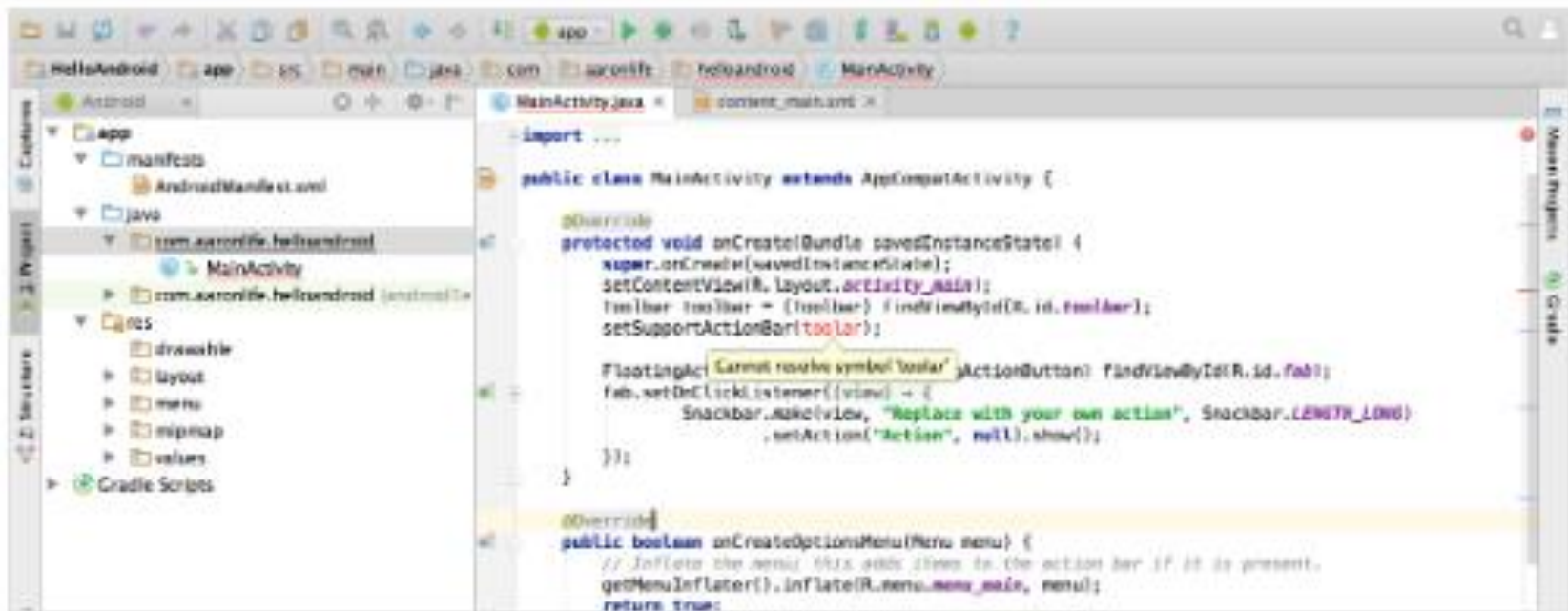
Example

- Simulate a SMS from 5566



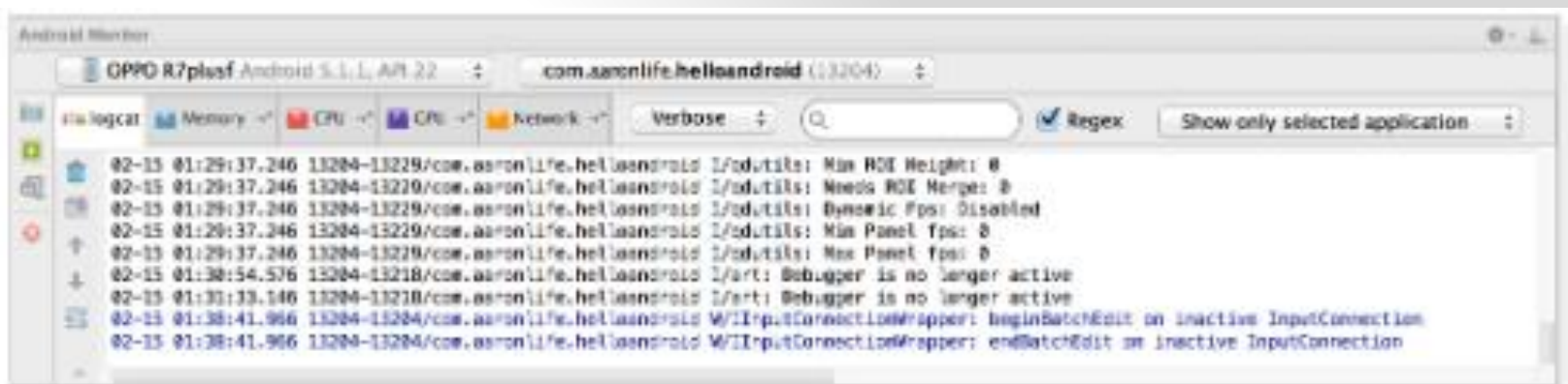
Debug tool

編輯器錯誤訊息：當出現程式語法錯誤時,Android Studio 將以紅色字體來表示,並且在右方 捲軸上用一紅色橫線標示錯誤的位置,當你跳到錯誤的位置,將滑鼠移到錯誤的程式碼上面,編輯器還會跳出錯誤的說明,讓你知道錯誤的原因,我們 就可以根據該訊息來修正錯誤。



Debug tutorial

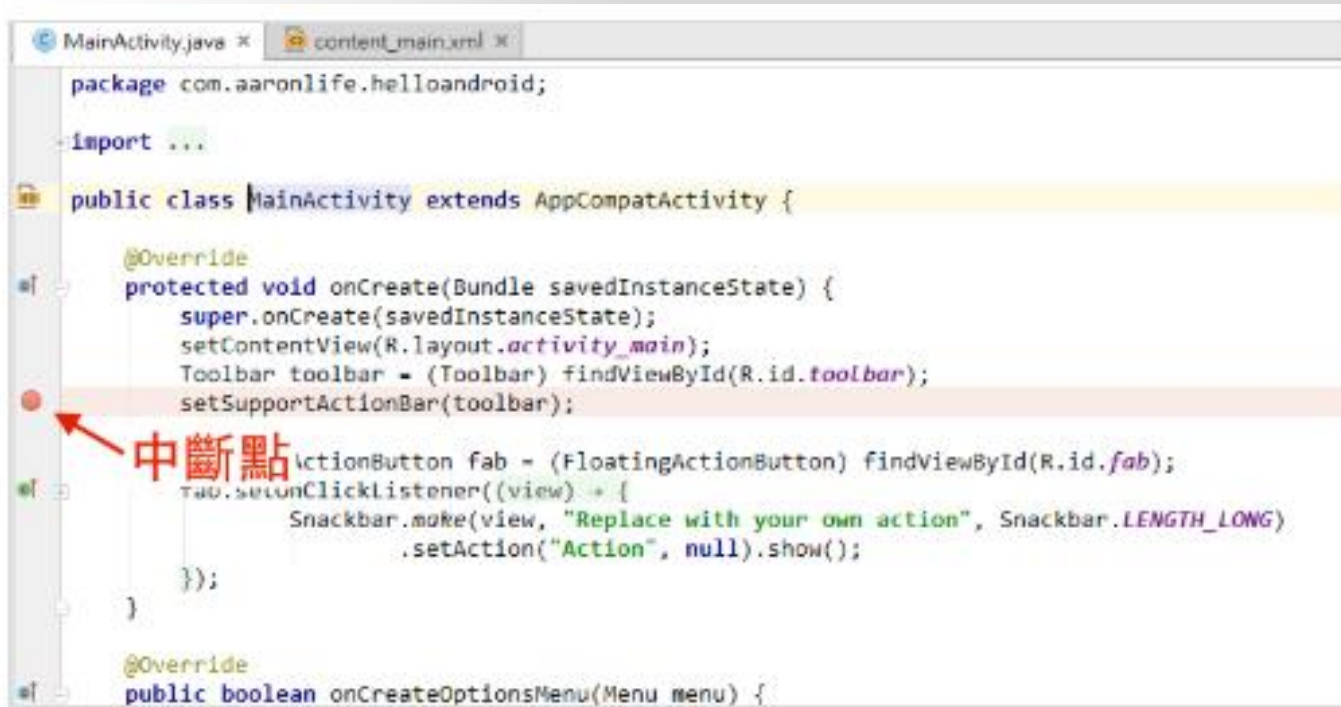
Android Monitor 視窗訊息：Android Monitor 視窗內包含許多功能，而可以用來除錯的為 logcat 標籤下的視窗，logcat 是 Android 系統提供的一個功能，可以用來在應用程式執行時丟出有用的資訊，再透過 logcat 來觀看該資訊，這些資訊在一般使用者手機上，是完全看不到的，只有在開發模式下，透過 logcat 才能取得，主要用在程式開發時的除錯。當程式因為執行錯誤產生例外而造成閃退時，也可以從這裡取得訊息，來得知發生什麼例外事件以及產生例外的原因。



編輯器錯誤訊息

Debug tool

設置中斷點，逐步偵錯



中斷點

ANDROID STUDIO專案介紹

Android的四大元件(Component)

1、activity

- 1 一個Activity通常就是一個單獨的螢幕（窗口）。
- 2 Activity之間通過Intent進行通信。
- 3 android應用中每一個Activity都必須要在AndroidManifest.xml配置文件中聲明，否則系統將不識別也不執行該Activity。

2、service

（a）started（啟動）：當應用程式組件（如activity）調用startService方法啟動服務時，服務處於started狀態。

（b）bound（綁定）：當應用程式組件調用bindService方法綁定到服務時，服務處於bound狀態。

=>(a)的服務可被其他程式調用,(b)的服務一旦綁定的程式終止即立即停用服務

Android的四大元件(Component)

3、content provider

android平台提供了Content Provider使一個應用程式的指定 數據集提供給其他應用程式。其他應用可以通過ContentResolver 類從該內容提供者中獲取或存入數據。

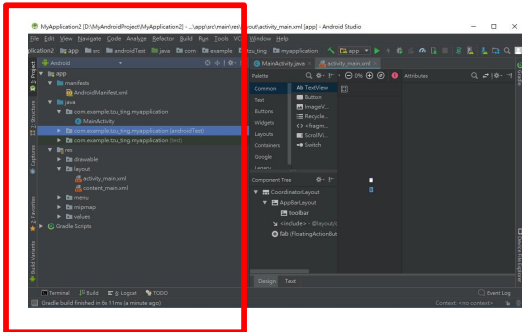
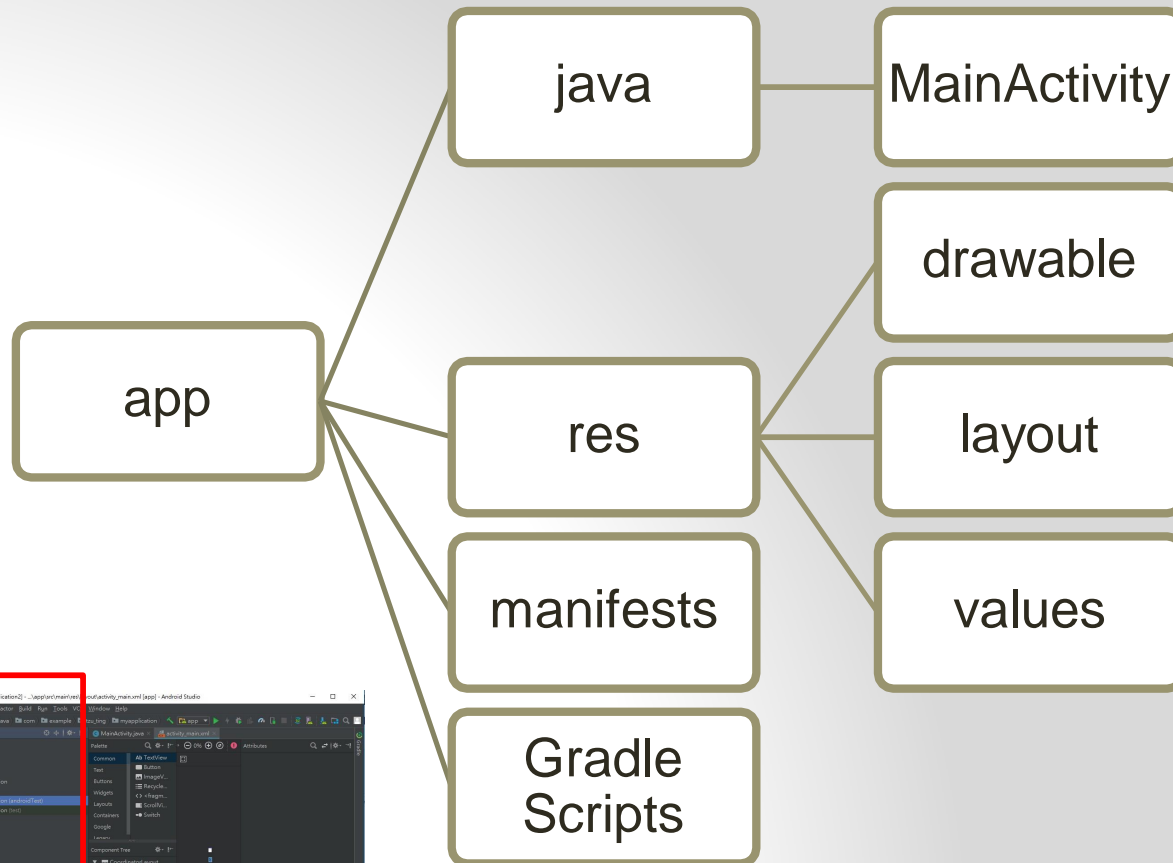
1只有需要在多個應用程式間共享數據是才需要內容提供者。例如，通訊錄數據被多個應用程式使用，且必須存儲在一個內容提供者中。它的好處是統一數據訪問方式。(提高效率)

4、broadcast receiver

你的應用可以使用它對外部事件進行過濾，只對感興趣的外部事件(如當電話呼入時，或者數據網絡可用時)進行接收並做出響應。廣播接收器沒有用戶介面。然而，它們可以啟動一個activity或service來響應它們收到的信息，或者用NotificationManager來通知用戶。通知可以用很多種方式來吸引用戶的注意力，例如閃動背燈、震動、播放聲音等。一般來說是在狀態欄上放一個持久的圖標，用戶可以打開它並獲取消息。

Android Studio程式目錄架構

- ▶ 在一個Android程式完整的套件目錄架構如下:



目錄架構介紹

- 一個專案(project)中可以包含多個模組(module)，每個module都可以獨立運行。

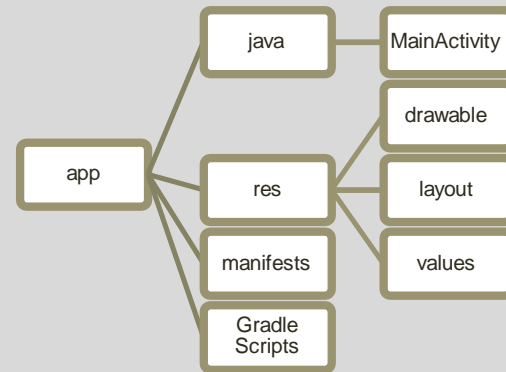
- Project中包含以下資料夾：

- java: 主要放code的地方。
 - 我們要寫code的地方也在這裡面。

- res: 放置APP需要用到的資源，如圖片、文字檔案等。
 - drawable: App所要用到的圖片。
 - layout: App所要用到畫面排版。

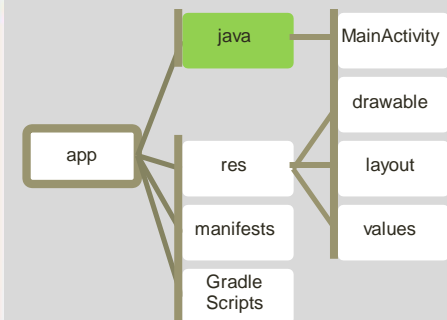
- manifest: 設定APP所需的權限，及程式的一些設定。

- gradle scripts: 應用程式建置系統，包含程式編譯等等相關的設定。

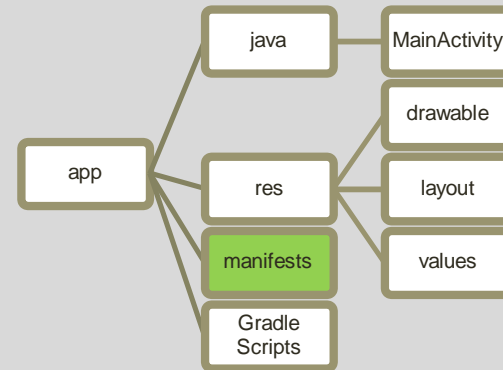


Android程式目錄架構-- java目錄

- 在java目錄下即是程式原始碼所存放的位置，
如我們所開發的主程式MainActivity.java => 可更動
- Android Project內還有一個重要的「R.java」，是**不可更動、由系統編譯自動產生的**
- 當我們在.xml檔案上建立一個Component(例如Button)，並且在屬性設定他的id(例如button1)，此時R.java內會自動產生一個**資源識別碼**，
為一組**16進位的數字**
- 系統藉由這個資源識別碼來.xml和背後的.java檔案連接起來，可以說R.java扮演了非常重要的橋梁



設定權限



- 設定專案權限
 - 權限設定同樣在 **AndroidManifest.xml**，以 **<uses-permission>** 標籤設定此Android程式要開放的權限。
 - 例如GoogleMap要正確執行，需有以下權限

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<!--
The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
Google Maps Android API v2, but are recommended.
-->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

程式運作指令

- 控制Android程式
 - 在Android中，控制程式執行流程的語法是java檔
 - 放在java目錄底下，藉由調整.java的code來操控整個程式
 - Android程式中，每個畫面都可以視為一個活動(activity)。
 - 藉由在AndroidManifest.xml中的<activity>標籤可決定此activity受哪個.java檔控制 (如下圖，此Activity控制java檔為”MapsActivity.java”
 - Activity標籤需為<application>的子標籤

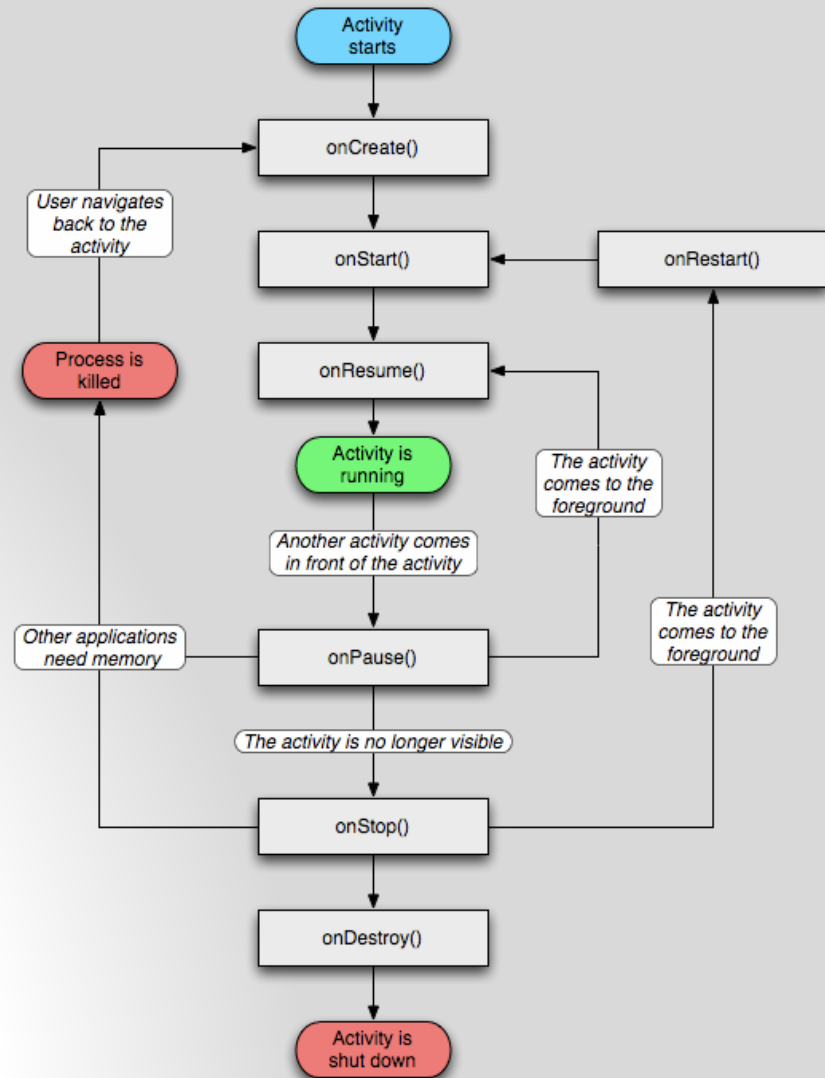
```
<activity
    android:name=".MapsActivity"
    android:label="Map" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

程式運作指令-How Activity work

- Android 生命週期
 - 根據Android生命週期，在程式正式執行之前(Activity is running)會依序呼叫onCreate(), onStart(), onResume() 等函式
 - 開啟一個activity，必須載入相對應的layout以配合程式行為
 - 通常會在onCreate()中利用 setContentView()載入layout
- setContentView()
 - 利用此函式載入Layout

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);
    setUpMapIfNeeded();
}
```



1) 一般啟動

onCreate → onStart → onResume

啟動一個Activity的基本流程是：分配資源給這個Activity(`onCreate`)，然後將Activity內容顯示到螢幕上(`onStart`)，在一切就緒後，取得螢幕的控制權(`onResume`)，使用者可以開始使用這個程式。

(2) 呼叫另一個Activity

onPause(1) → onCreate(2) → onStart(2) → onResume(2) → onStop(1)

先凍結原本的Activity，再交出直接存取螢幕控制權(onPause)的過程。直到Activity 2完成一般啟動流程後，Activity 1才會被停止。

(3) 回復原Activity

```
onPause(2) → onRestart(1) → onStart(1) → onResume(1) →
onStop(2) → onDestroy(2)
```

按Back鍵可以回到原本的Activity。

(4) 退出/結束

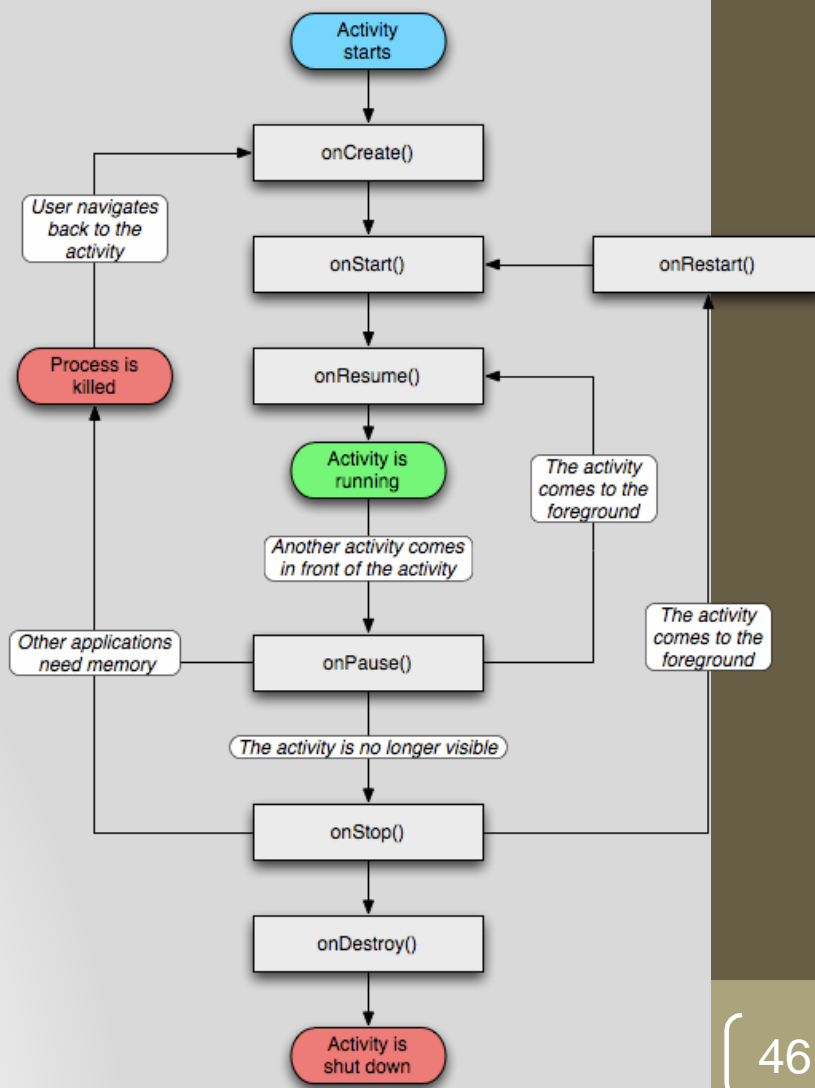
onPause → onStop → onDestroy

如果程式中有直接呼叫finish函式來關閉Activity的話，系統會暫停(onPause)後停止(onStop)，然後才是銷毀(onDestroy)。

(5) 回收後再啟動

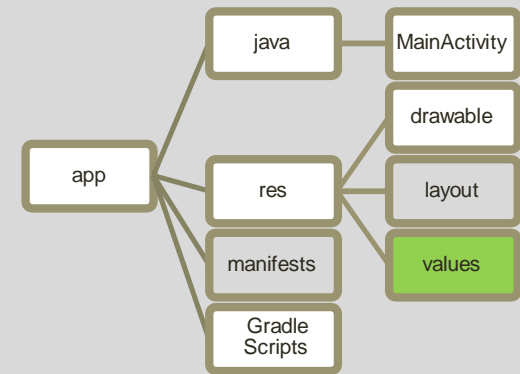
onCreate → onStart → onResume

被回收掉的Activity一旦又重新被呼叫時，會像一般啟動一樣再次呼叫Activity的onCreate函式。



Android程式目錄架構

res資源描述檔-values



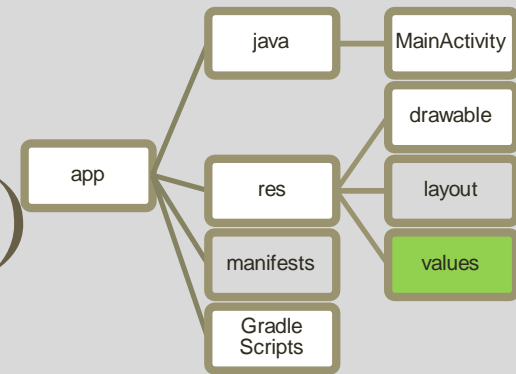
- 在res檔目錄下主要是儲存會使用到的資料以及資源,如:

Values目錄:

包含所有使用xml格式的參數值描述檔,及存放一些資源的目錄,如:字串、顏色、風格等;像strings.xml檔裡儲存的就是字串資源的xml描述檔,下張投影片則是string.xml程式之範例。

Android程式目錄架構

res資源描述檔-values(續)



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

```
<resources>
```

```
    <string name="app_name">導航</string>
```

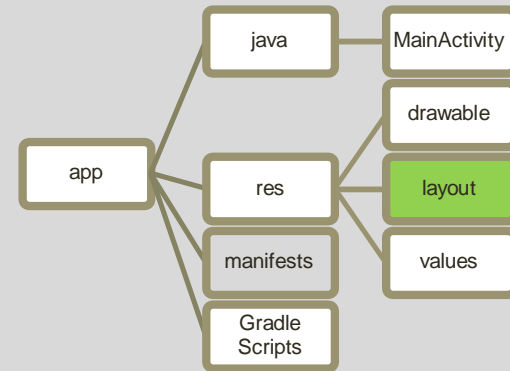
```
    <string name="destination">目的地</string>
```

```
    <string name="starting_point">出發點</string>
```

```
</resources>
```

//app_name裡所儲存的就是 “導航” 這個字串

Android程式目錄架構 res資源描述檔-layout



- **layout目錄:**

使用xml語法創建一個activity_main.xml，主要是來描述使用者介面視窗的描述檔，也就是所有介面如何佈局之樣式以及顯示方法皆在這邊設計、以下為程式範例:

```
<?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
```

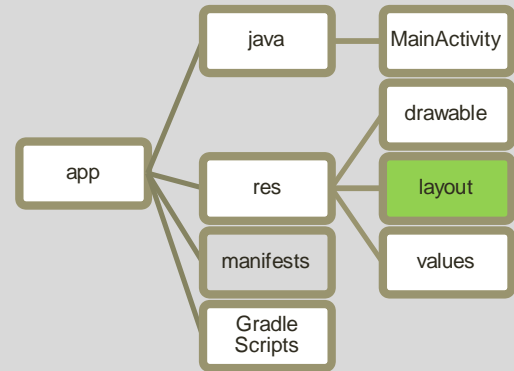
```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="@string/starting_point" />
```

Android程式目錄架構

res資源描述檔-layout(續)



<EditText

```
android:text=""
android:id="@+id/start"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:inputType="numberDecimal" />
```

<TextView

```
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="@string/destination" />
```

<EditText

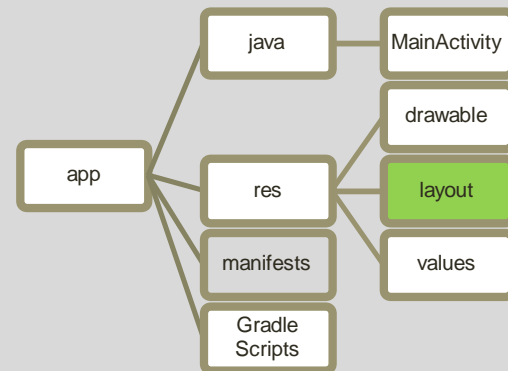
```
android:text=""
android:id="@+id/end"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:inputType="numberDecimal" />
```

<Button

```
android:text=" 導 航 開 始 "
android:id="@+id/run"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />
```

</LinearLayout>

Android程式目錄架構 res資源檔-layout(續)



- 程式解說:

<LinearLayout

`xmlns:android="http://schemas.android.com/apk/res/android"`

`android:orientation="vertical"`

`android:layout_width="fill_parent"`

`android:layout_height="fill_parent">`

● LinearLayout

- 表示介面是線性排列方法呈現,而下一行我們使用xmlns為開頭的程式碼,主要是來描述後面url網頁所提供的空間定義的網頁位置,接著以下若要使用android所提供的定義,則只要在最前面加上 android: 即可;

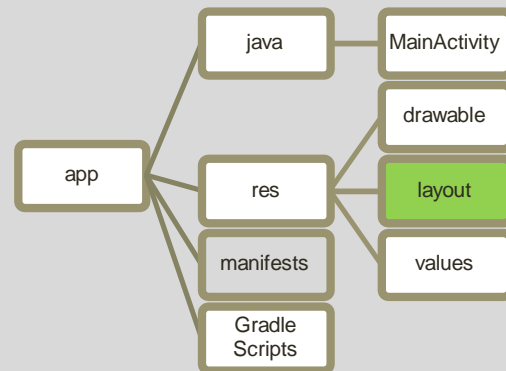
● android:orientation="vertical"

- 代表為垂直線性排列;

● android:layout_width="fill_parent"

- 表示此布局寬度為填滿手機寬度,下一行則式布局高度為填滿.

Layout類型



◆ **Linear Layout(線性佈局)**：
水平線或垂直線的排版設定。

◆ **Relative Layout(相對位置版面佈局)**：
在此版面的元件都需要參考一個對象，這個對象可以是任一個元件或是元件所在位置的元件。

◆ **TableLayout(表格欄位版面佈局)**：
讓物件像表格一樣排列。

◆ **AbsoluteLayout(絕對版面佈局)**：
用X軸與Y軸來配置版面。

◆ **FrameLayout(框架版面佈局)**：
重疊顯示，語法最先下的會出現在最後面。

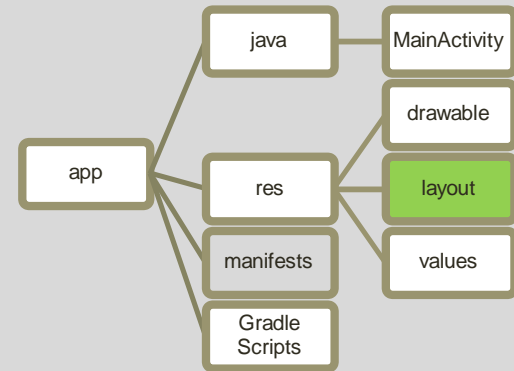
關於ConstraintLayout的部分屬性總結

<https://blog.csdn.net/u013187628/article/details/60751812>

ConstraintLayout 完全解析 快來優化你的佈局吧

<https://blog.csdn.net/lmj623565791/article/details/78011599>

Android程式目錄架構 res資源檔-layout(續)



- 程式解說:

<TextView

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/starting_point" />
```

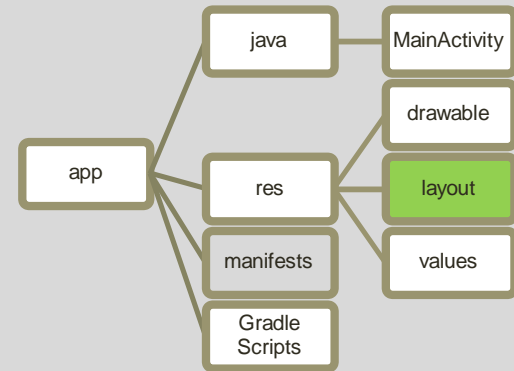
- Text View

- 元件資源標籤所包含的是文字顯示,在layout_height中我們設定為wrap_content,代表這個textview元件布置高度為與內文文字高度一樣 ;

- android:text="@string/starting_point" />

- 表示我們要顯示"@string/starting_point"中之字串;Xml解析器會將@後面的字串解析成識別符號,而使用在string.xml裡面鎖定義的字串starting_point

Android程式目錄架構 res資源檔-layout(續)



程式解說:

<EditText

android:text=""

android:id="@+id/start"

android:layout_width="fill_parent"

android:layout_height="wrap_content"

android:inputType="numberDecimal" />

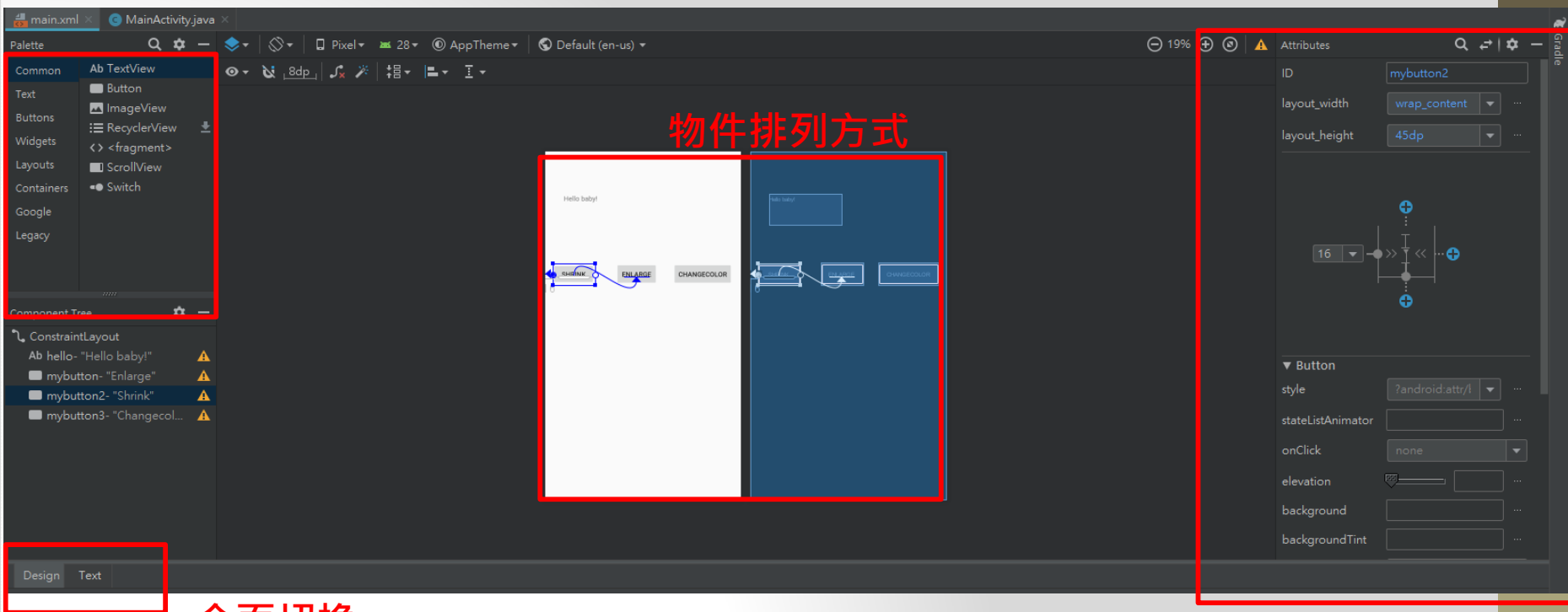
- EditText

- 一個提供修改文字的欄位，而在id=@後面我們加一個 '+' 符號，代表我們建立了一個名為start的識別符號，而把他歸類於values的id中。

設定Layout甚至可以直接拖拉介面

物件

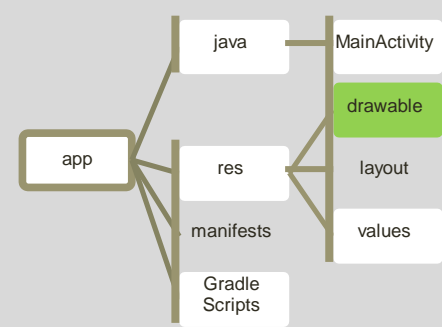
物件屬性



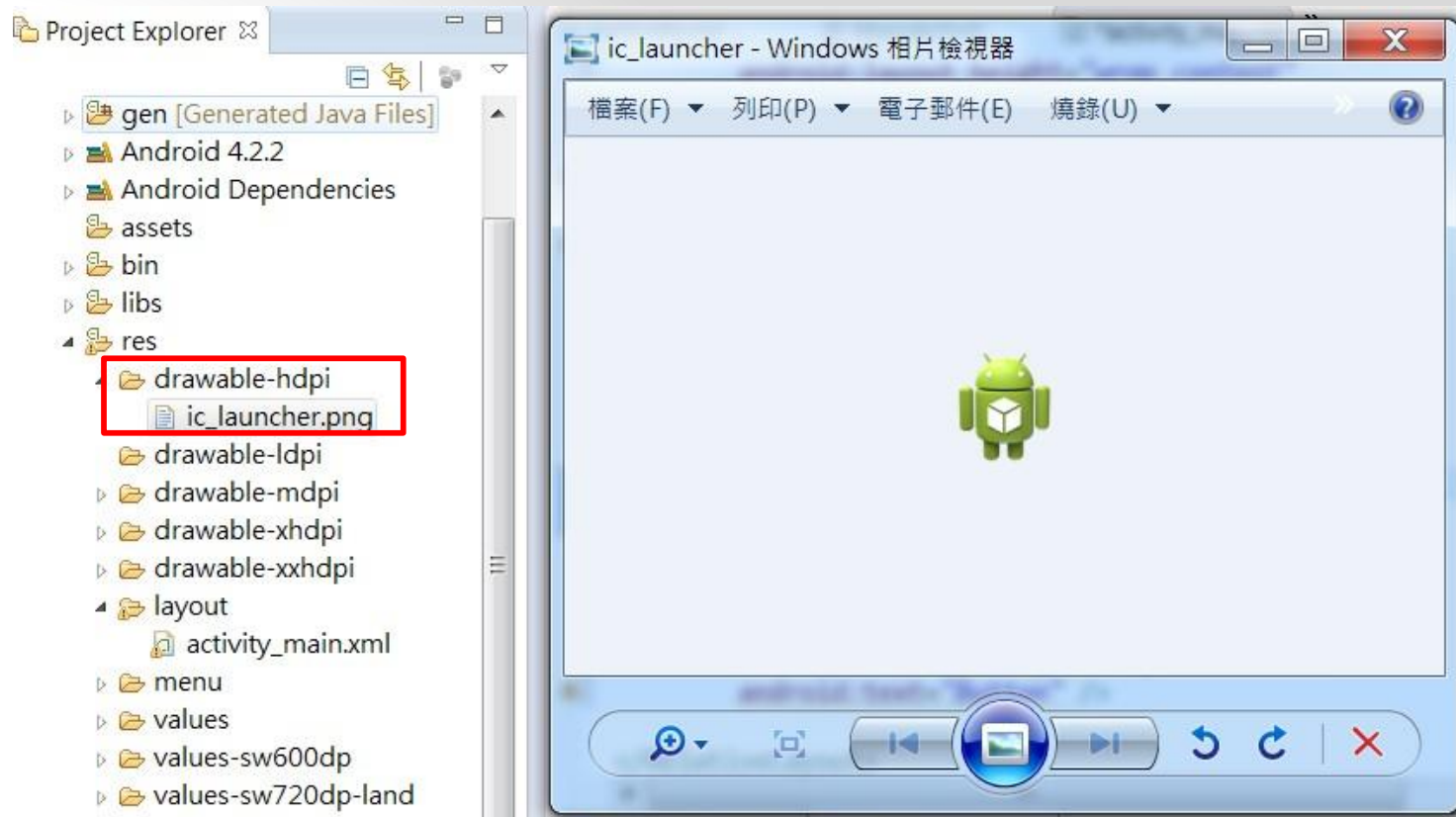
介面切換

Android程式目錄架構

res資源描述檔- drawable



- 在drawable下主要是儲存圖示檔案,讓程式開發者可以將圖示加在介面中,也就是當用模擬器跑結果時,便會將圖示顯是在畫面上。



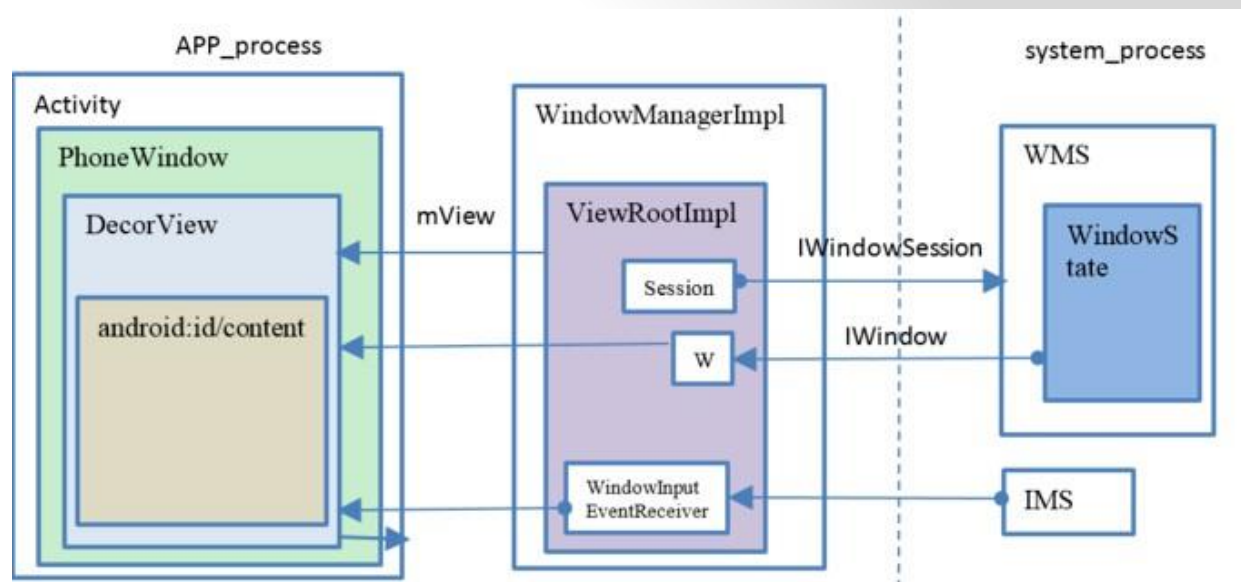
如想更深入了解，建議參考資源

- Activity是什麼？Java類別，沒有它就沒有互動了

<https://litotom.com/2017/03/12/android-effective-03-1st-app-2/>

- 圖解Android View的事件傳遞

<https://www.jianshu.com/p/bea1bb4aac95>

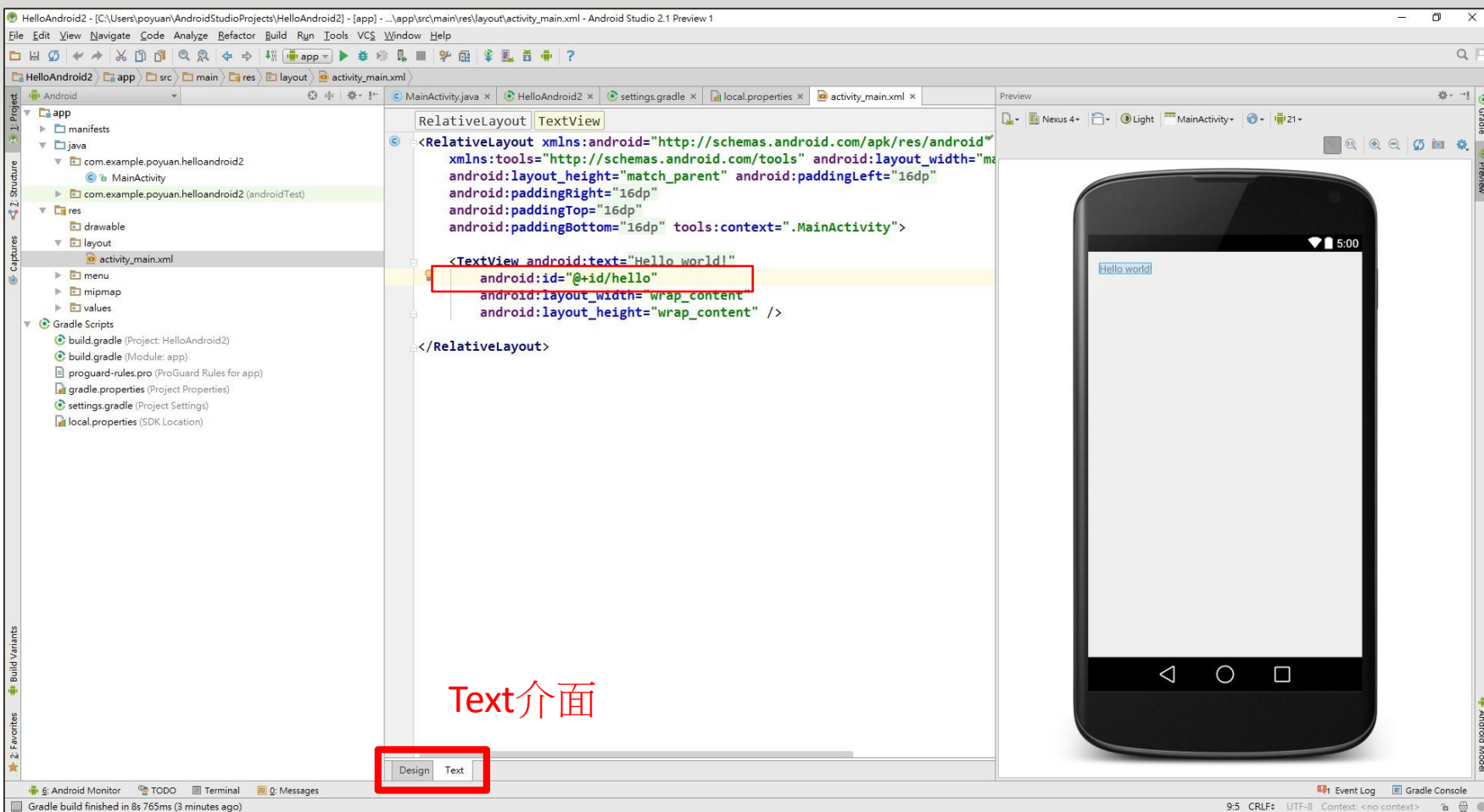


Android的Hello World!

Android : hello world

1. Application name: HelloAndroid
2. Project Name: Hello Android
3. Target SDK: Android lollipop

Android : hello world



Layout -> activity_main

為了能在code中控制layout中的物件，我們要幫物件取一個ID，才能在程式碼裡面控制此物件。

Android的Hello World!

```
ApplicationTest.java x MainActivity.java x activity_main.xml x strings.xml x
package com.example.poyuan.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView ;

public class MainActivity extends Activity {

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

        TextView tv = (TextView)findViewById(R.id.hello) ;
        tv.setText("Hello Android!");
    }
}
```

R檔是控管整個程式的資源的地方，會自動產生，不需要也不能手動更改。

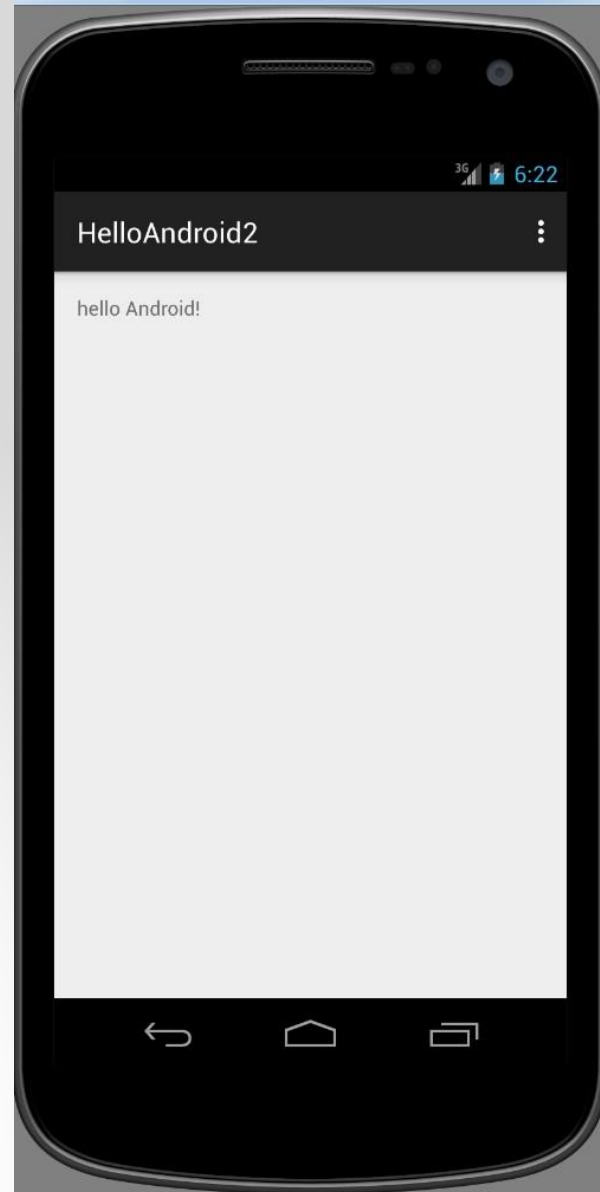
回到程式碼，宣告一個TextView物件，並將剛剛layout中的TextView指定過來。然後控制它所顯示的字。

Programming shortcuts

Action	Android Studio Key Command
Command look-up (autocomplete command name)	CTRL + SHIFT + A
Project quick fix	ALT + ENTER
Reformat code	CTRL + ALT + L (Win) OPTION + CMD + L (Mac)
Show docs for selected API	CTRL + Q (Win) F1 (Mac)
Show parameters for selected method	CTRL + P
Generate method	ALT + Insert (Win) CMD + N (Mac)
Jump to source	F4 (Win) CMD + down-arrow (Mac)
Delete line	CTRL + Y (Win) CMD + Backspace (Mac)
Search by symbol name	CTRL + ALT + SHIFT + N (Win) OPTION + CMD + O (Mac)

Android的Hello World!

- 模擬器執行畫面

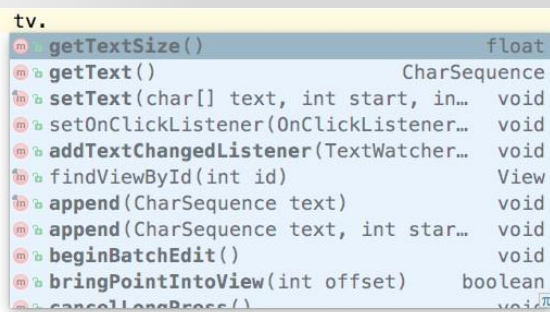


How to control your Objects?

- `findViewById(R.XXX.XXX)`
 - 在程式碼中，要先從R(resource)中找到你要控制的物件。

```
TextView tv = (TextView) findViewById(R.id.hello);  
Button enlarge = (Button) findViewById(R.id.btn_enlarge);
```

- 之後就可以從物件中的function操控物件。



```
tv.  
m getTextSize() float  
m getText() CharSequence  
m setText(char[] text, int start, in... void  
m setOnClickListener(OnClickListener... void  
m addTextChangedListener(TextWatcher... void  
m findViewById(int id) View  
m append(CharSequence text) void  
m append(CharSequence text, int star... void  
m beginBatchEdit() void  
m bringPointIntoView(int offset) boolean  
m cancelLongPress() void
```

- 監聽Button的動作
 - 需要設定一個listener，用來監聽使用者有沒有對Button做動作。

```
enlarge.setOnClickListener(...);
```


- 加入一個監聽**按壓(click)**的listener

```
enlarge.setOnClickListeners(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        //What to do when the button is pressed  
    }  
});
```

- 讓這個Button可以把剛剛的文字放大。

```
enlarge.setOnClickListeners(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        float size = tv.getTextSize() ;  
        tv.setTextSize(TypedValue.COMPLEX_UNIT_PX, size+5);  
    }  
});
```

Check Point-1

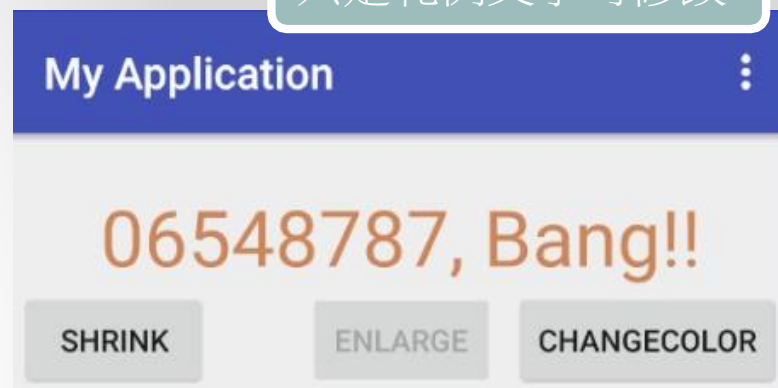
- 製作三個Button: Enlarge, Shrink, Change Color
- 控制畫面中的文字放大、縮小及換顏色。
- 條件：
 - Change Color可以隨機變換文字顏色。

- Hint

```
Random rnd = new Random();  
int color = Color.rgb(rnd.nextInt(256), rnd.nextInt(256), rnd.nextInt(256)) ;
```

- 文字放大到一定大小，Enlarge Button就不能按了。縮小功能也相同。
 - Hint: button.setEnabled(...)

只是範例文字可修改



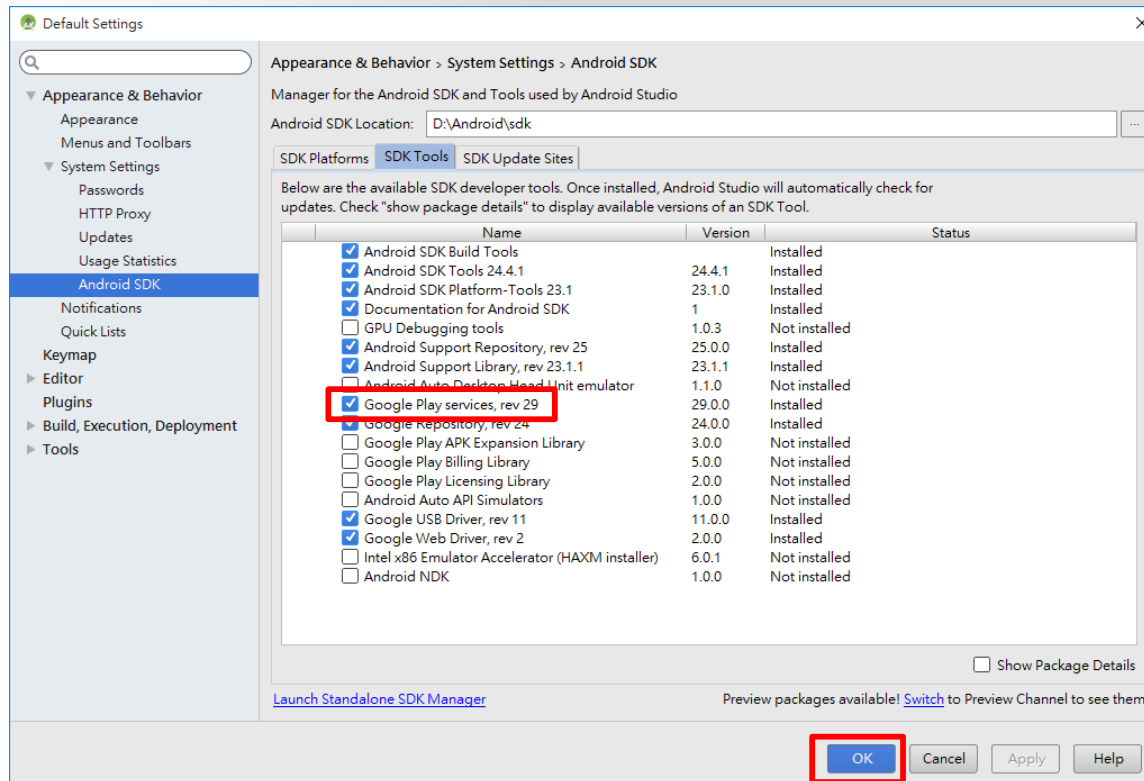
GOOGLE MAP API

Getting start

- 使用Google Map
 1. 安裝 Google Play service SDK
 2. 建立 Google Map project
 3. 取得 Google Maps API 金鑰
 4. 完成 Google Map App

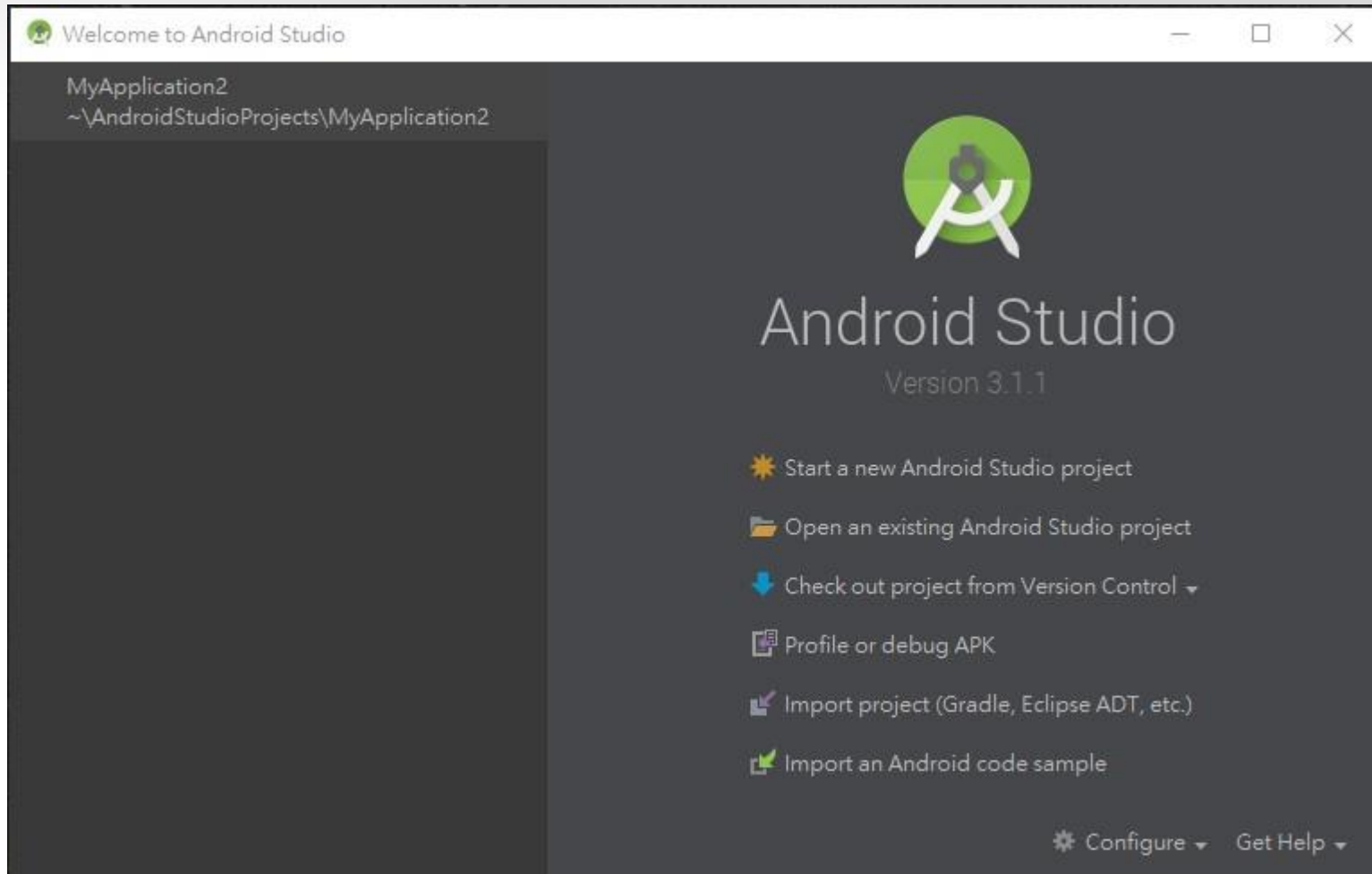
安裝 Google Play service SDK

- Tools -> Android -> SDK Manager
- SDK Tools -> google play service



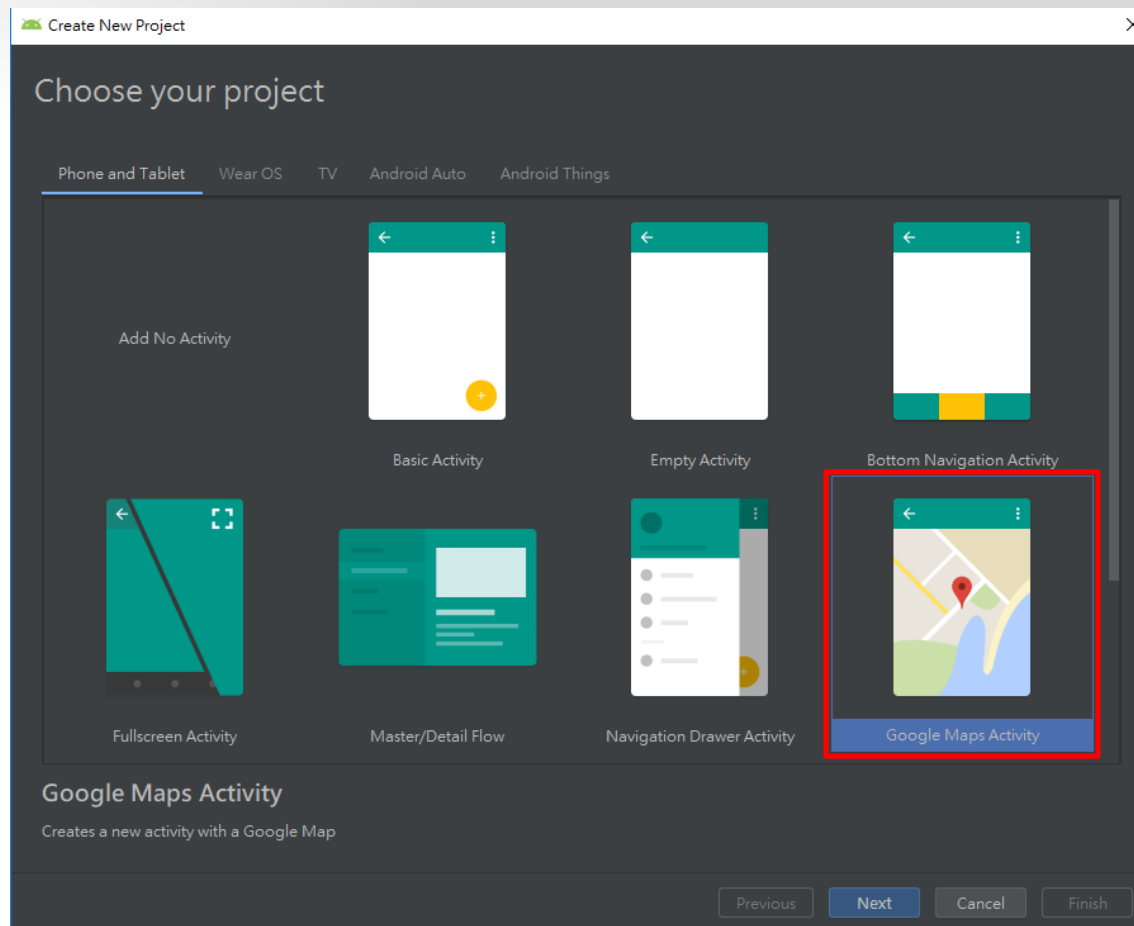
建立 Google Map project

- Start a new Android Studio project



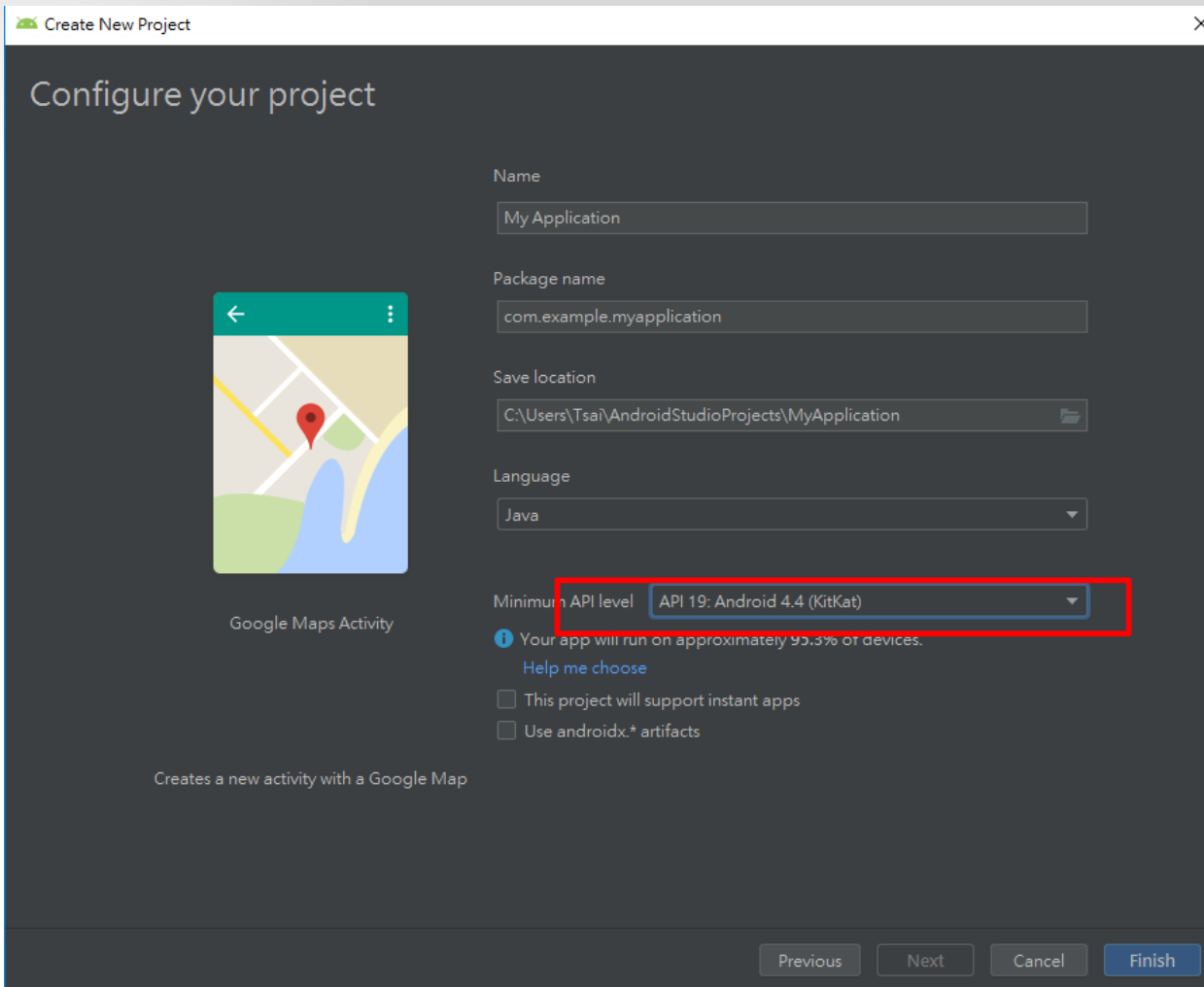
建立 Google Map project

- Configure your new project



建立 Google Map project

- Choose Minimum SDK version



The screenshot shows the 'Create New Project' dialog in Android Studio. The title bar says 'Create New Project'. The main heading is 'Configure your project'. On the left, there is a preview of a 'Google Maps Activity' with a map icon. The right side contains several configuration fields:

- Name:** My Application
- Package name:** com.example.myapplication
- Save location:** C:\Users\Tsai\AndroidStudioProjects\MyApplication
- Language:** Java
- Minimum API level:** API 19: Android 4.4 (KitKat) (highlighted with a red box)

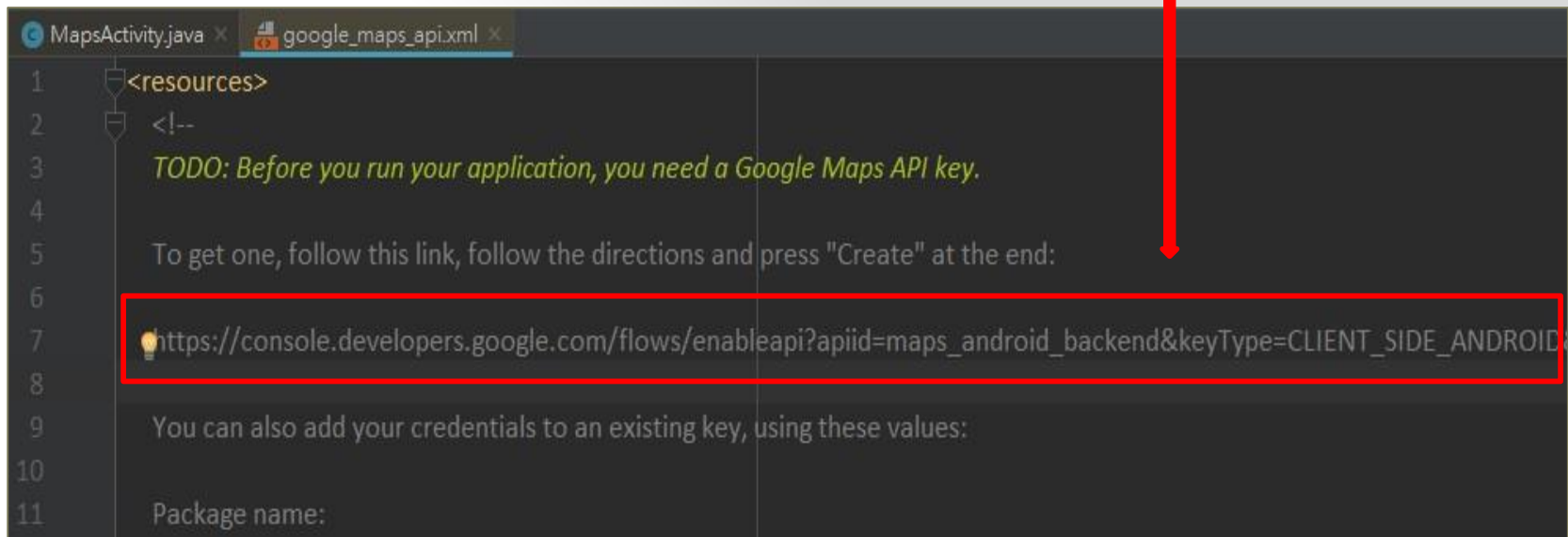
Below the API level, there is an information icon and text: 'Your app will run on approximately 95.3% of devices.' with a link 'Help me choose'. There are also two checkboxes:

- ☐ This project will support instant apps
- ☐ Use androidx.* artifacts

At the bottom, it says 'Creates a new activity with a Google Map'. The bottom navigation bar has buttons: 'Previous', 'Next', 'Cancel', and 'Finish'.

建立Google Map application

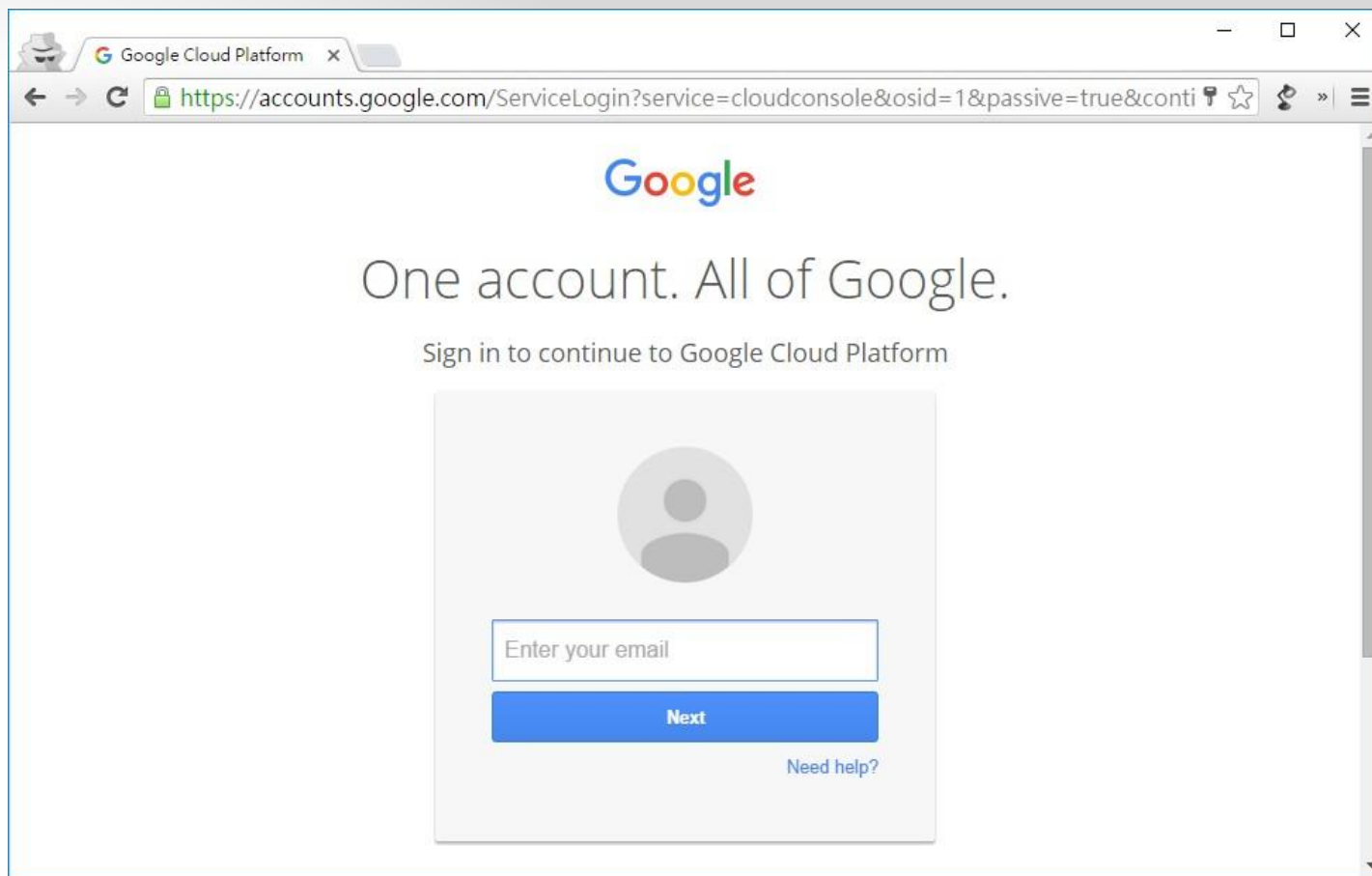
- 進入後說明文件有申請API Key的網址，複製到瀏覽器
- 此網址根據電腦金鑰產生，每台電腦不同



```
1 <resources>
2 <!--
3  TODO: Before you run your application, you need a Google Maps API key.
4
5  To get one, follow this link, follow the directions and press "Create" at the end:
6
7  https://console.developers.google.com/flows/enableapi?apiid=maps\_android\_backend&keyType=CLIENT\_SIDE\_ANDROID
8
9  You can also add your credentials to an existing key, using these values:
10
11  Package name:
```

取得 Google Maps API 金鑰

- 將網址貼入瀏覽器，並登入Google帳號



取得 Google Maps API 金鑰

- 同意在此Google帳號中，新增Google API，並放入新專案中



The screenshot shows a web browser window with the address bar displaying "https://console.developers.google.com/flows". The page title is "啟用 API" (Enable API). Below the address bar, there are several tabs: "從 Firefox 匯入的書籤" (Import bookmarks from Firefox), "Music", "New-NCTU", "Gaming", and "Stream". The main content area has a header with the Google APIs logo and a dropdown menu labeled "請選取專案" (Please select a project). The main text reads: "在 Google API Console 中為您的應用程式註冊 Google Maps Android API" (Register Google Maps Android API for your application in the Google API Console). Below this, it says: "Google API Console 可讓您管理應用程式及監控 API 用量。" (Google API Console lets you manage applications and monitor API usage). Then, it states: "您目前沒有任何專案，系統將建立名為「My Project」的新專案。" (You currently have no projects, the system will create a new project named "My Project"). Next, it says: "透過電子郵件通知我有關功能公告、效能建議、意見調查和特別優惠的最新消息。" (Notify me by email about new features, performance suggestions, surveys, and special offers). There are two radio buttons: "是" (Yes) and "否" (No), with "否" selected. Below this, it says: "我同意在使用任何服務和相關 API 時，均遵守適用的《服務條款》。" (I agree to comply with the applicable Terms of Service when using any service and related API). There are two radio buttons: "是" (Yes) and "否" (No), with "是" selected. At the bottom, there is a blue button labeled "同意並繼續" (Agree and continue).

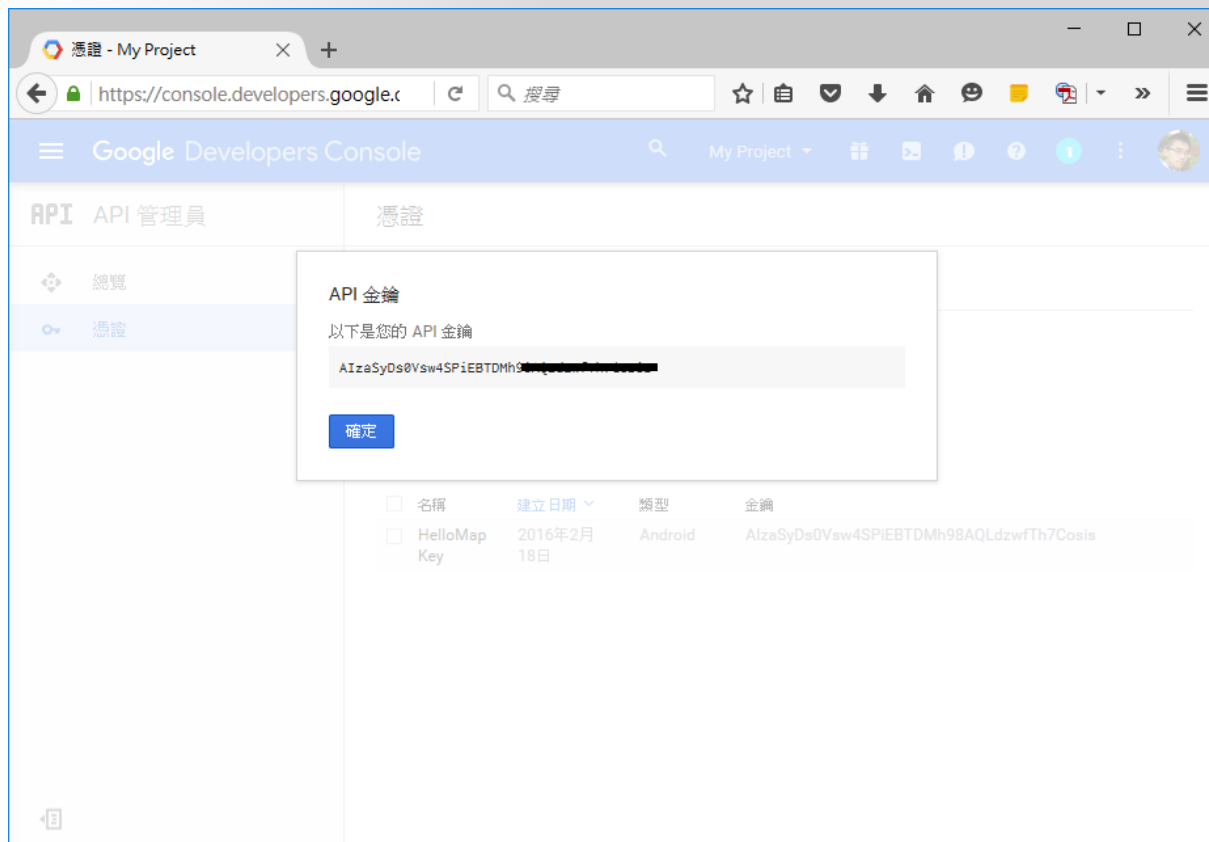
取得 Google Maps API 金鑰

- 啟用Google Map API成功，前往憑證產生API




取得 Google Maps API 金鑰

- 建立成功，取得的API金鑰將允許我們的Android project使用Google Map功能，若無API 金鑰，將無法正確執行Google Map



取得 Google Maps API 金鑰

- 按確定後開始設定，並按右邊的  設定此API Key
- 套件名稱、SHA-1憑證指紋為此專案與電腦的資訊，若要將此API Key共用至其他專案，點選新增套件名稱和指紋




名稱

HelloMap Key

金鑰限制

設定限制即可避免未授權的使用行為或配額遭到竊用。 [Learn more](#)

應用程式限制： Android 應用程式  API 限制： 無

應用程式限制 API 限制

應用程式限制可讓您指定有權使用這組金鑰的網站、IP 位址或應用程式。

應用程式限制

☐ 無

☐ HTTP 參照網址 (網站)

☐ IP 位址 (網路伺服器及 Cron 工作等)

☒ Android 應用程式

☐ iOS 應用程式

限制您的 Android 應用程式的使用量 (選項)

新增套件名稱和 SHA-1 簽署憑證指紋，即可限制您的 Android 應用程式的使用量。您可從 AndroidManifest.xml 檔案中找出套件名稱，然後利用以下指令取得指紋：

```
$ keytool -list -v -keystore mystore.keystore
```

套件名稱

tw.edu.nctu.hellomap

SHA-1 憑證指紋



+ 新增「套件名稱和指紋」

取得 Google Maps API 金鑰

- 設定API Key
 - AndroidManifest.xml 是一個用來描述 Android 應用程式「整體資訊」的設定檔，無論是設定權限、登記服務、放入API Key皆在此檔案中操作。
 - API Key 設定在Application標籤下，使用meta-data標籤告知Application指定名稱的數值為何。
 - 先將剛剛得到的KEY複製到value裡google_maps_api.xml中KEY的欄位
 - 再把第一個<meta-data />加入AndroidManifest.xml

註:第二個meta-data標籤中，value指向的值正是剛輸入KEY的地方

```
28
29
30
31
32
33
```

```
<meta-data
  android:name="com.google.android.gms.version"
  android:value="@integer/google_play_services_version" />
<meta-data
  android:name="com.google.android.geo.API_KEY"
  android:value="@string/google_maps_key" />
```

程式運作指令

- 宣告控制元件變數
 - Layout中我們宣告了許多元件，並利用id給各元件編號
 - 在java中，我們可以利用程式碼取得控制這些元件的變數
 - 利用findViewById(), findFragmentById()可透過ID找到變數
- 操控地圖元件

- 宣告變數

```
private GoogleMap mMap;
```

- 取得變數

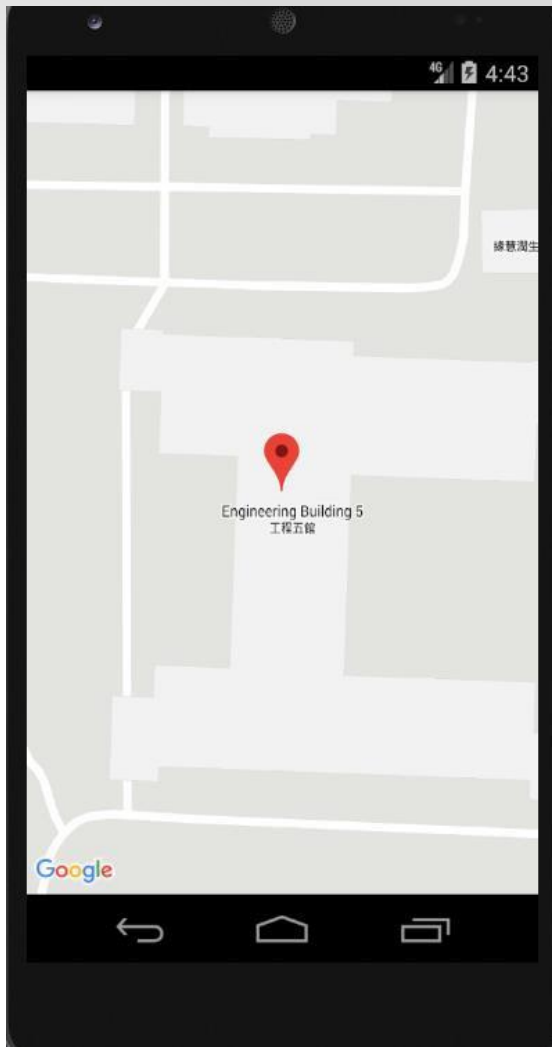
```
SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()  
    .findFragmentById(R.id.map);
```

- 操控地圖

```
mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));  
mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
```


Check Point-2

- 使Google Map能正確執行，並將Marker放在工程五館上



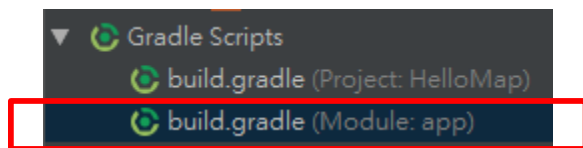
Hint: you can use
`mMap.animateCamera`
`(CameraUpdateFactory.zoomTo(,));`

若遇到update play-service問題

- 先確認前面的SDK Tools -> google play service是否安裝了
=>若未安裝 安裝後先測試能不能跑 不能跑則接續
- 開啟AVD Manger 並且新增一個AVD，選擇有PlayStore標誌的Nexus 5

Name ▾	Play Store	Size	Resolution	Density
Nexus 5X	▶	5.2"	1080x1920	420dpi
Nexus 5	▶	4.95"	1080x1920	xxhdpi

- Android版本選x86 images標籤裡4.4 KitKat (Google API)
(若原本沒有裝 先按Download更新)
- 修改Gradle Scriptes裡build.gradle(Module:app)



=>將implementation 'com.google.android.gms:play-services-maps:15.0.0'
改成implementation 'com.google.android.gms:play-services-maps:11.0.1'

- 開啟AVD Manger 將剛剛創好的AVD打開 然後再次Run App

參考資料及資源

- Android Developers, <http://developer.android.com>
- Developer Resources for Java Technology, <http://java.sun.com/>
- Eclipse.org home, <http://www.eclipse.org/>
- Android Market, <http://www.android.com/market/>
- Android、iOS、WP7三大系統有什麼差別
- , <http://www.betanews.com.cn/guide/2012-02-09/1620.shtml>
- Google Android 2.X應用開發實戰
- 百度, <http://baike.baidu.com>
- Android SDK 教學 | 孫傳雄談科技 ,
<http://blog.chinatimes.com/tomsun/>
- **Android Studio 筆記—Layout元件介紹**
<https://ithelp.ithome.com.tw/articles/10189264>
- Yu-Ja Chen's Android Lab

