

# Android App

## 程式設計教本 之無痛起步

引領入門

最簡單、最易懂的初學教材



## 第 11 章

# 播放音樂與影片

本投影片（下稱教用資源）僅授權給採用教用資源相關之旗標書籍為教科書之授課老師（下稱老師）專用，老師為教學使用之目的，得摘錄、編輯、重製教用資源（但使用量不得超過各該教用資源內容之80%）以製作為輔助教學之教學投影片，並於授課時搭配旗標書籍公開播放，但不得為網際網路公開傳輸之遠距教學、網路教學等之使用；除此之外，老師不得再授權予任何第三人使用，並不得將依此授權所製作之教學投影片之相關著作物移作他用。

# 前言

- 11-1 使用 Intent 來選取音樂或影片
- 11-2 用 MediaPlayer 播放音樂
- 11-3 用 VideoView 播放影片

# 11-1 使用 Intent 來選取音樂或影片

```
01 //選取音樂
02 Intent it = new Intent(Intent.ACTION_GET_CONTENT);
03 it.setType( "audio/*" );           ← 要選取所有音樂類型
04 startActivityForResult(it, 100);   ← 以識別編號 100 來啟動外部程式
05
06 //選取影片
07 Intent it = new Intent(Intent.ACTION_GET_CONTENT);
08 it.setType( "video/*" );          ← 要選取所有影片類型
09 startActivityForResult(it, 101);  ← 以識別編號 101 來啟動外部程式
```

# 取預存在程式中的多媒體檔案

```
//假設是要存取專案中 \res\raw\test.mp3 檔  
uri = Uri.parse("android.resource://" + getPackageName() + "/" +  
                R.raw.test);
```

# 範例11-1：讓使用者挑選影音檔

1 按此鈕來選取音樂



顯示選取的檔案名稱，預設為程式內建的音樂檔

顯示檔案的路徑

2 如果出現交談窗，請選擇選取音樂曲目程式

3 選取任一首音樂



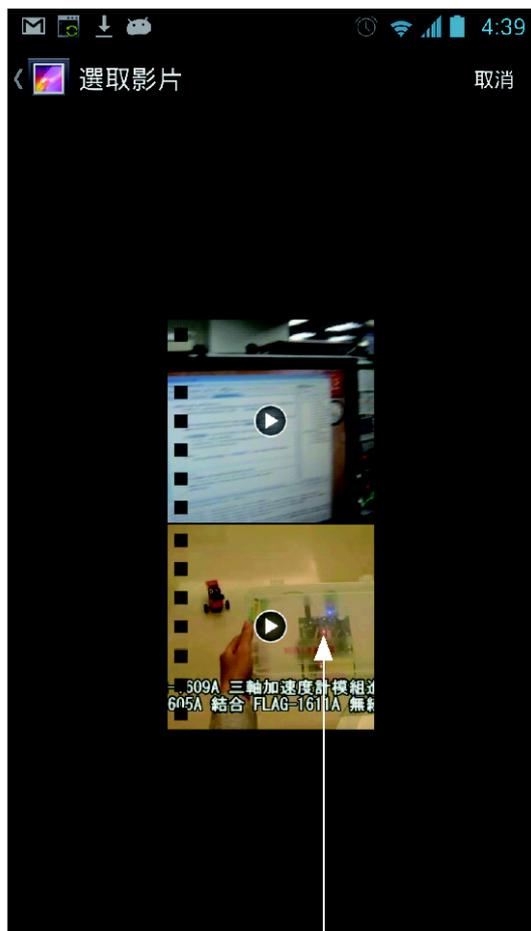
4 按此鈕確定

# 讓使用者挑選影音檔

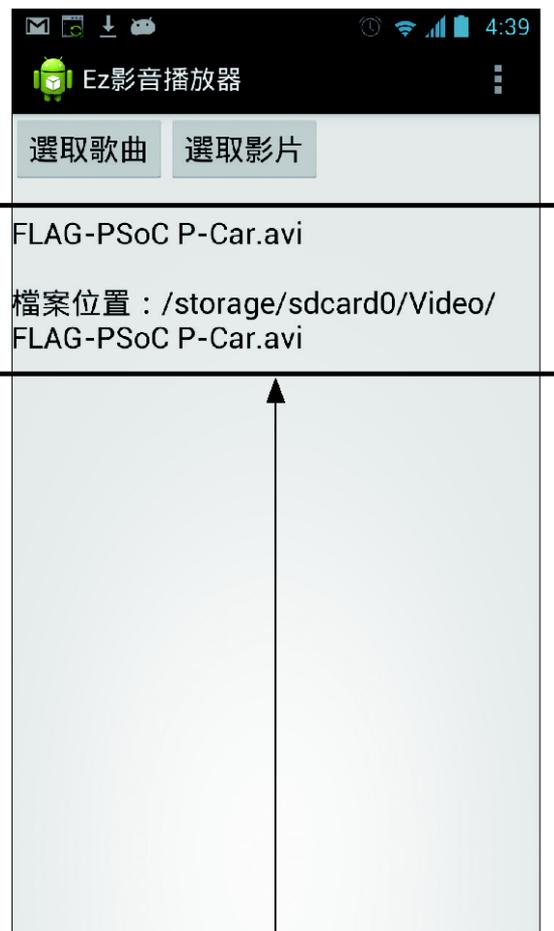
7 如果出現交談窗, 請選擇圖片庫(或媒體匣) 程式



# 讓使用者挑選影音檔



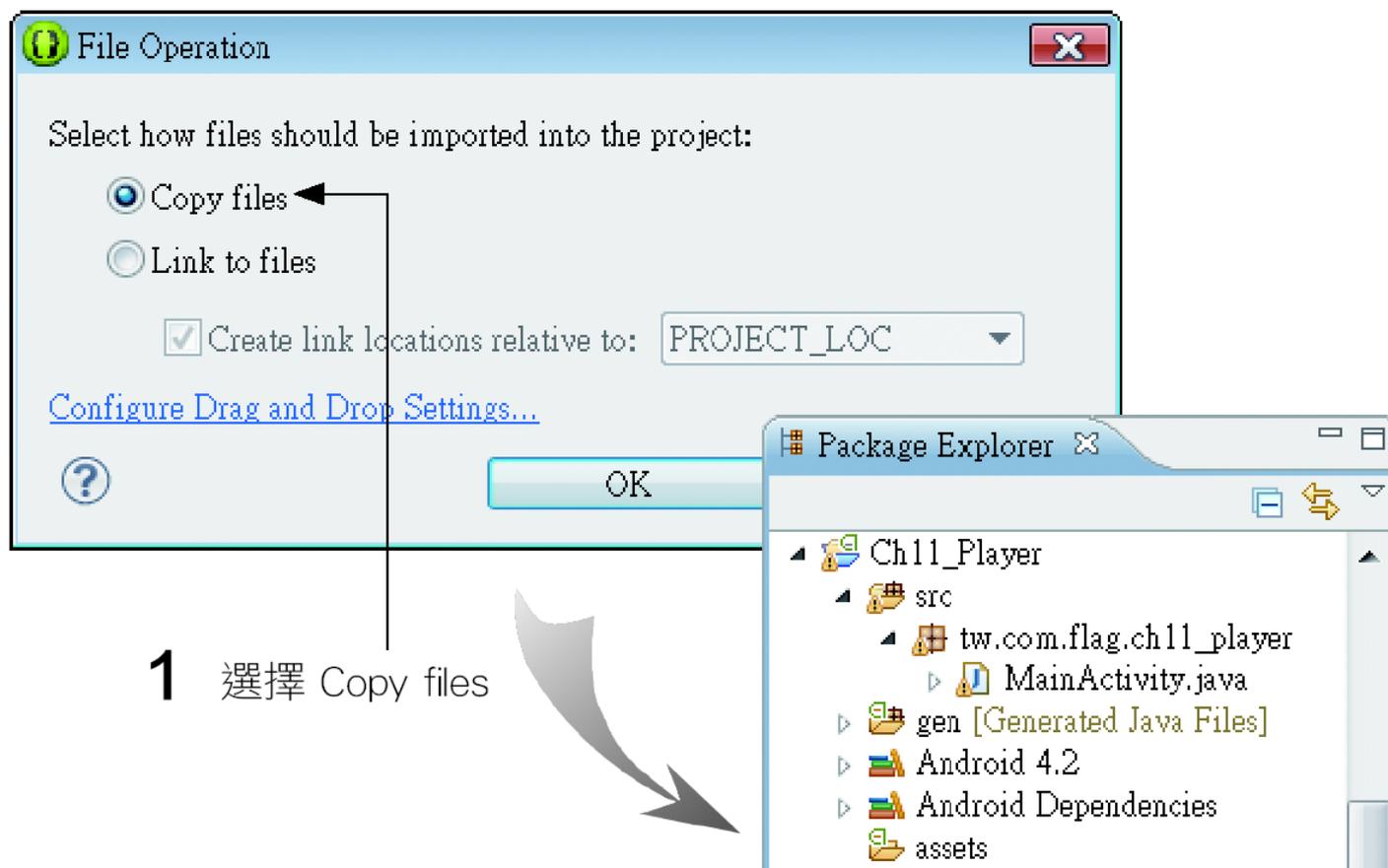
9 選取任一部影片



10 顯示選取影片的檔名及路徑

# 讓使用者挑選影音檔

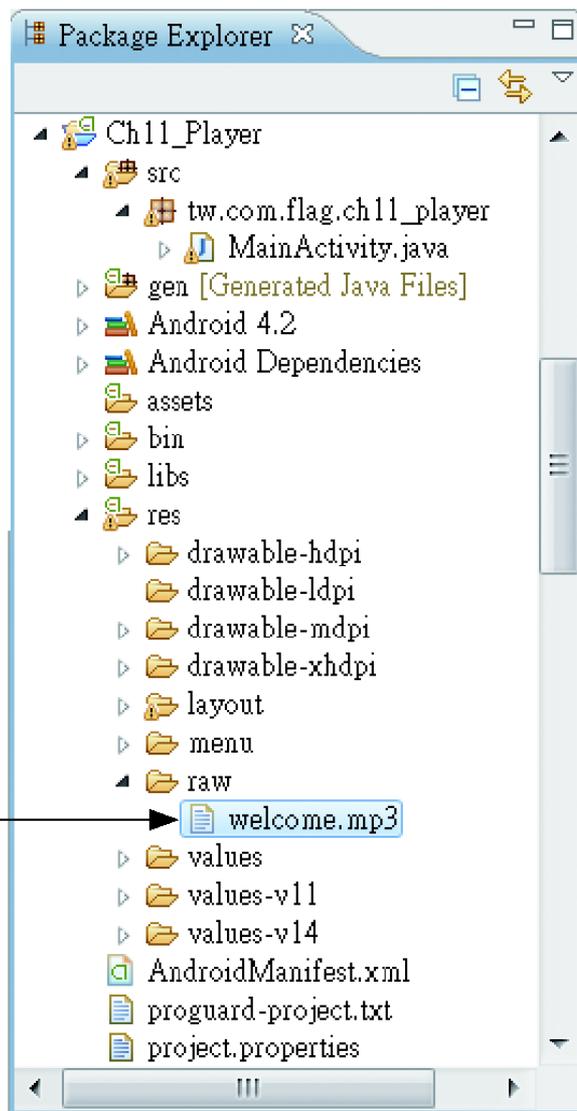
## step 1



# 讓使用者挑選影音檔

## step 2

2 已複製到 \res\raw 中了



# 讓使用者挑選影音檔

## step 3

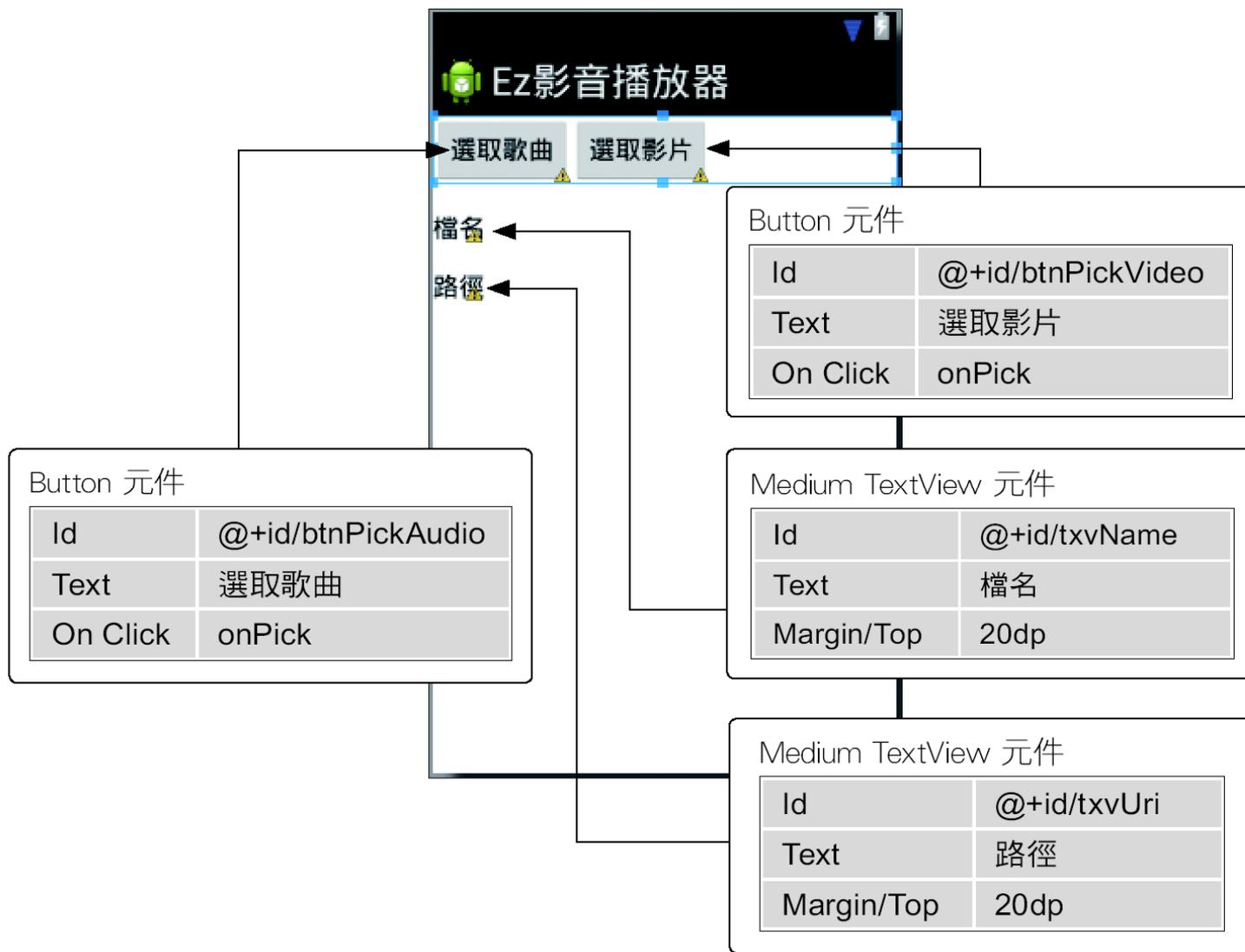
1 加入 1 個 LinearLayout(Horizontal)



2 在 LinearLayout 之內加入 2 個 Button 元件

3 在 LinearLayout 之下加入 2 個 Medium TextView 元件

# 讓使用者挑選影音檔



# 讓使用者挑選影音檔

## step 4

```
01 public class MainActivity extends Activity{  
02     Uri uri;           ← 儲存影音檔案的 Uri  
03     TextView txvName, txvUri;  
04     boolean isVideo = false; ← 記錄是否為影片檔 (否則為音樂檔)  
  
05  
06     @Override
```

Next

# 讓使用者挑選影音檔

```
07  protected void onCreate(Bundle savedInstanceState) {
08      super.onCreate(savedInstanceState);
09      setContentView(R.layout.activity_main);
10
11      //設定螢幕不隨手機旋轉、以及畫面直向顯示
12      setRequestedOrientation(ActivityInfo.
          SCREEN_ORIENTATION_NOSENSOR); ← 設定螢幕不隨手機旋轉
13      setRequestedOrientation(ActivityInfo.
          SCREEN_ORIENTATION_PORTRAIT); ← 設定螢幕直向顯示
14
15      txvName = (TextView)findViewById(R.id.txvName); ← 取得"檔案"文字元件
16      txvUri = (TextView)findViewById(R.id.txvUri); ← 取得"路徑"文字元件
```

# 讓使用者挑選影音檔

```
17
18     uri = Uri.parse("android.resource://" +
19                     packageName() + "/" + R.raw.welcome); ←
                                     預設會播放程式內的音樂檔 welcome.mp3
20     txvName.setText("welcome.mp3"); ← 在畫面中顯示檔名
21     txvUri.setText("程式內的樂曲：" + uri.toString()); ← 顯示內建音樂檔
                                                         的 Uri 路徑
22 }
```

# 讓使用者挑選影音檔

## step 5

```
01 public void onPick(View v) {
02     Intent it = new Intent(Intent.ACTION_GET_CONTENT ); ← 建立動作為 "選取內容" 的 Intent
03     if(v.getId() == R.id.btnPickAudio) { ← 如果是 "選取歌曲" 鈕
04         it.setType( "audio/*" ); ← 要選取所有音樂類型
05         startActivityForResult(it, 100); ← 以識別編號 100 來啟動外部程式
06     }
07     else { ← 否則就是 "選取影片" 鈕
08         it.setType( "video/*" ); ← 要選取所有影片類型
09         startActivityForResult(it, 101); ← 以識別編號 101 來啟動外部程式
10     }
11 }
```

# 讓使用者挑選影音檔

## step 6

```
01 protected void onActivityResult (int requestCode, int resultCode,
02                                 Intent data) {
03     super.onActivityResult(requestCode, resultCode, data);
04
05     if(resultCode == Activity.RESULT_OK) { ← 如果選取成功
06         isVideo = (requestCode == 101); ← 記錄是否選取了影片檔
                                                (當識別碼為 101 時)
07         uri = convertUri(data.getData()); ← 取得選取相片的 Uri
                                                並做 Uri 格式轉換
08         txvName.setText(uri.getLastPathSegment ()); ← 顯示檔名 (Uri
                                                        最後一段文字)
09         txvUri.setText("檔案位置：" + uri.getPath()); ← 顯示檔案的路徑
```

# 讓使用者挑選影音檔

```
10 }
11 }
12
13 Uri convertUri(Uri uri) { ← 將 "content://" 類型的 Uri 轉換為
    "file://" 的 Uri
14     if(uri.toString().substring(0, 7).equals("content")) { ←
        如果是以 "content" 開頭
15         String[] colName = { MediaColumns.DATA }; ← 宣告要查詢的欄位
16         Cursor cursor = getContentResolver() ← 以 uri 進行查詢
17             .query(uri, colName, null, null, null);
18         cursor.moveToFirst(); ← 移到查詢結果的第一筆記錄
19         uri = Uri.parse("file://" + cursor.getString(0)); ← 將路徑轉為 Uri
20     }
21     return uri; ← 傳回 Uri 物件
22 }
```

# 11-2 用 MediaPlayer 播放音樂

- MediaPlayer 的音樂播放流程
- MediaPlayer 可引發的 3 個重要事件
- 處理在播放音樂時切換到其他程式的狀況
- 讓螢幕不進入休眠狀態

# MediaPlayer 的音樂播放流程

1.

```
mper = new MediaPlayer();
```

 ← 建立 MediaPlayer 物件

2.

```
mper.reset();
```

 ← 如果之前有播放過其他音樂，則要先 reset  

```
mper.setDataSource(this, uri);
```

 ← 指定音樂檔的 Uri  

```
mper.prepareAsync();
```

 ← 準備播放，當準備好時會引發一個『音樂準備好了』的事件(稍後會介紹)

# MediaPlayer 的音樂播放流程

## 3.

```
mper.start();    ← 開始播放  
mper.pause();   ← 暫停播放  
mper.stop();    ← 停止播放  
mper.setLooping(true); ← 設定是否要重複播放 (true 為要)
```

## 4.

```
int len = mper.getDuration(); ← 取得音樂的總長度 (秒數)  
int pos = mper.getCurrentPosition(); ← 取得目前的播放位置 (秒數)  
mper.seekTo(pos); ← 移動播放位置到第 pos 秒的位置
```

# MediaPlayer 的音樂播放流程

5.

```
mper.release(); ← 釋放 MediaPlayer 物件
```

# MediaPlayer 可引發的 3 個重要事件

1.

```
public class MainActivity extends Activity implements  
    MediaPlayer.OnPreparedListener,      ← 音樂準備好時  
    MediaPlayer.OnErrorListener,        ← 發生錯誤時  
    MediaPlayer.OnCompletionListener { ← 播放完畢時
```

2.

```
mper.setOnPreparedListener(this); ← 設定音樂準備好時的事件監聽器  
mper.setOnErrorListener(this); ← 設定發生錯誤時的事件監聽器  
mper.setOnCompletionListener(this); ← 設定播放完畢時的事件監聽器
```

# MediaPlayer 可引發的 3 個重要事件

## 3.

```
public void onPrepared(MediaPlayer arg0) {    ← 執行 prepareAsync()
    //音樂準備好時要做的事情...              後，當音樂準備好了即會
}                                              引發此方法

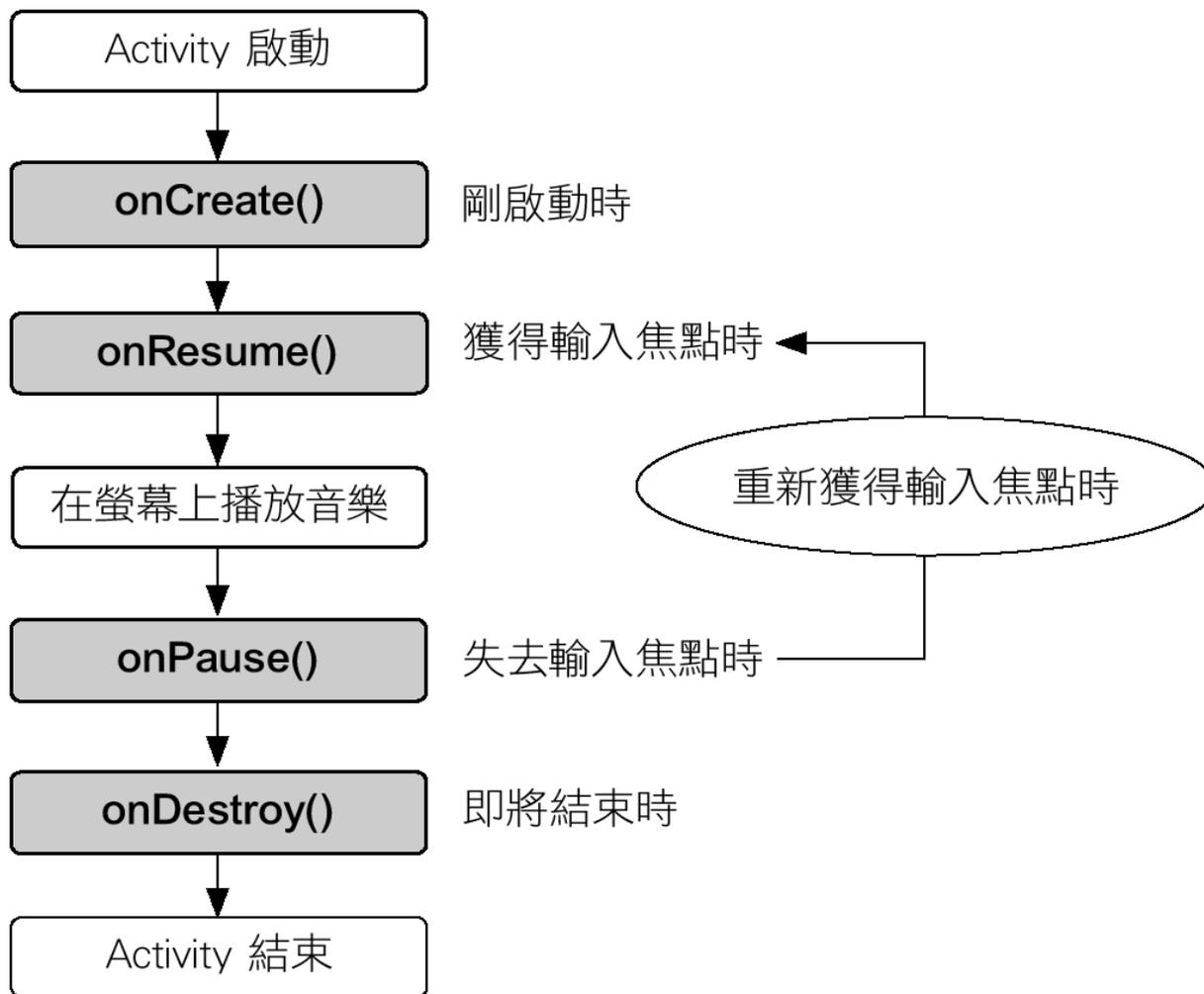
public boolean onError(MediaPlayer arg0, int arg1, int arg2) {
    //發生錯誤時要做的事情...
    return true; ← 傳回 true 表示錯誤已處理了
}

public void onCompletion(MediaPlayer arg0) {
    //播放完畢時要做的事情...
}
```

# 處理在播放音樂時切換到其他程式的狀況

事件	說明
onCreate()	當 Activity 啟動時
onResume()	當 Activity 獲得輸入焦點時
onPause()	當 Activity 失去輸入焦點時 (例如切換到手機的首頁或其他程式)
onDestroy()	當 Activity 結束時

# 處理在播放音樂時切換到其他程式的狀況



# 處理在播放音樂時切換到其他程式的狀況

```
01 protected void onPause() {
02     super.onPause(); ← 執行父類別的事件方法以處理必要事宜
03
04     //如果正在播放，就暫停...
05 }
06
07 protected void onDestroy() {
08     //釋放 MediaPlayer 物件...
09
10     super.onDestroy(); ← 執行父類別的事件方法以處理必要事宜
11 }
```

# 讓螢幕不進入休眠狀態

```
01 protected void onCreate(Bundle savedInstanceState) {
02     super.onCreate(savedInstanceState);
03     setContentView(R.layout.activity_main);
04
05     //設定螢幕不隨手機旋轉、畫面要直向顯示、以及螢幕不進入休眠
06     setRequestedOrientation(ActivityInfo.
                                SCREEN_ORIENTATION_NOSENSOR); ← 設定螢幕不
                                                                隨手機旋轉
07     setRequestedOrientation(ActivityInfo.
                                SCREEN_ORIENTATION_PORTRAIT); ← 設定螢幕
                                                                直向顯示
08     getWindow().addFlags(WindowManager.LayoutParams.
                                FLAG_KEEP_SCREEN_ON); ← 設定螢幕不進入休眠
09     ...
10 }
```

# 範例11-2：用 MediaPlayer 播放音樂

1 按播放鈕開始播放音樂



還沒播放時停止鈕不能按

2 播放鈕變成顯示"暫停", 請再按一下來暫停播放



停止鈕可以按了

# 用 MediaPlayer 播放音樂



- 3** 播放鈕變成顯示"繼續",  
請再按一下來繼續播放

方向及移動後的位置  
(目前秒數/全部秒數)



- 4** 可用這二個按鈕向前  
或向後移動 10 秒

# 用 MediaPlayer 播放音樂



當音樂播完時, 又會  
回到播放前的狀態

# 用 MediaPlayer 播放音樂

## step 1

## step 2

- 1 加入 1 個 LinearLayout(Horizontal), 在其中加入播放鈕、停止鈕、及重複播放多選鈕

LinearLayout	
Margin/Top	20dp
Background	#eee

LinearLayout(表5)	
Gravity	right (向右靠邊)

- 2 再加入 1 個 LinearLayout(Horizontal), 在其內加入後退 10 秒及前進 10 秒二個 ImageButton。

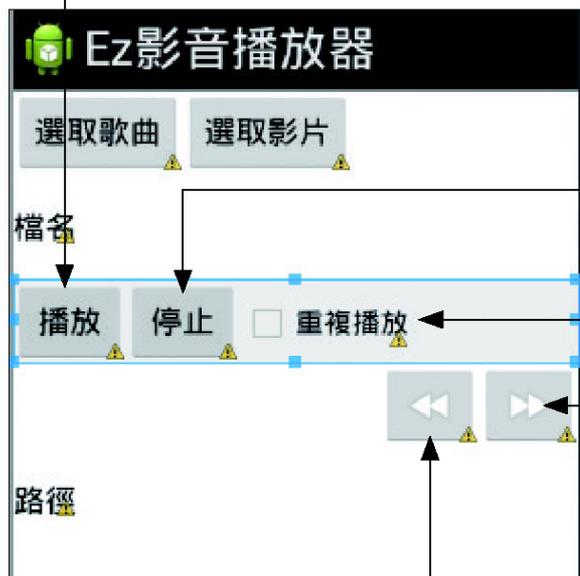
# 用 MediaPlayer 播放音樂

Button 元件

Id	@+id/btnPlay
Text	播放
On Click	onMpPlay

Button 元件

Id	@+id/btnStop
Text	停止
On Click	onMpStop



CheckBox 元件

Id	@+id/ckbLoop
Text	重複播放
On Click	onMpLoop
Checked	false

ImageButton 元件

Id	@+id/igbForward
Src	@android:drawable/ic_media_ff
On Click	onMpForward

ImageButton 元件

Id	@+id/igbBackward
Src	@android:drawable/ic_media_rew
On Click	onMpBackward

# 用 MediaPlayer 播放音樂

## step 3

```
01 public class MainActivity extends Activity implements
02     MediaPlayer.OnPreparedListener,
03     MediaPlayer.OnErrorListener,
04     MediaPlayer.OnCompletionListener {
05     Uri uri;           ← 儲存影音檔案的 Uri
06     TextView txvName, txvUri; ← 用來參照到畫面中的元件
07     boolean isVideo = false; ← 用來記錄是否為影片檔
08
09     Button btnPlay, btnStop; ← 用來參照播放鈕、停止鈕
10     CheckBox ckbLoop; ← 用來參照重複播放多選鈕
11     MediaPlayer mper; ← 用來參照 MediaPlayer 物件
12     Toast tos; ← 用來參照 Toast 物件 (顯示訊息之用)
```

# 用 MediaPlayer 播放音樂

## step 4

```
01 protected void onCreate(Bundle savedInstanceState) {
02     super.onCreate(savedInstanceState);
03     setContentView(R.layout.activity_main);
04
05     //設定螢幕不隨手機旋轉、畫面要直向顯示、以及螢幕不進入休眠
06     setRequestedOrientation(ActivityInfo.
                                SCREEN_ORIENTATION_NOSENSOR); ← 設定螢幕不
                                                                隨手機旋轉
07     setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_
                                PORTRAIT); ← 設定螢幕直向顯示
08     getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_
                                ON); ← 設定螢幕不進入休眠
09
10     txvName = (TextView)findViewById(R.id.txvName); ← 取得第1個文字元件
11     txvUri = (TextView)findViewById(R.id.txvUri); ← 取得第2個文字元件
```

# 用 MediaPlayer 播放音樂

```
12 btnPlay = (Button)findViewById(R.id.btnPlay);    ← 取得播放鈕
13 btnStop = (Button)findViewById(R.id.btnStop);   ← 取得停止鈕
14 ckbLoop = (CheckBox)findViewById(R.id.ckbLoop); ← 取得重複播放多選鈕
15
16 uri = Uri.parse("android.resource://" + ← 預設會播放程式內的音樂檔
17             getPackageName() + "/" + R.raw.welcome);
18 txvName.setText("welcome.mp3");                ← 在畫面中顯示檔名
19 txvUri.setText("程式內的樂曲：" + uri.toString()); ← 顯示 Uri
20
21 mper = new MediaPlayer();                        ← 建立 MediaPlayer 物件
22 mper.setOnPreparedListener(this);
23 mper.setOnErrorListener(this);                 ← 設定 3 個事件監聽器
24 mper.setOnCompletionListener(this);
25 tos = Toast.makeText(this, "", Toast.LENGTH_SHORT); ← 建立 Toast 物件
26
27 prepareMusic();    ← 準備預設音樂 (welcome.mp3) 的播放
28 }
```

# 用 MediaPlayer 播放音樂

## step 5

```
01 void prepareMusic() {
02     btnPlay.setText("播放");      ← 將按鈕文字顯示為 "播放"
03     btnPlay.setEnabled(false);   ← 使播放鈕不能按 (要等準備好才能按)
04     btnStop.setEnabled(false);   ← 使停止鈕不能按
05
06     try {
07         mper.reset();             ← 如果之前有播過歌, 必須 reset 後才能換歌
08         mper.setDataSource(this, uri); ← 指定音樂來源
09         mper.setLooping(ckbLoop.isChecked()); ← 設定是否重複播放
10         mper.prepareAsync();      ← 要求 MediaPlayer 準備播放指定的音樂
11     } catch (Exception e) {      ← 攔截錯誤並顯示訊息
12         tos.setText("指定音樂檔錯誤!" + e.toString());
13         tos.show();
14     }
15 }
```

# 用 MediaPlayer 播放音樂

## step 6

```
01 @Override
02 public void onPrepared(MediaPlayer arg0) {
03     btnPlay.setEnabled(true);    ← 當準備好時, 讓播放鈕有作用(可以按)
04 }
05
06 @Override
07 public void onCompletion(MediaPlayer arg0) {    ← 當音樂播完畢時
08     mper.seekTo(0);                ← 將播放位置歸 0
09     btnPlay.setText("播放");    ← 讓播放鈕顯示 "播放"
10     btnStop.setEnabled(false); ← 讓停止鈕禁用(不能按), 因為音樂已經停止播放
```

# 用 MediaPlayer 播放音樂

```
11 }
12
13 @Override
14 public boolean onError(MediaPlayer arg0, int arg1, int arg2) {
15     tos.setText("發生錯誤，停止播放"); ← 當發生錯誤時，顯示錯誤訊息
16     tos.show();
17     return true;
18 }
```

# 用 MediaPlayer 播放音樂

## step 7

```
01 protected void onActivityResult (int requestCode, int resultCode,
    Intent data) {
02     super.onActivityResult(requestCode, resultCode, data);
03
04     if(resultCode == Activity.RESULT_OK) {
05         isVideo = (requestCode == 101); ←
                記錄是否選取了影片檔 (當識別碼為101時)
06         uri = convertUri(data.getData()); ←
                取得選取檔案的 Uri 並做 Uri 格式轉換
07         txvName.setText(uri.getLastPathSegment ()); ←
                顯示檔名 (Uri 最後的一段文字)
08         txvUri.setText("檔案位置：" + uri.getPath()); ← 顯示檔案的路徑
09         if(!isVideo) prepareMusic(); ← 如果是音樂檔就準備播放
10     }
11 }
```

# 用 MediaPlayer 播放音樂

## step 8

```
01 public void onMpPlay(View v) { ← 按下播放鈕時
02     if (mper.isPlaying()) { ← 如果正在播, 就暫停
03         mper.pause(); ← 暫停播放
04         btnPlay.setText("繼續");
05     }
06     else { ← 如果沒有在播, 就開始播
07         mper.start(); ← 開始播放
08         btnPlay.setText("暫停"); ← 讓播放鈕顯示 "暫停"
09         btnStop.setEnabled(true); ← 音樂已經播放, 所以讓停止鈕有作用
10     }
```

Next

# 用 MediaPlayer 播放音樂

```
11 }
12
13 public void onMpStop(View v) { ← 按下停止鈕時
14     mper.pause(); ← 暫停播放
15     mper.seekTo(0); ← 移到音樂中 0 秒的位置
16     btnPlay.setText("播放");
17     btnStop.setEnabled(false); ← 讓停止鈕不能再按了
18 }
19
20 public void onMpLoop(View v) { ← 按下重複播放多選鈕時
21     if (ckbLoop.isChecked())
22         mper.setLooping(true); ← 設定要重複播放
23     else
24         mper.setLooping(false); ← 設定不要重複播放
```

# 用 MediaPlayer 播放音樂

```
25 }
26
27 public void onMpBackward(View v) { ← 按下倒退圖形鈕時
28     if(!btnPlay.isEnabled()) return; ←
                如果還沒準備好(播放鈕不能按), 則不處理
29     int len = mper.getDuration(); ← 讀取音樂長度
30     int pos = mper.getCurrentPosition(); ← 讀取目前播放位置
31     pos -= 10000; ← 倒退 10 秒 (10000ms)
32     if(pos < 0) pos = 0; ← 不可小於 0
33     mper.seekTo(pos); ← 移動播放位置
34     tos.setText("倒退10秒:" + pos/1000 + "/" + len/1000); ← 顯示訊息
35     tos.show();
36 }
37
38 public void onMpForward(View v) { ← 按下前進圖形鈕時
```

# 用 MediaPlayer 播放音樂

```
39     if(!btnPlay.isEnabled()) return;           ←
                                                如果還沒準備好(播放鈕不能按), 則不處理
40     int len = mper.getDuration();              ← 讀取音樂長度
41     int pos = mper.getCurrentPosition();      ← 讀取目前播放位置
42     pos += 10000;                             ← 前進 10 秒 (10000ms)
43     if(pos > len) pos = len;                  ← 不可大於總秒數
44     mper.seekTo(pos);                          ← 移動播放位置
45     tos.setText("前進10秒:" + pos/1000 + "/" + len/1000); ← 顯示訊息
46     tos.show();
47 }
```

# 用 MediaPlayer 播放音樂

## step 9

```
01 @Override
02 protected void onPause() {
03     super.onPause();    ← 執行父類別的事件方法以處理必要事宜
04
05     if (mper.isPlaying()) { ← 如果正在播，就暫停
06         btnPlay.setText("繼續");
07         mper.pause(); ← 暫停播放
08     }
09 }
10
11 @Override
12 protected void onDestroy() {
13     mper.release(); ← 釋放 MediaPlayer 物件
14     super.onDestroy(); ← 執行父類別的事件方法以處理必要事宜
15 }
```

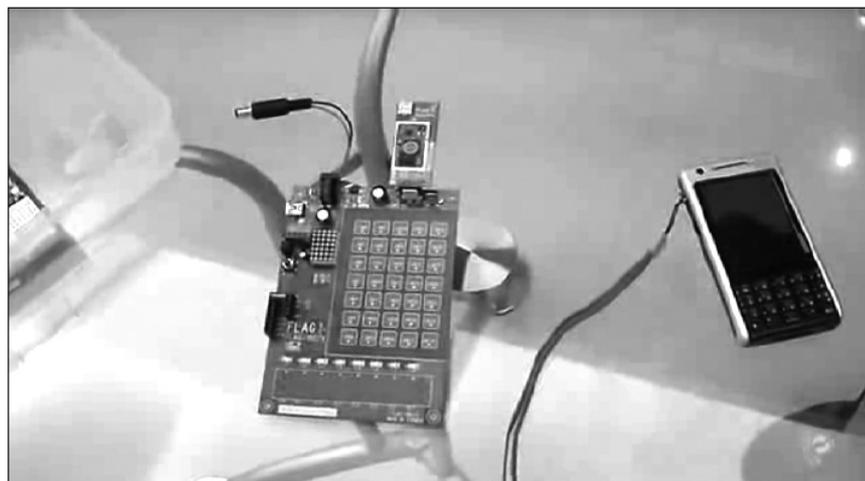
## 11-3 用 VideoView 播放影片

- 使用 VideoView 搭配 MediaController 來播放影片
- 用程式控制 VideoView 的影片播放
- 設定全螢幕顯示
- 處理在播放影片時切換到首頁或其他程式的狀況
- 處理在播放時旋轉手機的狀況

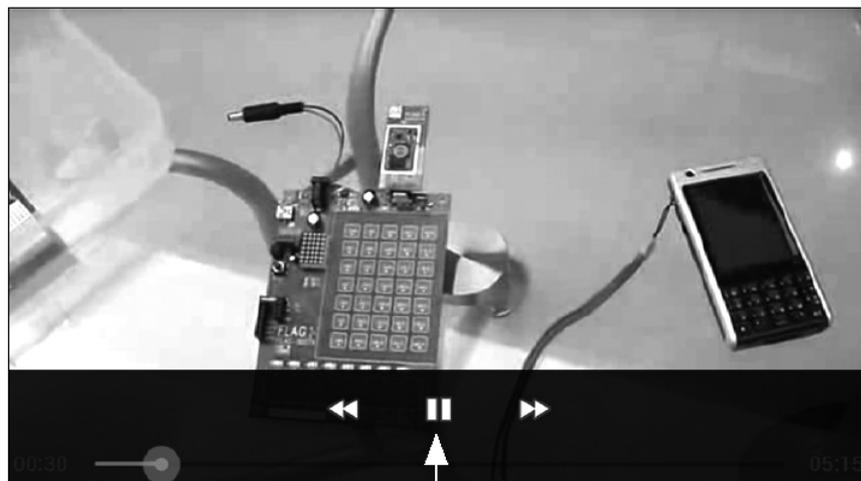
# 使用 VideoView 搭配 MediaController 來播放影片

```
vdv = (VideoView)findViewById(R.id.videoView1); ←取得畫面中的 VideoView 元件  
MediaController mediaCtrl = new MediaController(this); ←建立播放控制物件  
vdv.setMediaController(mediaCtrl); ←設定要使用播放控制物件
```

# 使用 VideoView 搭配 MediaController 來播放影片



在 VideoView 中播放的影片



按一下螢幕，即會出現  
MediaController 控制列

# 用程式控制 VideoView 的影片播放

`vdv.setVideoURI(uri);`      ← 設定要播放影片的 Uri

`vdv.start();`              ← 開始播放

`vdv.pause();`              ← 暫停播放

`vdv.stopPlayback();`      ← 停止播放

`boolean b = vdv.isPlaying();`      ← 是否在播放中

`int len = vdv.getDuration();`      ← 讀取影片長度 (秒)

`int pos = vdv.getCurrentPosition();` ← 讀取目前的播放位置 (秒)

`vdv.seekTo(pos);`              ← 設定播放位置 (秒)

# 設定全螢幕顯示



# 設定全螢幕顯示



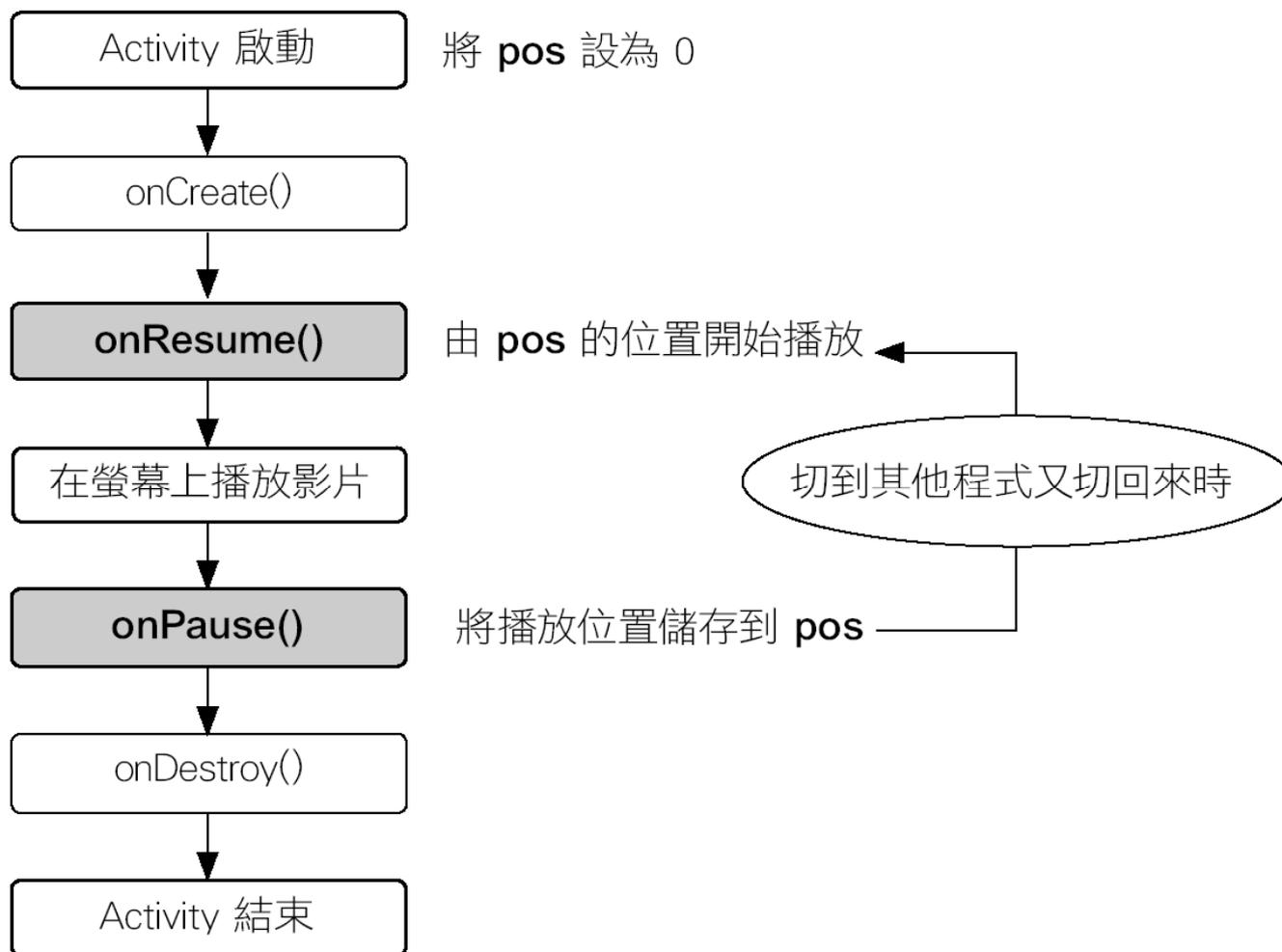
# 設定全螢幕顯示

```
01 getWindow().addFlags(WindowManager.LayoutParams.  
    FLAG_FULLSCREEN); ← 隱藏系統的狀態列  
02 requestWindowFeature(Window.FEATURE_NO_TITLE); ← 隱藏 Activity 的標題列  
03  
04 setContentView(R.layout.activity_video);  
05  
06 getWindow().addFlags(WindowManager.LayoutParams.  
    FLAG_KEEP_SCREEN_ON); ← 保持螢幕一直開著  
    (不要自動休眠)
```

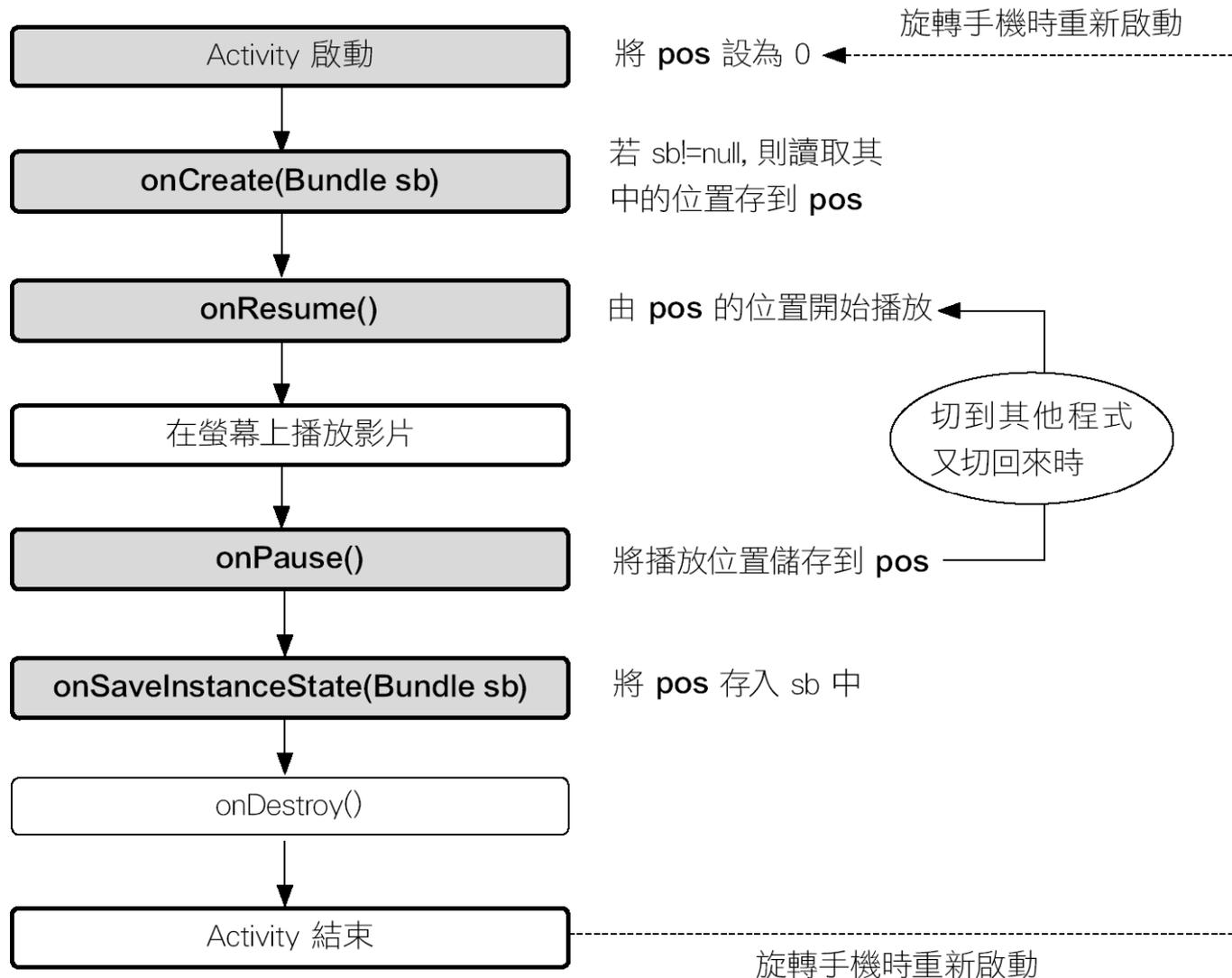
# 處理在播放影片時切換到首頁或其他程式的狀況

```
public class VideoActivity extends Activity {  
    int pos = 0; ← 用來記錄前次的播放位置, 預設為 0  
    ...  
    protected void onResume() { ← 當 Activity 獲得輸入焦點  
        super.onResume();  
        vdv.seekTo(pos); ← 移到 pos 的播放位置  
        vdv.start(); ← 開始播放  
    }  
  
    protected void onPause() { ← 當 Activity 失去輸入焦點時 (例如切換到其他程式)  
        super.onPause();  
        pos = vdv.getCurrentPosition(); ← 儲存播放位置  
        vdv.stopPlayback(); ← 停止播放  
    }  
}
```

# 處理在播放影片時切換到首頁或其他程式的狀況



# 處理在播放時旋轉手機的狀況



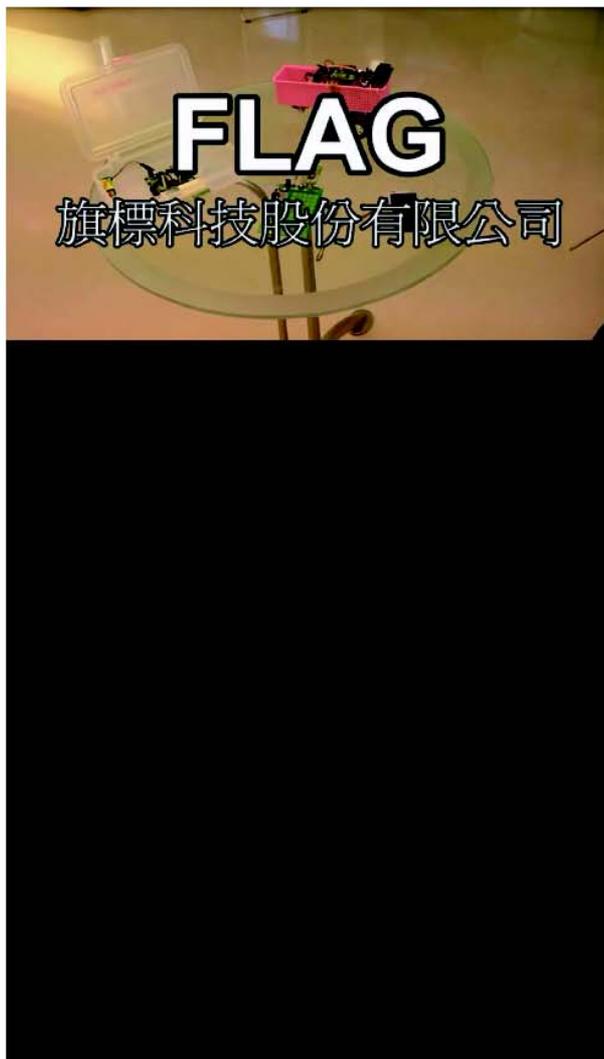
# 範例11-3：開啟新的 Activity 來播放影片



1 按此鈕選取影片

2 按此鈕播放影片

# 開啟新的 Activity 來播放影片



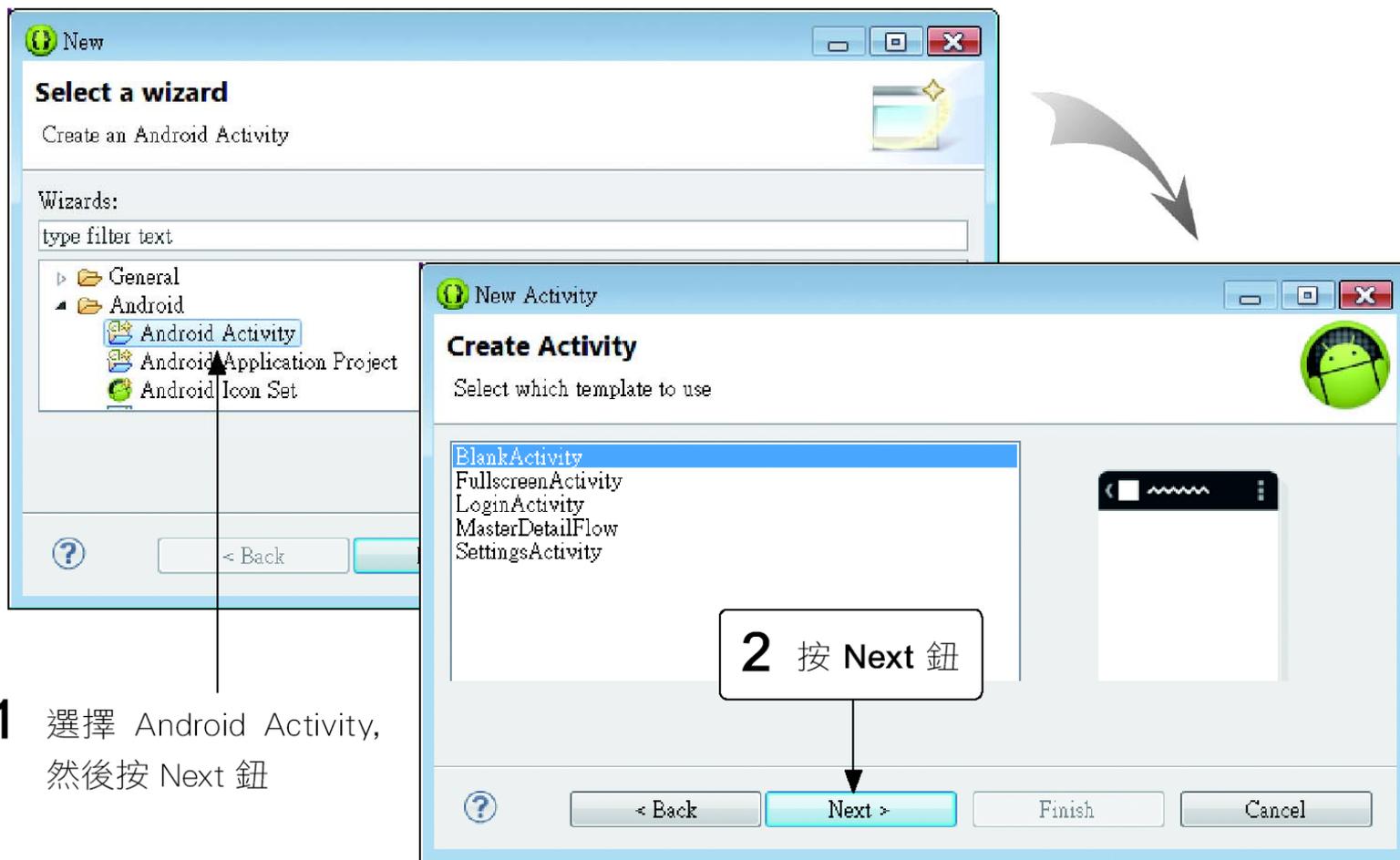
在全螢幕中播放影片，  
在播放中可以任意旋  
轉手機來觀賞



# 開啟新的 Activity 來播放影片

step 1

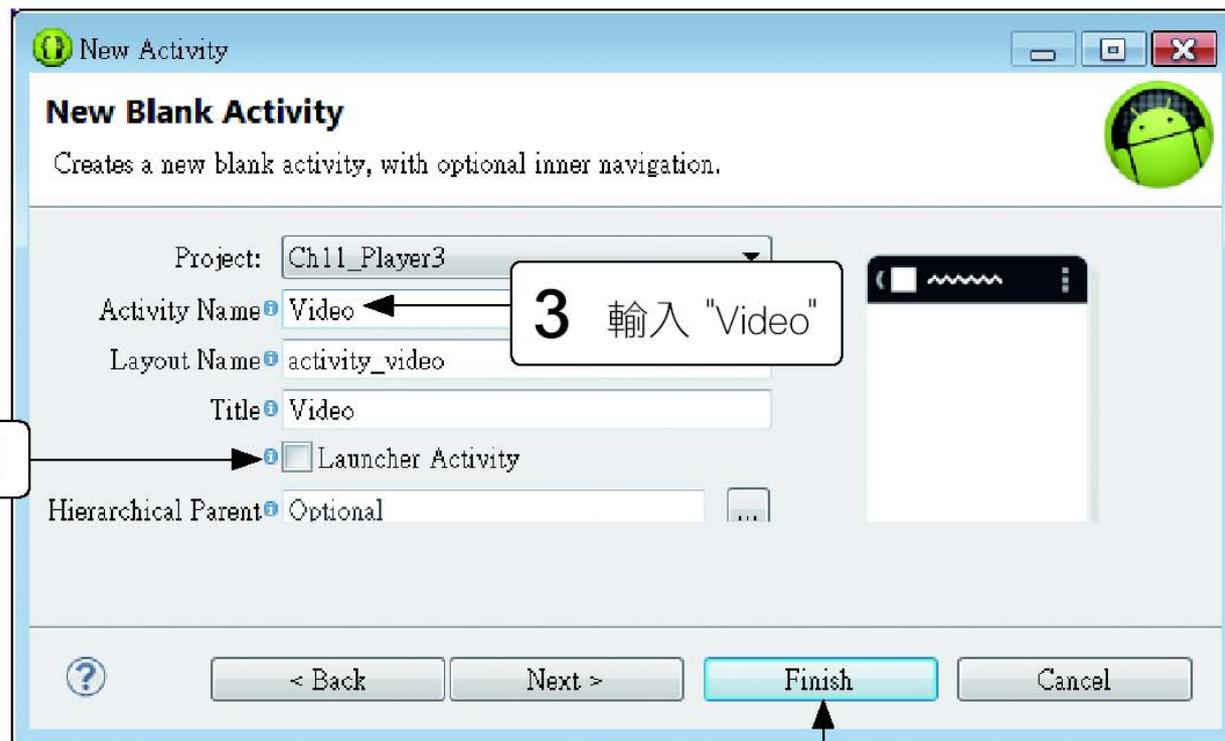
step 2



1 選擇 Android Activity,  
然後按 Next 鈕

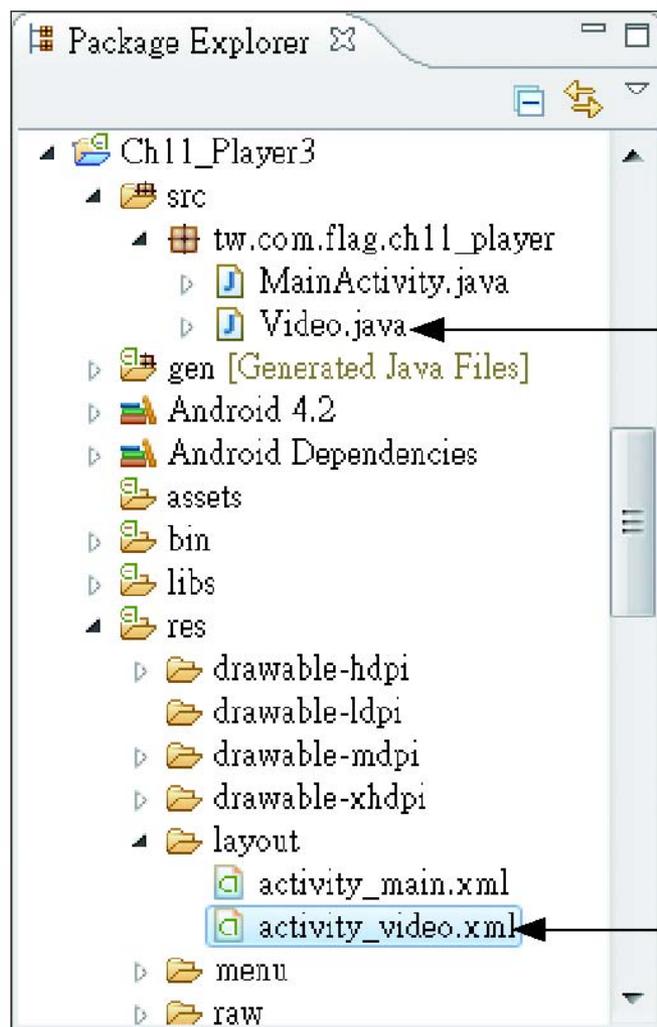
2 按 Next 鈕

# 開啟新的 **Activity** 來播放影片



4 按 Finish 鈕

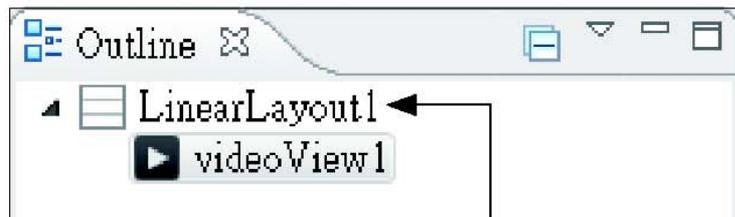
# 開啟新的 **Activity** 來播放影片



在專案中會產生  
新 Activity 的程式  
檔及 Layout 檔

# 開啟新的 **Activity** 來播放影片

## step 3

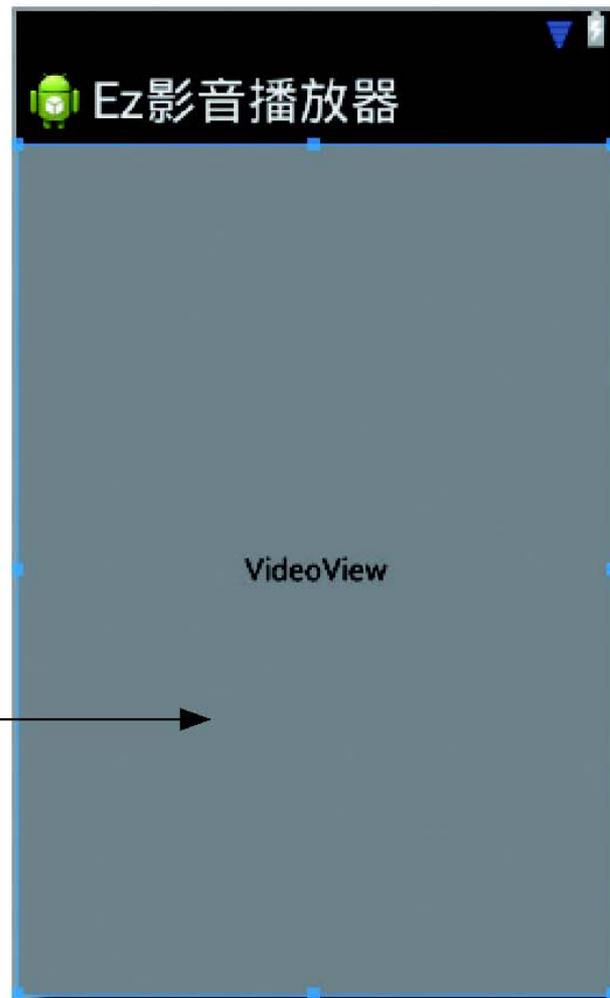


LinearLayout(Vertical)

background #000 (背景設為黑色)

VideoView 元件

Id	@+id/videoView1
Width	match_parent
Height	wrap_content



# 開啟新的 Activity 來播放影片

## step 4

```
01 public void onMpPlay(View v) {    ← 按下播放鈕時
02     if(isVideo) {    ← 如果是影片
03         Intent it = new Intent(this, Video.class); ← 建立開啟 Video
                                                    Activity 的 Intent
04         it.putExtra("uri", uri.toString()); ← 將影片的 Uri 以 "uri"
                                                    為名加入 Intent 中
05         startActivity(it);    ← 啟動 Video Activity
06         return;    ← 結束本方法
07     }
08
09     if (mper.isPlaying()) { ← 如果正在播, 就暫停
10         mper.pause();    ← 暫停播放
```

# 開啟新的 **Activity** 來播放影片

```
11     btnPlay.setText("繼續");
12 }
13 else { ← 如果沒有在播, 就開始播
14     mper.start(); ← 開始播放
15     btnPlay.setText("暫停");
16     btnStop.setEnabled(true);
17 }
18 }
```

# 開啟新的 **Activity** 來播放影片

## step 5

```
01 public class Video extends Activity {
02
03     VideoView vdv; ← 宣告 VideoView 物件
04     int pos = 0; ← 用來記錄前次的播放位置
05
06     @Override
07     protected void onCreate(Bundle savedInstanceState) {
08         super.onCreate(savedInstanceState);
09
10         getWindow().addFlags(WindowManager.LayoutParams.
11             FLAG_FULLSCREEN); ← 隱藏系統的狀態列
12         requestWindowFeature(Window.FEATURE_NO_TITLE); ← 隱藏 Activity 的標題列
13         setContentView(R.layout.activity_video); ← 以上2項設定必須在本方法之前呼叫
14     }
15 }
```

# 開啟新的 Activity 來播放影片

```
13 getWindow().addFlags(WindowManager.LayoutParams.  
    FLAG_KEEP_SCREEN_ON); ← 保持螢幕一直開著  
14                                不會自動休眠  
15 Intent it = getIntent(); ← 取得傳入的 Intent 物件  
16 Uri uri = Uri.parse(it.getStringExtra("uri")); ←  
    取出要播放影片的 Uri  
17 if(savedInstanceState != null) ← 如果是因旋轉而重新啟動 Activity  
18     pos = savedInstanceState.getInt("pos", 0); ← 取出旋轉前所  
19                                儲存的播放位置  
20 vdv = (VideoView)findViewById(R.id.videoView1); ←  
    取得到畫面中的 VideoView 元件  
21 MediaController mediaCtrl = new MediaController(this); ←  
    建立播放控制物件  
22 vdv.setMediaController(mediaCtrl); ← 設定播放控制物件  
23 vdv.setVideoURI(uri); ← 設定要播放影片的 Uri  
24 }
```

# 開啟新的 **Activity** 來播放影片

## step 6

```
01 @Override
02 protected void onResume() { ← 當 Activity 啟動、或由暫停狀態回到互動狀態時
03     super.onResume();
04     vdv.seekTo(pos); ← 移到 pos 的播放位置
05     vdv.start(); ← 開始播放
06 }
07
08 @Override
09 protected void onPause() { ← 當 Activity 進入暫停狀態時
10     super.onPause();
11     pos = vdv.getCurrentPosition(); ← 儲存播放位置
12     vdv.stopPlayback(); ← 停止播放
```

# 開啟新的 **Activity** 來播放影片

```
13 }
14
15 @Override
16 protected void onSaveInstanceState(Bundle outState) {
17     super.onSaveInstanceState(outState);
18     outState.putInt("pos", pos); ← 將 onPause() 中所取得的
                                   播放位置儲存到 bundle
19 }
```