



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

- 播放與錄製音訊
- 分貝感測
- 語音辨識



國立宜蘭大學

National Ilan University NIU



國立宜蘭大學

National Ilan University NIU

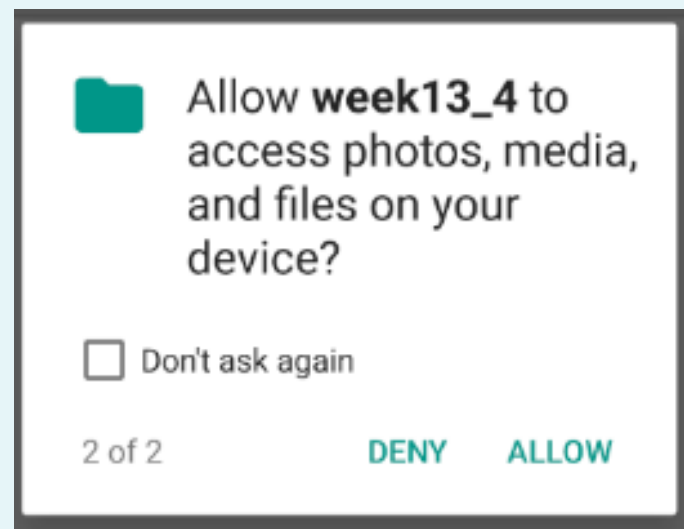
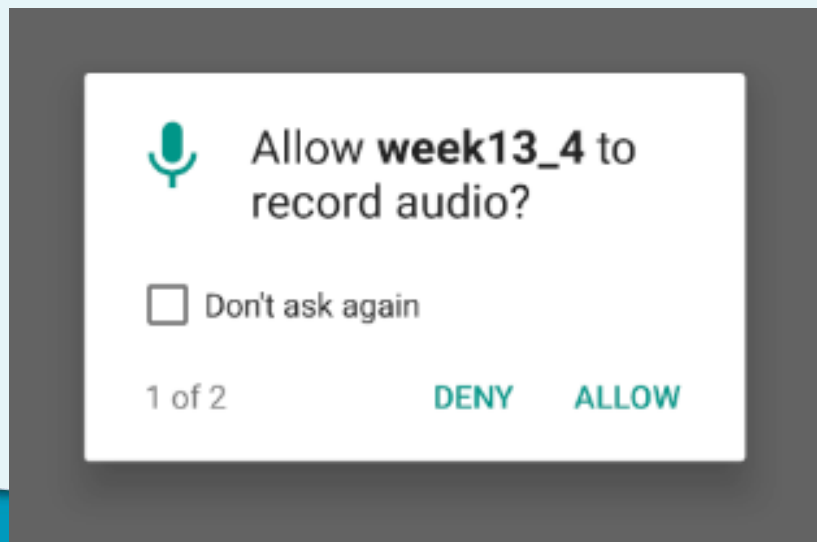
# 播放與錄製音訊

# 播放與錄製音訊範例



# 播放與錄製音訊說明

- Android可以使用**MediaPlayer**來**播放**音樂和影片，也可以使用**MediaRecorder**來**錄製**音樂，這些多媒體方式可以使得手機遊戲能更有趣。
- 因為**麥克風的權限**被歸類在**危險權限**，所以除了Manifest要給權限外，app仍要再跟使用者要一次權限



# AndroidManifest.xml

- 在執行程式前，記得要加入錄音和儲存至SD卡的權限。
- 錄音的權限名字：  
android.permission.RECORD\_AUDIO
- 儲存至SD卡：  
android.permission.WRITE\_EXTERNAL\_STORAGE

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

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

# UI介面



國立宜蘭大學

National Ilan University NIU



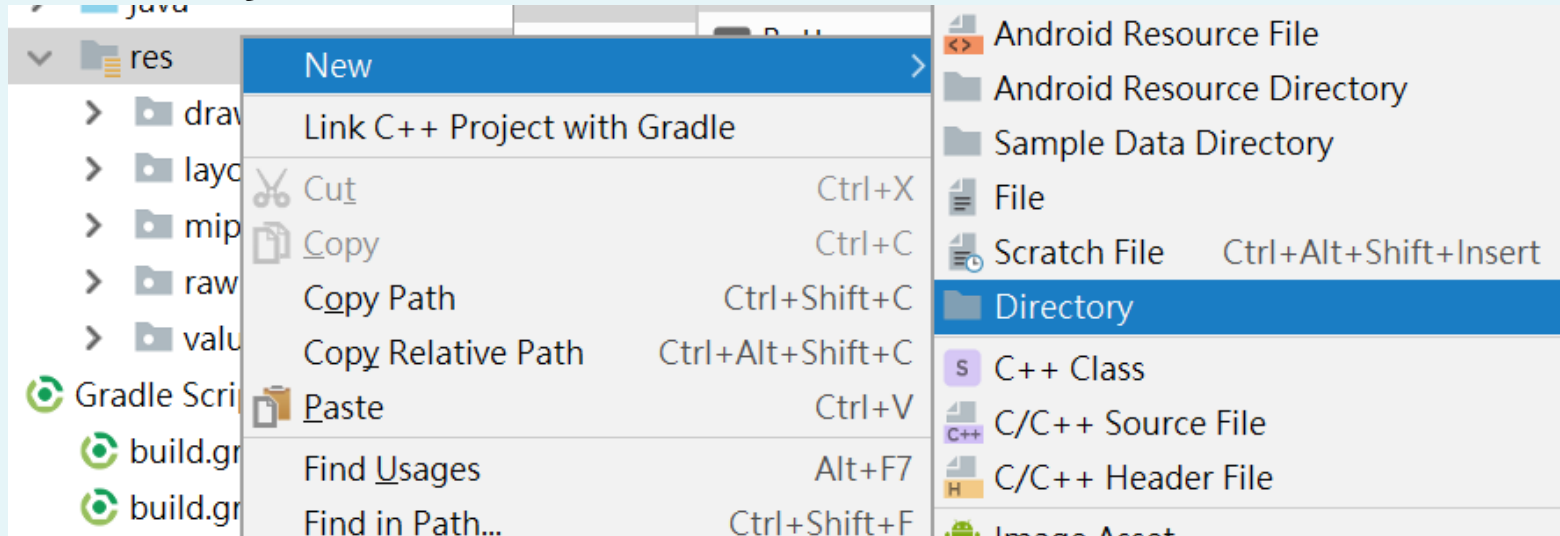
# activity\_main.xml

- 步驟1. 設定Layout成RelativeLayout。
- 步驟2. 拉五個Button元件。
- 步驟3. 將按鈕的text設定成與UI介面相同。

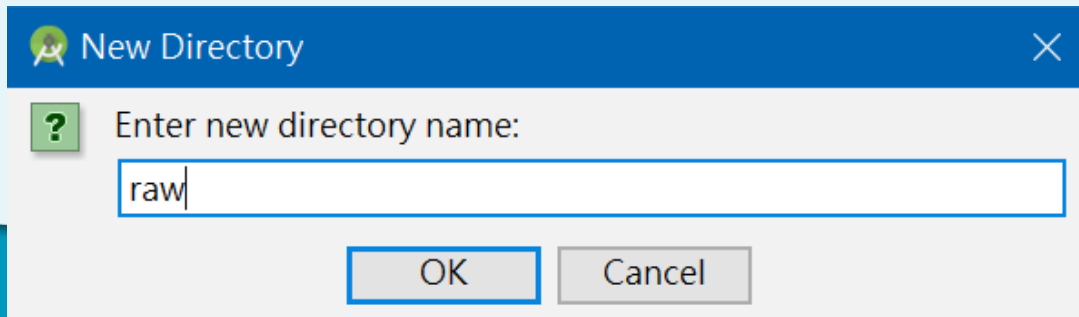


# 開始打程式 MainActivity.java

- 步驟1. 在res新增資料夾，右鍵 new > directory

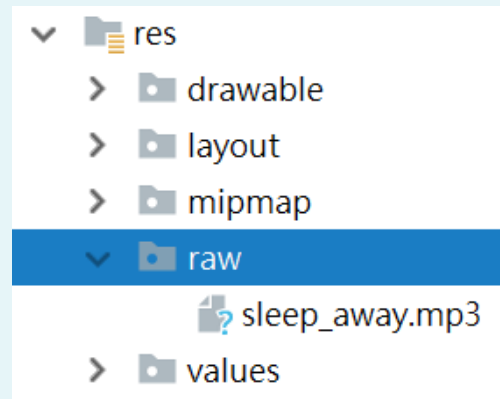


- 步驟2. 命名為raw。



# 開始打程式 MainActivity.java

- 步驟3. 載入音樂檔(助教提供)，複製貼上到raw



- 步驟4. 宣告UI元件。

```
Button mButton1, mButton2, mButton3, mButton4, mButton5;
```

- 步驟5. 宣告錄音和播放的物件。

```
MediaPlayer mp;  
MediaRecorder mr = new MediaRecorder();
```

# 開始打程式MainActivity.java

- 步驟6. 宣告按鈕轉變的布林值。

```
Boolean b = false;
```

- 步驟7. 宣告權限結果代號

```
private static final int MY_PERMISSIONS_REQUEST_CODE = 100;
```

- 步驟8. 建立程式與元件的關係。

```
btn1 = (Button) findViewById(R.id.button1);  
btn2 = (Button) findViewById(R.id.button2);  
btn3 = (Button) findViewById(R.id.button3);  
btn4 = (Button) findViewById(R.id.button4);  
btn5 = (Button) findViewById(R.id.button5);
```

# 開始打程式 MainActivity.java

- 步驟9. 宣告兩個變數內容是確定是否得到權限

```
int mPermissionRecord =  
ContextCompat.checkSelfPermission(  
    MainActivity.this,  
    Manifest.permission.RECORD_AUDIO);
```

```
int mPermissionWriteExternal = ContextCompat.checkSelfPermission(  
    MainActivity.this,  
    Manifest.permission.WRITE_EXTERNAL_STORAGE);
```

- 步驟10. 判斷是否得到權限

```
if (mPermissionRecord == PackageManager.PERMISSION_DENIED ||  
    mPermissionWriteExternal ==  
    PackageManager.PERMISSION_DENIED)
```

# 開始打程式 MainActivity.java

- 步驟9. 宣告兩個變數內容是確定是否得到權限

```
int mPermissionRecord =ContextCompat.checkSelfPermission(  
    MainActivity.this,  
    Manifest.permission.RECORD_AUDIO);
```

```
int mPermissionWriteExternal = ContextCompat.checkSelfPermission(  
    MainActivity.this,  
    Manifest.permission.WRITE_EXTERNAL_STORAGE);
```

- 步驟10. 判斷是否得到權限

```
if (mPermissionRecord == PackageManager.PERMISSION_DENIED ||  
    mPermissionWriteExternal ==PackageManager.PERMISSION_DENIED)
```



# 開始打程式 MainActivity.java

- 步驟11. 未取得權限則向使用者要求權限。

```
ActivityCompat.requestPermissions(MainActivity.this,  
    new String[]{Manifest.permission.RECORD_AUDIO,  
        Manifest.permission.WRITE_EXTERNAL_STORAGE},  
    MY_PERMISSIONS_REQUEST_CODE);
```

- 步驟12. 已取得權限則設定錄音音訊來源為手機的麥克風及輸出格式和音訊編碼。

```
mr.setAudioSource(MediaRecorder.AudioSource.MIC);  
mr.setOutputFormat(MediaRecorder.OutputFormat.DEFAULT);  
mr.setAudioEncoder(MediaRecorder.AudioSource.DEFAULT);
```



# 開始打程式 MainActivity.java

- 步驟13. 宣告錄音檔存的位置跟檔名。

```
final String recordName = "/sdcard/record.3gp";
```

- 步驟14. 建立五個按鈕的監聽事件。

```
mButton1.setOnClickListener(new Button.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        |  
    }  
});
```

# 開始打程式 MainActivity.java

- 步驟13. 宣告錄音檔存的位置跟檔名。

```
final String recordName = "/sdcard/record.3gp";
```

- 步驟14. 建立五個按鈕的監聽事件。

```
mButton1.setOnClickListener(new Button.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        |  
    }  
});
```



# 開始打程式 MainActivity.java

- 步驟9. 在播放的按鈕中建立播放檔，將播放設定成循環播放，如果已經設定好播放檔，就會進行播放音樂。

```
if (mp == null) {  
    mp = new MediaPlayer();  
    mp = MediaPlayer.create(context: MainActivity.this,  
        R.raw.sleep_away);  
    mp.setLooping(true);  
} else {  
    mp.start();  
}
```

- 步驟10. 在暫停的按鈕中讓音樂暫停播放。

```
if (mp != null)  
    mp.pause();
```

# 開始打程式 MainActivity.java

- 步驟11. 在停止按鈕設定將播放檔釋放資源，並設定為null，以方便播放按鈕重新建立播放檔。

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

- 步驟12. 在onPause()生命週期中也放入釋放資源，讓離開程式時可以把音樂停止，否則會關閉程式還是有音樂在播放。

```
if (mp != null)  
    mp.release();
```

# 完整播放按鈕程式碼

```
mButton1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        if (mp == null) {  
            mp = new MediaPlayer();  
            mp = MediaPlayer.create(context: MainActivity.this,  
                                    R.raw.sleep_away);  
            mp.setLooping(true);  
        } else {  
            mp.start();  
        }  
    }  
});
```

# 完整暫停按鈕程式碼

```
mButton2.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        if (mp != null) {  
            mp.pause();  
        }  
    }  
});
```

# 完整停止按鈕程式碼

```
mButton3.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        if (mp != null) {  
            mp.release();  
            mp = null;  
        }  
    }  
});
```



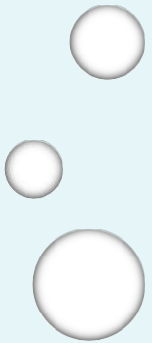
# 完整onPause()程式碼

```
@Override
protected void onPause() {
    // TODO Auto-generated method stub
    super.onPause();
    if (mp != null)
        mp.release();
}
```



# 開始打程式 MainActivity.java

- 步驟13. 在錄音按鈕設定錄音的路徑，並以 `prepare()` 讓錄音環境建立好在開始錄製，再將按鈕的 `text` 改成“儲存”，再按下第二次會再改成“錄音”，並停止錄音。





# 開始打程式 MainActivity.java

```
if (!b) {  
    try {  
        mr.setOutputFile(recordName);  
        mr.prepare();  
        mr.start();  
    } catch (java.io.IOException e) {  
        e.printStackTrace();  
    }  
  
    mButton4.setText("存檔");  
    b = true;  
} else {  
    try {  
        mr.stop();  
    } catch (IllegalStateException e) {  
        //no premission to store  
    }  
    mButton4.setText("錄音");  
    b = false;  
}
```

prepare在建立時會需要  
try{}  
catch{}來確認準備狀態，  
此 try catch只要輸入完後  
出現紅色底線，利用系統給予的  
方式建立即可。



# 開始打程式 MainActivity.java

- 步驟14. 最後將錄音檔播放出來，播放音樂也可以使用準備程序，當準備好再進行播放。

```
MediaPlayer mp = new MediaPlayer();  
try {  
    mp.setDataSource(recordName);  
    mp.prepare();  
    mp.start();  
  
} catch (java.io.IOException e) {  
    e.printStackTrace();  
}
```

# 完整錄音按鈕程式碼

```
mButton4.setOnClickListener((view) -> {  
    if (!b) {  
        try {  
            mr.setOutputFile(recordName);  
            mr.prepare();  
            mr.start();  
        } catch (java.io.IOException e) {  
            e.printStackTrace();  
        }  
  
        mButton4.setText("存檔");  
        b = true;  
    } else {  
        try {  
            mr.stop();  
        } catch (IllegalStateException e) {  
            //no premission to store  
        }  
        mButton4.setText("錄音");  
        b = false;  
    }  
});
```

# 完整播放錄音檔按鈕程式碼

```
mButton5.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        MediaPlayer mp = new MediaPlayer();  
        try {  
            mp.setDataSource(recordName);  
            mp.prepare();  
            mp.start();  
        } catch (java.io.IOException e) {  
            e.printStackTrace();  
        }  
    }  
});
```

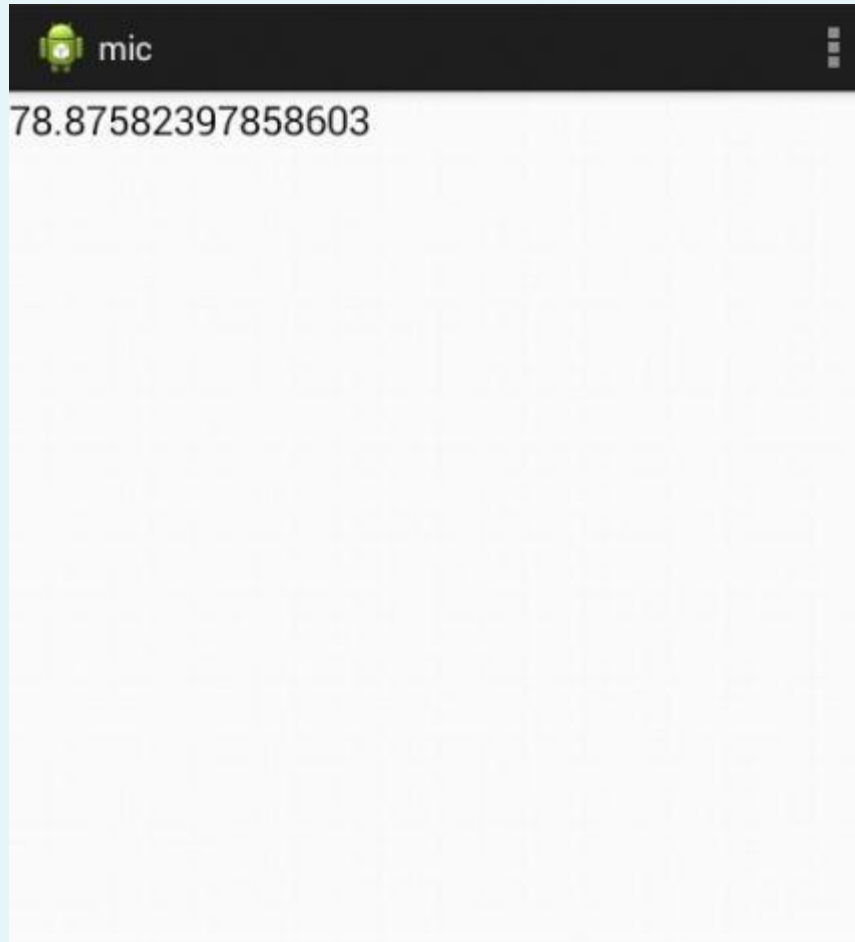


國立宜蘭大學

National Ilan University NIU

# 分貝感測

# 分貝感測

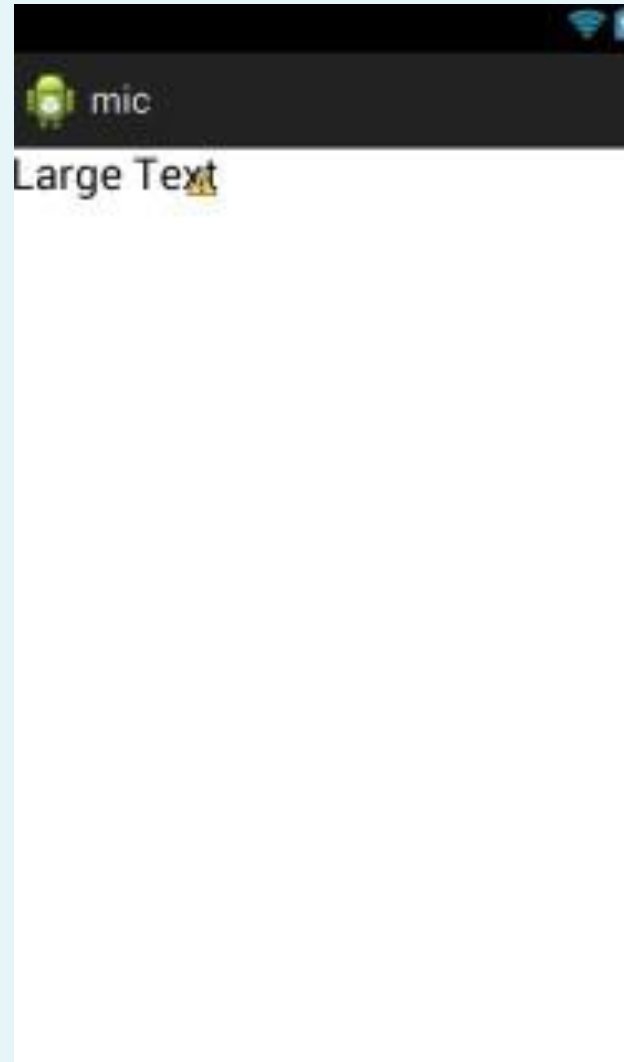




# 分貝感測

- 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





# 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

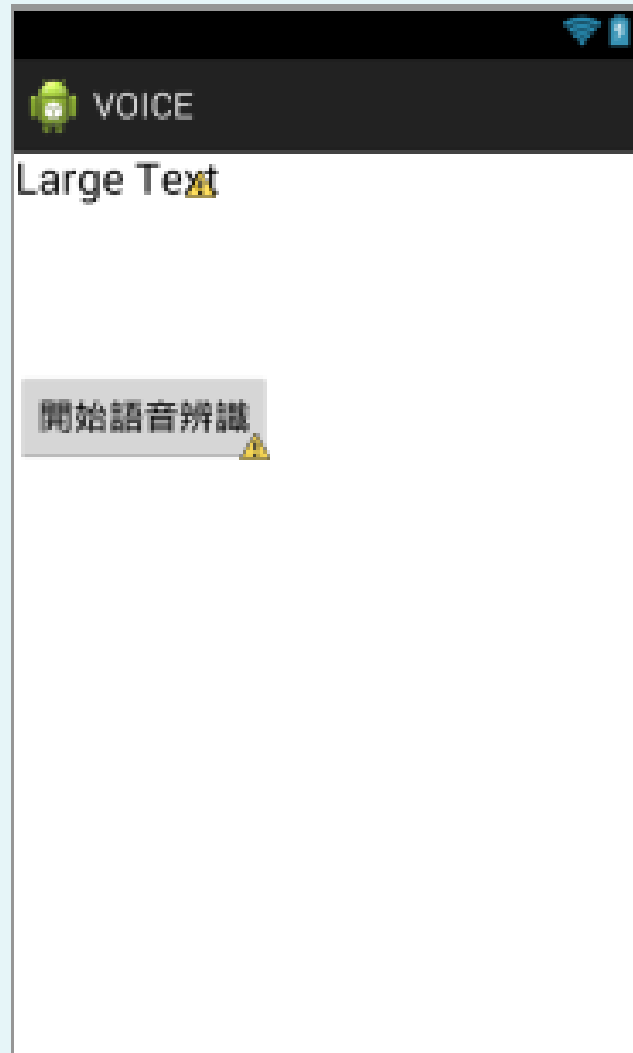
開始語音辨識

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);
        }
    }
}
```