



國立宜蘭大學

National Ilan University NIU

行動裝置互動系統設計與應用

授課老師：黃朝曦



教學單元

- 加速度感應器
- 光線感測器
- 分貝感測



感應器介紹

- **感應器**（sensor）就是專門感應外界事物變化，並將其變化轉為數值的一種接收器
- 目前API支援的感應器如下：

加速度感應器	Sensor.TYPE_ACCELEROMETER
重力感應器	Sensor.TYPE_GRAVITY
陀螺儀感應器	Sensor.TYPE_GYROSCOPE
光線感應器	Sensor.TYPE_LIGHT
線性加速度感應器	Sensor.TYPE_LINEAR_ACCELERATION
磁場感應器	Sensor.TYPE_MAGNETIC_FIELD
方位感應器	SensorManager.getOrientation()已取代 Sensor.TYPE_ORIENTATION
壓力感應器	Sensor.TYPE_PRESSURE
接近感應器	Sensor.TYPE_PROXIMITY
旋轉向量感應器	Sensor.TYPE_ROTATION_VECTOR
溫度感應器	Sensor.TYPE_TEMPERATURE

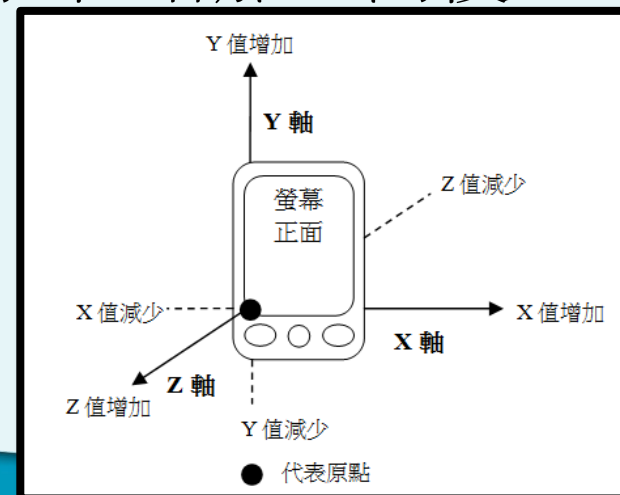


感應器的數值

- 感應器的數值以一個**float**陣列儲存，通常以 `values[i]` 來代表
- 不同的感應器，陣列的元素個數也會有所不同。
 - **加速度與方位感應器**都有X軸、Y軸、Z軸觀念，所以有**3**個數值：**`values[0]`**、**`values[1]`**、**`values[2]`**
 - **接近感應器**只有距離一個數值，所以只使用到**`values[0]`**
 - 數值雖不同，但**取得方式**大致相同

3D座標軸概念

- Android採用**OpenGL ES**的座標系統
 - 螢幕左下角頂點為原點（ $x=0, y=0, z=0$ ）
 - X軸為**水平**方向，向右X值增加，向左X值減少
 - Y軸為**垂直**方向，向上Y值增加，向下Y值減少
 - Z軸為**後向前**的方向，向前Z值增加，向後Z值減少





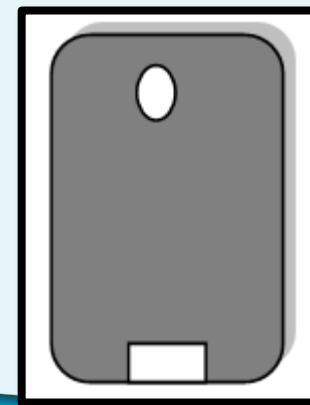
國立宜蘭大學

National Ilan University NIU

加速度感應器

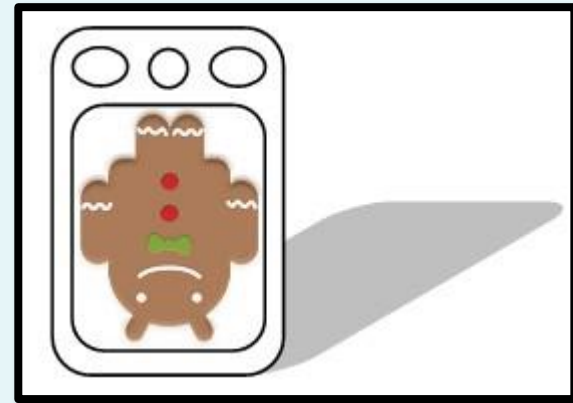
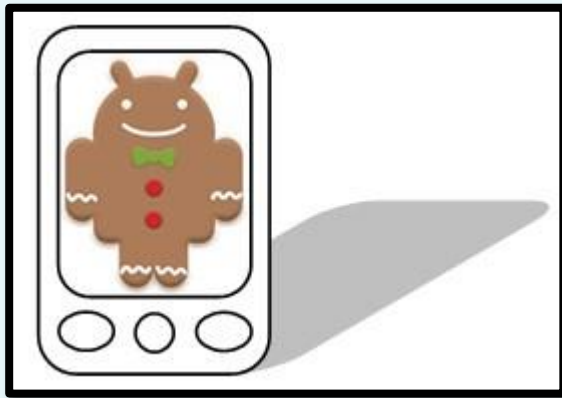
加速度感應器

- 重力方向恰與座標方向相反，所以若符合重力方向與座標方向**相反**，會得到**正**的值，反之會得到負的值
- 行動裝置**平躺**，Z軸受重力影響， $\text{values}[2] = 9.81$
- 行動裝置平躺但螢幕**正面朝下**， $\text{values}[2] = -9.81$



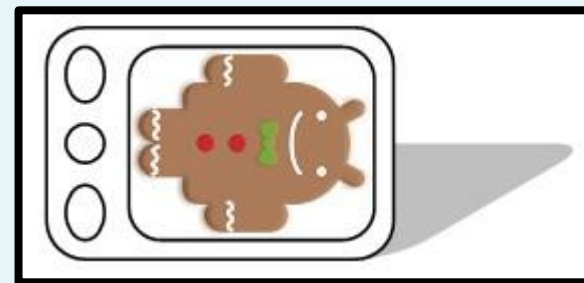
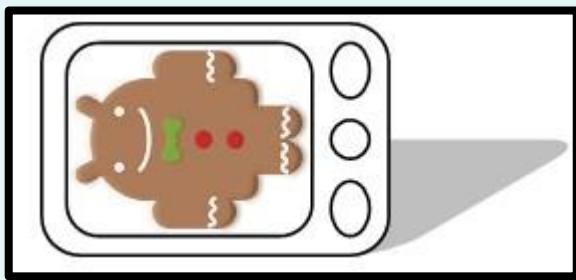
加速度感應器

- 行動裝置**縱向直立**，Y軸受重力影響， $\text{values}[1] = 9.81$
- 行動裝置直立方式**上下顛倒**， $\text{values}[1] = -9.81$

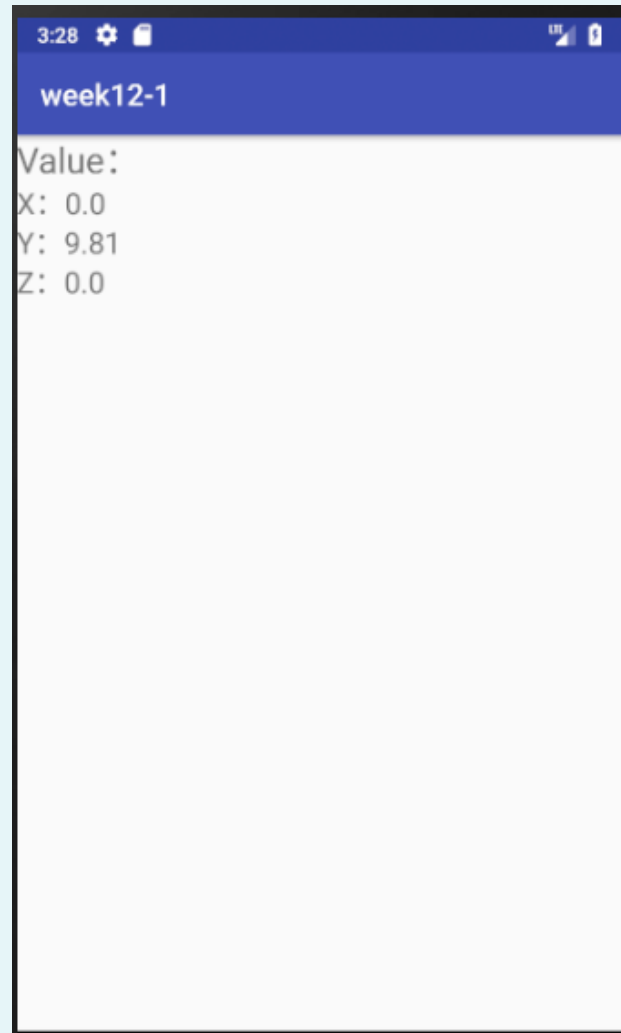


加速度感應器

- 行動裝置**橫向直立**，X軸受重力影響， $\text{values}[0] = 9.81$
- 行動裝置直立方式**左右顛倒**， $\text{values}[0] = -9.81$



UI介面

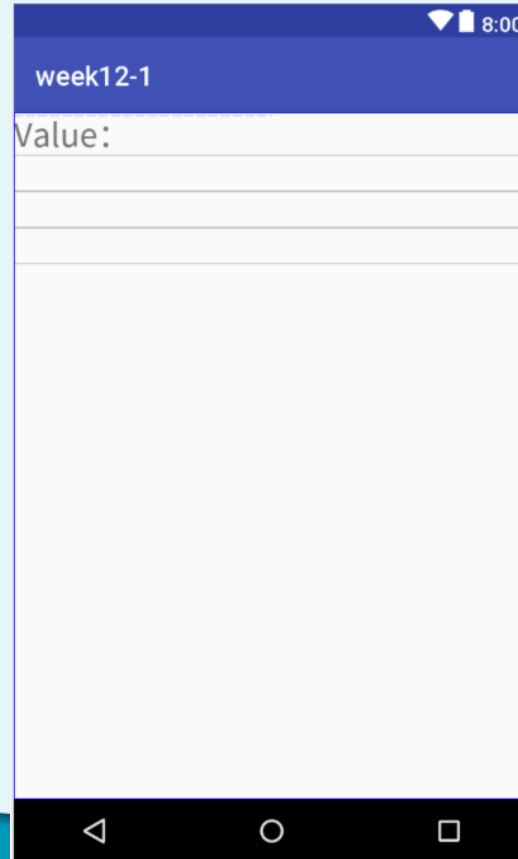


國立宜蘭大學

National Ilan University NIU

activity_main.xml

- 拉四個TextView。
- 第一個TextView的Text 改為 Value：
- 其餘三個的Text留空



開始打程式 MainActivity.java

- 步驟1. 宣告UI與Sensor物件。

```
TextView mTextView1,mTextView2,mTextView3;  
SensorManager mSensorManager;  
Sensor mSensor;
```

- 步驟2. 建立程式與元件的關係。

```
mTextView1 = findViewById(R.id.textview2);  
mTextView2 = findViewById(R.id.textview3);  
mTextView3 = findViewById(R.id.textview4);
```

- 步驟3. 建立Sensor Service並選擇Sensor類型。

```
mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);  
mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
```



選擇加速度感測器

- 步驟4. 建立SensorEventListener事件。

```
private SensorEventListener mysel = new SensorEventListener() {  
  
    @Override  
    public void onSensorChanged(SensorEvent sensorEvent) {  
        mTextView1.setText("X : "+sensorEvent.values[0]);  
        mTextView2.setText("Y : "+sensorEvent.values[1]);  
        mTextView3.setText("Z : "+sensorEvent.values[2]);  
    }  
  
    @Override  
    public void onAccuracyChanged(Sensor sensor, int i) {  
  
    }  
  
};
```

將Sensor改變的值存到value陣列，
再將其顯示在TextView。

開始打程式 MainActivity.java

步驟5. onPause()設定

- 1) 在空白處按下Ctrl+O 。
- 2) 找到onPause():取消註冊Sensor的監聽

```
@Override  
protected void onPause() {  
    super.onPause();  
    mSensorManager.unregisterListener(myself);  
}
```

開始打程式 MainActivity.java

步驟6. onResume()設定

- 1) 在空白處按下Ctrl+O。
- 2) 找到**onResume()**:重新註冊Sensor的監聽

```
@Override
protected void onResume() {
    super.onResume();
    mSensorManager.registerListener(myself, mSensor, mSensorManager.SENSOR_DELAY_NORMAL);
}
```



完整程式碼(1/2)

```
public class MainActivity extends AppCompatActivity {

    TextView mTextView1, mTextView2, mTextView3;
    SensorManager mSensorManager;
    Sensor mSensor;

    private SensorEventListener mysel = new SensorEventListener() {

        @Override
        public void onSensorChanged(SensorEvent sensorEvent) {
            mTextView1.setText("X : "+sensorEvent.values[0]);
            mTextView2.setText("Y : "+sensorEvent.values[1]);
            mTextView3.setText("Z : "+sensorEvent.values[2]);
        }

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

        }

    };

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

        mTextView1 = findViewById(R.id.textView2);
        mTextView2 = findViewById(R.id.textView3);
        mTextView3 = findViewById(R.id.textView4);

        mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
        mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    }
}
```




完整程式碼(2/2)

```
@Override
protected void onResume() {
    super.onResume();
    mSensorManager.registerListener(myself, mSensor, mSensorManager.SENSOR_DELAY_NORMAL);
}

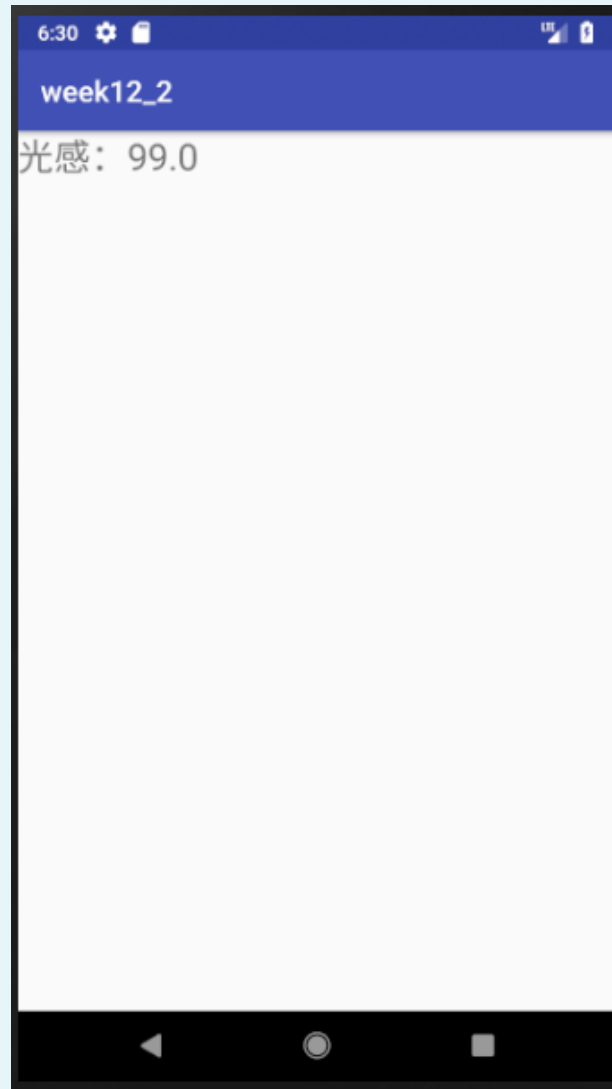
@Override
protected void onPause() {
    super.onPause();
    mSensorManager.unregisterListener(myself);
}
```



國立宜蘭大學
National Ilan University NIU

光線感測器

光線感測器



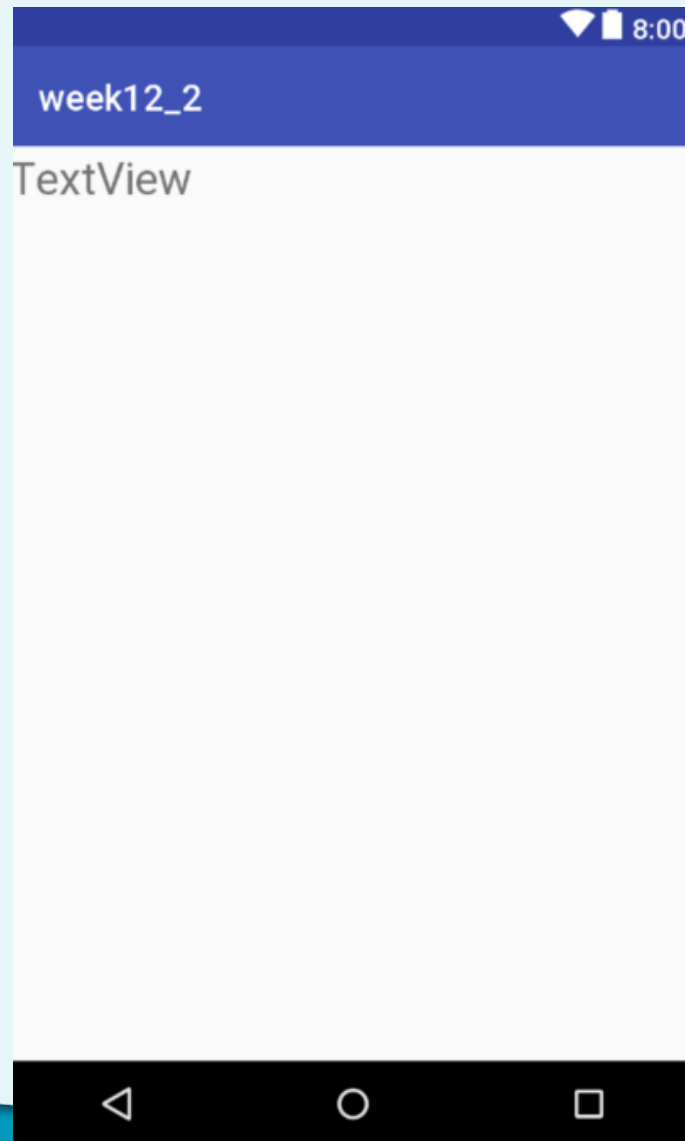
國立宜蘭大學

National Ilan University NIU

光線感測器

- 光線感應器主要是感應手機環境四週的光線強弱程度，一般稱為**照度**（**illuminance**），單位是**勒克斯**（**lux**，流明/平方公尺）
- **1 lux**大約等於**1燭光**在**1公尺**距離內的照度
- 光線感應器的數值只有一個，代表的就是**照度**

UI介面



國立宜蘭大學

National Ilan University NIU

activity_main.xml

- 拉一個TextView。



開始打程式 MainActivity.java

- 步驟1. 宣告UI與Sensor物件。

```
TextView tv;  
SensorManager mySm;  
Sensor myS;
```

- 步驟2. 建立程式與元件的關係。

```
tv = (TextView) findViewById(R.id.textView1);
```


- 步驟3. 建立Sensor Service並選擇Sensor類型。

```
mySm=(SensorManager) getSystemService(SENSOR_SERVICE);  
myS =mySm.getDefaultSensor(Sensor.TYPE_LIGHT);
```

選擇光線感測器

- 步驟4. 建立SensorEventListener事件。

```
private SensorEventListener mySel=new SensorEventListener() {  
  
    @Override  
    public void onSensorChanged(SensorEvent arg0) {  
        // TODO Auto-generated method stub  
        float[] value=arg0.values;  
        tv.setText(""+value[0]);  
    }  
  
    @Override  
    public void onAccuracyChanged(Sensor arg0, int arg1) {  
        // TODO Auto-generated method stub  
  
    }  
};
```



將Sensor改變的值存到value陣列，
再將其顯示在TextView。

開始打程式 MainActivity.java

步驟5. onPause()設定

- 1) 在空白處按下右鍵，選擇
Source→Override/Implement Methods。
- 2) 找到onPause():取消註冊Sensor的監聽

```
@Override  
protected void onPause() {  
    // TODO Auto-generated method stub  
    mySm.unregisterListener(mySel);  
    super.onPause();  
}
```

開始打程式 MainActivity.java

步驟6. onResume()設定

- 1) 在空白處按下右鍵，選擇
Source→Override/Implement Methods。
- 2) 找到**onResume()**:重新註冊Sensor的監聽

```
@Override
protected void onResume() {
    // TODO Auto-generated method stub
    super.onResume();
    mySm.registerListener(mySel, myS, SensorManager.SENSOR_DELAY_NORMAL);
}
```



完整程式碼(1/2)

```
public class MainActivity extends AppCompatActivity {

    TextView mTextView;
    SensorManager mySm;
    Sensor myS;

    private SensorEventListener mySel = new SensorEventListener() {
        @Override
        public void onSensorChanged(SensorEvent sensorEvent) {
            float[] value = sensorEvent.values;
            mTextView.setText("光感: "+value[0]);
        }

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

        }
    };

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

        mTextView = findViewById(R.id.textview);

        mySm = (SensorManager) getSystemService(SENSOR_SERVICE);
        myS = mySm.getDefaultSensor(Sensor.TYPE_LIGHT);
    }
}
```



完整程式碼(2/2)

```
@Override
protected void onPause() {
    super.onPause();
    mySm.unregisterListener(mySel);
}

@Override
protected void onResume() {
    super.onResume();
    mySm.registerListener(mySel, myS, SensorManager.SENSOR_DELAY_NORMAL);
}
}
```

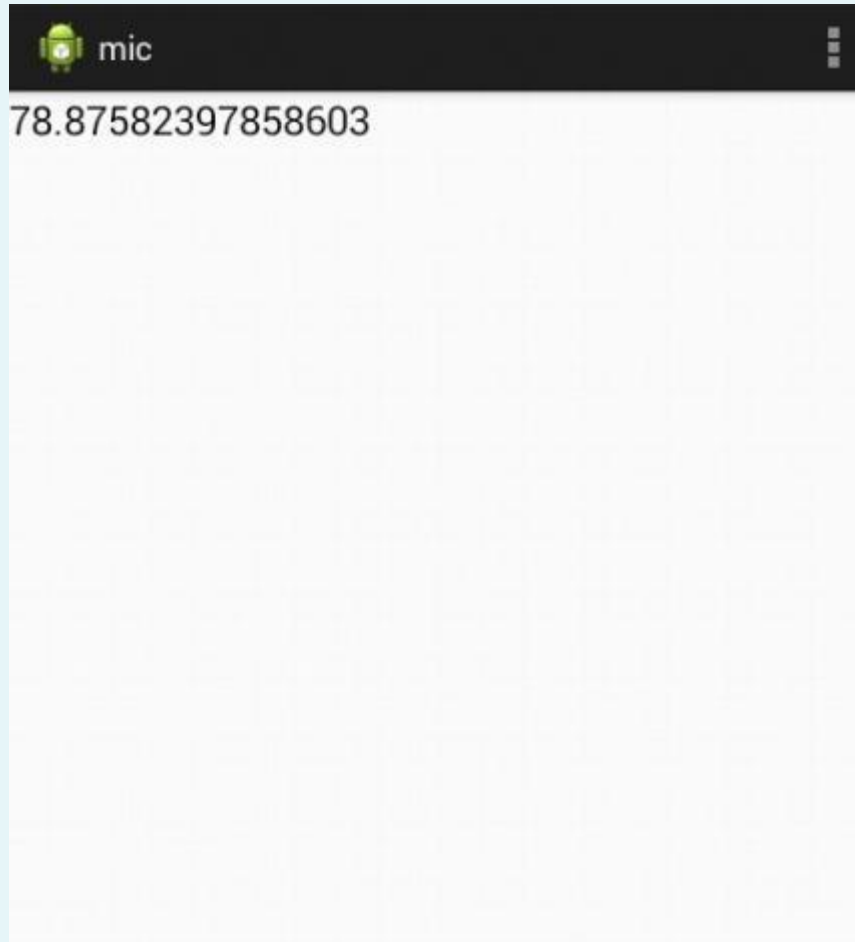


國立宜蘭大學

National Ilan University NIU

分貝感測

分貝感測



國立宜蘭大學

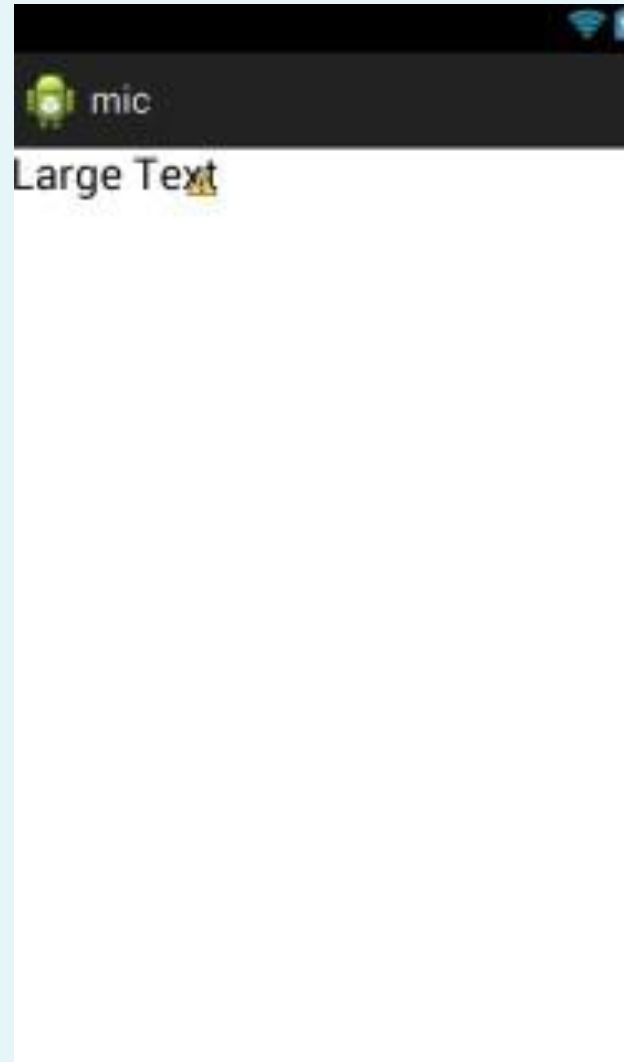
National Ilan University NIU



分貝感測

- Android並沒有實際的分貝感測器，所以必須利用麥克風的振幅+分貝轉換公式來製作分貝感測器。

UI介面



國立宜蘭大學

National Ilan University NIU

activity_main.xml

- 步驟1. 拉一個TextView。



開始打程式 MainActivity.java

- 步驟1. 宣告MediaRecorder物件。

```
private MediaRecorder mRecorder = null;
```

- 步驟2. 建立**handler**,負責接收麥克風最大的振幅量。

```
Handler handler = new Handler(){  
    @Override  
    public void handleMessage(Message msg) {  
        // TODO Auto-generated method stub  
        super.handleMessage(msg);  
        double db;  
        double ringerVol = mRecorder.getMaxAmplitude();  
        TextView tv = (TextView) findViewById(R.id.textView1);  
        db = 20 * Math.log10(ringerVol);  
        tv.setText(Double.toString(db));  
        handler.sendMessageDelayed(Message.obtain(), 300);  
    }  
};
```

獲得麥克風
振幅

分貝公式

物理複習-分貝公式

- 分貝的計算，依賴於是功率量還是場量而不同。

$$L_{\text{dB}} = 10 \log_{10} \left(\frac{A_1^2}{A_0^2} \right) = 20 \log_{10} \left(\frac{A_1}{A_0} \right).$$

開始打程式 MainActivity.java

- 步驟4. 建立Thread。

```
if (mRecorder == null) {  
    mRecorder = new MediaRecorder();  
    mRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);  
    mRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);  
    mRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);  
    mRecorder.setOutputFile("/dev/null");  
    try {  
        mRecorder.prepare();  
    } catch (IllegalStateException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
    mRecorder.start();  
}  
handler.sendMessageDelayed(0, 1000);
```

設置輸出檔的格式

設置音頻編碼

設為空裝置

空裝置:通常被用於丟棄不需要的輸出流，或作為用於輸入流的空檔案

開始打程式 MainActivity.java

步驟5. onDestroy()設定

- 1) 在空白處按下右鍵，選擇
Source→Override/Implement Methods。
- 2) 找到onDestroy():重新註冊Sensor的監聽。

```
@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();

    if (mRecorder != null) {
        mRecorder.stop();
        mRecorder.release();
        mRecorder = null;
    }
}
```

呼叫stop 結束錄音

呼叫release 會立即釋放

MediaRecorder 佔用的資源。
之後如果想要再使用
MediaRecorder 功能，必須
重新建立MediaRecorder 物件實體而無法重複使用已經
被釋放的MediaRecorder 物件。

AndroidManifest.xml

- 步驟6. 建立RECORD_AUDIO。

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.mic"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="19" />
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.mic.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>
    </application>

</manifest>
```

完整程式碼(1/3)

```
public class MainActivity extends ActionBarActivity {  
  
    private MediaRecorder mRecorder = null;  
  
    Handler handler = new Handler(){  
        @Override  
        public void handleMessage(Message msg) {  
            // TODO Auto-generated method stub  
            super.handleMessage(msg);  
            double db;  
            double ringerVol = mRecorder.getMaxAmplitude();  
            TextView tv = (TextView) findViewById(R.id.textView1);  
            db = 20 * Math.Log10(ringerVol);  
            tv.setText(Double.toString(db));  
            handler.sendMessageDelayed(Message.obtain(), 300);  
        }  
    };  
};
```

完整程式碼(2/3)

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

    if (mRecorder == null) {
        mRecorder = new MediaRecorder();
        mRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
        mRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
        mRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
        mRecorder.setOutputFile("/dev/null");
        try {
            mRecorder.prepare();
        } catch (IllegalStateException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        mRecorder.start();
    }

    handler.sendMessageDelayed(0, 1000);
}
```


完整程式碼(3/3)

```
@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();

    if (mRecorder != null) {
        mRecorder.stop();
        mRecorder.release();
        mRecorder = null;
    }
}
```



National Ilan University NIU
國立宜蘭大學



National Ilan University NIU
國立宜蘭大學



國立宜蘭大學

National Ilan University NIU

語音辨識

語音辨識



國立宜蘭大學

National Ilan University NIU



VOICE

回憶過去痛苦的相思忘不了

開始語音辨識

Start Speech



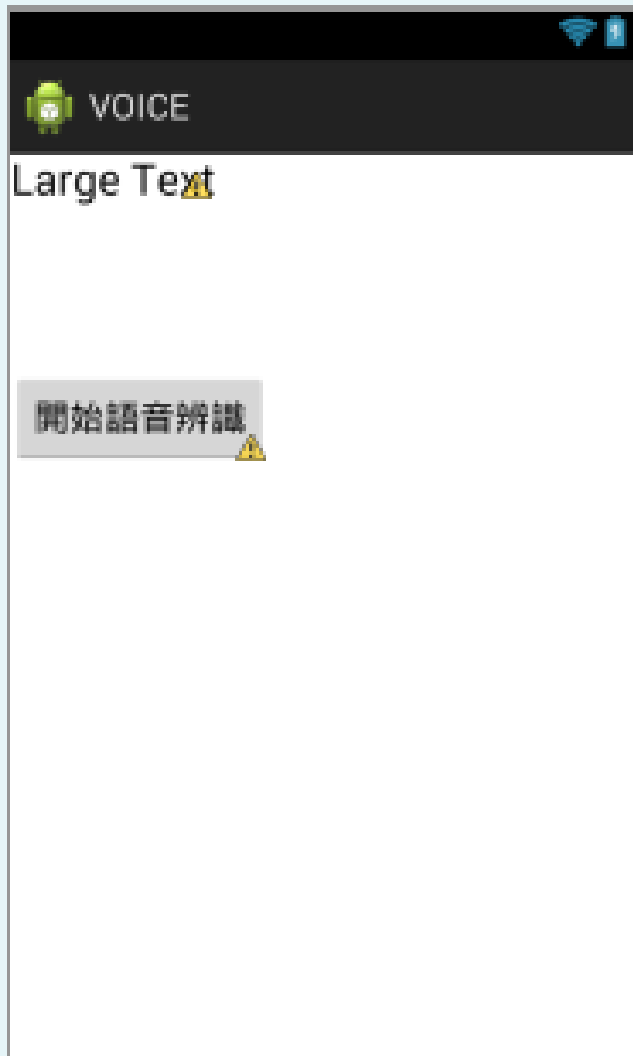
Google

語音辨識



國立宜蘭大學

National Ilan University NIU





activity_main.xml

- 步驟1. 拉一個TextView。
- 步驟2. 拉一個Button。

開始打程式 MainActivity.java

- 步驟1. 宣告UI物件。

```
TextView textResult;  
Button BTN;
```

- 步驟2. 建立程式與元件的關係。

```
BTN=(Button) findViewById(R.id.button1);  
textResult=(TextView) findViewById(R.id.textView1);
```


開始打程式 MainActivity.java

- 步驟3. 建立setOnClickListener事件。

通過Intent傳遞語音辨識的模式

```
BTN.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View arg0) {  
        // TODO Auto-generated method stub  
        Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);  
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,  
            RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);  
        intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Start Speech");  
        startActivityForResult(intent, RQS_VOICE_RECOGNITION);  
    }  
});
```

開始執行Intent→語音辨識

提示語音開始

語言模式和自由形式的語音辨識

- 步驟4. 當語音結束的回呼函數。

onActivityResult()設定

- 1) 在空白處按下右鍵，選擇
Source→Override/Implement Methods。
- 2) 找到onActivityResult():

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // TODO Auto-generated method stub
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode == RQS_VOICE_RECOGNITION){
        if(resultCode == RESULT_OK){
            ArrayList<String> result = data.getStringArrayListExtra
                (RecognizerIntent.EXTRA_RESULTS);
            String firstMatch = (String)result.get(0);
            textResult.setText(firstMatch);
        }
    }
}
```

取的語音字元

完整程式碼(1/2)

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

    BTN=(Button) findViewById(R.id.button1);
    textResult=(TextView) findViewById(R.id.textView1);

    BTN.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
            intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
                RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
            intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Start Speech");
            startActivityForResult(intent, RQS_VOICE_RECOGNITION);
        }
    });
}
```

完整程式碼(2/2)

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // TODO Auto-generated method stub
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode == RQS_VOICE_RECOGNITION){
        if(resultCode == RESULT_OK){
            ArrayList<String> result = data.getStringArrayListExtra
                (RecognizerIntent.EXTRA_RESULTS);
            String firstMatch = (String)result.get(0);
            textResult.setText(firstMatch);
        }
    }
}
```