

Android程式設計

定位服務

王昱景 Brian Wang

brian.wang.frontline@gmail.com

- Android行動裝置結合定位功能和Google地圖建立的“**位置感知服務**”(Location-based Service, **LBS**)
- LBS應用程式可以追蹤你的位置 and 提供一些額外服務，例如：找出附近的咖啡廳、停車場、自動櫃員機或加油站等
- LBS另一項常見的應用是路徑規劃的導航，除了行車導航外，對於大型展覽館、購物商場、城市觀光、野生動物園或主題樂園等，更可以提供導覽服務，結合定位功能來提供使用者更精確的位置資訊

- Android作業系統提供**LocationManager**類別的定位服務來幫助我們存取行動裝置目前的定位資料
- 包含：**緯度**(Latitude)、**經度**(Longitude)和高度(Altitude)等

Android提供的定位提供者

- Android作業系統的**定位提供者**(Provider)可以提供不同方式的定位服務
- **GPS定位提供者**：提供者名稱字串為”**gps**”，它是使用GPS(Global Positioning System)的衛星訊號來定位，可以提供精確的位置資訊，但是無法收到衛星訊號的室內並無法使用
- **網路定位提供者**：提供者名稱字串為”**network**”，它是直接使用電信公司基地台來執行**三角定位**，其提供的位置資訊較不精確，但是可以在室內使用

- 行動裝置並不一定支援GPS定位提供者
- 如果沒有電話或3G連網功能，就不會支援網路定位提供者
- 行動裝置是否支援定位服務需視硬體配備而定

經緯度座標

- 定位服務最主要的目的是找出行動裝置目前位置的經緯度座標
- 經緯度是經度與緯度合稱的座標系統，也稱為地理座標系統
- 使用三度空間的球面來定義地球表面各點的座標系統，能夠標示地球表面上的任何一個位置

- **緯度**：地球表面某一點距離地球**赤道**以南或以北的度數，其值為**0至90度**，赤道以北的緯度叫**北緯**(符號為**N**)；赤道以南的緯度稱為**南緯**(符號為**S**)
- **經度**：地球表面上某一點距離**本初子午線**(一條南北方向經過**倫敦格林威治天文台**舊址的子午線)以東或以西的度數，本初子午線的經度是0度，其他地點的經度是向東從0到180度，即**東經**(符號為**E**)或向西從0到180度，即**西經**(符號為**W**)

- 一般來說，在地球儀或地圖上描述經緯度座標是使用度(Degrees)、分(Minutes)和秒(Seconds)
- 例如：舊金山金門大橋的經緯度為 122°29' W, 37°49' N
- 上述經緯度是西經122度29分；北緯37度49分

- 每一度可以再分成60單位的分，分可以再細分60單位的秒
- 在電腦上表示經緯度通常是使用十進位方式表示
- N和E為正值；S和W為負值，分為小數點下2位，秒是之後2位
- 以上述經緯度為例，十進位表示法的經緯度為-122.29, 37.49

- 我在哪裡是定位服務的最簡單應用
- 可以顯示目前行動裝置的經緯度座標



11:49



我在哪裡

定位提供者(Provider): gps

緯度(Latitude): 25.046899999999997

經度(Longitude): 121.51609833333332

高度(Altitude): 0.0

顯示Google地圖



12:02



WhereAmIDemo

- 開啟和執行Android專案
- 建立我在哪裡使用介面的版面配置
- 建立Activity活動類別使用定位服務
- 在AndroidManifest.xml新增權限

I. 開啟和執行Android專案

- 請啟動 Eclipse IDE
- 建立 Android 專案
 - Project Name: **WhereAmIDemo**
 - Build Target: **Google API 4.2.2**
 - Package Name

2. 建立我在哪裡使用介面的版面配置

- 我在哪裡使用介面是定義在 `activity_main.xml` 版面配置檔
- 使用 `LinearLayout` 垂直編排 1 個 `TextView` 和 `Button` 元件

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/LinearLayout1"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical" >

  <TextView
    android:id="@+id/output"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

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

</LinearLayout>
```



3. 建立Activity活動類別使用定位服務

- 在活動類別的開頭宣告成員的
LocationManager和Location物件變數

```
public class MainActivity extends Activity {  
  
    private LocationManager manager;  
    private Location currentLocation;  
    private String best;  
  
}
```


onCreate() 方法

- 在覆寫的onCreate()方法載入版面配置後，可以取得系統服務的LocationManager物件，if條件檢查是否有啟用GPS
- if條件是呼叫LocationManager物件的isProviderEnabled()方法檢查是否有啟用GPS，如果沒有顯示對話方塊啟用GPS
- 如果使用者在對話方塊按啟用鈕，就建立Intent物件啟動設定程式來更改GPS設定

```

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

    // 取得系統服務的LocationManager物件
    manager = (LocationManager) getSystemService(LOCATION_SERVICE);

    // 檢查是否有啟用GPS
    if (!manager.isProviderEnabled(LocationManager.GPS_PROVIDER)) {

        // 顯示對話方塊啟用GPS
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("定位管理")
            .setMessage("GPS目前狀態是尚未啟用.\n" + "請問你是否現在就設定啟用GPS?")
            .setPositiveButton("啟用",
                new DialogInterface.OnClickListener() {

                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        // 使用Intent物件啟動設定程式來更改GPS設定
                        Intent i = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);

                        startActivity(i);
                    }

                })
            .setNegativeButton("不啟用", null).create().show();
    }

    Button btn = (Button) findViewById(R.id.btn);
    btn.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {
            // 取得經緯度座標
            float latitude = (float) currentLocation.getLatitude();
            float longitude = (float) currentLocation.getLongitude();

            // 建立URI字串
            String uri = String.format("geo:%f,%f?z=18", latitude, longitude);

            // 建立Intent物件
            Intent geoMap = new Intent(Intent.ACTION_VIEW, Uri.parse(uri));

            startActivity(geoMap); // 啟動活動
        }
    });
}

```

onResume()方法

- 在覆寫 onResume()方法建立Criteria物件設定如何選擇題供者，以便取得最佳或符合需求的定位提供者
- 呼叫getBestProvider()方法取得最佳提供者字串，參數是Criteria物件
- 更新位置頻率的條件，在if條件判斷是否有最佳提供者，如果有，就取得快取的最後位置，並且使用上述更新頻率，呼叫requestLocationUpdates()方法註冊傾聽者物件來周期性回報目前的位置

- if條件如果沒有最佳提供者，就使用GPS提供者，最後呼叫updatePosition()方法顯示更新位置
- requestLocationUpdates()方法有4個參數，第1個參數是定位提供者字串，第2個是參數是更新位置的間隔時間(以毫秒為單位)，第3個參數是更新位置的最短距離(以公尺為單位)，最後一個參數listener是LocationListener傾聽者物件


```
@Override
protected void onResume() {
    super.onResume();

    // 取得最佳的定位提供者
    Criteria criteria = new Criteria();
    best = manager.getBestProvider(criteria, true);

    // 更新位置頻率的條件
    int minTime = 5000; // 毫秒
    float minDistance = 5; // 公尺

    if (best != null) { // 取得快取的最後位置, 如果有的話
        currentLocation = manager.getLastKnownLocation(best);
        manager.requestLocationUpdates(best, minTime, minDistance, listener);
    } else { // 取得快取的最後位置, 如果有的話
        currentLocation = manager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        manager.requestLocationUpdates(LocationManager.GPS_PROVIDER, minTime, minDistance, listener);
    }
    updatePosition(); // 更新位置
}
```

onPause()方法

- 在覆寫onPause()方法呼叫
removeUpdates()方法取消周期更新位置

```
@Override  
protected void onPause() {  
    super.onPause();  
    manager.removeUpdates(listener);  
}
```

updatePosition() 方法

- 在自訂updatePosition()方法更新TextView元件顯示的位置資訊
- if條件判斷Location物件currentLocation，如果不是null，就呼叫getLocationInfo()方法顯示位置資訊

```
// 更新現在的位置
private void updatePosition() {
    TextView output = (TextView) findViewById(R.id.output);

    if (currentLocation == null) {
        output.setText("取得定位資訊中...");
    } else {
        output.setText(getLocationInfo(currentLocation));
    }
}
```


LocationListener傾聽者物件

- 在requestLocationUpdate()方法需要註冊定位服務的傾聽者物件
- 即使用匿名內層類別實作LocationListener介面來建立此物件，需要實作4個方法
- 4個方法只有使用onLocationChanged()方法，當位置更新時，呼叫updatePosition()方法來更新顯示的位置資訊

// 建立定位服務的傾聽者物件

```
private LocationListener listener = new LocationListener() {
```

```
    @Override
```

```
    public void onLocationChanged(Location location) {
```

```
        currentLocation = location;
```

```
        updatePosition();
```

```
    }
```

```
    @Override
```

```
    public void onProviderDisabled(String provider) {
```

```
    }
```

```
    @Override
```

```
    public void onProviderEnabled(String provider) {
```

```
    }
```

```
    @Override
```

```
    public void onStatusChanged(String provider, int status, Bundle extras) {
```

```
    }
```

```
};
```

getLocationInfo()方法

- 在自訂的getLocationInfo()方法可以從參數Location物件取得定位資訊
- 使用StringBuffer物件建立位置資訊，append()方法勢將參數字串新增至StringBuffer物件的最後
- 呼叫Location物件的getLatitude()方法取得緯度；getLongitude()方法取得經度；getAltitude()方法取得高度
- 最後使用toString()方法將StringBuffer物件轉換成字串傳回

// 取得定位資訊

```
private String getLocationInfo(Location location) {  
    StringBuffer str = new StringBuffer();  
    str.append("定位提供者(Provider): " + location.getProvider());  
    str.append("\n緯度(Latitude): " + Double.toString(location.getLatitude()));  
    str.append("\n經度(Longitude): " + Double.toString(location.getLongitude()));  
    str.append("\n高度(Altitude): " + Double.toString(location.getAltitude()));  
    return str.toString();  
}
```


4. 在AndroidManifest.xml新增權限

- 因為需要使用定位服務，所以在AndroidManifest.xml檔案需要新增2個權限
- ACCESS_COARSE_LOCATION是網路定位服務
- ACCESS_FINE_LOCATION是GPS定位服務

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