



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

- Timer類別
- ImageView顯示圖片
- xml動畫
- 播放與錄製音訊
- 小專題-翻牌遊戲

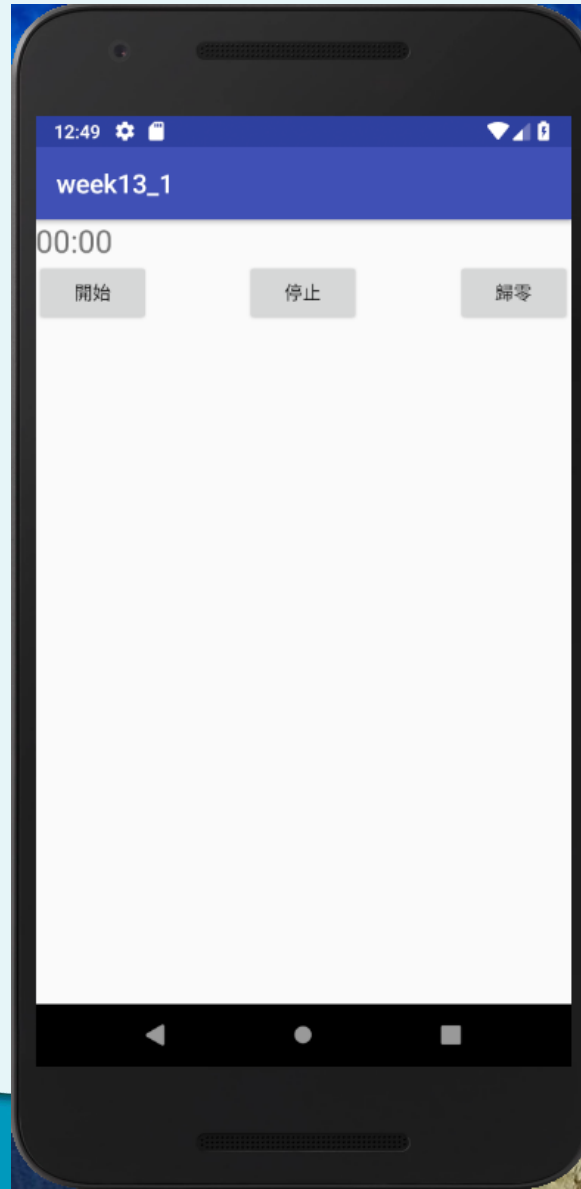


國立宜蘭大學

National Ilan University NIU

Timer類別

Timer範例



國立宜蘭大學

National Ilan University NIU



Timer

- **Timer類別**的功能可以每隔特定時間執行程式碼，且可以設定執行程式的間隔時間，也可以指定執行時間。
- 常使用在**碼錶**、**倒數計時器**或是需**定期更新**的程式中。
- Timer的計時單位為**毫秒(0.001秒)**。

UI介面



activity_main.xml

- 步驟1. 設定Layout成RelativeLayout。
- 步驟2. 拉一個TextView和三個Button。
- 步驟3. 在TextView的text設定成“00 : 00”。
- 步驟4. 左邊Button的text設定成“開始”，中邊Button的text設定成“停止”，右邊Button的text設定成“歸零”。

開始打程式 MainActivity.java

- 步驟1. 宣告UI物件。

```
TextView tv;  
Button btn1, btn2, btn3;
```

- 步驟2. 設定Timer類別。

```
Timer timer = new Timer();
```

- 步驟3. 宣告秒數和判定按鈕是否要啟動Timer的布林值。

```
int sec = 0;  
Boolean btn = false;
```


開始打程式 MainActivity.java

- 步驟4. 建立程式與元件的關係。

```
tv = (TextView) findViewById(R.id.textView1);  
btn1 = (Button) findViewById(R.id.button1);  
btn2 = (Button) findViewById(R.id.button2);  
btn3 = (Button) findViewById(R.id.button3);
```

- 步驟5. 執行Timer物件。

```
timer.schedule(new TimerTask() {  
  
    @Override  
    public void run() {  
        // TODO Auto-generated method stub  
        間隔時間(每隔幾毫秒執行一次)  
    }  
}, 0, 1000);
```

間隔時間(每隔幾毫秒執行一次)

延遲時間(開始執行時間)

開始打程式 MainActivity.java

- 步驟6. 在介紹TimerTask前，先設定Handler來更新UI元件的資料，因為TimerTask無法更新UI元件，因此使用Handler來輔助。

```
private Handler handler = new Handler() {  
  
    @Override  
    public void handleMessage(Message msg) {  
        // TODO Auto-generated method stub  
        super.handleMessage(msg);  
    }  
};
```

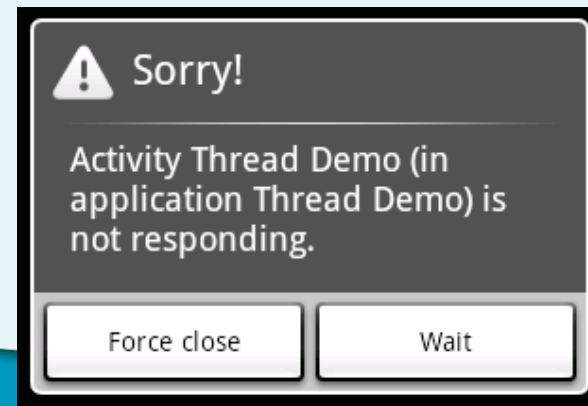
P.S.有super的皆可在此空間的空白處，按下右鍵
Source→Override/Implement Methods中找到。

Handler補充說明

- 每當開始啟動android應用程式，會有一個**main**的執行緒自動地被產生。
這個**主執行緒**（也叫**UI執行緒**）非常重要，因為它用來調度事件給適合的元件也包含繪製使用者介面的事件。它也是那個你與Android互動的**widget**所在的執行緒。

Handler補充說明

- 如果在一個單一執行緒發生的每件事都做了很久的運算，像是網路存取或者跟資料庫要資料，這樣的執行緒將會把使用者的畫面鎖住。當這些長時間的運算正在進行中時，將會沒有任何事件可以被觸發，包含重畫畫面的事件。當使用者看到畫面停住不動，甚至更壞的情況，像UI執行緒佔住畫面約超過5秒的時間，畫面就會跳出ANR (Application Not Responding)警告視窗。



Handler補充說明

- 因此我們需要Handler，Handler專門用來告知你的程式說：你的執行緒跑完了，該更新一下畫面了喔！所以利用Handler把message send出去處理，就可以解決因為超過五秒時間很出現的ANR警告視窗唷！

開始打程式 MainActivity.java

- 步驟7. **TimerTask**是**Timer**的主體，主要用來定義**Timer**定時要執行的工作內容，透過**Message**的方式來區別使用方法，並以**btn**的布林來確認是否有按按鈕。

```
Message message = new Message();  
if (btn)  
    message.what = 1;  
handler.sendMessage(message);
```

Timer完整程式碼

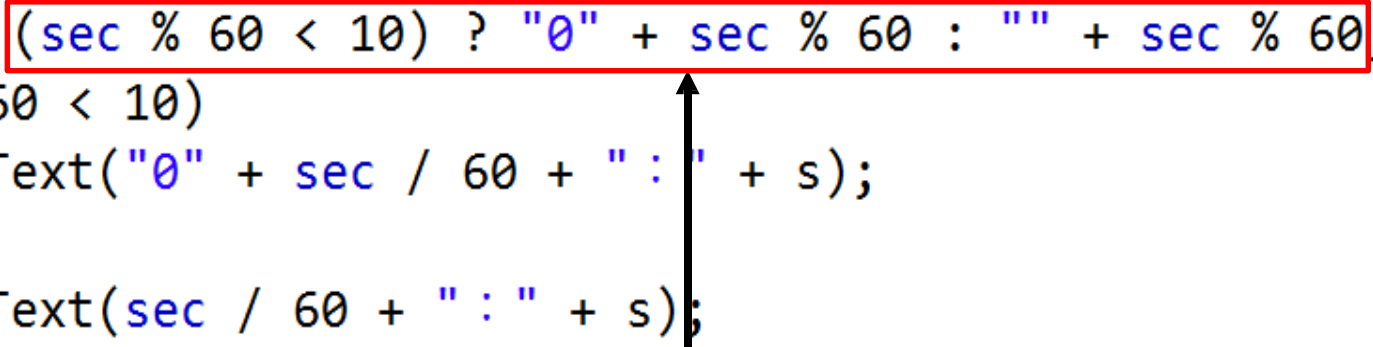
```
timer.schedule(new TimerTask() {  
  
    @Override  
    public void run() {  
        // TODO Auto-generated method stub  
        Message message = new Message();  
        if (btn)  
            message.what = 1;  
        handler.sendMessage(message);  
    }  
}, 0, 1000);
```

開始打程式 MainActivity.java

- 步驟8. 當按下啟動按鈕後，便會開始計時。

(判斷式) ? 符合時結果 : 不符合時結果

```
sec++;  
String s = (sec % 60 < 10) ? "0" + sec % 60 : "" + sec % 60;  
if (sec / 60 < 10)  
    tv.setText("0" + sec / 60 + ":" + s);  
else  
    tv.setText(sec / 60 + ":" + s);
```



判定秒數小於10時補足成雙位數字。

完整Handler程式碼

```
private Handler handler = new Handler() {  
  
    @Override  
    public void handleMessage(Message msg) {  
        // TODO Auto-generated method stub  
        super.handleMessage(msg);  
        if (msg.what == 1) {  
            sec++;  
            String s = (sec % 60 < 10) ? "0" + sec % 60 : "" + sec % 60;  
            if (sec / 60 < 10)  
                tv.setText("0" + sec / 60 + ":" + s);  
            else  
                tv.setText(sec / 60 + ":" + s);  
        }  
    }  
};
```

開始打程式 MainActivity.java

- 步驟9. 建立三個按鈕的監聽事件。

```
btn1.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View arg0) {  
        // TODO Auto-generated method stub  
  
    }  
});
```

- 步驟10. 在左邊的按鈕建立啟動Timer的布林，以及中間按鈕不啟動的布林。

```
btn = true;
```

```
btn = false;
```

完整左中按鈕程式碼

```
btn1.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View arg0) {  
        // TODO Auto-generated method stub  
        btn = true;  
    }  
});  
  
btn2.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        btn = false;  
    }  
});
```

開始打程式 MainActivity.java

- 步驟11. 在右邊按鈕設定成初始數值。

```
btn = false;  
sec = 0;  
tv.setText("00 : 00");
```

完整右按鈕程式碼

```
btn3.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        btn = false;  
        sec = 0;  
        tv.setText("00 : 00");  
    }  
});
```

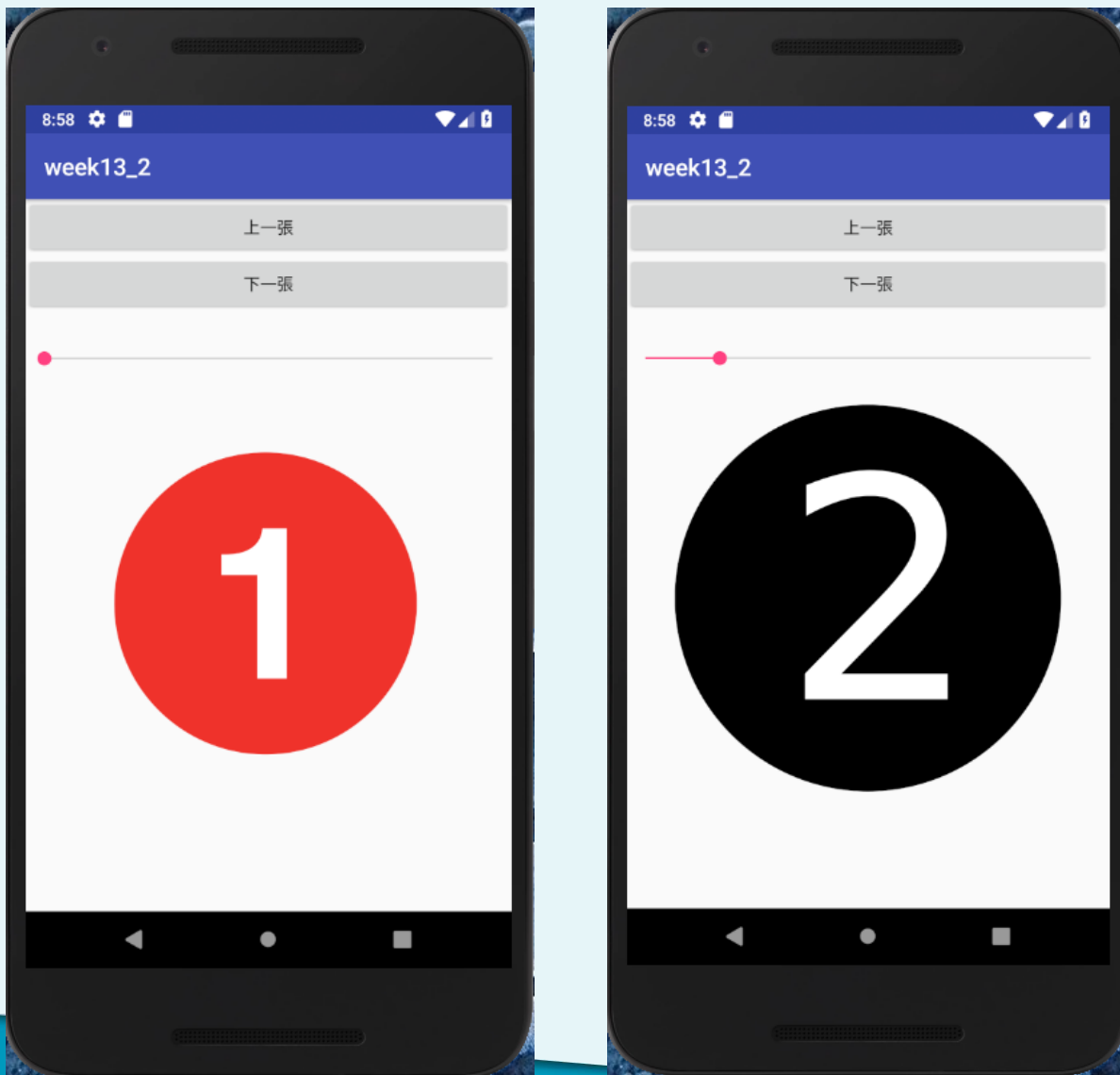


國立宜蘭大學

National Ilan University NIU

ImageView 顯示圖片

ImageView範例

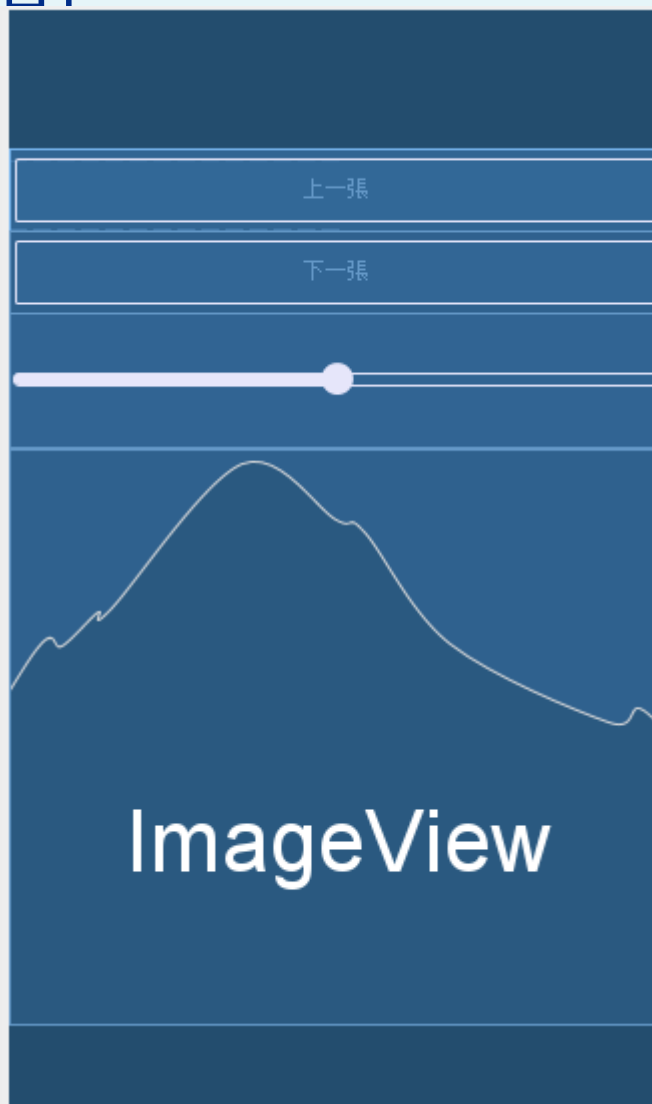
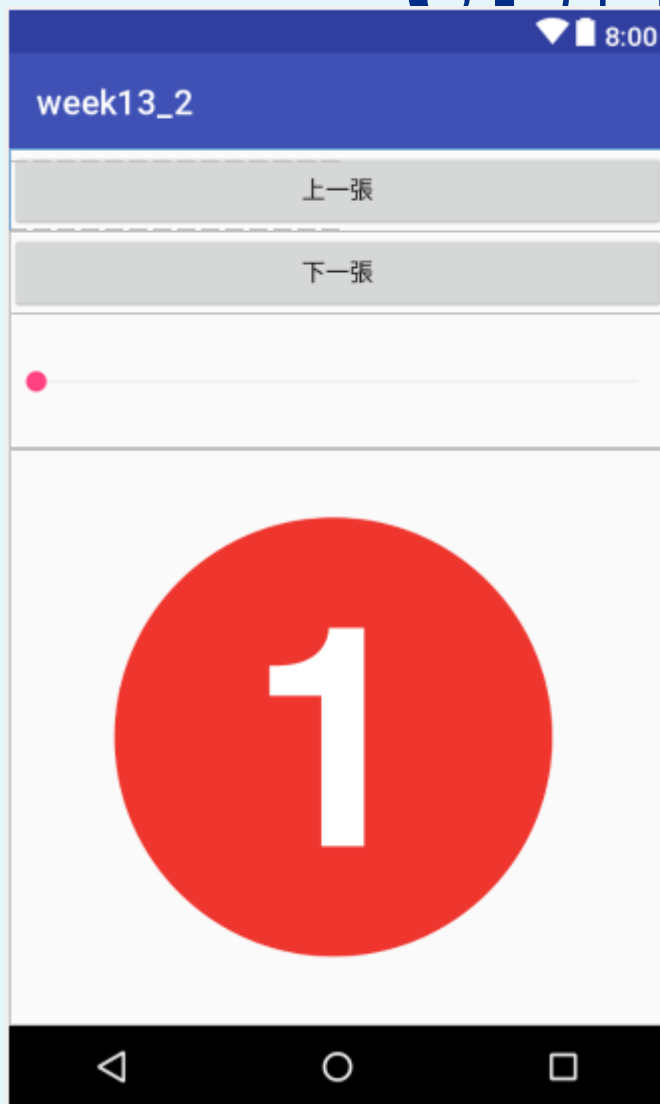




ImageView

- ImageView主要是用來顯示影像，讓畫面更加美觀。
- 程式的影像格式可以使用.png、.jpg或是.gif，特別要注意檔案名稱必須以英文小寫開頭，並且不能有空白。

UI介面



國立宜蘭大學

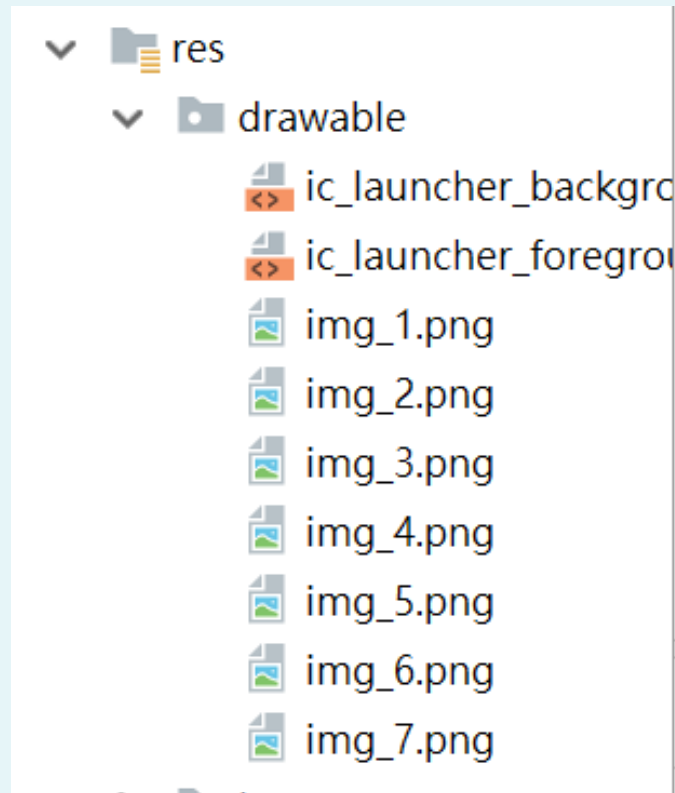
National Ilan University NIU

activity_main.xml

- 步驟1. 設定Layout成RelativeLayout。
- 步驟2. 拉兩個按鈕、拖動條(SeekBar)和ImageView，設定ImageView為預設圖片(ic_launcher)或是clear(設定為空會看不到元件位置比較不推薦)。
- 步驟3. 將Button的text設定成“上一張”和“下一張”。

開始打程式 MainActivity.java

- 步驟1. 載入圖片(助教提供)，直接複製貼上 drawable-的檔案格式中即可。



開始打程式 MainActivity.java

- 步驟2. 宣告UI物件。

```
Button mButton1, mButton2;  
SeekBar mSeekBar;  
ImageView mImageView;
```

- 步驟3. 宣告圖片陣列。

```
int[] img = new int[]{R.drawable.img_1, R.drawable.img_2, R.drawable.img_3,  
    R.drawable.img_4, R.drawable.img_5, R.drawable.img_6, R.drawable.img_7};
```

- 步驟4. 宣告Bitmap圖片格式。

```
Bitmap bitmap;
```

開始打程式 MainActivity.java

- 步驟5. 宣告取得圖片號碼變數。

```
int i = 0;
```

- 步驟6. 建立程式與元件的關係。

```
mButton1 = findViewById(R.id.button);  
mButton2 = findViewById(R.id.button2);  
mseekBar = findViewById(R.id.seekBar);  
mImageView = findViewById(R.id.imageView);
```

- 步驟7. 將圖片設定成第一張圖，由於i初始載入為0，因此為第一張圖。

```
im.setImageResource(img[i]);
```

開始打程式 MainActivity.java

- 步驟8. 設定SeekBar的最大值(6)與顯示位置。

```
mseekBar.setMax(6);  
mseekBar.setProgress(i);
```

- 步驟9. 建立兩個按鈕的監聽事件。

```
btn1.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View arg0) {  
        // TODO Auto-generated method stub  
  
    }  
});
```

開始打程式 MainActivity.java

- 步驟10. 將上面的按鈕設定更換上一張圖片，並將拉條條回初始位置，當按鈕到底時不會執行。

```
if (i>0){  
    i--;  
    mImageView.setImageResource(img[i]);  
    mseekBar.setProgress(i);  
}
```

- 步驟11. 將下面的按鈕設定更換下一張圖片，並將拉條條回初始位置，當按鈕到上限時不會執行

```
if (i<img.length-1){  
    i++;  
    mImageView.setImageResource(img[i]);  
    mseekBar.setProgress(i);  
}
```

← 因為陣列從0開始計算，所以判定要比陣列數量少1。

完整上一張按鈕程式碼

```
mButton1.setOnClickListener(new Button.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        if (i>0){  
            i--;  
            mImageView.setImageResource(img[i]);  
            mseekBar.setProgress(i);  
        }  
    }  
});
```


完整下一張按鈕程式碼

```
mButton2.setOnClickListener(new Button.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        if (i<img.length-1){  
            i++;  
            mImageView.setImageResource(img[i]);  
            mseekBar.setProgress(i);  
        }  
    }  
});
```

開始打程式 MainActivity.java

- 步驟12. 建立SeekBar動態拉動監聽。

```
sb.setOnSeekBarChangeListener(new OnSeekBarChangeListener() {  
  
    @Override  
    public void onStopTrackingTouch(SeekBar seekBar) {  
        // TODO Auto-generated method stub  
    }  
  
    @Override  
    public void onStartTrackingTouch(SeekBar seekBar) {  
        // TODO Auto-generated method stub  
    }  
  
    @Override  
    public void onProgressChanged(SeekBar seekBar, int progress,  
        boolean fromUser) {  
        // TODO Auto-generated method stub  
    }  
});
```

↑
放開時。

↑
按下時。

↑
移動時。

開始打程式 MainActivity.java

- 步驟13. 在SeekBar監聽內宣告取得拉條值的變數，並以此當為最小值。

```
int pg = 1;
```

- 步驟14. 在onProgressChanged內，將圖片進行等比例的縮放。

```
pg = a + 1 ;  
bitmap = BitmapFactory.decodeResource(getResources(),img[i]);  
int bmpwidth = bitmap.getWidth();  
int bmpheight = bitmap.getHeight();  
Matrix matrix = new Matrix();  
matrix.postScale( sx: (float)pg/10, sy: (float)pg/10);  
Bitmap resizeBmp = Bitmap.createBitmap(bitmap, x: 0, y: 0, bmpwidth, bmpheight,  
    matrix, filter: true);  
mImageView.setImageBitmap(resizeBmp);  
mImageView.setImageResource(img[a]);
```

圖片縮放說明

- 1) 取得拉條值，並以初始值為1開始計算。

```
pg = a + 1 ;
```

- 2) 轉換圖片格式為Bitmap。

```
bitmap = BitmapFactory.decodeResource(getResources(), img[i]);
```

- 3) 取得圖片的原始大小。

```
int bmpWidth = bitmap.getWidth();  
int bmpHeight = bitmap.getHeight();
```

- 4) 宣告Matrix陣列來儲存圖片模型。

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

圖片縮放說明

- 5) 將圖片以取得的值，並除初始值(10)來確認縮放比例，最後放入Matrix陣列中。

```
matrix.postScale((float) pg / 10, (float) pg / 10);
```

- 6) 將Matrix陣列重新會突出圖片成Bitmap。

```
Bitmap resizeBmp = Bitmap.createBitmap(bitmap, 0, 0, bmpWidth,  
    bmpHeight, matrix, true);
```

- 7) 將圖片換成新的Bitmap圖片。

```
im.setImageBitmap(resizeBmp);
```

- 8) 依照SeekBar位置設定圖片。

```
mImageView.setImageResource(img[a]);
```

完整onProgressChanged程式碼

@Override

```
public void onProgressChanged(SeekBar seekBar, int a, boolean b) {  
  
    int pg = 1;  
    pg = a + 1 ;  
    bitmap = BitmapFactory.decodeResource(getResources(),img[i]);  
    int bmpwidth = bitmap.getWidth();  
    int bmpheight = bitmap.getHeight();  
    Matrix matrix = new Matrix();  
    matrix.postScale( sx: (float)pg/10, sy: (float)pg/10);  
    Bitmap resizeBmp = Bitmap.createBitmap(bitmap, x: 0, y: 0, bmpwidth, bmpheight,  
        | matrix, filter: true);  
    mImageView.setImageBitmap(resizeBmp);  
    mImageView.setImageResource(img[a]);  
}
```



National Ilan University NIU
國立宜蘭大學



國立宜蘭大學
National Ilan University NIU

xml動畫

xml動畫範例



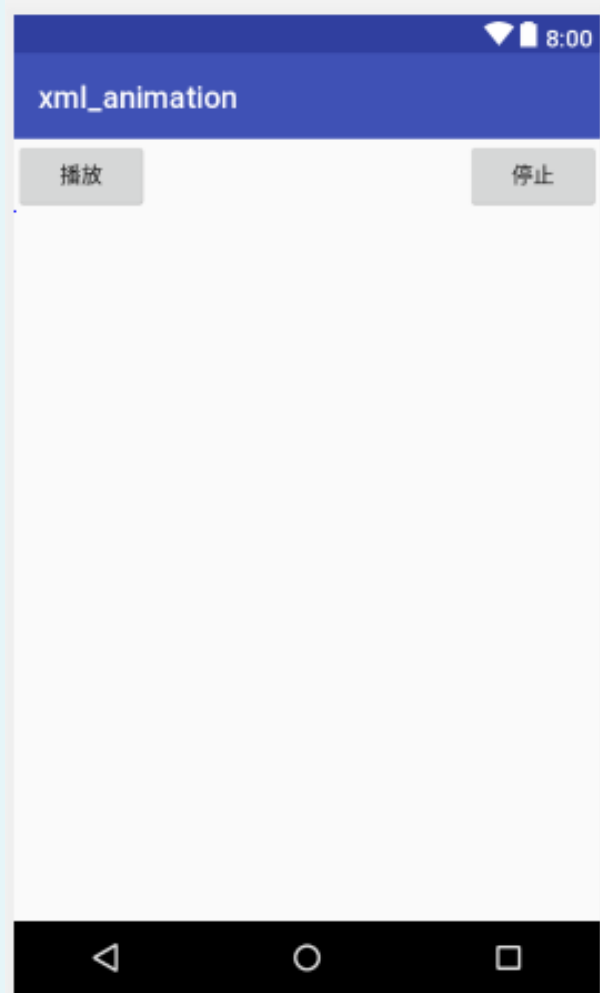
國立宜蘭大學

National Ilan University NIU

xml播放動畫方式

1. 補間動畫(tweening)讓第一個圖片逐漸轉變成第二張圖片。
 - 方法有位移、縮放、旋轉和透明化。
 - xml的檔案通常都會放在「**res/anim**」目錄。
2. Drawable動畫利用ImageView將圖片以一定的時間播放一連串的圖片。
 - Xml的檔案通常都會放在「**res/drawable**」目錄。
 - 需要搭配**AnimationDrawable**類別才可播放動畫。

UI介面



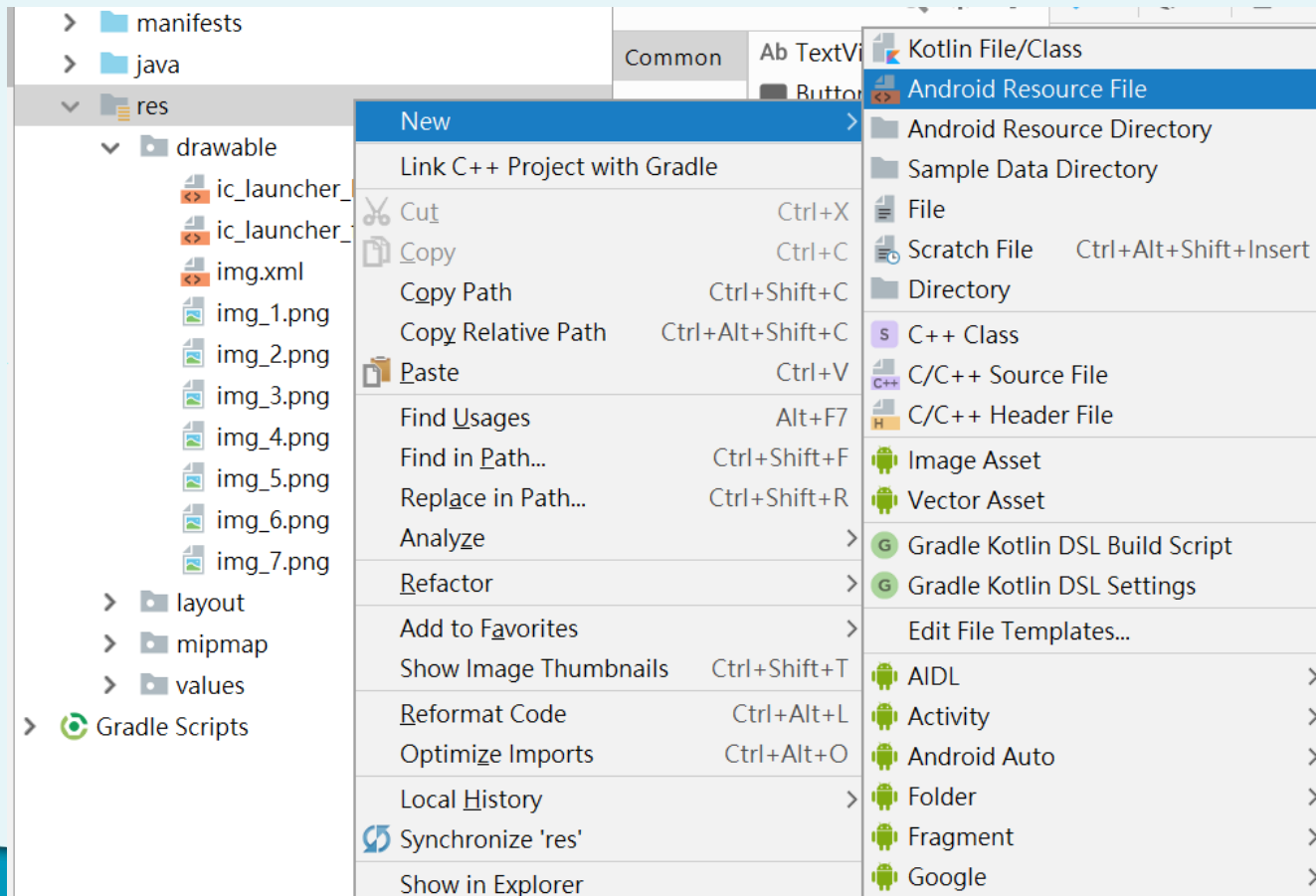
activity_main.xml

- 步驟1. 設定Layout成RelativeLayout。
- 步驟2. 拉兩個按鈕和ImageView，設ImageView為空(先任意選一個圖檔，再到srcCompat將設定清空)。
- 步驟3. 將Button的text設定成“播放”和“停止”。

Attributes	
ID	imageView
layout_width	wrap_content
layout_height	wrap_content
ImageView	
srcCompat	
contentDescription	
background	
scaleType	none
adjustViewBounds	<input type="checkbox"/>
cropToPadding	<input type="checkbox"/>
Favorite Attributes	
visibility	none
View all attributes	

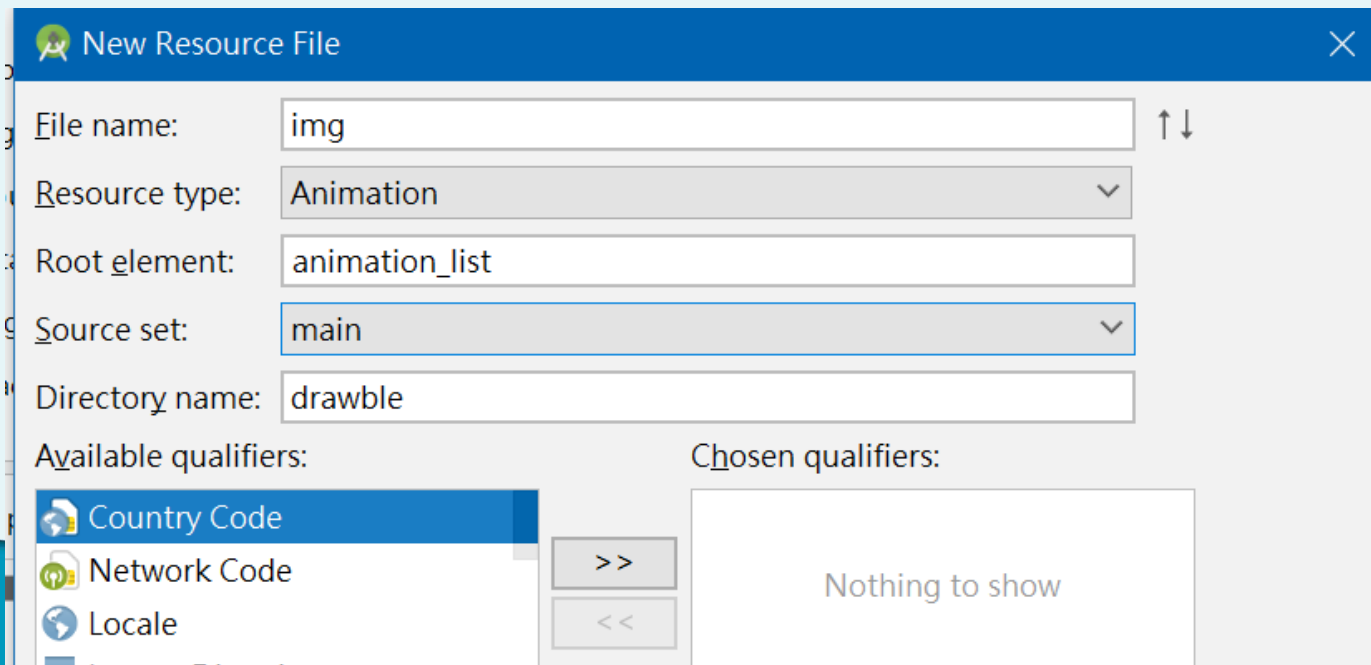
製作xml動畫格式

- 步驟1. 在res的資料夾，點擊滑鼠右鍵
New→Android XML File。



製作xml動畫格式

- 步驟2. 以下圖的方式輸入，
- File name 輸入 img，
- Resource type 選擇 Animation，
- Root Element 輸入 animation-list，最後OK。



製作xml動畫格式

- 步驟3. 打上下圖程式碼。

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<animation-list xmlns:android="http://schemas.android.com/apk/res/android" android:oneshot="false">
```

```
    <item android:drawable="@drawable/img_1" android:duration="150"/>
```

```
    <item android:drawable="@drawable/img_2" android:duration="150"/>
```

```
    <item android:drawable="@drawable/img_3" android:duration="150"/>
```

```
    <item android:drawable="@drawable/img_4" android:duration="150"/>
```

```
    <item android:drawable="@drawable/img_5" android:duration="150"/>
```

```
    <item android:drawable="@drawable/img_6" android:duration="150"/>
```

```
    <item android:drawable="@drawable/img_7" android:duration="150"/>
```

```
</animation-list>
```

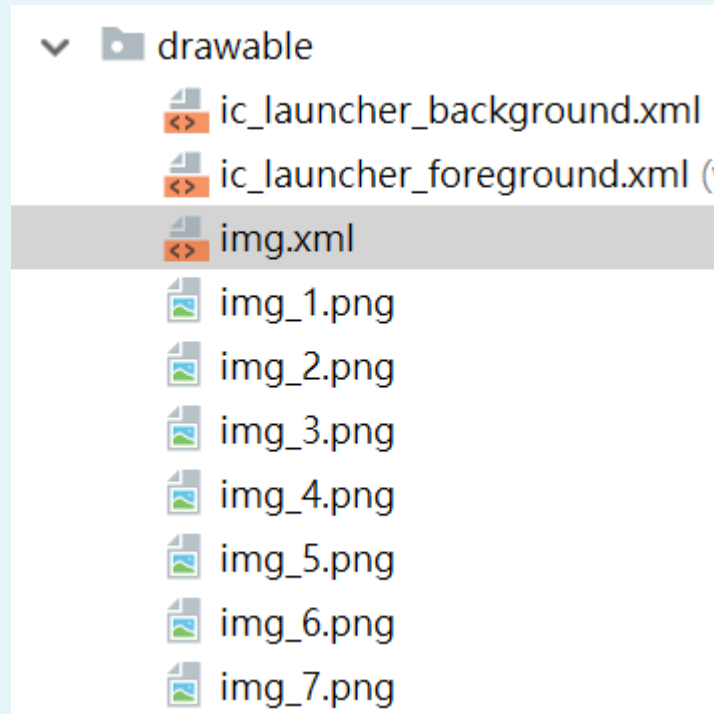
↑
是否要連續播放。

↑
更換的圖片名稱。

↑
切換的間隔時間。

開始打程式 MainActivity.java

- 步驟1. 載入圖片(助教提供)。



- 步驟2. MainActivity.java
- 宣告UI元件。

```
Button btn1, btn2;  
ImageView iv;
```


開始打程式 MainActivity.java

- 步驟3. 宣告播放動畫的類別AnimationDrawable。

```
AnimationDrawable animationDrawable;
```

- 步驟4. 建立程式與元件的關係。

```
iv=(ImageView) findViewById(R.id.imageView1);  
btn1=(Button) findViewById(R.id.button1);  
btn2=(Button) findViewById(R.id.button2);
```

- 步驟5. 將ImageView的背景設定成剛剛製作的xml檔。

```
iv.setBackgroundResource(R.drawable.img);
```

開始打程式 MainActivity.java

- 步驟6. 因為步驟5將ImageView的背景設定成xml的格式，所以這邊將AnimationDrawable與ImageView的背景建立關係。

```
animationDrawable=(AnimationDrawable) iv.setBackground();
```

- 步驟7. 建立兩個按鈕的監聽事件。

```
btn1.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View arg0) {  
        // TODO Auto-generated method stub  
  
    }  
});
```

開始打程式 MainActivity.java

- 步驟8. 在播放按鈕啟動動畫。

```
animationDrawable.start();
```

- 步驟9. 在停止按鈕關閉動畫。

```
animationDrawable.stop();
```



完整播放按鈕程式碼

```
btn1.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View arg0) {  
        // TODO Auto-generated method stub  
        animationDrawable.start();  
    }  
});
```



完整停止按鈕程式碼

```
btn2.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        animationDrawable.stop();  
    }  
});
```



National Ilan University NIU
國立宜蘭大學



國立宜蘭大學

National Ilan University NIU

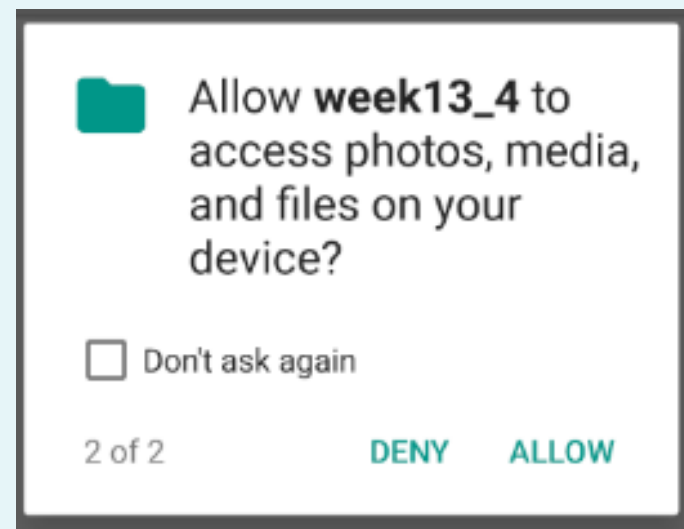
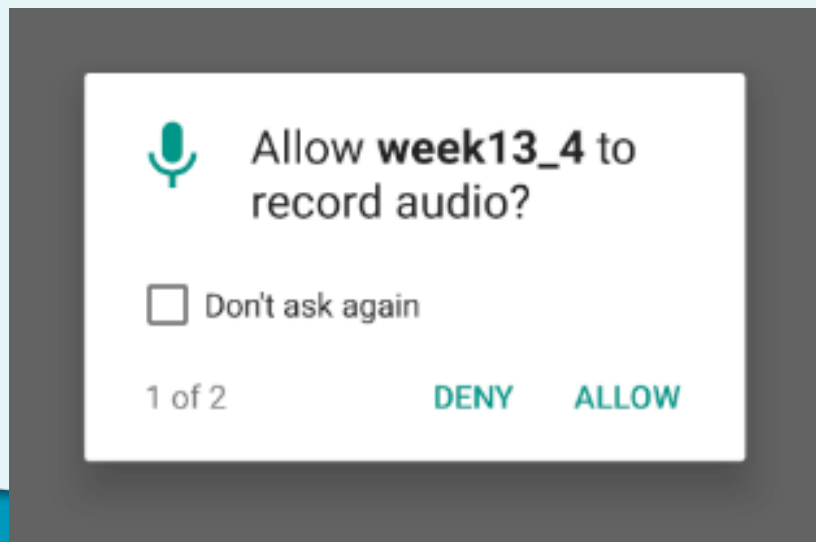
播放與錄製音訊

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



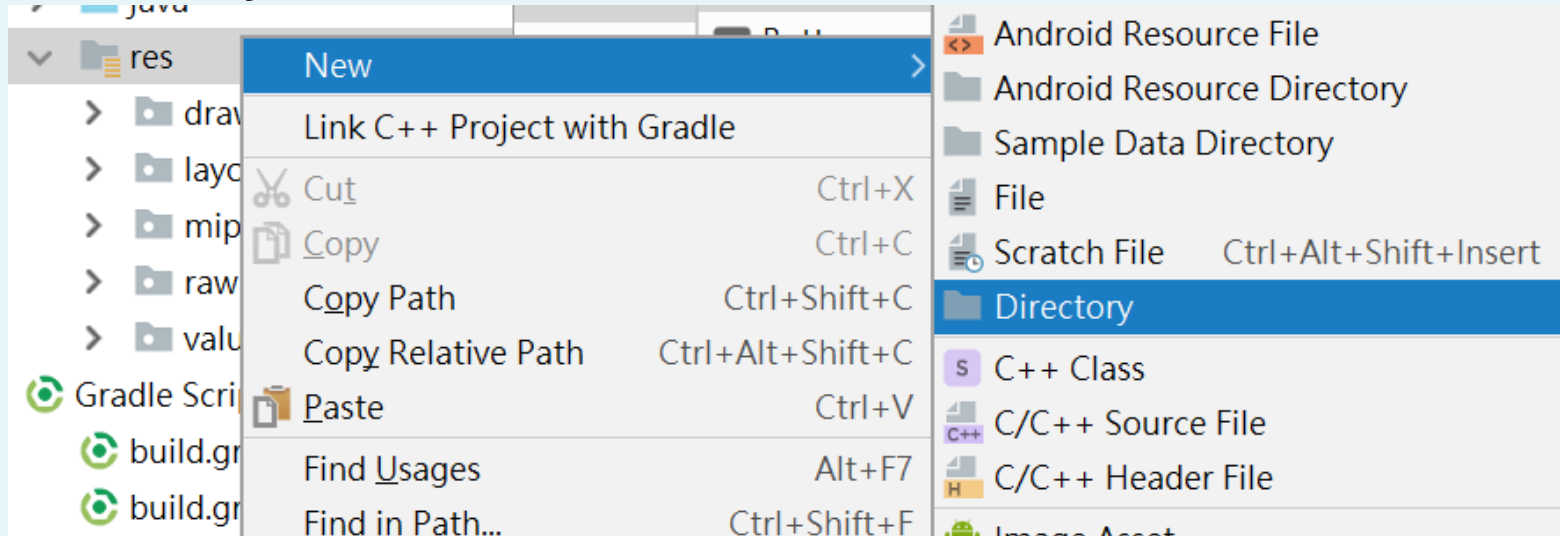


activity_main.xml

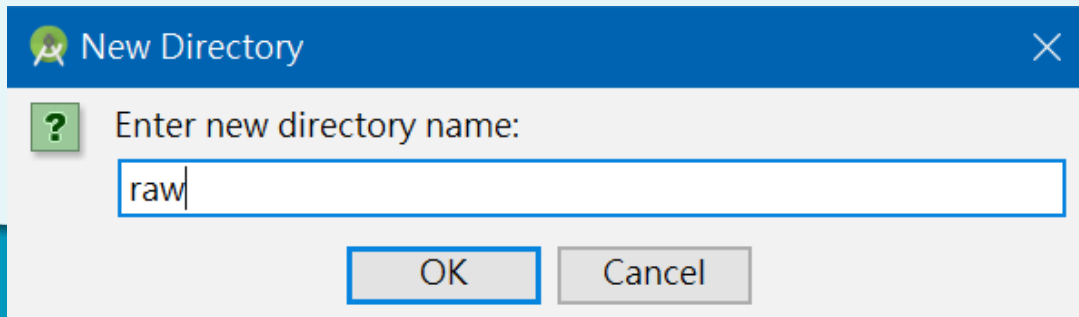
- 步驟1. 設定Layout成RelativeLayout。
- 步驟2. 拉五個Button元件。
- 步驟3. 將按鈕的text設定成與UI介面相同。

開始打程式 MainActivity.java

- 步驟1. 在res新增資料夾，右鍵 new > directory

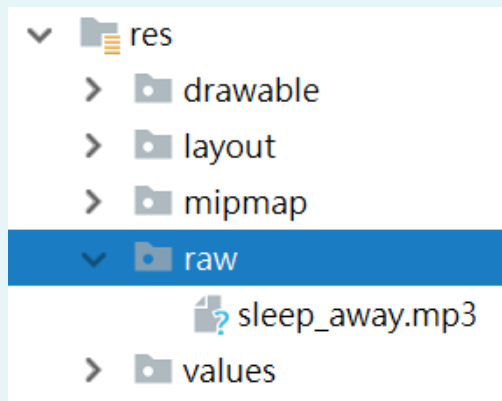


- 步驟2. 命名為raw。



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;  
        }  
    }  
});
```



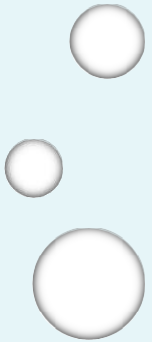
完整onPause()程式碼

```
@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

小專題- 翻牌遊戲

翻牌遊戲示範

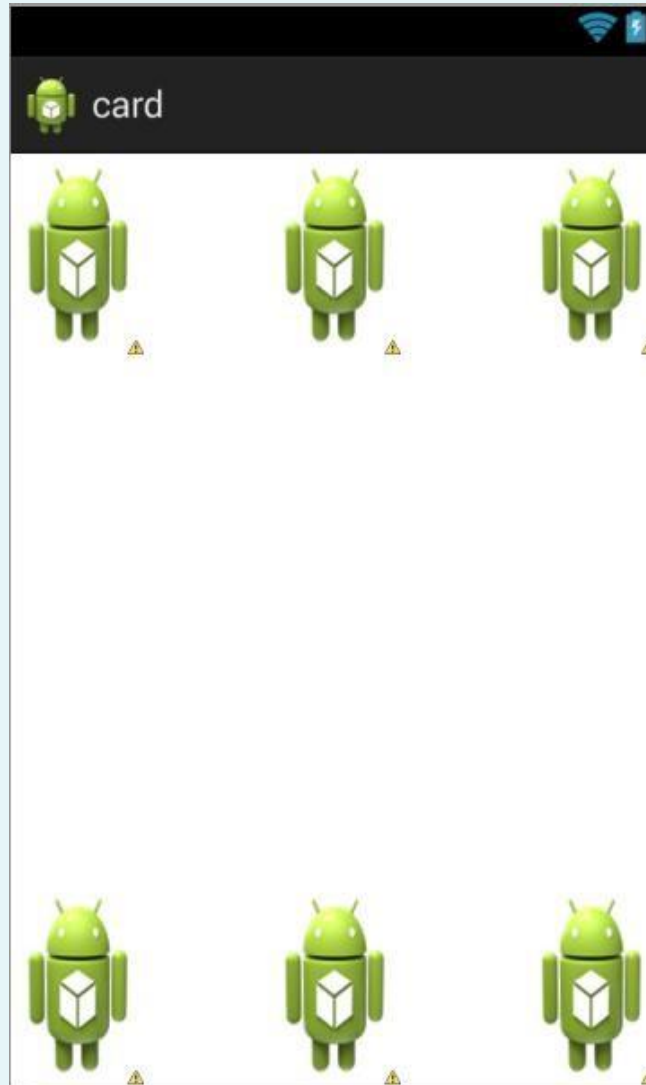


國立宜蘭大學

National Ilan University NIU



UI介面

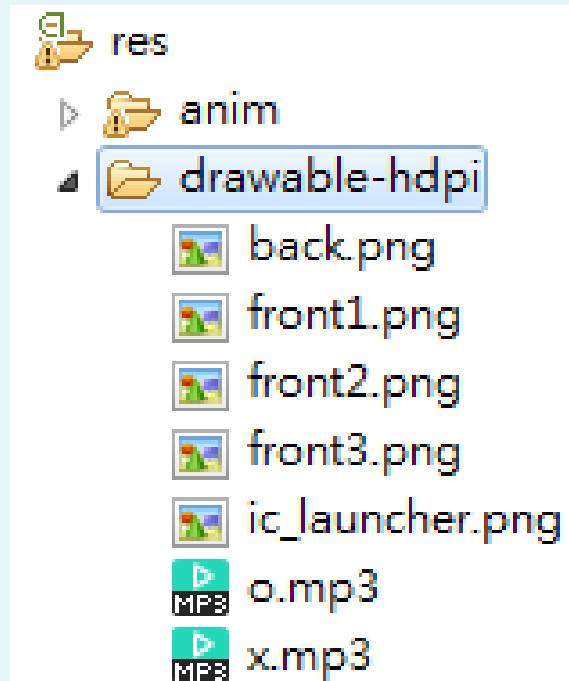


國立宜蘭大學

National Ilan University NIU

載入檔案

- 首先，先載入圖片和音樂檔案。



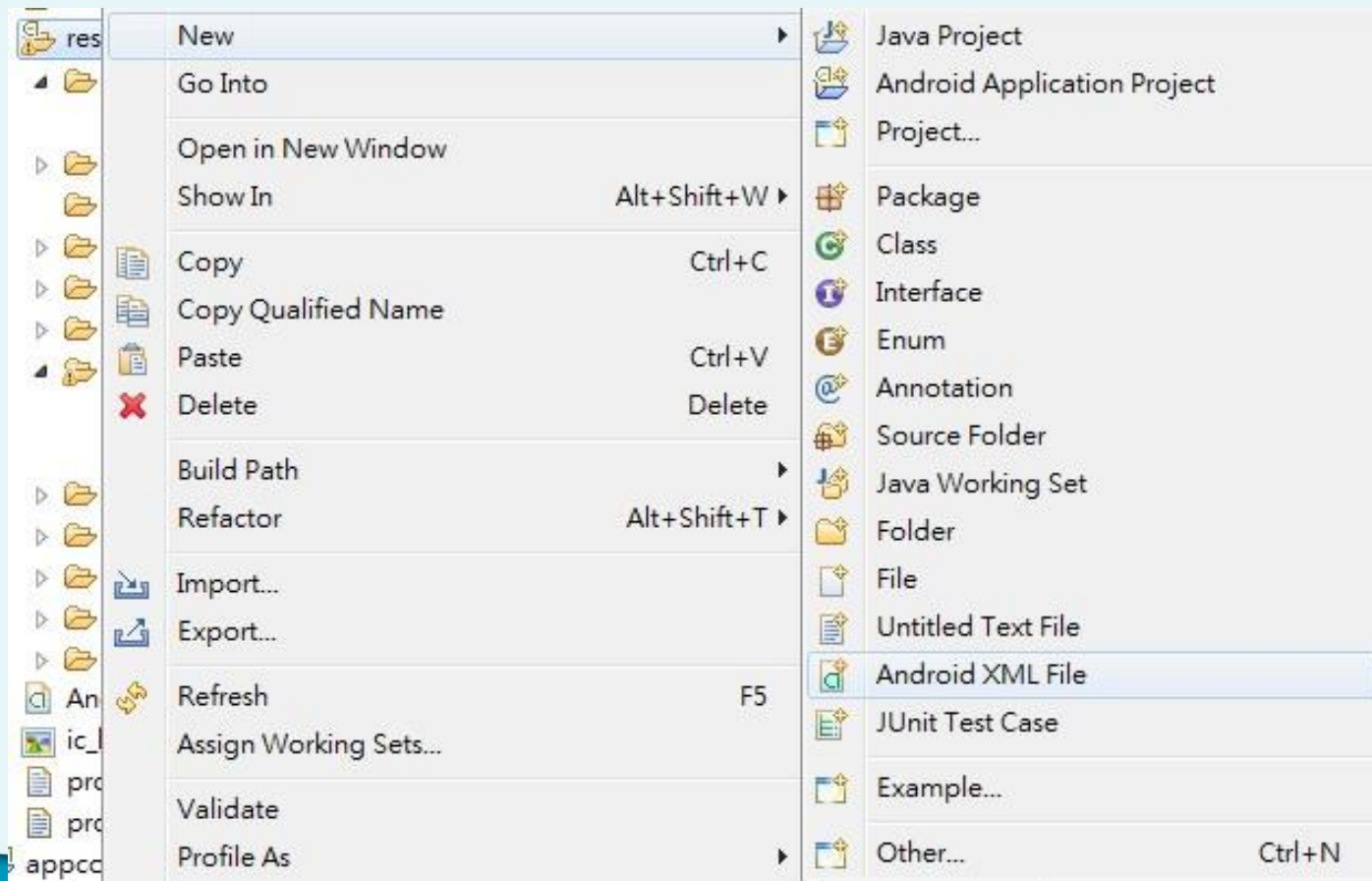


activity_main.xml

- 步驟1. 設定Layout成RelativeLayout。
- 步驟2. 拉六個ImageView，並將圖片設定成back。

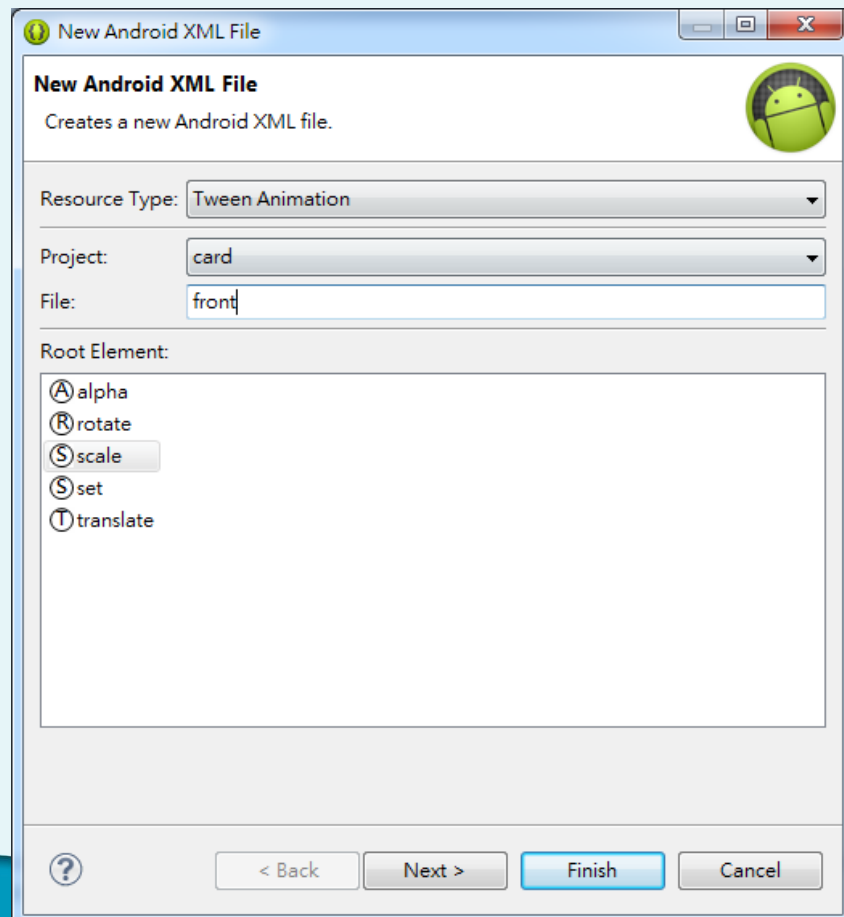
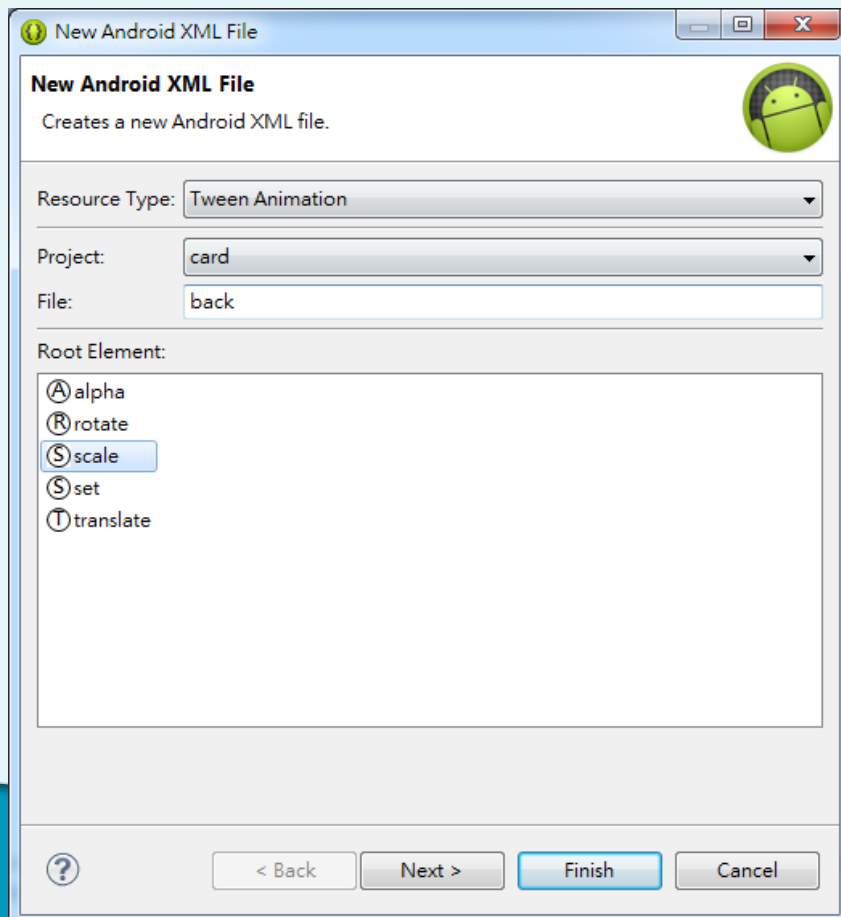
製作xml動畫格式

- 步驟1. 在res的資料夾，點擊滑鼠右鍵
New→Android XML File。



製作xml動畫格式

- 步驟2. 以下圖的方式輸入，Resource Type選擇 Tween Animation，File輸入back/front，Root Element選擇scale，最後Finish。





設計方法

- 透過back.xml將圖片從背面翻到正面，當快方到正面顯示時，會更改圖片的樣子，再透過front.xml將圖片繼續翻轉，並呈現出來。

back.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android" >

    <scale
        android:fromXScale="1.0"
        android:fromYScale="1.0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:toXScale="0.0"
        android:toYScale="1.0"
        android:duration="150" />

</set>
```

延遲時間。

初始狀態，1.0代表維持原本尺寸，這邊也同時代表顯示牌的狀態。大於1.0表示放大；小於1.0表示縮小。

翻轉的位置，我們透過相對座標指定中心點。

結束狀態，X表示0.0不顯示X的圖片，不過顯示Y的畫面，所以會有一條線的感覺。

front.xml

- 同理將排從一條線到顯示整張牌。

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android" >

    <scale
        android:fromXScale="0.0"
        android:fromYScale="1.0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:toXScale="1.0"
        android:toYScale="1.0"
        android:duration="150"/>

</set>
```

開始打程式 MainActivity.java

- 步驟1. 宣告變數。

```
ImageView iv1, iv2, iv3, iv4, iv5, iv6;  
int[] card = { R.drawable.front1, R.drawable.front1, R.drawable.front2,  
                R.drawable.front2, R.drawable.front3, R.drawable.front3 };  
boolean[] bool = { false, false, false, false, false, false };  
Timer timer = new Timer();  
int flop1, flop2, btncount = 0, time = 0, end = 0;  
ImageView fiv1, fiv2;  
MediaPlayer mp = new MediaPlayer();
```

- 步驟2. 建立元件與程式的關係。

```
iv1 = (ImageView) findViewById(R.id.imageView1);  
iv2 = (ImageView) findViewById(R.id.imageView2);  
iv3 = (ImageView) findViewById(R.id.imageView3);  
iv4 = (ImageView) findViewById(R.id.imageView4);  
iv5 = (ImageView) findViewById(R.id.imageView5);  
iv6 = (ImageView) findViewById(R.id.imageView6);
```

開始打程式 MainActivity.java

- 步驟3. 進行洗牌，將牌的位置亂數排列。

```
for (int i = 0; i < 10000; i++) {  
    int shuffle = (int) (Math.random() * 6);  
    int c = card[shuffle];  
    card[shuffle] = card[0];  
    card[0] = c;  
}
```

- 步驟4. 將六個ImageView設定按下的監聽事件。

```
iv1.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
  
    }  
});
```

- 步驟5. 因為會使用很多次播動畫的效果，所以以附程式方式設計，我們先取得到ImageView的值和它要翻轉的數值(iv1代表它是0號...).

```
private void animation(final ImageView img, final int ivValue) {  
  
}
```

- 步驟6. 在此附程式使用Tween動畫功能，並將牌從背面翻到一條線。

```
Animation animation = AnimationUtils.loadAnimation(MainActivity.this,  
    R.anim.back);
```

- 步驟7. 啟動動畫。

```
img.startAnimation(animation);
```

開始打程式 MainActivity.java

- 步驟8. 再啟動動畫的上方加入 `setAnimationListener` 監聽來取得開始到結束的資料。

```
animation.setAnimationListener(new AnimationListener() {  
  
    @Override  
    public void onAnimationStart(Animation arg0) {  
        // TODO Auto-generated method stub  
    }  
  
    @Override  
    public void onAnimationRepeat(Animation arg0) {  
        // TODO Auto-generated method stub  
    }  
  
    @Override  
    public void onAnimationEnd(Animation arg0) {  
        // TODO Auto-generated method stub  
    }  
});
```

↑
播放開始時。

↑
重播時。

↑
停止時。

開始打程式 MainActivity.java

- 步驟9. 在停止播放back.xml的時候將圖片轉變成正面的圖片，在執行front.xml讓牌呈現，並將布林值改成true為翻轉過的狀態，以方便確認第二張牌是否與第一張牌相同。

```
if (!bool[ivValue]) {  
    img.setImageResource(card[ivValue]);  
    bool[ivValue] = true;  
}  
img.startAnimation(AnimationUtils.LoadAnimation(  
    MainActivity.this, R.anim.front));
```

- 步驟10. 回到各ImageView將附程式需要的值附上去，就完成翻牌的動作。

```
animation(iv1, 0);
```

完整ImageView按鈕化程式碼

```
iv1.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        animation(iv1, 0);  
    }  
});
```

開始打程式 MainActivity.java

- 步驟11. 接下來回到onAnimationEnd，製作判別第一張和第二張是否相同，首先先記錄第一張的資料，並讓第一張牌不能再按下，以免會再進行一次翻牌動作。

```
btncount++;  
if (btncount == 1) {  
    fiv1 = img;  
    flop1 = ivValue;  
    img.setEnabled(false);  
}
```


開始打程式 MainActivity.java

- 步驟12. 再取得第二張的資料，並同時把全部的ImageView關閉點擊。

```
else if (btncount == 2) {  
    fiv2 = img;  
    flop2 = ivValue;  
    iv1.setEnabled(false);  
    iv2.setEnabled(false);  
    iv3.setEnabled(false);  
    iv4.setEnabled(false);  
    iv5.setEnabled(false);  
    iv6.setEnabled(false);  
}
```

開始打程式 MainActivity.java

- 步驟13. 在建立Timer方式前先建立Handler。

```
private Handler handler = new Handler() {  
  
    @Override  
    public void handleMessage(Message msg) {  
        // TODO Auto-generated method stub  
        super.handleMessage(msg);  
    }  
  
};
```

開始打程式 MainActivity.java

- 步驟14. 回到onCreat()建立Timer，並以200毫秒執行一次程式。

```
timer.schedule(new TimerTask() {  
  
    @Override  
    public void run() {  
        // TODO Auto-generated method stub  
        Message message = new Message();  
        message.what = 1;  
        handler.sendMessage(message);  
    }  
}, 0, 200);
```

- 步驟15. 建立好Timer後，再回到Handler，顯示遊戲時間。

因為是200毫秒，所以要除5才會等於1秒。

```
if (msg.what == 1) {  
    setTitle("遊戲時間：" + (time++) / 5 + " s");  
}
```

開始打程式 MainActivity.java

- 步驟16. 在關牌前，要顯示一段時間，我們隔200毫秒給予觀看，再判定是否為正確的牌。

```
if (btncount == 3) {  
    if (card[flop1] == card[flop2]) {  
        fiv1.setVisibility(View.INVISIBLE);  
        fiv2.setVisibility(View.INVISIBLE);  
    } else {  
        fiv1.setEnabled(true);  
        fiv1.setImageResource(R.drawable.back);  
        fiv2.setImageResource(R.drawable.back);  
        bool[flop1] = false;  
        bool[flop2] = false;  
    }  
    btncount = 0;  
    iv1.setEnabled(true);  
    iv2.setEnabled(true);  
    iv3.setEnabled(true);  
    iv4.setEnabled(true);  
    iv5.setEnabled(true);  
    iv6.setEnabled(true);  
} else if (btncount >= 2)  
    btncount++;
```

開始打程式 MainActivity.java

- 步驟17. 在最後判定是否正確加入正確或錯誤的音效，以及如果都翻完後就顯示出遊戲的記錄。

```
if (card[flop1] == card[flop2]) {
    fiv1.setVisibility(View.INVISIBLE);
    fiv2.setVisibility(View.INVISIBLE);
    end++;
    mp = MediaPlayer
        .create(MainActivity.this, R.drawable.o);
    if (card.length / 2 == end) {
        timer.cancel();
        setTitle("遊戲結束!! 記錄：" + time / 5 + " s");
    }
} else {
    fiv1.setEnabled(true);
    fiv1.setImageResource(R.drawable.back);
    fiv2.setImageResource(R.drawable.back);
    bool[flop1] = false;
    bool[flop2] = false;
    mp = MediaPlayer
        .create(MainActivity.this, R.drawable.x);
}
mp.start();
```

完整animation程式碼

```
private void animation(final ImageView img, final int ivValue) {
    Animation animation = AnimationUtils.LoadAnimation(MainActivity.this,
        R.anim.back);
    animation.setAnimationListener(new AnimationListener() {
        @Override
        public void onAnimationStart(Animation arg0) {
            // TODO Auto-generated method stub
        }

        @Override
        public void onAnimationRepeat(Animation arg0) {
            // TODO Auto-generated method stub
        }

        @Override
        public void onAnimationEnd(Animation arg0) {
            // TODO Auto-generated method stub
            if (!bool[ivValue]) {
                img.setImageResource(card[ivValue]);
                bool[ivValue] = true;
            }
            img.startAnimation(AnimationUtils.LoadAnimation(
                MainActivity.this, R.anim.front));

            btnccount++;
            if (btnccount == 1) {
                fiv1 = img;
                flop1 = ivValue;
                img.setEnabled(false);
            }
            else if (btnccount == 2) {
                fiv2 = img;
                flop2 = ivValue;
                iv1.setEnabled(false);
                iv2.setEnabled(false);
                iv3.setEnabled(false);
                iv4.setEnabled(false);
                iv5.setEnabled(false);
                iv6.setEnabled(false);
            }
        }
    });
    img.startAnimation(animation);
}
```



完整handler程式碼



```
@Override
public void handleMessage(Message msg) {
    // TODO Auto-generated method stub
    super.handleMessage(msg);
    if (msg.what == 1) {
        setTitle("遊戲時間：" + (time++) / 5 + " s ");

        if (btncount == 3) {
            if (card[flop1] == card[flop2]) {
                fiv1.setVisibility(View.INVISIBLE);
                fiv2.setVisibility(View.INVISIBLE);
                end++;
                mp = MediaPlayer
                    .create(MainActivity.this, R.drawable.o);
                if (card.length / 2 == end) {
                    timer.cancel();
                    setTitle("遊戲結束!! 記錄：" + time / 5 + " s ");
                }
            } else {
                fiv1.setEnabled(true);
                fiv1.setImageResource(R.drawable.back);
                fiv2.setImageResource(R.drawable.back);
                bool[flop1] = false;
                bool[flop2] = false;
                mp = MediaPlayer
                    .create(MainActivity.this, R.drawable.x);
            }
            mp.start();
            btncount = 0;
            iv1.setEnabled(true);
            iv2.setEnabled(true);
            iv3.setEnabled(true);
            iv4.setEnabled(true);
            iv5.setEnabled(true);
            iv6.setEnabled(true);
        } else if (btncount >= 2)
            btncount++;
    }
}

};
```