

# Android



## 第 10 章

# 創意 Android 程式設計



Linux公社（LinuxIDC.com）于2006年9月25日注册并开通网站，Linux现在已经成为一种广受关注和支持的一种操作系统，IDC是互联网数据中心，LinuxIDC就是关于Linux的数据中心。

LinuxIDC.com提供包括Ubuntu，Fedora，SUSE技术，以及最新IT资讯等Linux专业类网站。

并被收录到Google 网页目录-计算机 > 软件 > 操作系统 > Linux 目录下。

Linux公社（LinuxIDC.com）设置了有一定影响力的Linux专题栏目。

包括：

[Ubuntu专题](#)

[Fedora专题](#)

[RedHat专题](#)

[SUSE专题](#)

[红旗Linux专题](#)

[Android专题](#)

---

[Linux公社简介](#) - [广告服务](#) - [网站地图](#) - [帮助信息](#) - [联系我们](#)

本站（LinuxIDC）所刊载文章不代表同意其说法或描述，仅为提供更多信息，也不构成任何建议。

本站带宽由[\[6688.CC\]](#)友情提供

Copyright © 2006-2011 [Linux公社](#) All rights reserved

## 10-1

## 手機手電筒

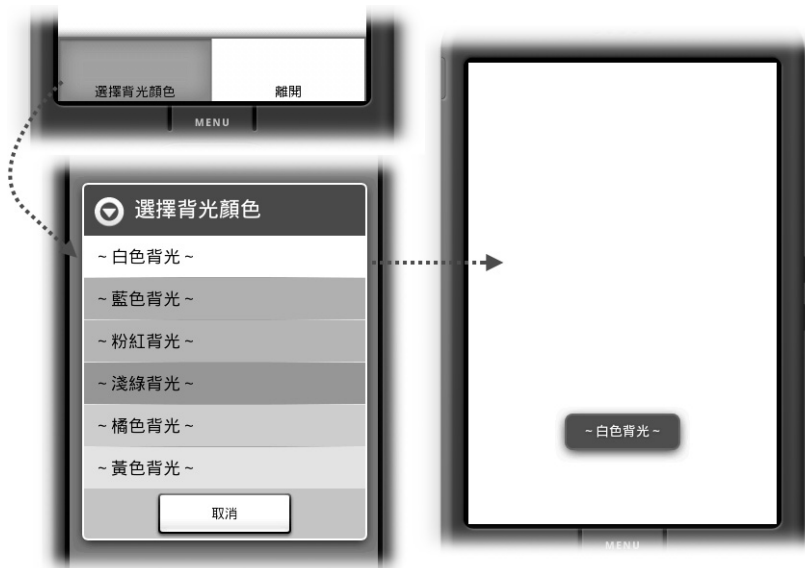
— PowerManager 控制 WakeLock 並改變手機亮度

### ▶ 範例說明

手電筒是每個家庭必備的用品，雖然平時用不到，但是如果突然停電，身邊又沒有手電筒，就只能在黑暗中度過。但是出門在外時，一般人並不會隨身攜帶手電筒，碰到臨時狀況需要照明時，如果手機也能當作手電筒使用，那就太棒了！

Android API 中的 PowerManager (android.os.PowerManager) 可以控制手機的 WakeLock (android.os.PowerManager.WakeLock)，讓手機螢幕保持在恆亮的狀態，再透過程式將手機亮度調整到最亮 (255)，手機就跟手電筒一樣可以拿來照明了！本範例將實作手機手電筒的功能，並可透過選單的選擇來改變手機背光顏色。

### ▶ 執行結果



▲ 圖 10-1 可選擇背光顏色的手機手電筒

## ► 範例程式

### src/irdc.ex10\_01/EX10\_01.java

主程式中在 `setContentView()` 之前先將螢幕設定為全螢幕顯示，以 `PowerManager.newWakeLock()` 取得 `WakeLock` 物件，並記錄 `Activity` 啟動前的螢幕亮度；當 `Activity` 啟動時，`onResume()` 會被呼叫，執行 `wakeLock()` 這個方法，將螢幕亮度設為 255，並將 `WakeLock` 喚起。當 `Activity` 被終止或暫停時，`onPause()` 會被呼叫，執行 `wakeUnlock()`，將螢幕亮度重新設為程式啟動前的亮度，並將 `WakeLock` 釋放。

程式中實作了兩個 `Menu` 按鍵，按下任一按鍵時，`onOptionsItemSelected()` 會被執行，如果按下的是「選擇背光顏色」的 `Menu`，會跳出選擇顏色的 `AlertDialog`。選完顏色後，則以 `LinearLayout.setBackgroundResource()` 來設定 `Layout` 的背景顏色；如果按下的是「離開」，則終止 `Activity` 的運行。

`setBrightness()` 為自訂的方法，可以調整螢幕的亮度，其中使用的 `IHardwareService` 類別則為自行 `import` 的 `jar` 檔內提供的類別。

```
package irdc.ex10_01;

/* import 相關 class */
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.os.Bundle;
import android.os.PowerManager;
import android.os.ServiceManager;
import android.provider.Settings;
import android.view.Menu;
import android.view.MenuItem;
import android.view.Window;
import android.view.WindowManager;
import android.widget.LinearLayout;
import android.widget.Toast;
import android.os.IHardwareService;

public class EX10_01 extends Activity
{
    private boolean ifLocked = false;
    private PowerManager.WakeLock mWakeLock;
    private PowerManager mPowerManager;
    private LinearLayout mLinearLayout;
    /* 儲存程式啟動前的手機亮度 */
    private int mUserBrightness=0;
```

## Google Android SDK 開發範例大全

```

/* 獨一無二的 menu 選項 identifier，用以識別事件 */
static final private int M_CHOOSE = Menu.FIRST;
static final private int M_EXIT = Menu.FIRST+1;
/* 顏色選單的顏色與文字陣列 */
private int[] color={R.drawable.white,R.drawable.blue,
                    R.drawable.pink,R.drawable.green,
                    R.drawable.orange,R.drawable.yellow};
private int[] text={R.string.str_white,R.string.str_blue,
                   R.string.str_pink,R.string.str_green,
                   R.string.str_orange,R.string.str_yellow};

@Override
public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);

    /* 必須在 setContentView 之前呼叫全螢幕顯示 */
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    getWindow().setFlags
    (
        WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN
    );

    setContentView(R.layout.main);
    /* 初始化 mLinearLayout */
    mLinearLayout=(LinearLayout) findViewById(R.id.myLinearLayout1);
    /* 取得 PowerManager */
    mPowerManager = (PowerManager)
        getSystemService(Context.POWER_SERVICE);
    /* 取得 WakeLock */
    mWakeLock = mPowerManager.newWakeLock
    (
        PowerManager.SCREEN_BRIGHT_WAKE_LOCK, "BackLight"
    );

    try
    {
        /* 取得程式啟動時的手機設定亮度 */
        mUserBrightness = Settings.System.getInt(getContentResolver(),
            Settings.System.SCREEN_BRIGHTNESS);
    }
    catch (Exception e)
    {
        Toast.makeText(EX10_01.this, ""+e, Toast.LENGTH_LONG).show();
        e.printStackTrace();
    }
}

```

```
@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    /* menu 群組 ID */
    int idGroup1 = 0;
    /* menuItemID */
    int orderMenuItem1 = Menu.NONE;
    int orderMenuItem2 = Menu.NONE+1;
    /* 建立 menu */
    menu.add(idGroup1,M_CHOOSE,orderMenuItem1,R.string.str_title);
    menu.add(idGroup1,M_EXIT,orderMenuItem2,R.string.str_exit);
    menu.setGroupCheckable(idGroup1, true, true);

    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    switch(item.getItemId())
    {
        case (M_CHOOSE):
            /* 跳出選擇背光顏色的 AlertDialog */
            new AlertDialog.Builder(EX10_01.this)
                .setTitle(getResources().getString(R.string.str_title))
                .setAdapter(new MyAdapter(this,color,text),listener1)
                .setPositiveButton("取消",
                    new DialogInterface.OnClickListener()
                    {
                        public void onClick(DialogInterface dialog, int which)
                        {
                        }
                    })
                .show();
            break;
        case (M_EXIT):
            /* 離開程式 */
            this.finish();
            break;
    }
    return super.onOptionsItemSelected(item);
}

/* 選擇背光顏色的 AlertDialog 的 OnClickListener */
OnClickListener listener1=new DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog,int which)
    {
        /* 更改背景顏色 */
        mLinearLayout.setBackgroundResource(color[which]);
    }
}
```

```

        /* 以 Toast 顯示設定的顏色 */
        Toast.makeText(EX10_01.this,
                        getResources().getString(text[which]),
                        Toast.LENGTH_LONG).show();
    }
};

@Override
protected void onResume()
{
    /* onResume()時呼叫 wakeLock() */
    wakeLock();
    super.onResume();
}

@Override
protected void onPause()
{
    /* onPause()時呼叫 wakeUnlock() */
    wakeUnlock();
    super.onPause();
}

/* 喚起 WakeLock 的方法 */
private void wakeLock()
{
    if (!ifLocked)
    {
        setBrightness(255);
        ifLocked = true;
        mWakeLock.acquire();
    }
    setBrightness(255);
}

/* 釋放 WakeLock 的方法 */
private void wakeUnlock()
{
    if (ifLocked)
    {
        mWakeLock.release();
        ifLocked = false;
        setBrightness(mUserBrightness);
    }
}

/* 設定手機亮度的方法 */
private void setBrightness(int brightness)
{
    /* 取得 IHardwareService */

```

```

    IHardwareService hardware = IHardwareService.Stub.asInterface(
        ServiceManager.getService("hardware"));
    if (hardware != null)
    {
        try
        {
            /* 設定亮度 */
            Settings.System.putInt(getContentResolver(),
                Settings.System.SCREEN_BRIGHTNESS,
                brightness);
            hardware.setScreenBacklight(brightness);
        }
        catch (Exception e)
        {
            Toast.makeText(EX10_01.this, ""+e, Toast.LENGTH_LONG).show();
            e.printStackTrace();
        }
    }
}
}

```

### src/irdc.ex10\_01/MyAdapter.java

背光顏色選單所使用的 Adapter，繼承 android.widget.BaseAdapter，使用 change\_color.xml 作為 Layout。

```

package irdc.ex10_01;

/* import 相關 class */
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

/* 自定義的 Adapter，繼承 android.widget.BaseAdapter */
public class MyAdapter extends BaseAdapter
{
    private LayoutInflater mInflater;
    private int[] color;
    private int[] text;

    public MyAdapter(Context context, int[] _color, int[] _text)
    {
        mInflater = LayoutInflater.from(context);
        color = _color;
        text = _text;
    }
}

```



```
/* 因繼承 BaseAdapter，需覆寫以下方法 */
@Override
public int getCount()
{
    return text.length;
}

@Override
public Object getItem(int position)
{
    return text[position];
}

@Override
public long getItemId(int position)
{
    return position;
}

@Override
public View getView(int position, View convertView, ViewGroup par)
{
    ViewHolder holder;

    if (convertView == null)
    {
        /* 使用自定義的 change_color 作為 Layout */
        convertView = mInflater.inflate(R.layout.change_color, null);
        /* 初始化 holder 的 text */
        holder = new ViewHolder();
        holder.mText = (TextView) convertView.findViewById(R.id.myText);
        convertView.setTag(holder);
    }
    else
    {
        holder = (ViewHolder) convertView.getTag();
    }
    holder.mText.setText(text[position]);
    holder.mText.setBackgroundResource(color[position]);

    return convertView;
}

/* class ViewHolder */
private class ViewHolder
{
    TextView mText;
}
}
```

### res/layout/changecolor.xml

選擇背景顏色的選單中，每一個選項所使用的 Layout，定義了顯示顏色名稱的 TextView。

```
<?xml version="1.0" encoding="utf-8"?>
<TextView android:id="@+id/myText"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_gravity="center_vertical"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="10px"
    android:textSize="18px"
    android:textColor="@drawable/black"
/>
```

### AndroidManifest.xml

在 AndroidManifest.xml 裡，必須要有 android.permission.WAKE\_LOCK（喚起 WakeLock）、android.permission.WRITE\_SETTINGS（寫入設定值）與 android.permission.HARDWARE\_TEST（硬體測試）三種權限。

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="irdc.ex10_01"
    android:versionCode="1"
    android:versionName="1.0.0">
    <application
        android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity
            android:name=".EX10_01"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.WRITE_SETTINGS"/>
    <uses-permission android:name="android.permission.HARDWARE_TEST"/>
</manifest>
```

## ► 延伸學習

在 Android API 1.0(1.0r1)之前的版本，都包含 `IHardwareService` (`android.os.IHardwareService`) 這個類別，可用來改變手機硬體的相關設定，不過，這個類別在 1.0 之後的版本被移除了，但有好心的國外網友將 Android API 0.9 版的相關類別整理成一個 jar 檔，範例中 import 了這個 jar，並使用內含的 `IHardwareService` 類別來實作調整螢幕亮度的功能，在 HTC G1 實機測試是可以正常執行的。

本範例中取得 `WakeLock` 時傳入 `PowerManager.SCREEN_BRIGHT_WAKE_LOCK` 這個 Flag，它能让手機 CPU 持續運作，讓螢幕保持恆亮；`WakeLock` 提供了四種 Flag 供開發者使用，且一次只能傳入一個 Flag：

Flag	CPU	Screen	Keyboard
<code>PARTIAL_WAKE_LOCK</code>	On	Off	Off
<code>SCREEN_DIM_WAKE_LOCK</code>	On	Dim	Off
<code>SCREEN_BRIGHT_WAKE_LOCK</code>	On	Bright	Off
<code>FULL_WAKE_LOCK</code>	On	Bright	Bright

使用 `WakeLock` 會十分耗電，所以要盡可能的減少使用 `WakeLock`，或是在使用時盡快的釋放 `WakeLock`，如搭配先前介紹的電池記量的偵測程式也是個不錯的方法，至於將 `WakeLock` 與 `Intent.ACTION_BATTERY_CHANGED` 混搭的應用，就留給你自行練習吧！

## 10-2 GPS 軌跡記錄器

— 利用 `LocationListener` 在地圖上畫圖並換算距離

### ► 範例說明

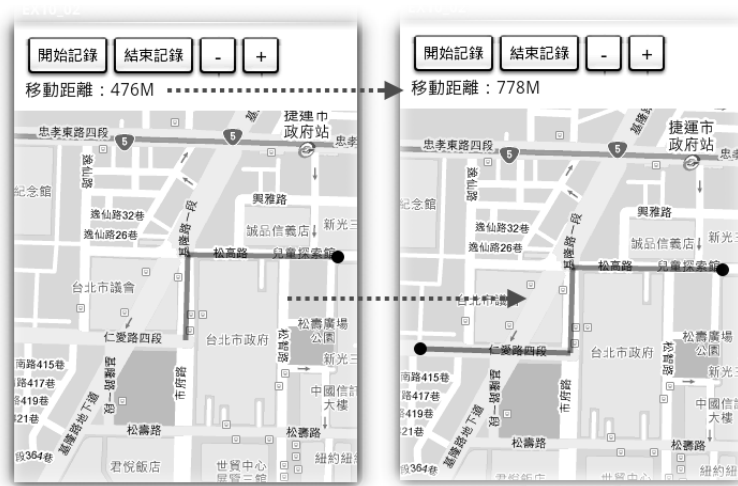
之前有一則新聞，一位竹科工程師為了贏得女友的芳心，花了八天的時間沿著中央山脈騎單車，行車的路線剛好串聯成女友的名字，讓她感動不已，點頭答應求婚，這位工程師就是使用 GPS 軌跡記錄器來記下這八天的行車路線。

在 Android 手機上要如何實作軌跡記錄器的功能呢？前面的範例曾經介紹過利用覆寫 `LocationListener` 的 `onLocationChanged()` 來追蹤目前所在的座標位置，只要延用這個方法，在偵測到 `Location` 座標變更時，利用變更前後 `GeoPoint` 座標的不同，在 `MapView` 上畫一條線，就變成了行車軌跡的記錄。

Android API 中的 Overlay (`com.google.android.maps.Overlay`) 類別可以在 `MapView` 上繪製圖形或加入圖片，本範例實作了繼承 `com.google.android.maps.Overlay` 的 `MyOverLay` 類別，並覆寫 `onDraw()` 這個方法，以達成在 `MapView` 上加入軌跡記錄的效果。

範例中設計了「開始記錄」與「結束記錄」的按鈕，當「開始記錄」按鈕被按下時，會先在起點處標註一個圓圈，隨著手機 GPS 座標的改變，會在 `MapView` 上繪製移動的軌跡，當「結束記錄」按鈕被按下時則終止記錄，並在終點處標記一個圓圈；範例中同時在座標改變時計算移動的距離，並即時顯示在手機螢幕上。

## ▶ 執行結果



▲ 圖 10-2 GPS 軌跡記錄並計算行走距離

## ▶ 範例程式

### src/irdc.ex10\_02/EX10\_02.java

主程式中自訂了幾個方法，功能都是透過自訂的 `MyOverLay` 類別在 `MapView` 裡畫上標記：`setStartPoint()` 會在「開始記錄」按鈕被按下時執行，在 `MapView` 裡畫下起點的圓形標記；`setEndPoint()` 會在「結束記錄」按鈕被按下時執行，在 `MapView` 裡畫下終點的圓形標記；`setRoute()` 則會在座標移動時被執行，利用執行前後兩個不同的座標點，在 `MapView` 裡畫出移動的軌跡。

程式中覆寫了 `LocationListener` 的 `onLocationChanged()`，當 `onLocationChanged()` 被觸發時，先取得目前的座標位置，除了透過 `setRoute()` 畫出移動軌跡外，並使用 `GetDistance(gp1, gp2)`

---

Google Android SDK 開發範例大全

---

來計算出移動的距離，並即時顯示在 `TextView` 中；`getDistance()` 計算移動距離的方式，是將地球視為一個圓形，運用地球的圓周來計算兩個座標點間的距離。

```
package irdc.ex10_02;

import java.text.DecimalFormat;
import java.text.NumberFormat;
import java.util.List;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;

public class EX10_02 extends MapActivity
{
    private TextView mTextView;
    private Button mButton01;
    private Button mButton02;
    private Button mButton03;
    private Button mButton04;
    private MapView mMapView;
    private MapController mMapController;
    private LocationManager mLocationManager;
    private Location mLocation;
    private String mLocationPrvider="";
    private int zoomLevel=0;
    private GeoPoint gp1;
    private GeoPoint gp2;
    private boolean _run=false;
    private double distance=0;

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

```
/* 建立 MapView 物件 */
mMapView = (MapView) findViewById(R.id.myMapView1);
mMapController = mMapView.getController();
/* 物件初始化 */
mTextView = (TextView) findViewById(R.id.myText1);
mButton01 = (Button) findViewById(R.id.myButton1);
mButton02 = (Button) findViewById(R.id.myButton2);
mButton03 = (Button) findViewById(R.id.myButton3);
mButton04 = (Button) findViewById(R.id.myButton4);
/* 設定預設的放大層級 */
zoomLevel = 17;
mMapController.setZoom(zoomLevel);

/* Provider 初始化 */
mLocationManager = (LocationManager)
    getSystemService(Context.LOCATION_SERVICE);
/* 取得 Provider 與 Location */
getLocationPrvider();
if(mLocation!=null)
{
    /* 取得目前的經緯度 */
    gp1=getGeoByLocation(mLocation);
    gp2=gp1;
    /* 將 MapView 的中點移至目前位置 */
    refreshMapView();
    /* 設定事件的 Listener */
    mLocationManager.requestLocationUpdates(mLocationPrvider,
        2000, 10, mLocationListener);
}
else
{
    new AlertDialog.Builder(EX10_02.this).setTitle("系統訊息")
        .setMessage(getResources().getString(R.string.str_message))
        .setNegativeButton("確定",new DialogInterface.OnClickListener()
        {
            public void onClick(DialogInterface dialog, int which)
            {
                EX10_02.this.finish();
            }
        })
        .show();
}

/* 開始記錄的 Button */
mButton01.setOnClickListener(new Button.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
```

```

        gp1=gp2;
        /* 清除 Overlay */
        resetOverlay();
        /* 畫起點 */
        setStartPoint();
        /* 更新 MapView */
        refreshMapView();
        /* 重設移動距離為 0，並更新 TextView */
        distance=0;
        mTextView.setText("移動距離：0M");
        /* 啟動畫路線的機制 */
        _run=true;
    }
});

/* 結束記錄的 Button */
mButton02.setOnClickListener(new Button.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        /* 畫終點 */
        setEndPoint();
        /* 更新 MapView */
        refreshMapView();
        /* 終止畫路線的機制 */
        _run=false;
    }
});

/* 縮小地圖的 Button */
mButton03.setOnClickListener(new Button.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        zoomLevel--;
        if(zoomLevel<1)
        {
            zoomLevel = 1;
        }
        mMapController.setZoom(zoomLevel);
    }
});

/* 放大地圖的 Button */
mButton04.setOnClickListener(new Button.OnClickListener()
{
    @Override

```

```
public void onClick(View v)
{
    zoomLevel++;
    if (zoomLevel > mMapView.getMaxZoomLevel())
    {
        zoomLevel = mMapView.getMaxZoomLevel();
    }
    mMapController.setZoom(zoomLevel);
}
});
}

/* MapView 的 Listener */
public final LocationListener mLocationListener =
    new LocationListener()
{
    @Override
    public void onLocationChanged(Location location)
    {
        /* 如果記錄進行中，就畫路線並更新移動距離 */
        if (_run)
        {
            /* 記下移動後的位置 */
            gp2=getGeoByLocation(location);
            /* 畫路線 */
            setRoute();
            /* 更新 MapView */
            refreshMapView();
            /* 取得移動距離 */
            distance+=GetDistance(gp1, gp2);
            mTextView.setText("移動距離：" + format(distance) + "M");

            gp1=gp2;
        }
    }

    @Override
    public void onProviderDisabled(String provider)
    {
    }
    @Override
    public void onProviderEnabled(String provider)
    {
    }
    @Override
    public void onStatusChanged(String provider, int status,
                                Bundle extras)
    {
    }
}
};
```



```
/* 取得 GeoPoint 的方法 */
private GeoPoint getGeoByLocation(Location location)
{
    GeoPoint gp = null;
    try
    {
        if (location != null)
        {
            double geoLatitude = location.getLatitude()*1E6;
            double geoLongitude = location.getLongitude()*1E6;
            gp = new GeoPoint((int) geoLatitude, (int) geoLongitude);
        }
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
    return gp;
}

/* 取得 LocationProvider */
public void getLocationPrvider()
{
    Criteria mCriteria01 = new Criteria();
    mCriteria01.setAccuracy(Criteria.ACCURACY_FINE);
    mCriteria01.setAltitudeRequired(false);
    mCriteria01.setBearingRequired(false);
    mCriteria01.setCostAllowed(true);
    mCriteria01.setPowerRequirement(Criteria.POWER_LOW);

    mLocationPrvider = mLocationManager
        .getBestProvider(mCriteria01, true);
    mLocation = mLocationManager
        .getLastKnownLocation(mLocationPrvider);
}

/* 設定起點的方法 */
private void setStartPoint()
{
    int mode=1;
    MyOverlay mOverlay = new MyOverLay(gp1, gp2, mode);
    List<Overlay> overlays = mMapView.getOverlays();
    overlays.add(mOverlay);
}

/* 設定路線的方法 */
private void setRoute()
{
    int mode=2;
    MyOverlay mOverlay = new MyOverLay(gp1, gp2, mode);
```

```
List<Overlay> overlays = mMapViews.getOverlays();
overlays.add(mOverlay);
}
/* 設定終點的方法 */
private void setEndPoint()
{
    int mode=3;
    MyOverLay mOverlay = new MyOverLay(gp1,gp2,mode);
    List<Overlay> overlays = mMapViews.getOverlays();
    overlays.add(mOverlay);
}
/* 重設 Overlay 的方法 */
private void resetOverlay()
{
    List<Overlay> overlays = mMapViews.getOverlays();
    overlays.clear();
}
/* 更新 MapView 的方法 */
public void refreshMapView()
{
    mMapViews.displayZoomControls(true);
    MapController myMC = mMapViews.getController();
    myMC.animateTo(gp2);
    myMC.setZoom(zoomLevel);
    mMapViews.setSatellite(false);
}

/* 取得兩點間的距離的方法 */
public double GetDistance(GeoPoint gp1,GeoPoint gp2)
{
    double Lat1r = ConvertDegreeToRadians(gp1.getLatitudeE6()/1E6);
    double Lat2r = ConvertDegreeToRadians(gp2.getLatitudeE6()/1E6);
    double Long1r= ConvertDegreeToRadians(gp1.getLongitudeE6()/1E6);
    double Long2r= ConvertDegreeToRadians(gp2.getLongitudeE6()/1E6);
    /* 地球半徑(KM) */
    double R = 6371;
    double d = Math.acos(Math.sin(Lat1r)*Math.sin(Lat2r)+
        Math.cos(Lat1r)*Math.cos(Lat2r)*
        Math.cos(Long2r-Long1r))*R;
    return d*1000;
}

private double ConvertDegreeToRadians(double degrees)
{
    return (Math.PI/180)*degrees;
}

/* format 移動距離的方法 */
public String format(double num)
{

```

## Google Android SDK 開發範例大全

```

        NumberFormat formatter = new DecimalFormat("###");
        String s=formatter.format(num);
        return s;
    }

    @Override
    protected boolean isRouteDisplayed()
    {
        return false;
    }
}

```

**src/irdc.ex10\_02/MyOverLay.java**

本程式的功能為在 MapView 上繪製圖形，繼承 com.google.android.maps.Overlay，程式中先以 mapView.getProjection() 取得 Projection 物件，再以 Projection.toPixels(GeoPoint,Point) 將 GeoPoint 轉換成 Point 物件，再利用 Point 物件的對應位置來繪製圖形。

MyOverLay 的建構子傳入移動前的 GeoPoint、移動後的 GeoPoint 與 mode 三個參數，當傳入 mode 為 1 時，會做繪製起點標記的動作；傳入 mode 為 2 時，會做繪製移動路線的動作；傳入 mode 為 3 時，則會做繪製終點標記的動作。

```

package irdc.ex10_02;

import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Point;
import android.graphics.RectF;
import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;
import com.google.android.maps.Projection;

public class MyOverLay extends Overlay
{
    private GeoPoint gp1;
    private GeoPoint gp2;
    private int mRadius=6;
    private int mode=0;

    /* 建構子，傳入起點與終點的 GeoPoint 與 mode */
    public MyOverLay(GeoPoint gp1,GeoPoint gp2,int mode)
    {
        this.gp1 = gp1;
        this.gp2 = gp2;
        this.mode = mode;
    }
}

```

```
@Override
public boolean draw
(Canvas canvas, MapView mapView, boolean shadow, long when)
{
    Projection projection = mapView.getProjection();
    if (shadow == false)
    {
        /* 設定筆刷 */
        Paint paint = new Paint();
        paint.setAntiAlias(true);
        paint.setColor(Color.BLUE);

        Point point = new Point();
        projection.toPixels(gp1, point);
        /* mode=1: 建立起點 */
        if(mode==1)
        {
            /* 定義 RectF 物件 */
            RectF oval=new RectF(point.x - mRadius, point.y - mRadius,
                                point.x + mRadius, point.y + mRadius);

            /* 繪製起點的圓形 */
            canvas.drawOval(oval, paint);
        }
        /* mode=2: 畫路線 */
        else if(mode==2)
        {
            Point point2 = new Point();
            projection.toPixels(gp2, point2);
            paint.setColor(Color.BLACK);
            paint.setStrokeWidth(5);
            paint.setAlpha(120);
            /* 畫線 */
            canvas.drawLine(point.x, point.y, point2.x,point2.y, paint);
        }
        /* mode=3: 建立終點 */
        else if(mode==3)
        {
            /* 避免誤差，先畫最後一段的路線 */
            Point point2 = new Point();
            projection.toPixels(gp2, point2);
            paint.setStrokeWidth(5);
            paint.setAlpha(120);
            canvas.drawLine(point.x, point.y, point2.x,point2.y, paint);

            /* 定義 RectF 物件 */
            RectF oval=new RectF(point2.x - mRadius,point2.y - mRadius,
                                point2.x + mRadius,point2.y + mRadius);

            /* 繪製終點的圓形 */
        }
    }
}
```

## Google Android SDK 開發範例大全

```

        paint.setAlpha(255);
        canvas.drawOval(oval, paint);
    }
}
return super.draw(canvas, mapView, shadow, when);
}
}

```

### AndroidManifest.xml

由於使用了 `google.android.maps.MapView` Widget，在 `AndroidManifest.xml` 裡宣告了要使用的自訂 library（uses-library）的敘述，並加入網路連線與取得 Location 座標的權限。

```

<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="irdc.ex10_02"
    android:versionCode="1"
    android:versionName="1.0.0">
    <application
        android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity
            android:name=".EX10_02"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <uses-library android:name="com.google.android.maps" />
    </application>
    <uses-permission
        android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission
        android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />
</manifest>

```

### ► 延伸學習

本範例利用覆寫 `LocationListener` 中的 `onLocationChanged()` 來追蹤目前所在的座標位置，進而計算出移動的距離，如果要同時計算目前的時速，只要在記錄 `GeoPoint` 座標的同時，記下當前的時間，透過前後時間差與移動距離的計算，就可以算出目前時速是多少，以下提供範例給各位參考：

```

/* 參數說明
   m1：移動前的時間(微秒)
   m2：移動後的時間(微秒)
   gp1：移動前的 GeoPoint
   gp2：移動後的 GeoPoint
*/

/* 取得移動距離 */
double dis=GetDistance(gp1, gp2);
/* 取得時間差 */
double time=(m2-m1)/1000F;
/* 取得時速(單位為 KM/HR) */
double speed=dis/time*60*60/1000;

```

除了透過實作 `com.google.android.maps.Overlay` 類別，可在 `MapView` 上繪製圖形外，Android API 中亦提供了一個抽象（abstract）類別 `ItemizedOverlay`（`com.google.android.maps.ItemizedOverlay`），可以透過繼承 `ItemizedOverlay` 來實作一個類別，同樣可以達到在 `MapView` 上標註記的效果，繼承 `ItemizedOverlay` 的類別範例如下：

```

/* 繼承 ItemizedOverlay */
private class MyItemOverlay extends ItemizedOverlay<OverlayItem>
{
    private List<OverlayItem> items = new ArrayList<OverlayItem>();
    /* 建構子，傳入 Drawable 與 GeoPoint */
    public MyItemOverlay(Drawable defaultMarker, GeoPoint gp)
    {
        super(defaultMarker);
        /* 加入一個 OverlayItem */
        items.add(new OverlayItem(gp, "Title", "Snippet"));
        populate();
    }

    @Override
    protected OverlayItem createItem(int i)
    {
        return items.get(i);
    }

    @Override
    public int size()
    {
        return items.size();
    }
    /* 按下該座標時會做的動作 */
    @Override
    protected boolean onTap(int pIndex)
    {
        Toast.makeText

```

## Google Android SDK 開發範例大全

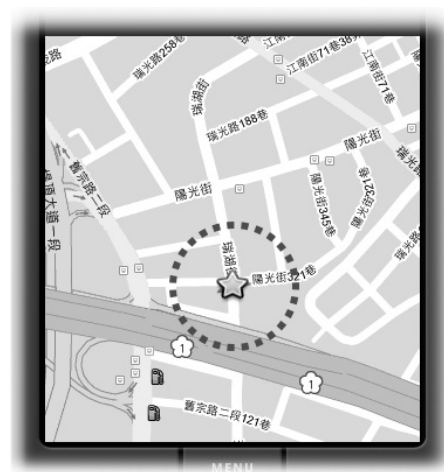
```
(
    EX10_02_1.this,items.get(pIndex).getSnippet(),
    Toast.LENGTH_LONG
).show();
return true;
}
}
```

在程式中使用這個類別標註記的方式如下：

```
GeoPoint gp = new GeoPoint((int)geoLatitude,(int)geoLongitude);
Drawable dr=getResources().getDrawable
(
    android.R.drawable.btn_star_big_on
);
dr.setBounds(-15,-15, 15, 15);

MyItemOverlay mOverlay01 = new MyItemOverlay(dr,gp);
List<Overlay> overlays = mMapView.getOverlays();
overlays.add(mOverlay01);
```

程式執行後會在座標點放上一個 Drawable 圖形，執行結果如下：



▲ 圖 10-3 使用 ItemizedOverlay 在地圖上做標記