



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

- 影片播放器
- 如何設計Android UI 滿足各不同的螢幕尺寸

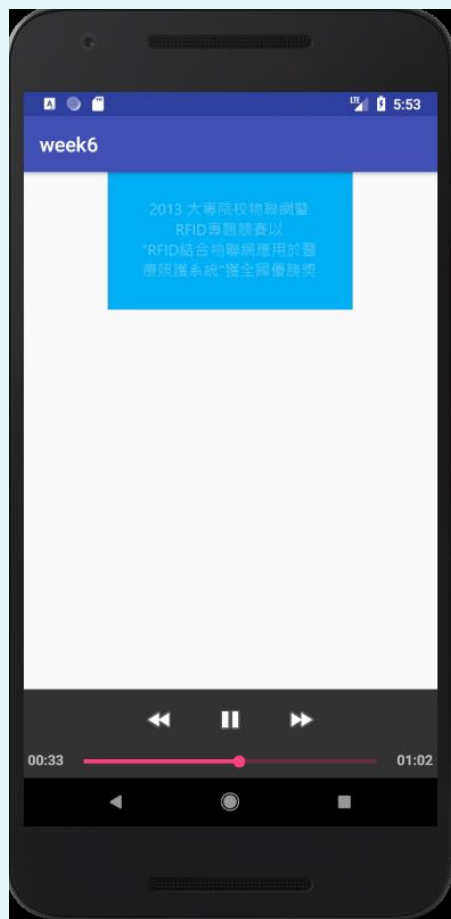


國立宜蘭大學

National Ilan University NIU

# 影片播放器

# UI介面

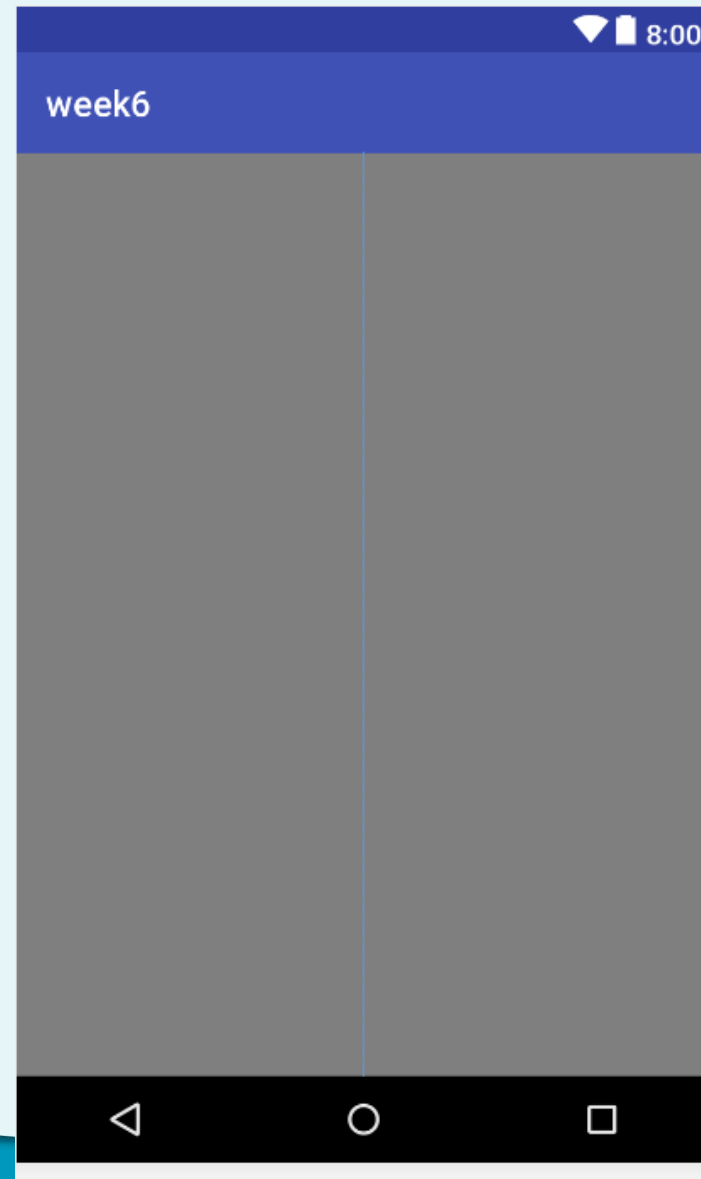
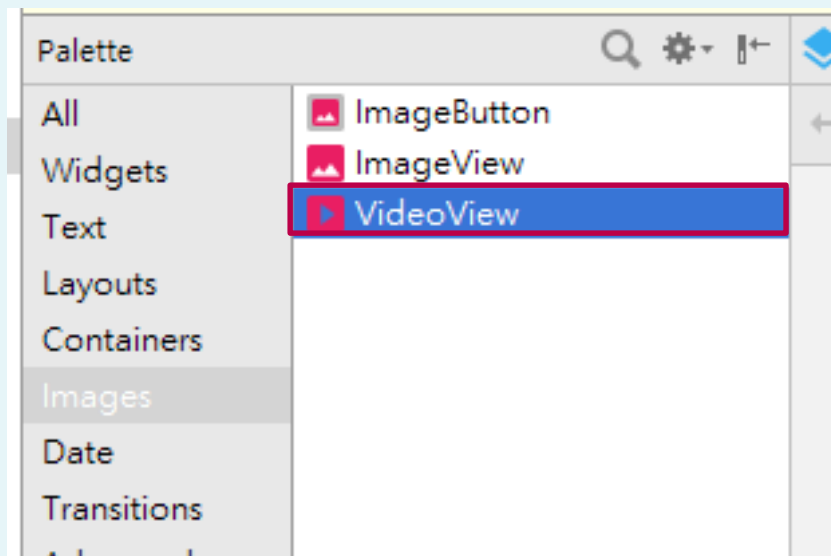


國立宜蘭大學

National Ilan University NIU

# activity\_main.xml

- 拉一個VideoView。



# 開始打程式 MainActivity.java

- 步驟1. 宣告VideoView& MediaController物件。

```
VideoView vidView;  
MediaController vidControl;
```

- 步驟2. 設定好影片路徑

```
String vidAddress = "http://rfid.niu.edu.tw/web/cch_converted.mp4";
```

- 步驟3. 建立程式與元件的關係(基本功)。

```
vidView = (VideoView) findViewById(R.id.videoView1);
```

- 步驟4. 設定先前宣告好的MediaController物件

```
vidControl = new MediaController(this);  
vidControl.setAnchorView(vidView);  
vidView.setMediaController(vidControl);
```

# 開始打程式 MainActivity.java

- 步驟5. 將先前宣告好的字串路徑轉成URI (Universal Resource Identifier)。

```
Uri vidUri = Uri.parse(vidAddress);
```

- 步驟6. 將VideoView的URI設置好，並開始。

```
vidView.setVideoURI(vidUri);  
vidView.start();
```

# 完整程式碼

```
VideoView vidView;  
MediaController vidControl;  
String vidAddress = "http://rfid.niu.edu.tw/web/cch_converted.mp4";  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    vidView = (VideoView) findViewById(R.id.videoView1);  
  
    vidControl = new MediaController(this);  
    vidControl.setAnchorView(vidView);  
    vidView.setMediaController(vidControl);  
  
    Uri vidUri = Uri.parse(vidAddress);  
  
    vidView.setVideoURI(vidUri);  
    vidView.start();  
}
```



# 加入網路權限

**Step:**進入Manifest --->輸入

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

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="week5_3.niu.edu.tw.week5_3">  
  
    <uses-permission android:name="android.permission.INTERNET" />  
  
    <application  
        android:allowBackup="true"  
        android:icon="@mipmap/ic_launcher"  
        android:label="week5_3"
```

# 試著將**apk**載入自己手機看看

- 你會遇到一個問題!?
- 為什麼我一旋轉，影片就會停止，並重新開始?

# 因為

- 當螢幕旋轉，原先方向的Activity會先被結束掉，之後會重新產生一個旋轉後的Activity，因此影片才會停止重開始。
- 生命週期:  
onPause → onSaveInstanceState → onStop →  
onDestroy → onCreate → onStart →  
onRestoreInstanceState → onResume

# 那該如何解決呢？

- 方法一:限制螢幕方向。
- 方法二:旋轉後，不刷新原先的Activity

# 方法一、限制螢幕方向

- 步驟1.進入Activity
- 步驟2.在 onCreate() 中，添加一條屬性資訊:
- 如果要強制直屏→輸入:

**setRequestedOrientation(ActivityInfo.SCREEN\_ORIENTATION\_PORTRAIT);**

```
setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
```

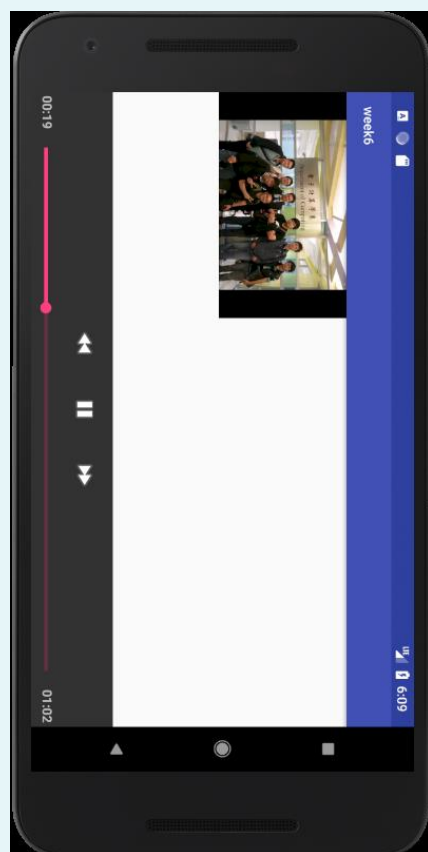
- 如果要強制橫屏→輸入:

**setRequestedOrientation(ActivityInfo.SCREEN\_ORIENTATION\_LANDSCAPE);**

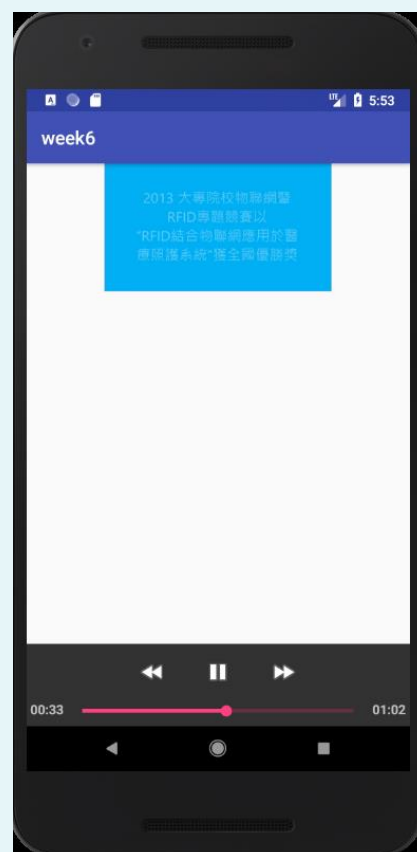
```
setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
```

# 畫面效果

- 強制橫向:



- 強制直向



# 方法二、不刷新原先的Activity

- 步驟1.進入AndroidManifest.xml
- 步驟2.在 <activity> </activity> 中，添加一條屬性資訊：

android:configChanges="orientation|keyboardHidden|screenSize"

```
<activity android:name=".MainActivity" android:configChanges="orientation|keyboardHidden|screenSize">
```

- **註：**這個屬性指的是，當後面屬性值代表的事件發生時，Activity 會執行某個函數，  
orientation 指的是螢幕旋轉時，  
keyboardHidden 指的是鍵盤協助工具改變  
screenSize指的是因旋轉螢幕大小改變

# 方法二、不刷新原先的Activity

- 步驟3.在對應 Activity 中重寫 onConfigurationChanged() 方法：

```
@Override
public void onConfigurationChanged(Configuration newConfig){
    super.onConfigurationChanged(newConfig);

    // Checks the orientation of the screen
    if(newConfig.orientation ==Configuration.ORIENTATION_LANDSCAPE){
        Toast.makeText(this,"landscape",Toast.LENGTH_SHORT).show();
    }else if(newConfig.orientation ==Configuration.ORIENTATION_PORTRAIT){
        Toast.makeText(this,"portrait",Toast.LENGTH_SHORT).show();
    }
}
```

- 註:此方法為當螢幕旋轉時會自動觸發的函式，因此當我們改寫後(override)，就能按照我們的方是觸發了！





國立宜蘭大學

National Ilan University NIU

# 如何設計Android UI 滿足各款手機

# 螢幕尺寸(Screen Size)

- 螢幕的物理大小，可以直接使用量尺工具來量測，一般會測量螢幕的對角線長度。
- Android 螢幕尺寸範圍：
  - small：3.3吋~3.5吋。
  - normal：3.5吋~5吋。
  - large：5吋~7.5吋。
  - xlarge：7.5吋以上。

# 螢幕密度(Screen Density)

- 指在一塊螢幕物理面積中的像素數量，但是通常不以面積而是以長度做計算，常用的單位是 **DPI**(Dots Per Inch)，也就是一吋的長度要使用幾個點才連接而成。
- 螢幕密度範圍：
  - ldpi大約為**120**dpi
  - mdpi大約為**160**dpi
  - hdpi大約為**240**dpi
  - xhdpi大約為**320**dpi

# 解析度(Resolution)

- 用來表示組成一塊平面所需要的像素(Pixel)的數量。例如解析度**1920x1200**，就表示這個平面的寬需要使用**1920個像素**來組成，高需要使用**1200個像素**來組成，因此要填滿這塊平面，必須使用 $1920 \times 1200 = 2304000$ 個像素。



# 螢幕方向(Orientation)

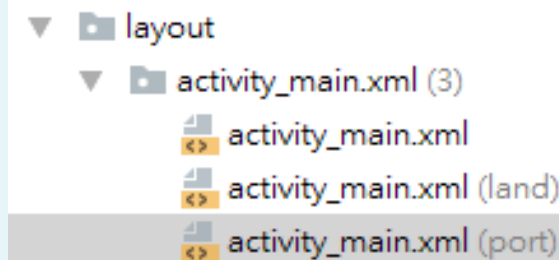
- 「橫的」螢幕畫面表示成「Landscape」。
- 「直的」螢幕畫面則表示成「Portrait」。

# Android App如何支援多種螢幕？

- 方法一、
  - 替不同螢幕尺寸和不同螢幕方向指定不同的Layout
- 方法二、
  - 替不同螢幕密度指定不同尺寸的Drawable圖片

# 方法一:指定不同的Layout

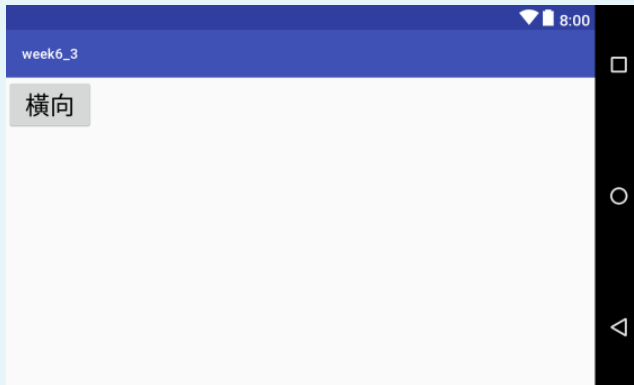
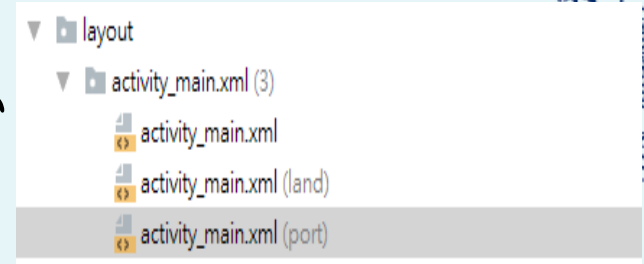
- 步驟1. 將原本layout複製到res下並命名為**layout-land** & **layout-port**  
會多出兩個activity\_main.xml



- 步驟2. 即可開始設定**橫向**與**縱向**的不同介面設計

# 方法一:指定不同的Layout

- 步驟3. 在layout-land & layout-port 的 activity\_main 各拉一個Button
- 註:此處故意加強兩個Button大小差，好讓清楚了解不同Layout能有完全不同的布局。





## 方法二:指定不同尺寸的**Drawable**圖片

- 圖片儲存在「res」目錄中的「drawable」目錄下:
  - drawable-ldpi
  - drawable-mdpi
  - drawable-hdpi
  - drawable-xhdpi

註:雖然只需要將圖片存入任一目錄下，Android 會自動縮放大小，但解析度方面，會因為縮放而失真，因此根據不同密度的手機，會取用的資源也不同，如此一來才能達到圖片解析度一樣大。

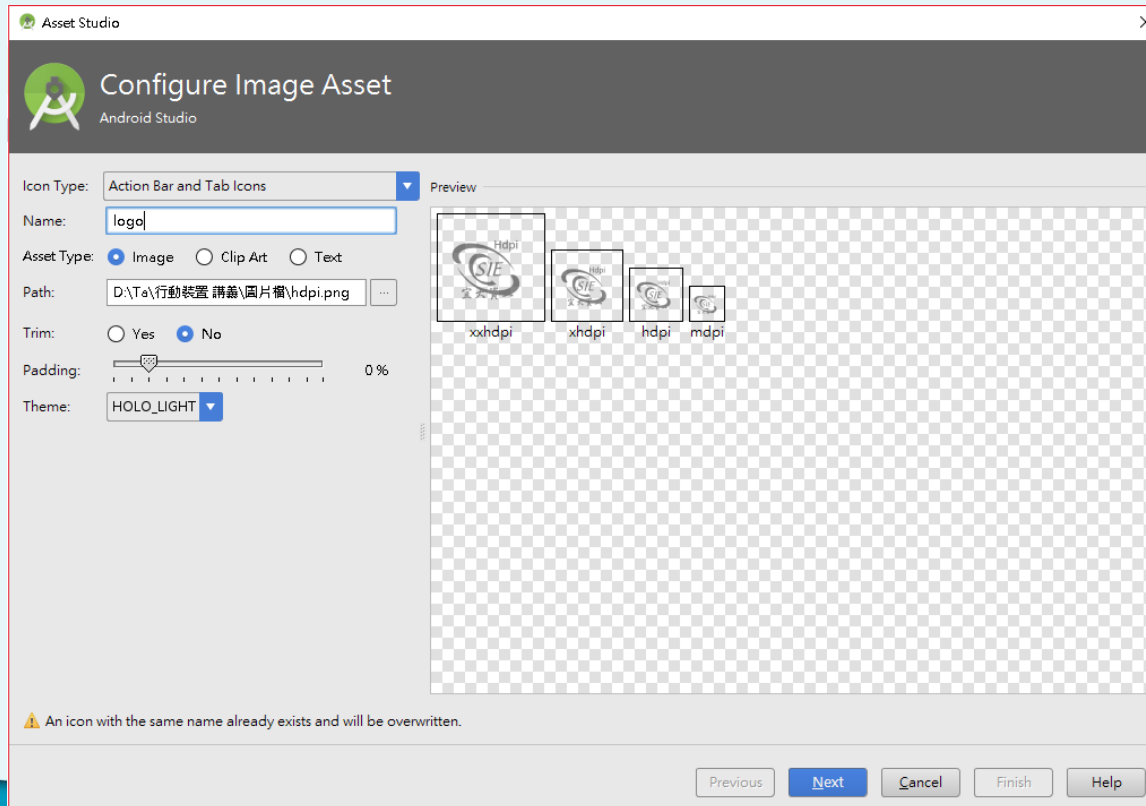
## 方法二:指定不同尺寸的**Drawable**圖片

- 由於解析度比較難以肉眼看出，此處Demo手機會根據**DPI**大小去選擇個目錄下的圖片。
- 以下兩張圖左為XHdpi，右為Ldpi。



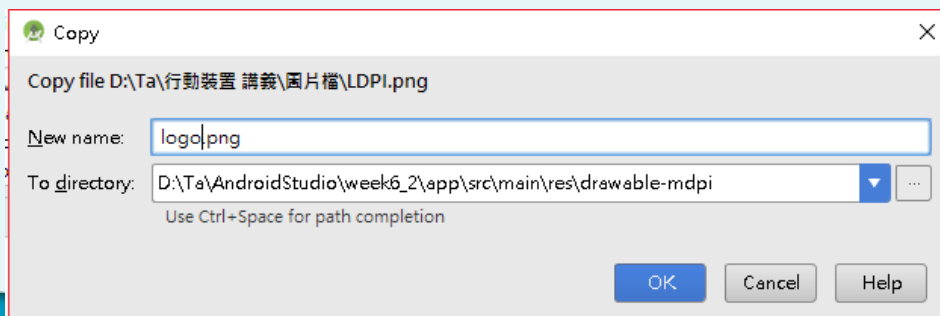
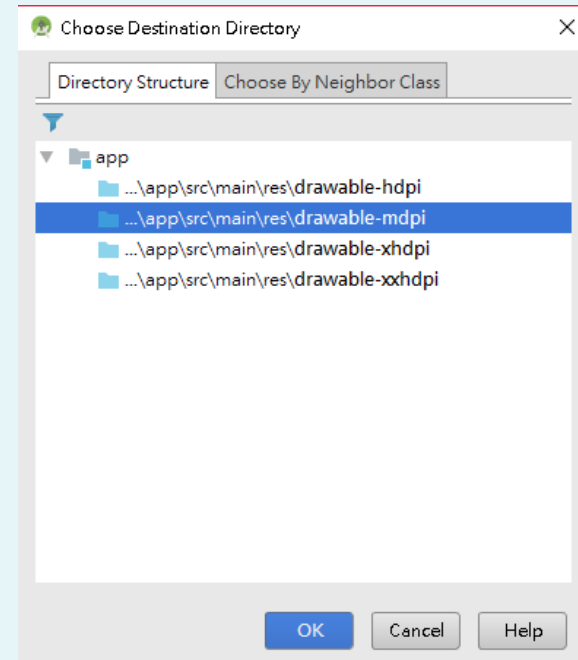
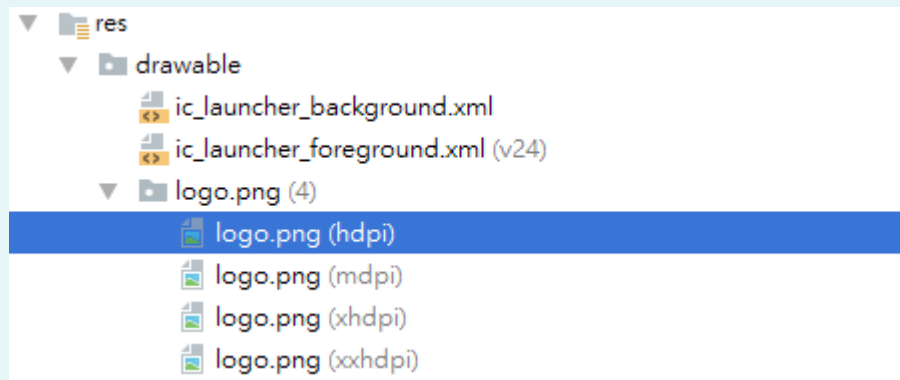
## 方法二:指定不同尺寸的**Drawable**圖片

- 步驟1 . Res下的Drawable 右鍵 new -> Image Asset  
Icon type 選 Action Bar and Tab Icons  
Asset Type 選 Image , Path 選擇圖片



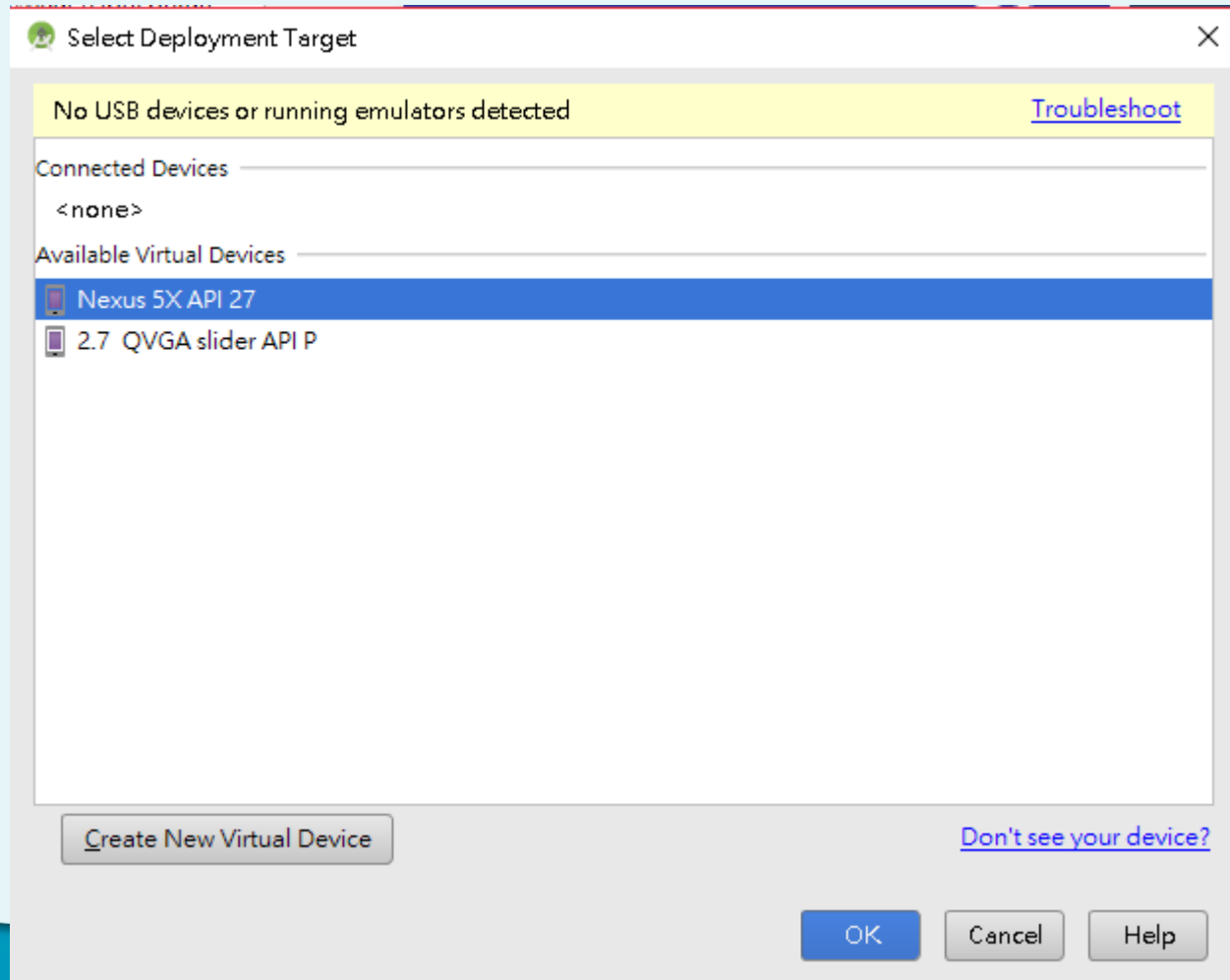
## 方法二:指定不同尺寸的**Drawable**圖片

- 步驟2. 在drawable得到一資料夾，分別將不同dpi的圖片複製到資料夾內覆蓋不同dpi的檔案，並重新命名為相同的檔名

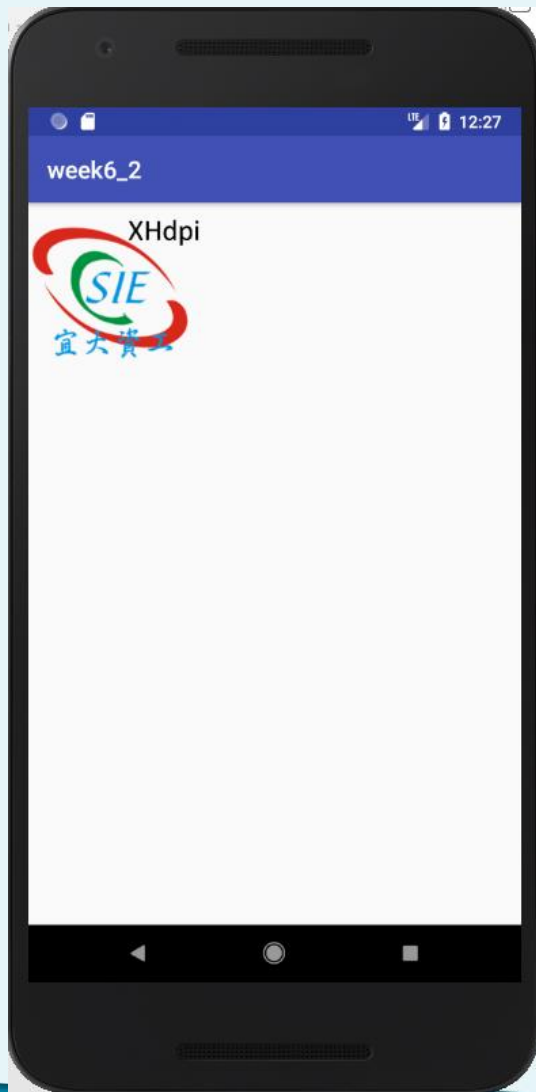


## 方法二:指定不同尺寸的**Drawable**圖片

- 步驟3. 開啟兩個不同DPI的模擬器。



# 完成



國立宜蘭大學

National Ilan University NIU

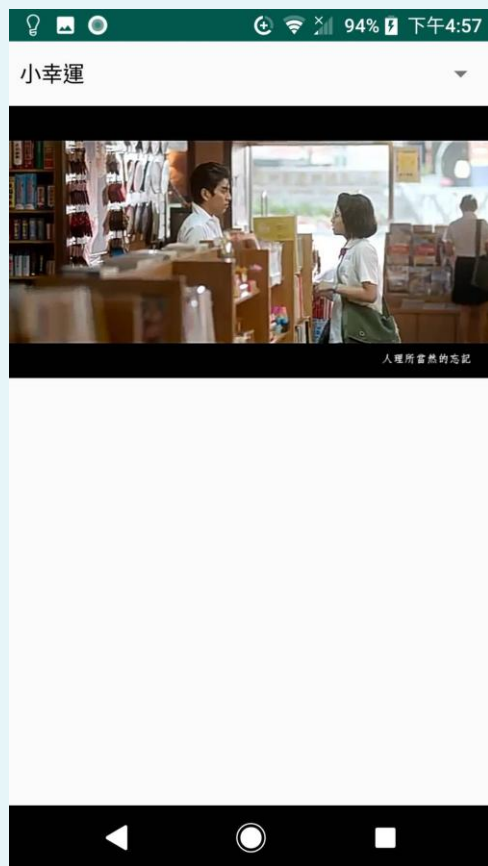


國立宜蘭大學

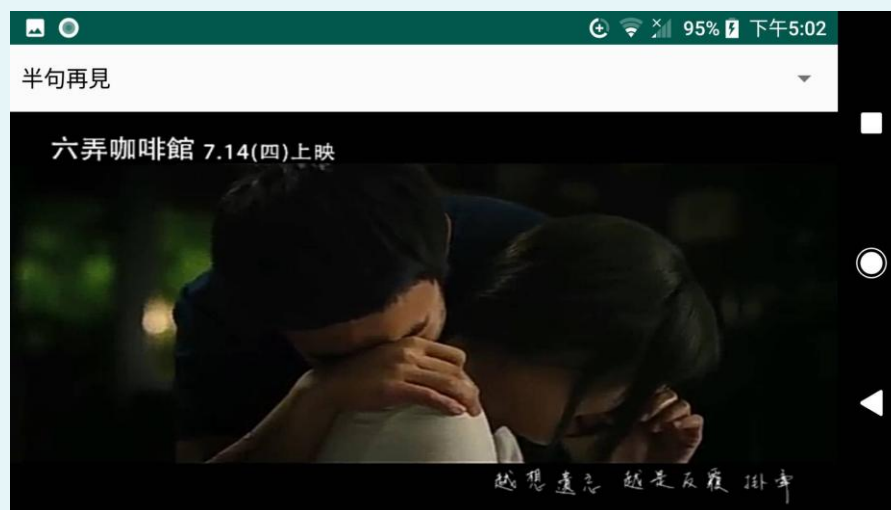
National Ilan University NIU

# 小專題- MV播放系統

# MV播放系統完成畫面



直式



橫式



# activity\_main.xml

- 步驟1. 加入元件。

元件名
Spinner
VideoView



# MainActivity.java

- 步驟1. 宣告元件與MV歌單內容。

```
Spinner mSpinner;  
VideoView mVideoView;  
String videoitem[]={"請選擇","小幸運","半句再見"};  
MediaController mMediaController;
```

# MainActivity.java

- 步驟2. 建立程式與元件的關係。

```
mVideoView=(VideoView) findViewById(R.id.videoView);  
mSpinner=(Spinner) findViewById(R.id.spinner);
```

- 步驟3. 建立ArrayAdapter給Spinner裝上

```
ArrayAdapter<String> adapter=new ArrayAdapter<String>  
    ( context: MainActivity.this, R.layout.support_simple_spinner_dropdown_item,videoitem);  
mSpinner.setAdapter(adapter);
```

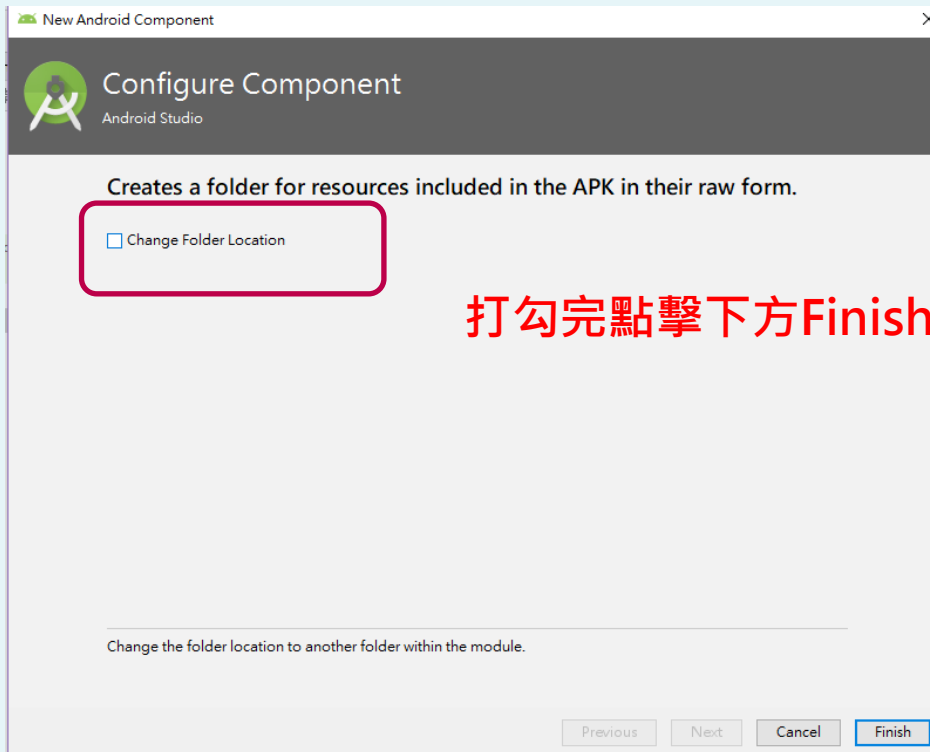
# MainActivity.java

- 步驟4.將Spinner建立一個監聽動作。

```
mSpinner.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
    }  
  
    @Override  
    public void onNothingSelected(AdapterView<?> adapterView) {  
    }  
});
```

# MainActivity.java

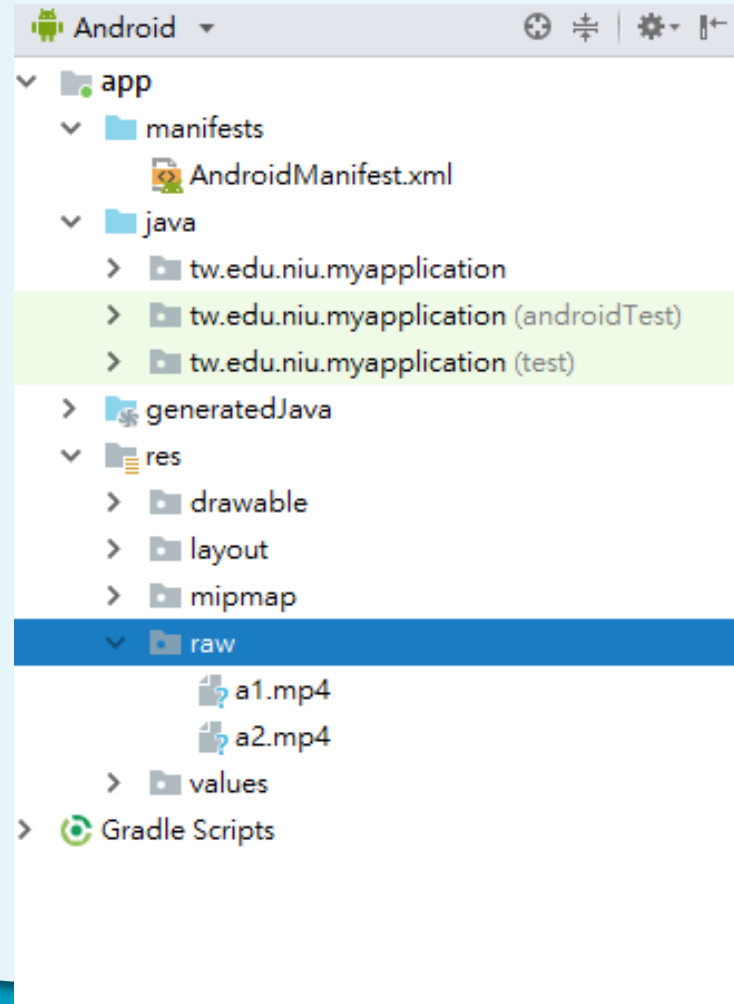
- 步驟5. 新增Raw資料夾，找到res資料夾右鍵選擇
- new → Folder → Raw Resource Folder



打勾完點擊下方Finish，即可建立Raw資料夾

# MainActivity.java

- 步驟6. 自行下載兩部MV，放入raw資料夾當中



# MainActivity.java

- 步驟7. Spinner依據選擇播放對應的影片

```
public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {  
    if(i!=0)  
    {  
        Toast.makeText( context: MainActivity.this, text: "正在播放"+videoitem[i],Toast.LENGTH_SHORT).show();  
    }  
    if(i==1)  
    {  
        mMediaController=new MediaController( context: MainActivity.this);  
        mMediaController.setAnchorView(mVideoView);  
        mVideoView.setMediaController(mMediaController);  
        Uri uri= Uri.parse("android.resource://" +getPackageName()+"/"+R.raw.a1);  
        mVideoView.setVideoURI(uri);  
        mVideoView.start();  
    }  
    else if(i==2)  
    {  
        mMediaController=new MediaController( context: MainActivity.this);  
        mMediaController.setAnchorView(mVideoView);  
        mVideoView.setMediaController(mMediaController);  
        Uri uri= Uri.parse("android.resource://" +getPackageName()+"/"+R.raw.a2);  
        mVideoView.setVideoURI(uri);  
        mVideoView.start();  
    }  
}
```

# 加入網路權限

- 步驟8. 進入**manifest**加入權限

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="week5_3.niu.edu.tw.week5_3">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="week5_3"
```