

Android APP手機程式設計實務



講師：賴貴平



本節課程內容

- 本節課程內容將包含以下教學內容:

Android 硬體裝置與應用

- 音樂撥放、影片撥放
- 相機、螢幕繪圖



ANDROID 硬體裝置與應用

- **MediaPlayer物件**

- Android 內建 MediaPlayer物件來播放音樂，可以直接播放儲存在專案「raw」目錄的MP3音樂檔
- 「原料資源」(raw resources) 存放在Android Studio專案的「res\raw」目錄
 - 此目錄的檔案是不壓縮的原始資料檔案
 - 音樂
 - 影片檔
- 將任意一個MP3檔案複製到「res\raw」目錄
- 呼叫MediaPlayer.create()類別方法來建立MediaPlayer物件



ANDROID 硬體裝置與應用

- **MediaPlayer**物件

- MediaPlayer.create()類別方法
- 第1個參數是活動本身，第2個參數是音樂檔案的URI
 - `player = MediaPlayer.create(this, R.raw.xxxx);`
- 呼叫setOnCompletionListener()方法，註冊音樂播放完畢的傾聽者物件
 - `player.setOnCompletionListener(this);`



ANDROID 硬體裝置與應用

- **MediaPlayer物件**

- 呼叫`prepareAsync()`方法準備MediaPlayer物件，開始播放音樂
 - `try {`
 - `player.prepareAsync();`
 - `} catch (Exception ex) {`
 - `Log.d("APP01", "onCreate: " + ex.getMessage());`
 - `}`



ANDROID 硬體裝置與應用

- **MediaPlayer**物件常用方法

方法	定義
start()	開始或繼續播放音樂
pause()	暫停播放音樂
stop()	停止播放音樂，當呼叫後，需要呼叫prepareAsync()方法準備後，才能再次呼叫start()方法播放音樂
prepareAsync()	準備MediaPlayer物件的播放器來播放音樂
isPlaying()	音樂是否正在播放中，true是；false為不是
seekTo(int)	將音樂跳至參數時間（毫秒）的位置
release()	釋放MediaPlayer物件佔用的資源



ANDROID 硬體裝置與應用

- **MediaPlayer**物件

- 實作OnCompletionListener介面作為傾聽者物件
- 介面擁有1個onCompletion()方法，當音樂播放完畢，顯示訊息文字，並呼叫seekTo()方法將播放時間歸零
 - @Override
 - public void onCompletion(MediaPlayer mediaPlayer) {
 - output.setText("音樂已經播放完畢...");
 - player.seekTo(0);
 - }



ANDROID 硬體裝置與應用

- 開啟 **AndroidManifest.xml.xml**

- `<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-
permission>`
- `<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"></uses-
permission>`
- `<application`
 - `android:allowBackup="true"`
 - `android:icon="@mipmap/ic_launcher"`
 - `android:label="@string/app_name"`
 - `android:supportsRtl="true"`
 - `android:theme="@style/AppTheme"`
 - `>`
- `<activity android:name=".MusicActivity"></activity>`
- `</application>`



ANDROID 硬體裝置與應用

- 開啟 **MusicActivity.java**
 - `package com.example.android.application2;`
 - `import android.media.MediaPlayer;`
 - `import android.support.v7.app.AppCompatActivity;`
 - `import android.os.Bundle;`
 - `import android.view.View;`
 - `import android.widget.TextView;`
 - `public class MusicActivity extends AppCompatActivity`
 - `implements MediaPlayer.OnCompletionListener {`
 - `private TextView output;`
 - `private MediaPlayer player;`



ANDROID 硬體裝置與應用

- 開啟 **MusicActivity.java**
 - `@Override`
 - `protected void onCreate(Bundle savedInstanceState) {`
 - `super.onCreate(savedInstanceState);`
 - `setContentView(R.layout.activity_music);`
 - `// 取得TextView元件`
 - `output = (TextView) findViewById(R.id.lblOutput);`
 - `player = MediaPlayer.create(this, R.raw.piano01);`
 - `player.setOnCompletionListener(this);`
 - `}`
 - `@Override`
 - `public void onCompletion(MediaPlayer mediaPlayer) {`
 - `output.setText("音樂已經播放完畢...");`
 - `player.seekTo(0); // 從頭開始`
 - `}`



ANDROID 硬體裝置與應用

- 開啟 **MusicActivity.java**
 - // Button元件的事件處理
 - `public void button_Click(View view) {`
 - `if (player != null) {`
 - `if (player.isPlaying() == false) {`
 - `output.setText("音樂播放中...");`
 - `player.start(); // 播放`
 - `}`
 - `}`
 - `}`
 - `public void button2_Click(View view) {`
 - `output.setText("音樂暫停中...");`
 - `if (player != null) {`
 - `player.pause(); // 暫停`



ANDROID 硬體裝置與應用

- 開啟 **MusicActivity.java**

```
○ }  
○ }  
○ public void button3_Click(View view) {  
○     output.setText("音樂已經停止播放...");  
○     if (player != null) {  
○         player.stop();  
○         player.prepareAsync(); // 準備  
○     }  
○ }
```



ANDROID 硬體裝置與應用

- 開啟 **MusicActivity.java**
 - @Override
 - public void onDestroy() {
 - if (player != null) {
 - player.release(); // 釋放MediaPlayer物件
 - }
 - super.onDestroy();
 - }
 - }



ANDROID 硬體裝置與應用

- 成果





ANDROID 硬體裝置與應用

- **VideoView**元件、**MediaController**物件
 - 提供控制影片播放功能，建立一個介面簡單的視訊播放器
 - 將一個mp4影片檔案複製到「res/raw」目錄
 - 在佈局檔中使用VideoView元件標籤語法建立影片播放區
 - 在Activity類別的onCreate()方法取得VideoView物件
 - 使用Uri類別的parse()方法，將URI字串轉換成URI物件
 - 呼叫setVideoURI()方法指定影片檔所在的URI物件



ANDROID 硬體裝置與應用

- **VideoView**元件、**MediaController**物件
 - 使用**setMediaController()**方法，實體化 **MediaController** 物件來控制影片播放
 - 呼叫**start()**方法來播放影片
 - `video = (VideoView) findViewById(R.id.videoView);`
 - `int id = getResources().getIdentifier(videoFile, "raw",`
`this.getPackageName());`
 - `final String path = "android.resource://" + this.getPackageName() + "/" +`
`id;`
 - `video.setVideoURI(Uri.parse(path));`
 - `video.setMediaController(new MediaController(this));`
 - `video.start();`



ANDROID 硬體裝置與應用

- **VideoView**元件、**MediaController**物件
 - 覆寫onPause() 方法
 - 呼叫stopPlayback()方法停止影片播放
 - `public void onPause() {`
 - `super.onStop();`
 - `video.stopPlayback();`
 - `}`
 - 執行專案就可以開始播放影片



ANDROID 硬體裝置與應用

- **VideoView**物件常用方法

方法	定義
start()	開始播放
pause()	暫停播放
resume()	恢復暫停播放
stopPlayback()	停止播放
setMediaController()	指定使用的MediaController物件
isPlaying()	判斷目前是否正在播放中，傳回值true為是；false為否



ANDROID 硬體裝置與應用

- 開啟 **VedioActivity.java**
 - `package com.example.android.application2;`
 - `import android.net.Uri;`
 - `import android.support.v7.app.AppCompatActivity;`
 - `import android.os.Bundle;`
 - `import android.widget.MediaController;`
 - `import android.widget.VideoView;`
 - `public class VedioActivity extends AppCompatActivity {`
 - `private VideoView video;`
 - `private String videoFile = "point21";`



ANDROID 硬體裝置與應用

- 開啟 **VedioActivity.java**
 - `@Override`
 - `protected void onCreate(Bundle savedInstanceState) {`
 - `super.onCreate(savedInstanceState);`
 - `setContentView(R.layout.activity_vedio);`
 - `// 取得VideoView元件`
 - `video = (VideoView)findViewById(R.id.videoView);`
 - `// 指定影片的URI`
 - `int id = getResources().getIdentifier(videoFile, "raw",`
 - `this.getPackageName());`
 - `final String path = "android.resource://" +`
 - `this.getPackageName() + "/" + id;`



ANDROID 硬體裝置與應用

- 開啟 **VedioActivity.java**

- `video.setVideoURI(Uri.parse(path));`
- `// 指定MediaController`
- `video.setMediaController(new MediaController(this));`
- `// 開始播放`
- `video.start();`
- `}`
- `@Override`
- `public void onPause() {`
- `super.onStop();`
- `// 停止播放`
- `video.stopPlayback();`
- `}`
- `}`



ANDROID 硬體裝置與應用

- 成果





ANDROID 硬體裝置與應用

- 行動裝置相機硬體
 - 使用Intent啟動相機進行拍攝
 - 使用ImageView元件顯示結果相片
- 建立Intent，動作類型是ACTION_IMAGE_CAPTURE
 - `Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);`
 - `startActivityForResult(intent, REQUEST_IMAGE);`
- 呼叫startActivityForResult()方法開啟有回傳值的活動
- 設定回傳碼REQUEST_IMAGE，開啟內建相機程式



ANDROID 硬體裝置與應用

- 行動裝置相機硬體

- 利用onActivityResult()方法取得從startActivityForResult()方法回傳的相片(回傳值)
- 建立Bitmap物件，顯示在ImageView元件中
 - @Override
 - protected void onActivityResult(int requestCode,
 - int resultCode, Intent data) {
 - if (requestCode == REQUEST_IMAGE &&
 - resultCode == Activity.RESULT_OK) {
 - Bitmap userImage = (Bitmap) data.getExtras().get("data");
 - image.setImageBitmap(userImage);
 - }
 - }



ANDROID 硬體裝置與應用

- 開啟 **AndroidManifest.xml**

- `<uses-feature android:name="android.hardware.camera" />`
- `<uses-feature android:name="android.hardware.camera.autofocus" />`
- `<uses-permission android:name="android.permission.CAMERA"></uses-permission>`
- `<application`
- `android:allowBackup="true"`
- `android:icon="@mipmap/ic_launcher"`
- `android:label="@string/app_name"`
- `android:supportsRtl="true"`
- `android:theme="@style/AppTheme"`
- `>`
- `<activity android:name=".CameraActivity"></activity>`
- `</application>`



ANDROID 硬體裝置與應用

- 開啟 **CameraActivity.java**
 - `package com.example.android.application2;`
 - `import android.app.Activity;`
 - `import android.content.Intent;`
 - `import android.graphics.Bitmap;`
 - `import android.provider.MediaStore;`
 - `import android.support.v7.app.AppCompatActivity;`
 - `import android.os.Bundle;`
 - `import android.view.View;`
 - `import android.widget.ImageView;`
 - `public class CameraActivity extends AppCompatActivity {`
 - `private static final int REQUEST_IMAGE = 100;`



ANDROID 硬體裝置與應用

- 開啟 **CameraActivity.java**

- `private ImageView image;`
- `@Override`
- `protected void onCreate(Bundle savedInstanceState) {`
- `super.onCreate(savedInstanceState);`
- `setContentView(R.layout.activity_camera);`
- `// 取得ImageView元件`
- `image = (ImageView)findViewById(R.id.imageView);`
- `}`
- `// Button元件的事件處理`
- `public void button_Click(View view) {`
- `Intent intent = new Intent(`
- `MediaStore.ACTION_IMAGE_CAPTURE);`



ANDROID 硬體裝置與應用

- 開啟 **CameraActivity.java**

- `startActivityForResult(intent, REQUEST_IMAGE);`
- `}`
- `@Override`
- `protected void onActivityResult(int requestCode,`
- `int resultCode, Intent data) {`
- `if (requestCode == REQUEST_IMAGE &&`
- `resultCode == Activity.RESULT_OK) {`
- `Bitmap userImage = (Bitmap) data.getExtras().get("data");`
- `image.setImageBitmap(userImage);`
- `}`
- `}`
- `}`



ANDROID 硬體裝置與應用

• 成果





ANDROID 硬體裝置與應用

- 2D繪圖

- 利用繼承View類別覆寫的onDraw()方法進行繪圖
- 利用Canvas物件建立畫布和繪圖
 - public class Draw2D extends View {
 - public Draw2D(Context context) {
 - super(context);
 - }
 - @Override
 - protected void onDraw(Canvas canvas) {
 - // 繪圖
 - }
 - }



ANDROID 硬體裝置與應用

- 2D繪圖

- 覆寫 onCreate() 方法以建立Draw2D物件
- 利用 setContentView() 方法顯示Draw2D物件
 - @Override
 - public void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - Draw2D d = new Draw2D(this);
 - setContentView(d);
 - }



ANDROID 硬體裝置與應用

- 2D繪圖

- 建立畫筆 Paint物件
- 利用 `setStyle()` 方法設定畫筆樣式
- 利用 `setColor()` 方法指定畫筆色彩
- 利用 `drawPaint ()` 方法將Canvas物件填滿色彩
 - `Paint paint = new Paint();`
 - `paint.setStyle(Paint.Style.FILL);`
 - `paint.setColor(Color.WHITE);`
 - `canvas.drawPaint(paint);`



ANDROID 硬體裝置與應用

- 2D繪圖

- 利用 `drawCircle()` 方法畫出圓形
- 前2個參數是(x, y)座標，參數3是半徑，參數4是畫筆 `Paint` 物件
- `setAntiAlias()` 設定線條反鋸齒功能
 - `paint.setAntiAlias(true);`
 - `paint.setColor(Color.RED);`
 - `canvas.drawCircle(80, 30, 25, paint);`
- `drawRect()` 方法畫出長方形



ANDROID 硬體裝置與應用

- 2D繪圖

- drawRect()方法畫出長方形
- 前4個參數為兩組(x,y)是左上角和右下角座標，參數5是畫筆物件
 - paint.setColor(Color.BLUE);
 - canvas.drawRect(20, 15, 50, 100, paint);
- drawText()方法畫文字內容
- 參數1為要顯示的字串，參數2和3為(x,y)座標，參數4是畫筆Paint物件
 - paint.setTextSize(30);
 - canvas.drawText("Android應用程式APP!", 50, 180, paint);



ANDROID 硬體裝置與應用

- 2D繪圖

- rotate()方法設定旋轉角度
- 參數1是旋轉角度，參數2和3是旋轉軸座標(x,y)，以此座標為軸心，將Canvas物件旋轉參數1的角度
 - canvas.rotate(-45, 200, 200);
 - canvas.drawText(str, 200, 200, paint);
- drawBitmap()方法畫出點陣圖檔
- 參數1是Bitmap物件，參數2和3是位置(x,y)座標，參數4是畫筆Paint物件
 - Bitmap bitmap = BitmapFactory.decodeResource(res, R.drawable.main);
 - canvas.drawBitmap(bitmap, 50 ,200 , paint);



ANDROID 硬體裝置與應用

- 開啟 **Painter2D.java**
 - `package com.example.android.application2;`
 - `import android.content.Context;`
 - `import android.content.res.Resources;`
 - `import android.graphics.Bitmap;`
 - `import android.graphics.BitmapFactory;`
 - `import android.graphics.Canvas;`
 - `import android.graphics.Color;`
 - `import android.graphics.Paint;`
 - `import android.view.View;`
 - `public class Painter2D extends View {`
 - `public Painter2D(Context context) {`



ANDROID 硬體裝置與應用

- 開啟 Painter2D.java

- `super(context);`
- `}`
- `@Override`
- `protected void onDraw(Canvas canvas) {`
- `super.onDraw(canvas);`
-
- `// 建立Paint物件`
- `Paint paint = new Paint();`
- `paint.setStyle(Paint.Style.FILL);`
-
- `// 在整個Canvas物件的背景填滿色彩`
- `paint.setColor(Color.WHITE);`
- `canvas.drawPaint(paint);`



ANDROID 硬體裝置與應用

- 開啟 Painter2D.java

- // 畫圓
- `paint.setAntiAlias(true);`
- `paint.setColor(Color.RED);`
- `canvas.drawCircle(80, 30, 25, paint);`
- // 畫長方形
- `paint.setColor(Color.BLUE);`
- `canvas.drawRect(20, 15, 50, 100, paint);`
- // 畫出資源圖形
- `Resources res = this.getResources();`
- `Bitmap bitmap = BitmapFactory.decodeResource(res, R.drawable.main);`
- `canvas.drawBitmap(bitmap, 50, 200, paint);`



ANDROID 硬體裝置與應用

- 開啟 **Painter2D.java**

- // 畫出文字內容
- `paint.setColor(Color.GREEN);`
- `paint.setStyle(Paint.Style.FILL);`
- `paint.setAntiAlias(true);`
- `paint.setTextSize(30);`
- `canvas.drawText("Android應用程式!", 50, 180, paint);`

- // 畫出旋轉的文字
- `paint.setColor(Color.BLACK);`
- `paint.setTextSize(25);`
- `String str = "旋轉Android文字物件!";`



ANDROID 硬體裝置與應用

- 開啟 Painter2D.java

- // 旋轉繪出文字內容
 - canvas.rotate(-45, 200, 200);
 - paint.setStyle(Paint.Style.FILL);
 - canvas.drawText(str, 200, 200, paint);
 - }
 - }



ANDROID 硬體裝置與應用

- 成果





 **THE END**