

Android終端系統匯流應用設計

行車碰撞通報

Brian Wang 王昱景

brian.wang.frontline@gmail.com

感測器

- Android 支援多種感測器來監測行動裝置目前的狀態，例如：數位羅盤、加速度感測器、重力感測器、趨近感測器、陀螺儀和環境光線感測器等
- 行動裝置可能只支援其中幾項感測器，而且 Android 模擬器不支援感測器，只能使用實機來測試感測器
- 在實務上最常使用加速感測器 (Accelerometer)

- 碰撞監測是使用加速感測器判斷車輛是否發生事故
- 當取得感測器系統服務的 `ServiceManager` 物件後，就可以註冊 `SensorEventListener` 傾聽者物件來取得感測器的偵測資料
- 為了避免耗用過多電力，建議在 `onResume()` 方法註冊，`onPause()` 方法取消註冊

簡訊處理

- 行動裝置的主要功能就是對外通訊，除了使用語音通話外，另一個常用功能是簡訊 (Short Message Service, SMS) 即手機的文字訊息服務

儲存偏好設定

- 在 Android 提供 SharedPreferences 物件儲存簡單的應用程式資料，實際上，它就是使用 XML 格式的偏好設定檔來儲存資料
- SharedPreferences 類別屬於 android.content 套件，可以用來儲存少量且簡單的應用程式資料，這些資料是儲存成類似 Bundle 物件的鍵值與對應值，支援 Boolean、Float、Integer、Long 和 String 型態的資料

- 在活動存取偏好設定需要取得 `SharedPreferences` 物件，通常我們是在 `onCreate()` 覆寫方法取得此物件
- 使用 `Activity` 類別的 `getPreferences()` 方法取得 `SharedPreferences` 物件，參數是操作模式常數，`MODE_PRIVATE` 只允許建立偏好設定檔的活動存取，屬於活動層級（`Activity-level`）的偏好設定，只允許同一個活動來存取

- 在 `Activity` 活動類別讀取偏好設定建議是在 `onResume()` 覆寫方法，在取得 `SharedPreferences` 物件後，可以使用 `getString()`、`getFloat()` 或 `getInt()` 等方法取得儲存的字串、浮點數和整數等值
- 方法的第1個參數是字串型態的鍵值，第2個參數是預設值

- 在 `Activity` 活動類別儲存偏好設定建議是在 `onPause()` 覆寫方法，我們是使用 `SharedPreferences.Editor` 物件來編輯存入的資料
- 使用 `putString()`、`putInt()` 和 `putFloat()` 等方法存入字串、整數和浮點數等資料

行車碰撞通報

- 利用加速感測器當行動裝置偵測到有碰撞發生時，利用預存的電話號碼及簡訊內容，發送簡訊通知求救

SOSDemo

- 開啟和執行 Android 專案
- 建立使用介面的版面配置
- 建立 Activity 活動類別
- 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: **SOSDemo**
 - Build Target: **Android API 4.2.2**
 - Package Name: tw.edu.uch

2.建立使用介面的版面配置

- 使用 `LinearLayout` 垂直編排 2 個 `TextView` 和 `EditText` 元件，3 個 `Button` 元件



```
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/phoneno" />

<EditText
    android:id="@+id/txtPhoneNo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="phone" >

    <requestFocus />
</EditText>

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/message" />

<EditText
    android:id="@+id/txtMessage"
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:ems="10"
    android:gravity="top"
    android:inputType="textMultiLine" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <Button
        android:id="@+id/startBtn"
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:onClick="startBtn_Click"
        android:text="@string/start" />

    <Button
        android:id="@+id/stopBtn"
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:onClick="stopBtn_Click"
        android:text="@string/stop" />

    <Button
        android:id="@+id/saveBtn"
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:onClick="saveBtn_Click"
        android:text="@string/save" />

</LinearLayout>
```

3. 建立 Activity 活動類別

- 在活動類別的開頭宣告成員變數

```
public class MainActivity extends Activity implements SensorEventListener {  
  
    private static final String PHONE_NO = "phoneNo";  
    private static final String MESSAGE = "message";  
    private static final String IS_SAVE = "isSave";  
  
    private EditText txtPhoneNo;  
    private EditText txtMessage;  
    private SensorManager sensorManager;  
    private SharedPreferences prefs;  
  
    private String phoneNo;  
    private String message;  
    private boolean isSave = false;  
    private long lastUpdate;
```

onCreate() 方法

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

    txtPhoneNo = (EditText) findViewById(R.id.txtPhoneNo);
    txtMessage = (EditText) findViewById(R.id.txtMessage);

    sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
    sensorManager.unregisterListener(this);

    prefs = getPreferences(MODE_PRIVATE);
    lastUpdate = System.currentTimeMillis();
}
```


onResume() 方法

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

    txtPhoneNo.setText(prefs.getString(PHONE_NO, ""));
    txtMessage.setText(prefs.getString(MESSAGE, ""));
    phoneNo = prefs.getString(PHONE_NO, "");
    message = prefs.getString(MESSAGE, "");
    isSave = prefs.getBoolean(IS_SAVE, false);

    if (isSave) {
        sensorManager.registerListener(this,
            sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
            SensorManager.SENSOR_DELAY_NORMAL);
    }
}
```


onPause() 方法

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

    SharedPreferences.Editor prefEdit = prefs.edit();
    prefEdit.putString(PHONE_NO, phoneNo);
    prefEdit.putString(MESSAGE, message);
    prefEdit.putBoolean(IS_SAVE, isSave);
    prefEdit.commit();
}
```

onStop() 方法

```
@Override  
protected void onStop() {  
    super.onStop();  
  
    sensorManager.unregisterListener(this);  
}
```

SensorEventListener

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

    sensorManager.unregisterListener(this);
}

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
}

@Override
public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        getAccelerometer(event);
    }
}
```

startBtn_Click() 事件

```
public void startBtn_Click(View view) {  
    if (!isSave) {  
        Toast.makeText(this, "請輸入電話號碼和訊息內容！", Toast.LENGTH_SHORT).show();  
        return;  
    }  
  
    sensorManager.registerListener(this,  
        sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),  
        SensorManager.SENSOR_DELAY_NORMAL);  
}
```


stopBtn_Click() 事件

```
public void stopBtn_Click(View view) {  
    sensorManager.unregisterListener(this);  
}
```

saveBtn_Click() 事件

```
public void saveBtn_Click(View view) {  
    String pNo = txtPhoneNo.getText().toString();  
    String msg = txtMessage.getText().toString();  
  
    if (pNo.length() > 0 && msg.length() > 0) {  
        phoneNo = pNo;  
        message = msg;  
        isSave = true;  
    } else {  
        isSave = false;  
  
        Toast.makeText(this, "請輸入電話號碼和訊息內容！", Toast.LENGTH_SHORT).show();  
    }  
}
```

getAccelerometer() 方法

```
private void getAccelerometer(SensorEvent event) {  
    float[] values = event.values;  
  
    float x = values[0];  
    float y = values[1];  
    float z = values[2];  
  
    float accelationSquareRoot = (x * x + y * y + z * z) / (SensorManager.GRAVITY_EARTH * SensorManager.GRAVITY_EARTH);  
    long actualTime = System.currentTimeMillis();  
  
    if (accelationSquareRoot >= 2) {  
        if (actualTime - lastUpdate < 200) {  
            return;  
        }  
  
        sendSMS();  
        lastUpdate = actualTime;  
    }  
}
```

sendSMS() 方法

```
private void sendSMS() {  
    SmsManager.getDefault().sendTextMessage(phoneNo, null, message, null, null);  
}
```


4. 在 AndroidManifest.xml 新增權限

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