

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 10 章

拍照與顯示相片

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

- 10-1 使用 Intent 啟動系統的相機程式
- 10-2 要求相機程式存檔
- 10-3 解決相片過大問題
- 10-4 旋轉手機與旋轉相片
- 10-5 使用 Intent 瀏覽並選取相片

10-1 使用 Intent 啟動系統的相機程式

```
Intent it = new Intent(MediaStore.ACTION_IMAGE_CAPTURE); ← 建立 Intent  
startActivityForResult(it, 100); ← 用 Intent 啟動程式, 並要求傳回資料
```

```
protected void onActivityResult(int reqCode, int resCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
    if(resCode == Activity.RESULT_OK && reqCode==100) { ← 有拍到照片時  
        //處理傳回的資料  
    }  
    else { ← 沒拍到照片時  
        Toast.makeText(this, "沒有拍到照片", Toast.LENGTH_LONG).show();  
    }  
}
```

利用 Bundle 取出 Intent 中附帶的 Bitmap 物件

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode == Activity.RESULT_OK && requestCode==100) { ← 有拍到照片時
        Bundle bdl = data.getExtras(); ← 將 Intent 的附加資料轉
                                         為 Bundle 物件
        Bitmap bmp = (Bitmap) bdl.get("data"); ←
                                                由 Bundle 取出名為 "data" 的 Bitmap 資料
        ImageView imv = (ImageView) findViewById(R.id.imageView1);
        imv.setImageBitmap(bmp); ← 將 Bitmap 資料顯示在 ImageView 中
    }
    else { ← 沒拍到照片時
        Toast.makeText(this, "沒有拍到照片", Toast.LENGTH_LONG).show();
    }
}
```

範例10-1：利用系統的相機程式來拍照

1 按此鈕啟動手機的相機程式

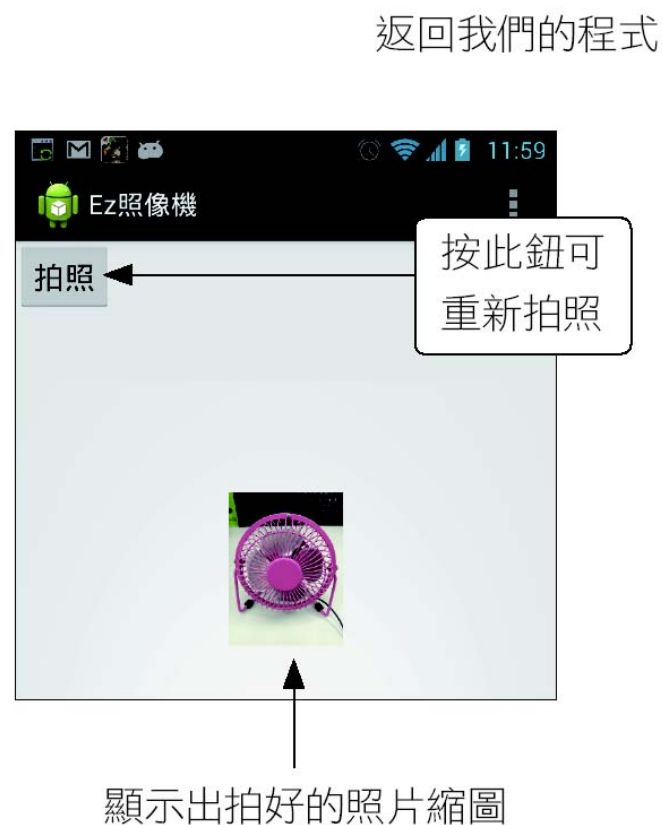


2 按此鈕拍照

利用系統的相機程式來拍照



3 按此鈕表示拍好了



利用系統的相機程式來拍照

step 1

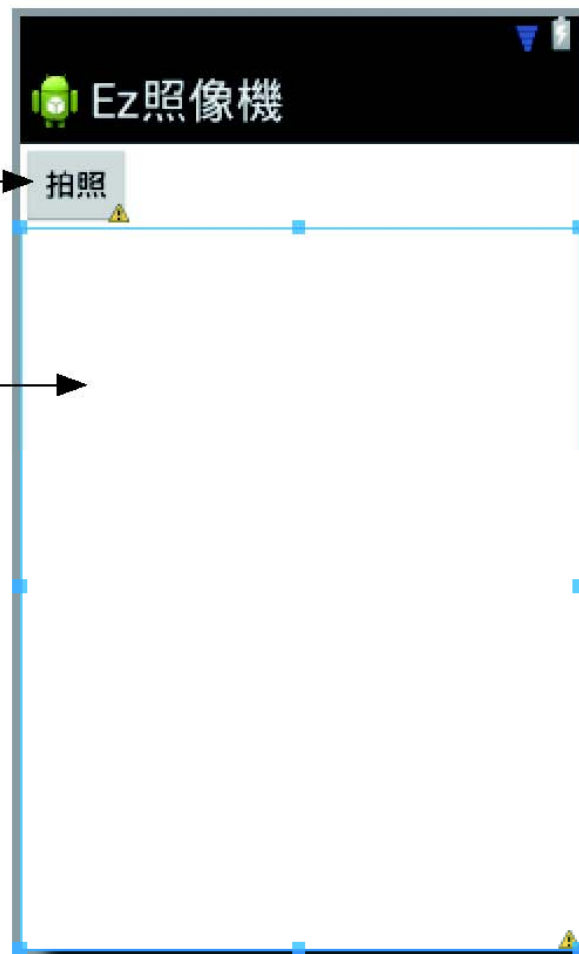
step 2

1 加入 Button 元件

Id	@+id/button1
Text	拍照
On Click	onGet

2 加入 ImageView 元件 (在加入時會出現 **Resource Chooser** 交談窗, 請按 **Clear** 鈕表示不指定預設圖片)

Id	@+id/imageView1
Width	match_parent
Height	match_parent
Scale Type	centerInside



利用系統的相機程式來拍照

step 3

```
01 package tw.com.flag.ch10_camera;  
02  
03 import android.app.Activity;  
04 import android.view.Menu;  
05 import android.view.View;
```

Next

利用系統的相機程式來拍照

```
06 import android.content.Intent;
07 import android.graphics.Bitmap;
08 import android.os.Bundle;
09 import android.provider.MediaStore;
10 import android.widget.ImageView;
11 import android.widget.Toast;
12
13 public class MainActivity extends Activity {
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_main);
19     }
20
```

利用系統的相機程式來拍照

```
21     public void onGet(View v) {
22         Intent it = new Intent(MediaStore.ACTION_IMAGE_CAPTURE); ←
                                                    建立動作為拍照的 Intent
23         startActivityForResult(it, 100); ← 啟動 Intent 並要求傳回資料
24     }
25
26     protected void onActivityResult (int requestCode, int resultCode,
                                                    Intent data) {
27         super.onActivityResult(requestCode, resultCode, data);
28
29         if(resultCode == Activity.RESULT_OK && requestCode==100) {
30             Bundle extras = data.getExtras(); ←
                                                    將 Intent 的附加資料轉為 Bundle 物件
31             Bitmap bmp = (Bitmap) extras.get("data"); ←
                                                    由 Bundle 取出名為 "data" 的 Bitmap 資料
```

利用系統的相機程式來拍照

```
32         ImageView imv = (ImageView) findViewById (
33             R.id.imageView1);
34         imv.setImageBitmap(bmp); ← 將 Bitmap 資料顯示在
35         ImageView 中
36     }
37     else {
38         Toast.makeText(this, "沒有拍到照片", Toast.LENGTH_LONG).
39             show();
40     }
41 }
```

Next

利用系統的相機程式來拍照

```
39
40     @Override
41     public boolean onCreateOptionsMenu(Menu menu) {
42         // Inflate the menu;
43         getMenuInflater().inflate(R.menu.activity_main, menu);
44         return true;
45     }
46 }
```

10-2 要求相機程式存檔

- 準備代表圖檔路徑的 Uri
- 用 BitmapFactory 類別讀取圖檔

準備代表圖檔路徑的 Uri

```
01 String dir = Environment.getExternalStoragePublicDirectory( ←
02         Environment.DIRECTORY_PICTURES).toString(); 取得系統的公
                                                    用圖檔路徑
03 String fname = "p" + System.currentTimeMillis() + ".jpg" ←
                                                    利用目前時間組合出一個不會重複的檔名
04 Uri uri = Uri.parse("file://" + dir + "/" + fname); ←
                                                    依前面的路徑及檔
                                                    名建立 Uri 物件
05
06 Intent it = new Intent("android.media.action.IMAGE_CAPTURE");
07 it.putExtra(MediaStore.EXTRA_OUTPUT, uri); ← 將 uri 加到拍照
                                                    Intent 的額外資料中
08 startActivityForResult(it, 100);
```

用 BitmapFactory 類別讀取圖檔

```
Bitmap bmp = BitmapFactory.decodeFile(imgUri.getPath()); ←  
                                                    讀取圖檔內容並儲存為 Bitmap 物件  
imv.setImageBitmap(bmp); ← 將 Bitmap 物件顯示在 ImageView 中
```

範例10-2：要求相機程式存檔並在程式中顯示出來

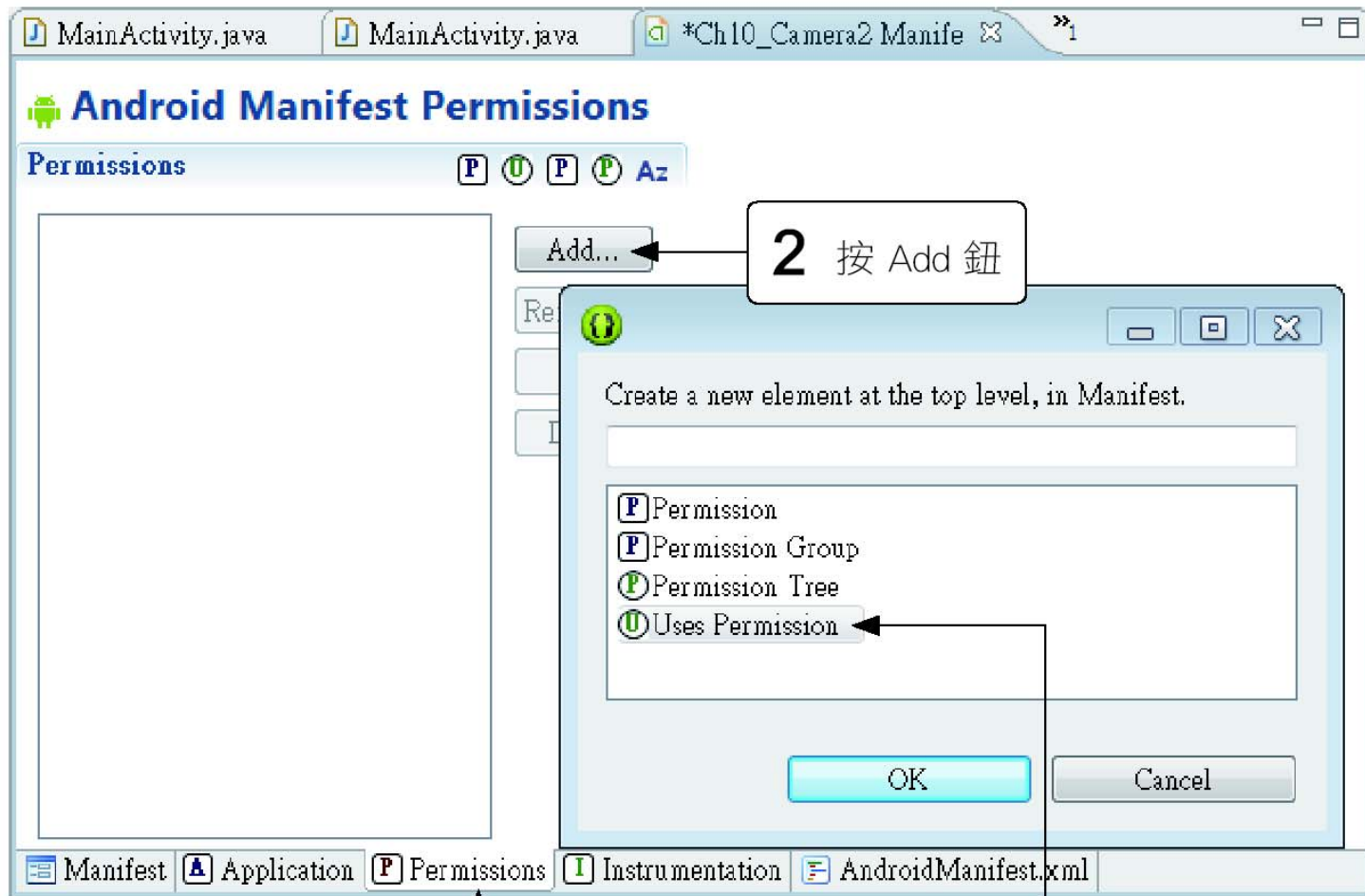
剛拍好的照片，
不再只是一張
小縮圖了



要求相機程式存檔並在程式中顯示出來

step 1

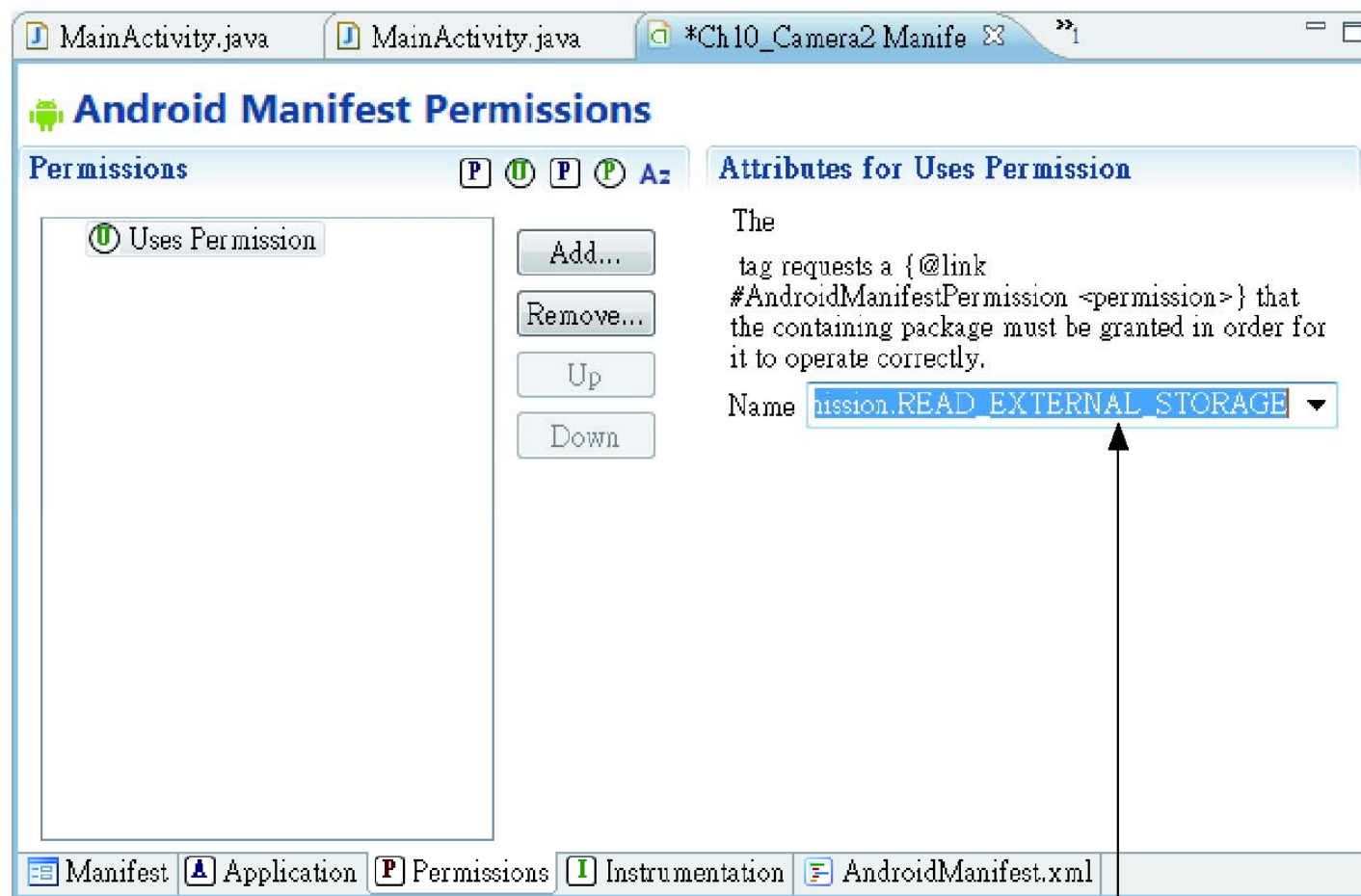
step 2



1 切到此頁次

3 雙按 Uses Permission 項目

要求相機程式存檔並在程式中顯示出來



4 選取 android.permission.READ_EXTERNAL_STORAGE

要求相機程式存檔並在程式中顯示出來

step 3

```
01 public class MainActivity extends Activity {  
02     Uri imgUri;    ← 拍照存檔的 Uri 物件  
03     ImageView imv; ← ImageView 物件
```

要求相機程式存檔並在程式中顯示出來

step 4

```
01  @Override
02  protected void onCreate(Bundle savedInstanceState) {
03      super.onCreate(savedInstanceState);
04      setContentView(R.layout.activity_main);
05      imv = (ImageView) findViewById(R.id.imageView1);
06  }
07
08  public void onGet(View v) { ← 按下拍照鈕時執行的方法
09      String dir = Environment.getExternalStoragePublicDirectory(← 取得系統的公
10          Environment.DIRECTORY_PICTURES).toString();          用圖檔路徑
```

Next

要求相機程式存檔並在程式中顯示出來

```
11     String fname = "p" + System.currentTimeMillis() + ".jpg"; ←
                                     利用目前時間組合出一個不會重複的檔名
12     imgUri = Uri.parse("file://" + dir + "/" + fname); ←
13                                     依前面的路徑及檔名建立 Uri 物件
14     Intent it = new Intent("android.media.action.IMAGE_CAPTURE");
15     it.putExtra(MediaStore.EXTRA_OUTPUT, imgUri); ←
                                     將 uri 加到拍照 Intent 的額外資料中
16     startActivityForResult(it, 100); ← 啟動 Intent
17 }
18
19 protected void onActivityResult (int requestCode, int resultCode,
                                     Intent data) {
```

要求相機程式存檔並在程式中顯示出來

```
20     super.onActivityResult(requestCode, resultCode, data);
21
22     if(resultCode == Activity.RESULT_OK && requestCode==100) {
23         Bitmap bmp = BitmapFactory.decodeFile(imgUri.getPath()); ← 讀取圖檔內容轉換為 Bitmap 物件
24         imv.setImageBitmap(bmp); ← 將 Bitmap 物件顯示在 ImageView 中
25     }
26     else {
27         Toast.makeText(this, "沒有拍到照片", Toast.LENGTH_LONG).show();
28     }
29 }
```

10-3 解決相片過大問題

- 用 `BitmapFactory.Options` 設定載入圖檔的選項

```
01 //查詢圖檔的寬、高
02 BitmapFactory.Options option = new BitmapFactory.Options(); ← 建立選項物件
03 option.inJustDecodeBounds = true; ← 設定選項：只讀取圖檔資訊而不載入圖檔
04 BitmapFactory.decodeFile(imgUri.getPath(), option); ← 讀取圖檔資訊存入 Option 中
05 iw = option.outWidth; ← 由 option 中讀出圖檔寬度
06 ih = option.outHeight; ← 由 option 中讀出圖檔高度
```

用 BitmapFactory.Options 設定載入圖檔的選項

```
option.inSampleSize = 2; ← 設定縮小比例為 2, 則寬高都將縮小為原來的 1/2  
Bitmap bmp = BitmapFactory.decodeFile(imgUri.getPath(), option); ← 載入圖檔
```

```
option.inSampleSize = 2;  
option.inPurgeable = true; ← 設定在記憶體不夠時, 允許  
系統將圖片內容暫時刪除  
Bitmap bmp = BitmapFactory.decodeFile(imgUri.getPath(), option);
```


範例10-3：依顯示尺寸來載入縮小的圖檔

step 1

step 2

```
01  protected void onActivityResult (int requestCode, int resultCode,
                                Intent data) {
02      super.onActivityResult(requestCode, resultCode, data);
03
04      if(resultCode == Activity.RESULT_OK && requestCode==100) {
05          showImg();
06      }
07      else {
08          Toast.makeText(this, "沒有拍到照片", Toast.LENGTH_LONG).
                show();
09      }
10 }
```

依顯示尺寸來載入縮小的圖檔

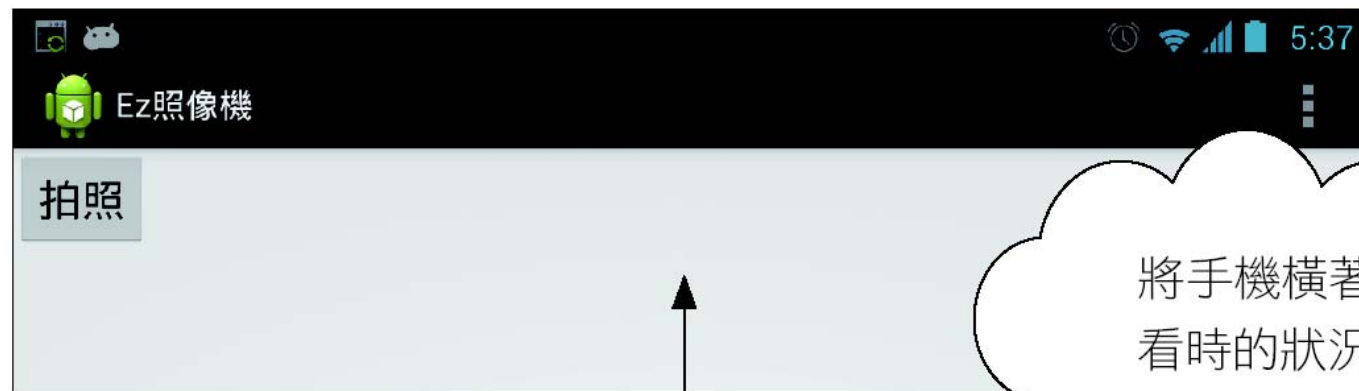
```
11
12 void showImg() {      Imageview 元件的寬高
13     int iw, ih, vw, vh;
14         Image 圖片的寬高
15     BitmapFactory.Options option = new BitmapFactory.Options();
16         建立選項物件
17     option.inJustDecodeBounds = true; ← 設定選項：只讀取圖檔資訊而
        不載入圖檔
18     BitmapFactory.decodeFile(imgUri.getPath(), option); ←
        讀取圖檔資訊存入 Option 中
19     iw = option.outWidth; ← 由 option 中讀出圖檔寬度
20     ih = option.outHeight; ← 由 option 中讀出圖檔高度
    vw = imv.getWidth(); ← 取得 ImageView 的寬度
```

Next

依顯示尺寸來載入縮小的圖檔

```
21     vh = imv.getHeight(); ← 取得 ImageView 的高度
22
23     int scaleFactor = Math.min(iw/vw, ih/vh); ← 計算縮小比率
24
25     option.inJustDecodeBounds = false; ← 關閉只載入圖檔資訊的選項
26     option.inSampleSize = scaleFactor; ← 設定縮小比例, 例如 3 則長
                                           寬都將縮小為原來的 1/3
27     option.inPurgeable = true; ← 設定在記憶體不夠時,
                                           允許系統將圖片內容刪除
28     Bitmap bmp = BitmapFactory.decodeFile(imgUri.getPath(),
                                           option); ← 載入圖檔
29     imv.setImageBitmap(bmp); ← 顯示照片
30 }
```

10-4 旋轉手機與旋轉相片



剛才拍的照片不見了!

將手機橫著
看時的狀況

旋轉手機與旋轉相片



若是橫拍的照片，
就由程式自己將照
片旋轉 90 度顯示



將手機橫著看時，
畫面不會跟著旋轉

旋轉手機與旋轉相片

- 關閉自動旋轉功能並設定螢幕為直向顯示
- 用Matrix 物件來旋轉圖片

關閉自動旋轉功能並設定螢幕為直向顯示

```
setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_NOSENSOR); ◀  
設定螢幕不隨手機旋轉  
setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT); ◀  
設定螢幕直向顯示
```

用 Matrix 物件來旋轉圖片

```
Matrix matrix = new Matrix();
```

← 建立 Matrix 矩陣物件

```
matrix.postRotate(90);
```

← 設定矩陣的順時針旋轉角度

```
bmp = Bitmap.createBitmap(bmp , 0, 0,
```

← 用原來的 Bitmap

```
        bmp.getWidth(),
```

產生一個新的 Bitmap

```
        bmp.getHeight(),
```

```
        matrix, true);
```

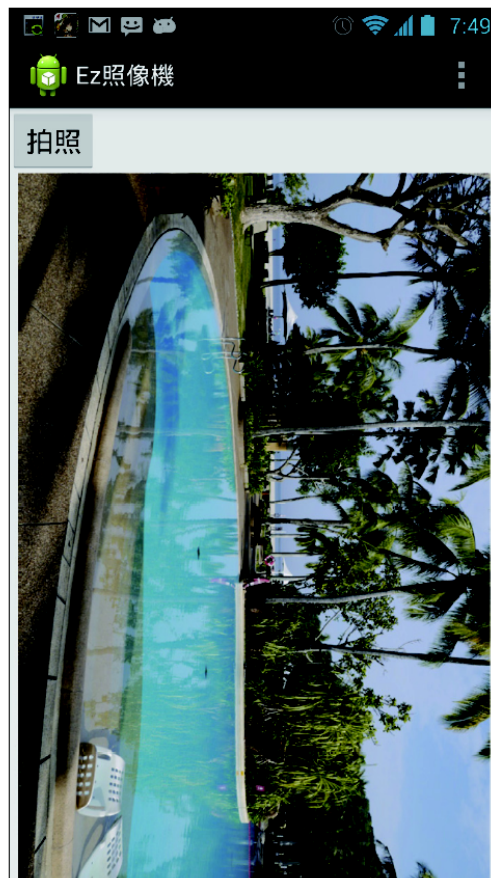
```
createBitmap (Bitmap src, int x, int y, int width, int height,
```

```
        Matrix m, boolean filter)
```


範例10-4：依相片是直拍或橫拍而自動旋轉相片



直拍的相片



橫拍的相片

依相片是直拍或橫拍而自動旋轉相片

step 1

step 2

```
01  protected void onCreate(Bundle savedInstanceState) {  
02      super.onCreate(savedInstanceState);  
03      setContentView(R.layout.activity_main);  
04  
05      setRequestedOrientation(                                設定螢幕不隨手機旋轉  
        ActivityInfo.SCREEN_ORIENTATION_NOSENSOR); ←  
06      setRequestedOrientation(                                設定螢幕直向顯示  
        ActivityInfo.SCREEN_ORIENTATION_PORTRAIT); ←  
07      imv = (ImageView) findViewById(R.id.imageView1); ←  
08  }                                                         取得 Layout 中的 ImageView 元件
```

依相片是直拍或橫拍而自動旋轉相片

step 3

```
01 void showImg() {
02     int iw, ih, vw, vh;
03     boolean needRotate; ← 用來儲存是否需要旋轉
04
05     BitmapFactory.Options option = new BitmapFactory.Options();
                                ↑ 建立選項物件
06     option.inJustDecodeBounds = true; ← 設定選項：只讀取圖檔
                                           資訊而不載入圖檔
07     BitmapFactory.decodeFile(imgUri.getPath(), option); ←
                                                         讀取圖檔資訊存入 Option 中
```

依相片是直拍或橫拍而自動旋轉相片

```
08     iw = option.outWidth;    ← 由 option 中讀出圖檔寬度
09     ih = option.outHeight;   ← 由 option 中讀出圖檔高度
10     vw = imv.getWidth();     ← 取得 ImageView 元件的寬度
11     vh = imv.getHeight();    ← 取得 ImageView 元件的高度
12
13     int scaleFactor;
14     if(iw < ih) {           ← 如果圖片的寬度小於高度
15         needRotate = false; ← 不需要旋轉
```

Next

依相片是直拍或橫拍而自動旋轉相片

```
16     scaleFactor = Math.min(iw/vw, ih/vh); ← 計算縮小比率
17 }
18 else {
19     needRotate = true; ← 需要旋轉
20     scaleFactor = Math.min(ih/vw, iw/vh); ←
//                                         改用旋轉後的圖片寬、
21 }                                         高來計算縮小比例

22
23 option.inJustDecodeBounds = false; ← 關閉只載入圖檔資訊的選項
24 option.inSampleSize = scaleFactor; ← 設定縮小比例，例如 2 則長
                                         寬都將縮小為原來的 1/2
25 option.inPurgeable = true; ← 設定在記憶體不夠時，
                                         允許系統將圖片內容刪除
```

依相片是直拍或橫拍而自動旋轉相片

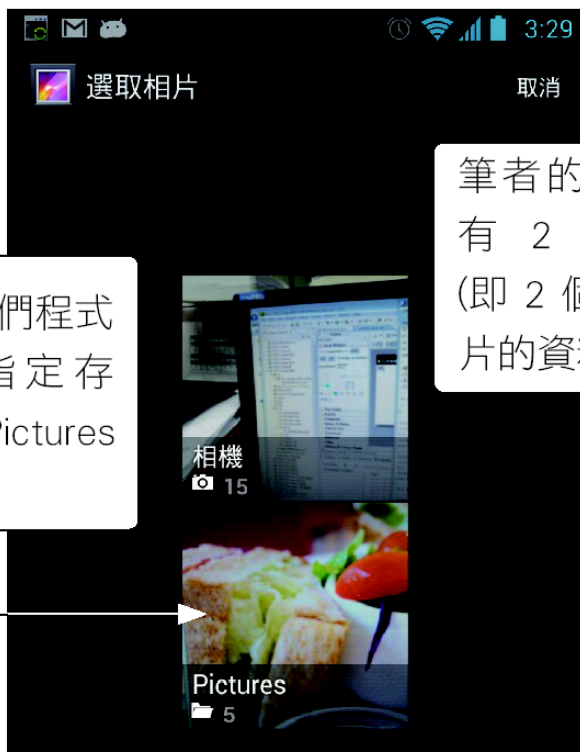
```
26     Bitmap bmp = BitmapFactory.decodeFile(  
        imgUri.getPath(), option);    ← 載入圖檔  
27  
28     if (needRotate) {    ← 如果需要旋轉  
29         Matrix matrix = new Matrix(); ← 建立 Matrix 物件  
30         matrix.postRotate(90);    ← 設定旋轉角度 (順時針 90°)  
31         bmp = Bitmap.createBitmap(bmp, ← 用原來的圖片產生  
            一個新的圖片  
32             0, 0, bmp.getWidth(), bmp.getHeight(), matrix, true);  
33     }  
34     imv.setImageBitmap(bmp);    ← 顯示圖片  
35 }
```

10-5 使用 Intent 瀏覽並選取相片

```
Intent it = new Intent(Intent.ACTION_GET_CONTENT); ← 動作設為 "選取內容"  
it.setType("image/*"); ← 設定選取的媒體類型為『所有類型的圖片』  
startActivityForResult(it, 101); ← 啟動 Intent, 並要求傳回選取的圖檔
```

使用 Intent 瀏覽並選取相片

1 點選我們程式中所指定存放的 Pictures 資料夾



筆者的手機中有 2 個相簿 (即 2 個存放圖片的資料夾)



2 選擇一張相片來傳回程式中顯示



使用 Intent 瀏覽並選取相片



回到程式並顯示選取的相片

使用 Intent 瀏覽並選取相片

```
01  protected void onActivityResult (int requestCode, int resultCode,
                                Intent data) {
02      super.onActivityResult(requestCode, resultCode, data);
03
04      if(resultCode == Activity.RESULT_OK) { ← 要求的 Intent
05          switch(requestCode) {           執行了
06              case 100: ← 拍照
07                  showImg();
08                  break;
09              case 101: ← 選取相片
10                  imgUri = data.getData(); ← 取得選取相片的 Uri
11                  showImg();
12                  break;
13          }
14      }
15  }
```

將相片改為可供系統共用的檔案

```
Intent it = new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE,  
                        imgUrl); ← 將 imgUrl 所指的檔案  
                               設定為系統共享媒體檔  
sendBroadcast(it); ← 用廣播方式將 Intent 傳送給系統
```

```
content://media/external/images/media/7 ← SD 卡中編號為 7 的圖檔  
content://media/internal/video/media/12 ← 內部記憶體中編號為 12 的影片檔
```

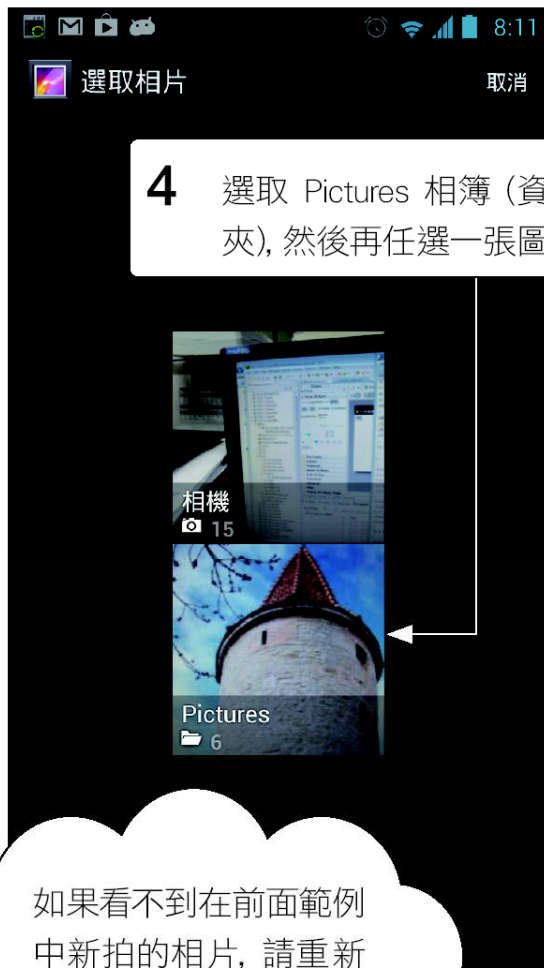
將相片改為可供系統共用的檔案

```
Uri convertUri(Uri uri) {  
    if(uri.toString().substring(0, 7).equals("content")) { ←  
        如果是以 "content" 開頭  
        String[] colName = { MediaColumns.DATA }; ← 宣告要查詢的欄位  
  
        Cursor cursor = getContentResolver().query(uri, colName,  
            以 uri 進行查詢 ↑  
            null, null, null);  
  
        cursor.moveToFirst(); ← 移到查詢結果的第一筆記錄  
        uri = Uri.parse("file://" + cursor.getString(0)); ←  
        將路徑轉為 Uri  
    }  
    return uri; ← 傳回 Uri 物件  
}
```

範例10-5：利用 Intent 瀏覽並選取已拍好的相片



利用 Intent 瀏覽並選取已拍好的相片



4 選取 Pictures 相簿 (資料夾), 然後再任選一張圖片

如果看不到在前面範例中新拍的相片, 請重新啟動手機然後再試一次

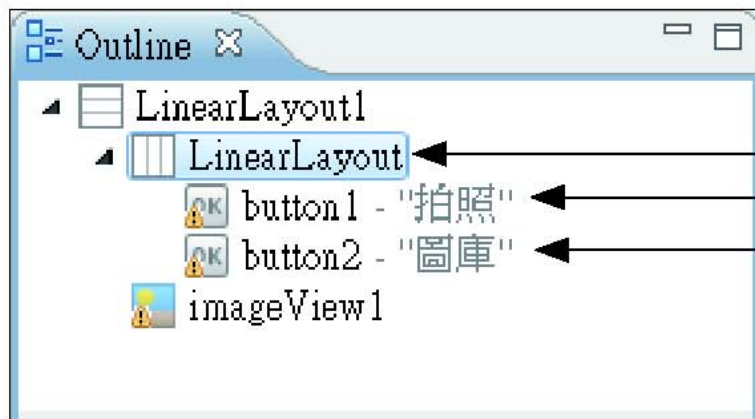


返回程式並顯示選取的相片

利用 Intent 瀏覽並選取已拍好的相片

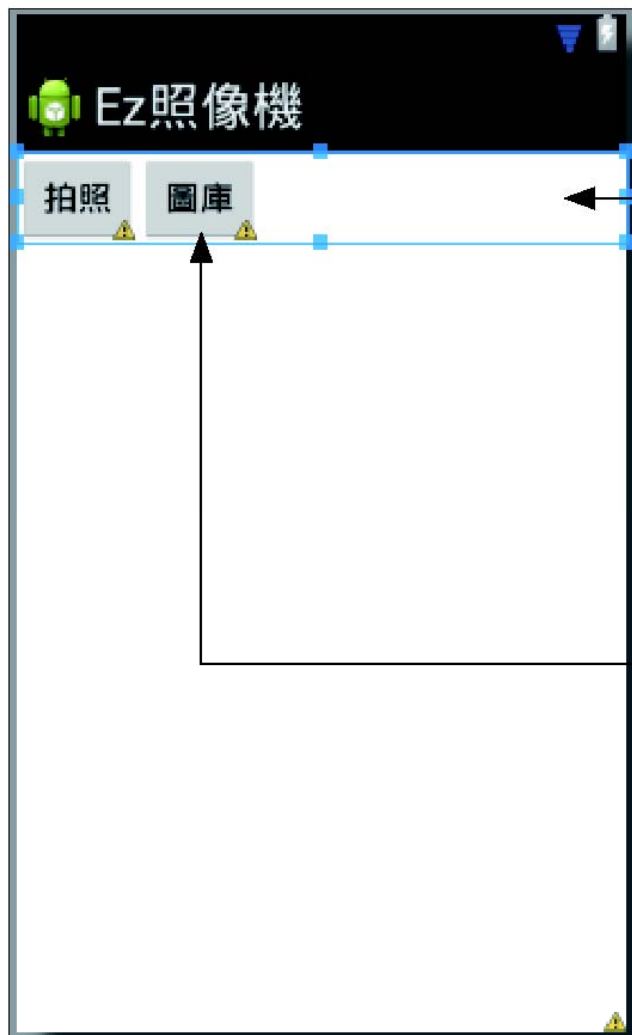
step 1

step 2



- 1 加入 LinearLayout(Horizontal)
- 2 將原來的**拍照**鈕拉曳到新的 LinearLayout 中
- 3 再加入一個**圖庫**按鈕

利用 Intent 瀏覽並選取已拍好的相片



LinearLayout 元件

Width	match_parent
Height	wrap_content

Button 元件

Id	@+id/button2
Text	圖庫
On Click	onPick

利用 Intent 瀏覽並選取已拍好的相片

step 3

```
01  protected void onActivityResult(int requestCode, int resultCode,
                                Intent data) {
02      super.onActivityResult(requestCode, resultCode, data);
03
04      if(resultCode == Activity.RESULT_OK) { ← 要求的Intent成功了
05          switch(requestCode) {
06              case 100: ← 拍照
07                  Intent it = new Intent(Intent.ACTION_MEDIA_SCANNER_
                                SCAN_FILE, imgUri); ←
08                  sendBroadcast(it); ← 設為系統共享媒體檔
09                  break;
10              case 101: ← 選取相片
```

利用 Intent 瀏覽並選取已拍好的相片

```
11         imgUri = convertUri(data.getData()); ←
12         break;                               取得選取相片的 Uri
13     }                                         並做 Uri 格式轉換
14     showImg(); ← 顯示相片
15 }
16 else { ← 要求的Intent沒有成功
17     Toast.makeText(this, requestCode==100? "沒有拍到照片":
18         "沒有選取相片", Toast.LENGTH_LONG)
19         .show();
20 }
```

利用 Intent 瀏覽並選取已拍好的相片

step 4

```
01 Uri convertUri(Uri uri) {
02     if(uri.toString().substring(0, 7).equals("content")) { ← 如果是以 "content" 開頭
03         String[] colName = { MediaColumns.DATA }; ← 宣告要查詢的欄位
04         Cursor cursor = getContentResolver().query(uri, colName,
05                                                     null, null, null); ← 以 uri 進行查詢
06         cursor.moveToFirst(); ← 移到查詢結果的第一筆記錄
07         uri = Uri.parse("file://" + cursor.getString(0)); ← 將路徑轉為 Uri
08     }
09     return uri; ← 傳回 Uri 物件
10 }
```

利用 Intent 瀏覽並選取已拍好的相片

step 5

```
01 public void onPick(View v) {                                動作設為 "選取內容"  
02     Intent it = new Intent(Intent.ACTION_GET_CONTENT);      ←  
03     it.setType("image/*");  ← 設定要選取的媒體類型為：所有類型的圖片  
04     startActivityForResult(it, 101);  ← 啟動 Intent, 並  
05 }                                                         要求傳回選取的圖檔
```