序

從撰寫第一本《Android 2.x 手機程式開發教戰手冊》到此本《Android 5.x App 開發教戰手冊》歷經了 Android 四大版本（2.x, 3.x, 4.x, 5.x），Android 系統也歷經了許多重大改變：

- 系統的成長與改良：
  - 2.x 版僅支援手機，而且操作流暢度差強人意。
  - 3.0 版開始支援平板，但卻不支援手機，苦了開發者要分別熟悉 2 套 API。
  - 4.0 版系統穩定性大幅提升，而且手機、平板共用 API 而不再分家，方便開發者開發。
  - 5.0 版提倡 material design 將 UI 設計提升至高水準，即便對 iOS 系統也毫不遜色。
- 市佔率大幅提升與各種裝置綻放光芒：
  - 每年的銷售市佔率從 2.x 版的 3 成到現在接近 8.5 成。
  - 從當初只有手機裝置到平板、電子書裝置，到現在有行動電視與穿戴裝置。

對 App 開發者而言最重大的改變莫過於 Android Studio 於 2014/12/8 正式成為開發 Android 應用程式的官方 IDE（Integrated Development Environment）工具，Android 開發官網也很明確指出原先 Eclipse with ADT 的開發模式已經被 Android Studio 取代，建議開發者盡早改用 Android Studio 以獲得最好的支援。

用過 Eclipse with ADT 的開發者都知道一旦匯入 Android 專案，不明原因的編譯失敗常常伴隨發生，而且往往需要重啟 Eclipse 或用一些很奇怪的方式才能解決。這些問題在 Android Studio 上完全不見了，而且 UI 設計工具比以

前更聰明，XML 原始碼與 UI 畫面可以同時顯示；許多貼心的提示功能可以讓開發者避免一些不必要的錯誤發生。使用 Android Studio 更可以讓初學者很容易就進入到 Android App 開發世界。本書所有範例都已經更新成 Android Studio 版本。

CardView 與 RecyclerView 是 Android 5.0 時發表的 2 個新 UI 元件，讓 UI 呈現更多樣化。因為屬於 support 函式庫，所以舊版的 Android 裝置也可以呈現這 2 個 UI 元件，「5-5 CardView 與 RecyclerView」會詳加說明如何使用它們。

除了前述更新的部分外，藉此機會將範例內容調整為更符合對應的主題，讓讀者可以透過主題與範例緊密結合地說明，更清楚了解 Android App 的開發技巧。除此之外還增加一些新的主題使原本內容更加充實，例如說明當裝置處於休眠狀態時仍舊可以執行 Service 的技巧，如何在裝置一開機時就啟動 App，以及陀螺儀感應器的應用。

最後特別要感謝碁峰資訊編輯團隊的積極協助，讓這本書能夠順利出版。

黃彬華

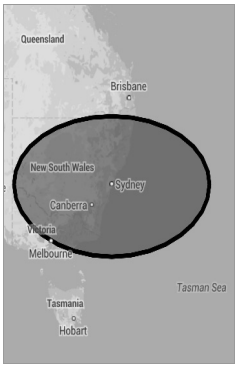
Chapter 10

Google 地圖

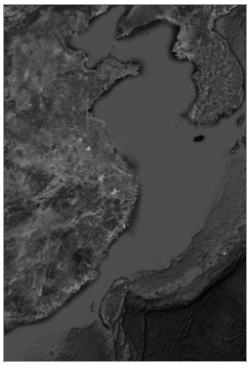
10-1 Google 地圖功能介紹

欲建立帶有 Google 地圖功能的 Android 應用程式，必須使用 Google Maps Android API（以下簡稱為 Maps API），目前版本為第 2 版（全名為 Google Maps Android API v2）。透過 Maps API 可以存取 Google 地圖伺服器上的資料用以呈現在 Android 裝置上；除此之外，Maps API 還可以協助開發者在 Google 地圖上加標記（markers）、繪製線條與多邊形（polygons），以及加上圖層（overlays）以達到更豐富的圖資呈現；甚至使用者可以改變地圖呈現方式（例如：地圖傾斜度、切換交通圖與衛星圖等），如圖 10-1，並可與地圖做多樣的互動。

加標記



多邊形



衛星圖

圖 10-1

想要在 Android 應用程式加上 Google 地圖功能，可以按照下列步驟；而各步驟的詳細說明則可繼續參看之後接續的各節。

1. 產生數位憑證指紋
2. 申請 Google Maps Android API Key（以下簡稱 API 金鑰）
3. Google Play services 安裝與匯入
4. 建立基本 Google 地圖

10-2 產生數位憑證指紋

Android 應用程式如果需要呼叫 Maps API，就必須使用 API 金鑰。欲取得 API 金鑰的必須提供：

- 數位憑證指紋（certificate fingerprint）
- Android 應用程式套件名稱。

以下說明產生除錯憑證[1]指紋的步驟：

1 STEP 找到除錯憑證指紋的位置：

- Windows 7－C:\Users\[使用者名稱]\.android\
- MacOS 或 Linux－[使用者名稱]/.android/

可以找到預設 debug.keystore 檔案[2]位置，如圖 10-2，而 debug.keystore 檔案內儲存的就是除錯憑證。

---

1 應用程式的數位憑證有 2 種，開發階段使用除錯憑證（debug certificate），又稱作除錯金鑰（debug key）；如果要將應用程式發佈到 Play 商店，就必須使用發佈憑證（release certificate），又稱為發佈金鑰（publish key）。因為目前還在開發階段，所以本章說明以除錯憑證為主；發佈憑證則在第 14 章說明。

2 執行一次 Android 應用程式，Android SDK 工具就會自動產生 debug.keystore 檔案並放在預設位置，並且用 debug.keystore 來簽署應用程式。debug.keystore 是金鑰庫，其內儲存著金鑰。

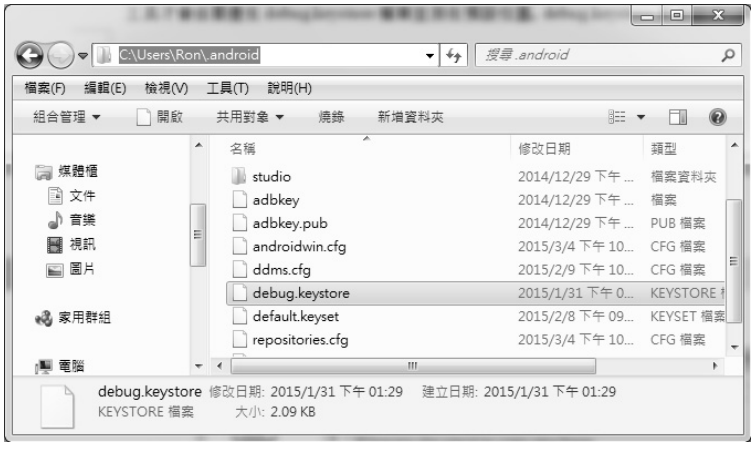


圖 10-2

2 STEP 將 JDK 的 keytool 工具（keytool.exe）之路徑加入 path 環境變數[3]，如圖 10-3。

圖 10-3

---

3 Path 環境變數設定說明如下：Windows 7 系統：控制台 > 系統 > 進階系統設定 > 進階 > 視窗下半部系統變數找到 Path 後按下「編輯」按鈕 > 在「變數值」欄位的最後輸入「;」（前後不可有任何空白字元），接著將「keytool.exe」所在目錄路徑－例如「C:\Program Files\Java\jdk1.7.0_05\bin」輸入。

3 STEP 請依照 debug.keystore 檔案（假設在 C:\Users\RON\.android 目錄）所在路徑下指令：

```
C:\Users\RON\.android>keytool -list -keystore debug.keystore
```

4 STEP 接著會要求輸入 keystore 密碼，預設為「android」，也可以不輸入直接按 Enter 按鈕，最後會產生 SHA1 憑證指紋如下所示，下個階段就會利用 SHA1 憑證指紋去申請 API 金鑰。

```
C:\Users\Ron\.android>keytool -list -keystore debug.keystore
輸入金鑰儲存庫密碼: （預設為 android）
金鑰儲存庫類型: JKS
金鑰儲存庫提供者: SUN
您的金鑰儲存庫包含 1 項目
androiddebugkey, 2012/6/19, PrivateKeyEntry,
憑證指紋 (SHA1): 93:99:DF:BA:6A:13:71:ED:03:75:AD:96:E9:46:5A:D0:31:C9:43:E4
```

10-3 申請 API 金鑰

要申請 API 金鑰，必須透過 Google APIs Console 管理介面，申請步驟如下：

1 STEP 開啟瀏覽器（建議使用 Google Chrome 瀏覽器），網址列輸入「https://code.google.com/apis/console/」，接下來輸入 Google 帳號密碼，如圖 10-4。

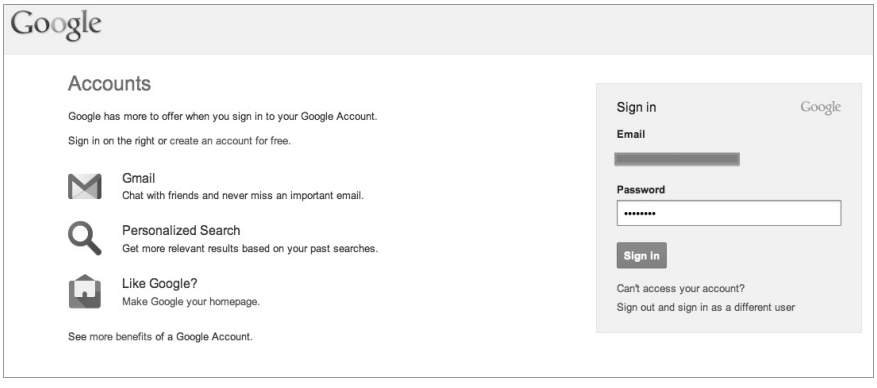


圖 10-4

2 STEP 接著會跳出如圖 10-5 畫面，按下「Create project」按鈕繼續。

圖 10-5

3 STEP 點擊左邊導覽列的「API 和驗證」>「API」，在右邊點擊「Google Maps Android API」，如圖 10-6；接受條款後即可啟動該服務。

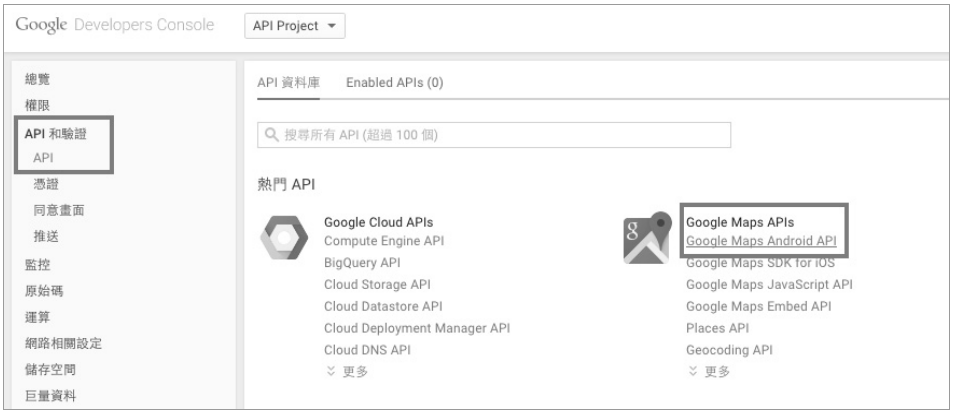


圖 10-6

4 STEP 點擊左邊導覽列「憑證」選項，並按下「建立新的金鑰」>「Android 金鑰」，如圖 10-7。

圖 10-7

5 STEP 申請 API 金鑰需要「憑證指紋」與「應用程式套件名稱」。請依照圖 10-8 所示，在文字輸入方塊中先貼上憑證指紋，加上分號，再貼上套件名稱，格式為「憑證指紋;套件名稱」(分號作為分隔)，然後按下「建立」按鈕。

圖 10-8

6 STEP 最後會產生 API 金鑰，如圖 10-9。之後建立 Android 專案時，manifest 檔案內的套件名稱要與申請 API 金鑰時所輸入的套件名稱相同；<meta-data>也必須加上申請到的 API 金鑰，如圖 10-10。

圖 10-9

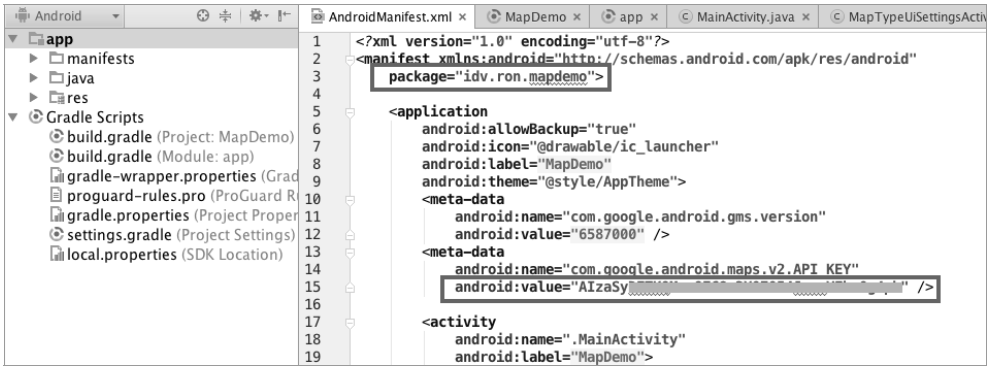


圖 10-10

10-4 Google Play Services 安裝與匯入

Maps API 現在已經成為 Google Play Services API 一部分，所以要使用 Google 地圖相關功能，必須先安裝 Google Play Services API 並匯入到專案內，方便專案程式引用。一般而言，安裝好 Android Studio 就會連同 Google Play Services API 一起安裝好。安裝與匯入步驟為：

1 STEP 安裝 Google Play Services API：點選 Android Studio 主選單 Tools > Android > SDK Manager 會開啟 Android SDK Manager，如圖 10-11；勾選「Google Repository」[4]按下右下角「Install packages」按鈕即可開始安裝。

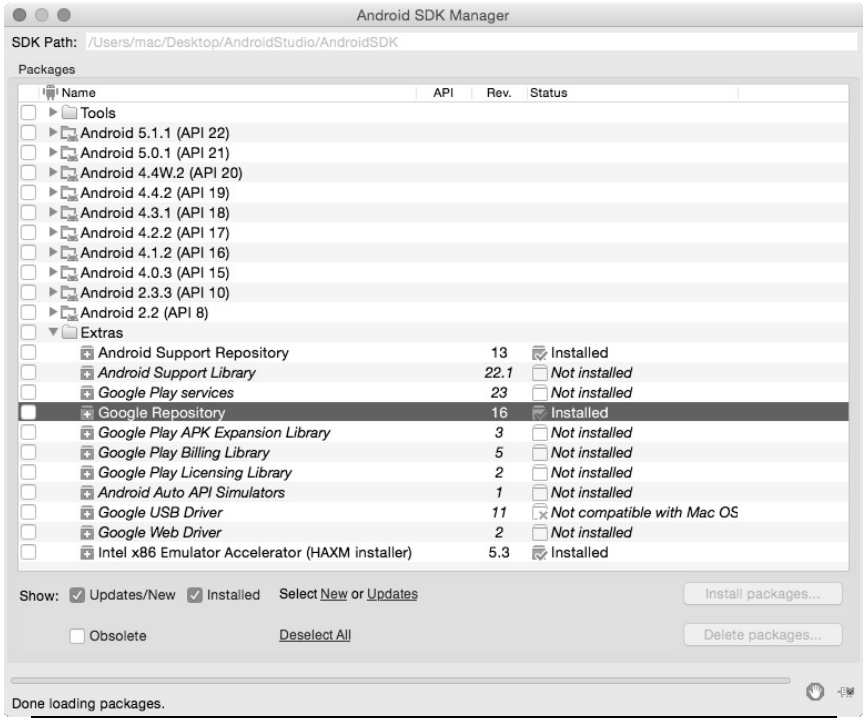


圖 10-11

2 STEP 專案匯入 Google Play services API：Android Studio 主選單 File > Project Structure > 左邊導覽列 app > Dependencies 會顯示出已經加入的 API。如果沒有加入 Google Play Services API，可以按下「+」按鈕 > Library dependency > 選擇 play-services (com.google.android.gms:play-services)。

[4] 圖 10-11 同樣在 Extras 目錄的「Google Play services」項目是給 Eclipse 開發工具安裝，而非給 Android Studio 安裝。

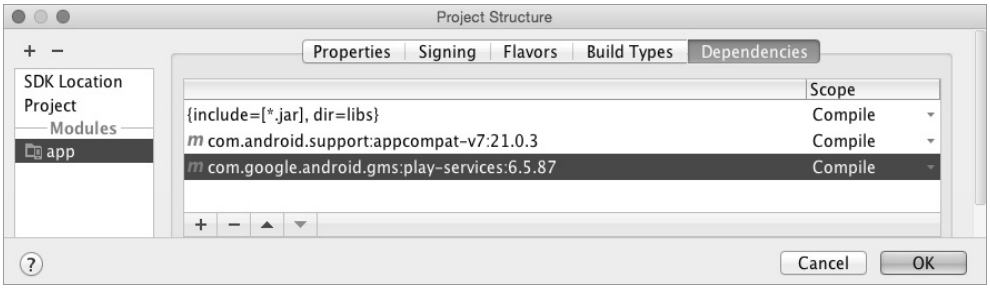


圖 10-12

加入完畢後也可以在專案導覽列的 Gradle Scripts > build.gradle (Module: app)看到對應加入的內容，如圖 10-13。

圖 10-13

3 STEP 新增 meta-data 標籤。要正常使用 Google Play Services API，必須加上 android:name="com.google.android.gms.version" 與 android:value="@integer/google_play_services_version"這 2 個屬性。

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

10-5 建立基本 Google 地圖

想要建立具備基本 Google 地圖的 Android 應用程式，必須先在 manifest 檔案內作權限設定並加入 API 金鑰，layout 檔案也要加入 SupportMapFragment 以呈現地圖，可參看下面範例步驟。Android 官方模擬器[5]與實機都支援 Google 地圖，所以可以執行具有 Google 地圖功能的應用程式。

範例 MapDemo/BasicMapActivity

圖 10-14

[5] Genymotion 模擬器沒有支援 Google 地圖功能，需要安裝 ARM Translation Installer 與 Google Apps for Android 二個 zip 檔案，步驟如下：（1）搜尋並下載 ARM Translation Installer 最新版，然後將檔案拖曳到模擬器安裝後重啟（2）搜尋並下載 Google Apps for Android（最好符合模擬器的 Android 版本），然後將檔案拖曳到模擬器安裝後重啟（3）許多 Google 的應用程式會更新，可能有應用程式會跳出錯誤訊息，不用理會（4）此時可以進入 Play 商店，下載 Google 地圖應用程式以做測試。

範例說明：

- 呈現最簡單的 Google 地圖。

建立步驟：

1 STEP 套件名稱必須與申請 API 金鑰時所用的套件名稱相同。使用 Google 地圖需要加上許多權限設定。需要填入申請的 API 金鑰。

MapDemo > manifests > AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- 套件名稱必須與申請 API 金鑰時所用的套件名稱相同 -->
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="idv.ron.mapdemo">

<!-- 允許應用程式透過 internet 下載地圖資訊 -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<!-- 允許應用程式將地圖資訊暫存到 Android 裝置的外部儲存體 -->
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<!-- 允許應用程式透過 WiFi 或行動網路來定位[6] -->
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<!-- 允許應用程式透過 GPS 來定位 -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

<!-- Google 地圖需要使用 OpenGL ES v2 版本，加上<uses-feature>設定可以讓已經放在 Play 商店的此
應用程式不會被不支援此功能的裝置看到，具有過濾效果 -->
    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">
```

6 其實 Google 地圖應用程式可以不用加上 android.permission.ACCESS_COARSE_LOCATION 與 android.permission.ACCESS_FINE_LOCATION 這 2 個權限設定，但官方文件仍建議加上。

```
        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />
<!-- 填入申請的API 金鑰 -->
        <meta-data
            android:name="com.google.android.maps.v2.API_KEY"
            android:value="申請的API 金鑰" />

        <activity
            android:name=".MainActivity"
            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=".BasicMapActivity"
            android:label="@string/title_BasicMap" />
        <activity
            android:name=".MapTypeUiSettingsActivity"
            android:label="@string/title_MapTypeUiSettings" />
        <activity
            android:name=".MarkersActivity"
            android:label="@string/title_Markers" />
        <activity
            android:name=".PolylinesPolygonsActivity"
            android:label="@string/title_PolylinesPolygons" />
        <activity
            android:name=".GeocoderActivity"
            android:label="@string/title_Geocoder" />
    </application>

</manifest>
```

2 STEP 建立帶有 Google 地圖的 layout 檔案：加上<fragment>標籤，如果 class 屬性是 SupportMapFragment，可以支援舊版本的 Android 系統。如果改用 MapFragment，僅支援 API 12（Android 3.1）以後的 Android 系統。

MapDemo > res > layout > basic_map_activity.xml

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/fmMap"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    class="com.google.android.gms.maps.SupportMapFragment" />
```

3 STEP 建立 Activity，將帶有 Google 地圖的 layout 檔案載入。

MapDemo > java > BasicMapActivity.java

```
public class BasicMapActivity extends ActionBarActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.basic_map_activity);
    }
}
```

10-6 地圖種類與 UI 設定

10-6-1 地圖種類設定

在 Google 地圖上可以設定下列 4 種地圖種類，參看圖 10-15：

- 一般圖（normal）：典型的道路地圖，重要的人造設施或天然景觀如河流、湖泊的形狀會顯示；另外重要道路與重要設施的名稱也會以文字顯示。
- 混合圖（hybrid）：就是一般圖與衛星圖的混合體。
- 衛星圖（satellite）：衛星所拍攝的空照圖，但不會以文字顯示道路或設施的名稱。
- 地形圖（terrain）：顯示地形走勢，也會以文字顯示一些道路或設施名稱。

一般圖

衛星圖

地形圖

圖 10-15

要設定地圖種類可以呼叫 GoogleMap.setMapType(int) 方法並搭配 GoogleMap 的地圖種類常數來達成，例如：

```
GoogleMap map;
...
// 將地圖種類設定為混合圖
map.setMapType(GoogleMap.MAP_TYPE_HYBRID);
```

10-6-2 地圖 UI 設定

Google 地圖可以顯示交通資訊（traffic）與自己位置圖層[7]（my location layer），只要透過呼叫 GoogleMap.setMyLocationEnabled(boolean) 與 GoogleMap.setMyLocationEnabled(boolean)方法即可達成，例如：

```
GoogleMap map;
...
// 顯示交通資訊
map.setTrafficEnabled(true);
// 顯示自己位置
map.setMyLocationEnabled(true);
```

[7] Google 地圖上可以新增許多不同的圖層，例如自己位置圖層、標記圖層等以方便管理；如果使用者不想看到某一圖層資訊，只要移除該圖層即可。

另外還有 UiSettings 儲存著地圖 UI 設定與操作手勢設定，可以呼叫 getUiSettings()取得該物件，並呼叫 setter 方法加以設定：

```
GoogleMap map;
...
// 取得地圖 UI 設定物件
UiSettings uiSettings = map.getUiSettings();

// 顯示縮放按鈕
uiSettings.setZoomControlsEnabled(true)
// 顯示指北針
uiSettings.setCompassEnabled(true)
// 顯示自己位置按鈕
uiSettings.setMyLocationButtonEnabled(true)

// 開啟地圖捲動手勢
uiSettings.setScrollGesturesEnabled(true)
// 開啟地圖縮放手勢
uiSettings.setZoomGesturesEnabled(true)
// 開啟地圖傾斜手勢
uiSettings.setTiltGesturesEnabled(true)
// 開啟地圖旋轉手勢
uiSettings.setRotateGesturesEnabled(true)
```

範例 MapDemo/MapTypeUiSettingsActivity

圖 10-16

範例說明：

- 點擊右上角自己位置按鈕可以將裝置本身位置[8]顯示在地圖畫面中央。
- 點擊右下角縮放按鈕可以縮放地圖。
- 點擊左下角下拉選單可以改變地圖種類；勾選下方選項可以開啟/關閉個別 UI 顯示或操作功能。

建立步驟：

1 STEP layout 檔案建立帶有 Google 地圖的 SupportMapFragment，並建立 Spinner 與 CheckBox 讓使用者可以做地圖上的 UI 設定。

MapDemo > res > layout > map_type_ui_settings_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <fragment
        android:id="@+id/fmMap"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        class="com.google.android.gms.maps.SupportMapFragment" />

    <ScrollView
        android:layout_width="wrap_content"
        android:layout_height="175dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:background="#FFFFFF"
```

[8] 模擬器必須先設定裝置本身的位置，點擊自己位置按鈕才會有效果。

```
    android:orientation="vertical"
    android:paddingRight="5dp">

    <Spinner
        android:id="@+id/spMapType"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <CheckBox
        android:id="@+id/cbTraffic"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:onClick="onTrafficClick"
        android:text="@string/text_cbTraffic" />

    <CheckBox
        android:id="@+id/cbMyLocationButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:checked="true"
        android:onClick="onMyLocationButtonClick"
        android:text="@string/text_cbMyLocationButton" />

    <CheckBox
        android:id="@+id/cbMyLocationLayer"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:checked="true"
        android:onClick="onMyLocationLayerClick"
        android:text="@string/text_cbMyLocationLayer" />

    <CheckBox
        android:id="@+id/cbZoomControls"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:checked="true"
        android:onClick="onZoomControlsClick"
```

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

    <CheckBox
        android:id="@+id/cbCompass"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:checked="true"
        android:onClick="onCompassClick"
        android:text="@string/text_cbCompass" />

    <CheckBox
        android:id="@+id/cbScrollGestures"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:checked="true"
        android:onClick="onScrollGesturesClick"
        android:text="@string/text_cbScrollGestures" />

    <CheckBox
        android:id="@+id/cbZoomGestures"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:checked="true"
        android:onClick="onZoomGesturesClick"
        android:text="@string/text_cbZoomGestures" />

    <CheckBox
        android:id="@+id/cbRotateGestures"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:checked="true"
        android:onClick="onRotateGesturesClick"
        android:text="@string/text_cbRotateGestures" />

    <CheckBox
        android:id="@+id/cbTiltGestures"
        android:layout_width="wrap_content"
```

```
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:checked="true"
                android:onClick="onTiltGesturesClick"
                android:text="@string/text_cbTiltGestures" />
        </LinearLayout>
    </ScrollView>

</RelativeLayout>
```

2 STEP 建立 Activity 載入 Google 地圖，並處理使用者對地圖所做的 UI 相關設定。

MapDemo > java > MapTypeUiSettingsActivity.java

```
public class MapTypeUiSettingsActivity extends ActionBarActivity {
    private final static String TAG = "MapTypeUiSettings";

    private GoogleMap map; // 儲存著地圖資訊
    private UiSettings uiSettings; // 儲存著地圖 UI 設定

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.map_type_ui_settings_activity);
        initMap();
        setMyMapType();
    }

    // 初始化地圖
    private void initMap() {
        // 檢查 GoogleMap 物件是否存在
        if (map == null) {
            // 從 SupportMapFragment 取得 GoogleMap 物件
            map = ((SupportMapFragment) getSupportFragmentManager()
                    .findFragmentById(R.id.fmMap)).getMap();
            if (map != null) {
                setUpMap();
            }
        }
```