行動裝置數位匯流應用實務

相機與感測器的應用 – 聰明相機

王昱景 Brian Wang
brian.wang.frontline@gmail.com

- 相機與感測器的整合應用
- 使用加速感測器偵測的數值來判斷行動裝置的相機是否拿歪

SmartCamDemo

- 開啟和執行 Android 專案
- 建立版面配置
- 建立 Activity 活動類別
- 在 AndroidManifest.xml 新增權限

1.開啟和執行Android專案

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

2.建立版面配置

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <SurfaceView
        android:id="@+id/cameraview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1" >
    </SurfaceView>

    <TextView
        android:id="@+id/output"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp" />

</FrameLayout>
```

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 SurfaceHolder.Callback, OnClickListener {

    private Camera camera;
    private boolean isPreviewRunning = false;
    private SurfaceView surfaceview;
    private SurfaceHolder surfaceHolder;
    private SensorManager manager;
    private Sensor accelerometer;
    private MySensorListener mListener;
    private TextView output;

}
```

onCreate() 方法

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);

     requestWindowFeature(Window.FEATURE_NO_TITLE);

     getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);

     setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
     setContentView(R.layout.activity_main);

     // 取得SurfaceView元件
     surfaceview = (SurfaceView) findViewById(R.id.cameraview);
     surfaceview.setOnClickListener(this);

     surfaceHolder = surfaceview.getHolder();
     surfaceHolder.addCallback(this);
     surfaceHolder.~~setType~~(SurfaceHolder.~~SURFACE_TYPE_PUSH_BUFFERS~~);

     // 取得感測器的系統服務
     manager = (SensorManager) getSystemService(SENSOR_SERVICE);

     // 使用加速感測器
     accelerometer = manager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);

     mListener = new MySensorListener();

     // 取得TextView元件
     output = (TextView) findViewById(R.id.output);
}
```

onResume() 方法

```
@Override
protected void onResume() {
    super.onResume();
    manager.registerListener(mListener, accelerometer, SensorManager.SENSOR_DELAY_UI);
}
```

onPause() 方法

```
@Override
protected void onPause() {
    super.onPause();
    manager.unregisterListener(mListener);
}
```

onStop() 方法

```java
@Override
protected void onStop() {
   manager.unregisterListener(mListener);
   super.onStop();
}
```

建立 PictureCallback 物件

```java
// 建立PictureCallback物件
private Camera.PictureCallback pictureCallback = new Camera.PictureCallback() {

    public void onPictureTaken(byte[] imageData, Camera c) {
        if (imageData != null) {
            saveImage(MainActivity.this, imageData, 50);
            // 相片預覽
            camera.startPreview();
        }
    }
};
```

實作 SurfaceHolder.Callback 介面方法

```java
// 實作SurfaceHolder.Callback介面方法
public void surfaceCreated(SurfaceHolder holder) {
    camera = Camera.open();
}

public void surfaceChanged(SurfaceHolder holder, int format, int w, int h) {
    if (isPreviewRunning) {
        camera.stopPreview(); // 停止預覽
    }

    Camera.Parameters p = camera.getParameters();
    p.setPreviewSize(w, h);

    camera.setParameters(p);

    try {
        camera.setPreviewDisplay(holder);
    } catch (IOException e) {
        Log.e("SmartCamDemo", e.getMessage());
    }

    camera.startPreview();

    isPreviewRunning = true;
}

public void surfaceDestroyed(SurfaceHolder holder) {
    camera.stopPreview();
    isPreviewRunning = false;
    camera.release();
}
```

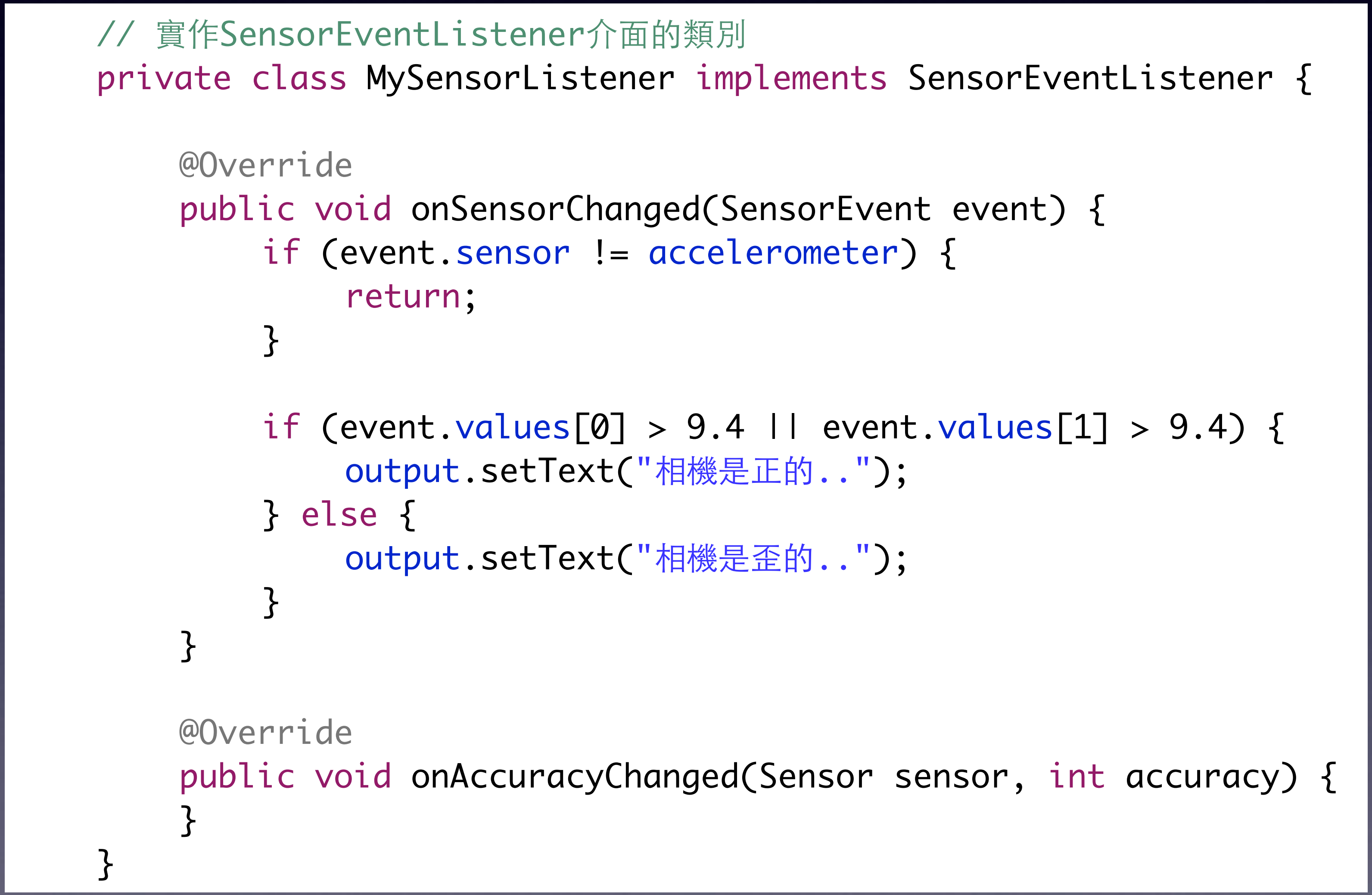
實作 OnClickListener 介面方法

```
// 實作OnClickListener介面方法
public void onClick(View arg0) {
    camera.takePicture(null, pictureCallback, pictureCallback);
}
```

儲存 JPEG 格式的圖片

```java
// 儲存JPEG格式的圖片
public Bitmap saveImage(Context mContext, byte[] imageData, int quality) {
    // 建立File物件的儲存路徑
    File path = new File("/sdcard");
    Bitmap image = null;

    try {
        BitmapFactory.Options options = new BitmapFactory.Options();
        options.inSampleSize = 5;

        image = BitmapFactory.decodeByteArray(imageData, 0, imageData.length, options);

        FileOutputStream fos = new FileOutputStream(path.toString() + "/picture.jpg");

        BufferedOutputStream bos = new BufferedOutputStream(fos);

        // 圖檔格式JPEG
        image.compress(CompressFormat.JPEG, quality, bos);
        bos.flush();
        bos.close();
    } catch (Exception e) {
        Log.e("SmartCamDemo", e.getMessage());
    }

    return image;
}
```

實作 SensorEventListener 介面的類別



```java
// 實作SensorEventListener介面的類別
private class MySensorListener implements SensorEventListener {

    @Override
    public void onSensorChanged(SensorEvent event) {
        if (event.sensor != accelerometer) {
            return;
        }

        if (event.values[0] > 9.4 || event.values[1] > 9.4) {
            output.setText("相機是正的..");
        } else {
            output.setText("相機是歪的..");
        }
    }

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

4. 在 AndroidManifest.xml 新增權限

```
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
```