

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:

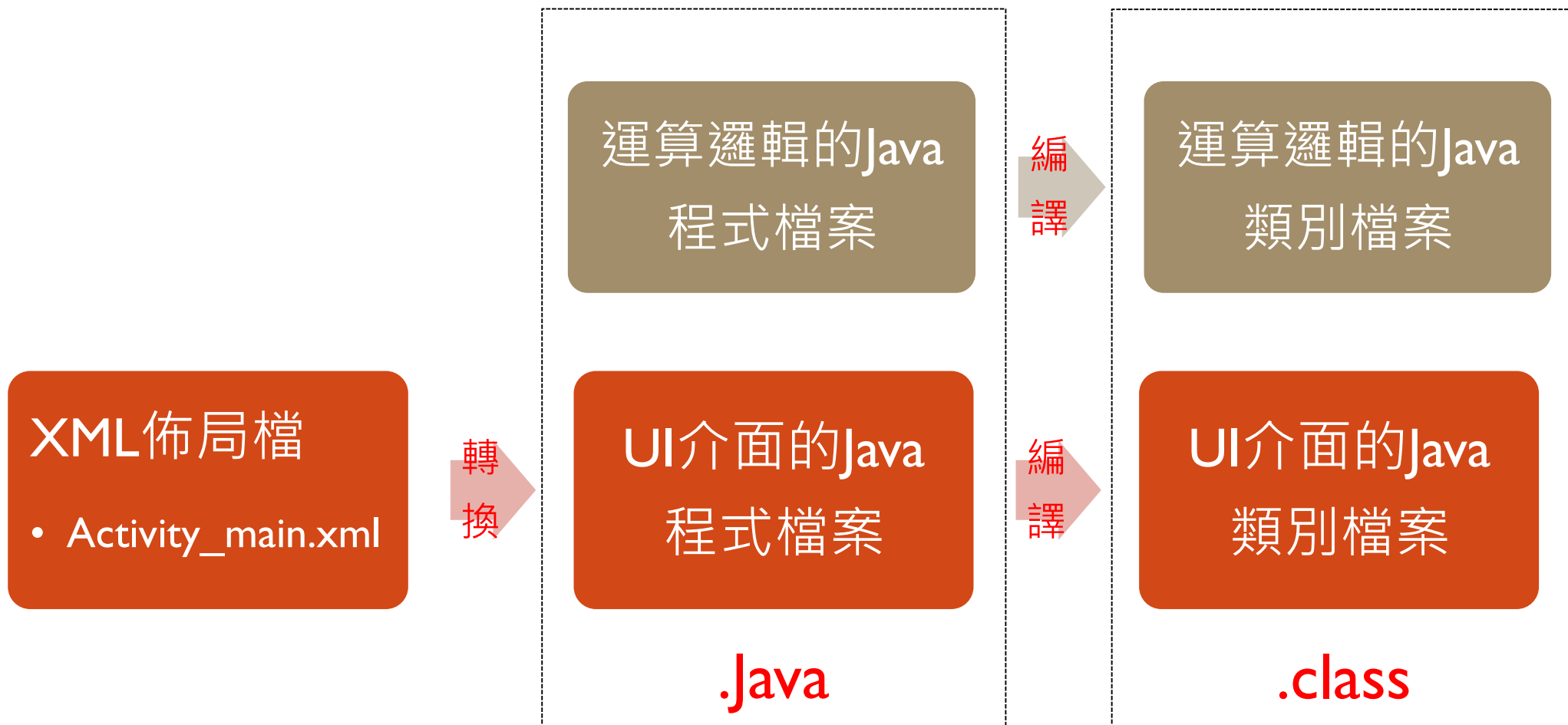
Android 介面元件

- TextView
- ImageView
- Button
- EditText
- Checkbox
- RadioButton



ANDROID 介面元件

- **Android** 應用程式的使用者介面:





ANDROID 介面元件

- **xmlns:**
 - 代表XML namespace 的縮寫
- **命名空間(namespace)**
 - 命名空間提供避免元素命名衝突的方法
 - 命名空間存放特定屬性的集合
- **實例:**
 - 甲班有個學生方大同，乙班也有個學生方大同，老師一叫方大同，兩人同時喊有!!
 - 甲班:方大同
 - 乙班:方大同



ANDROID 介面元件

- 命名空間：**Android**
- **xmlns:android="http://schemas.android.com/apk/res/android"**
 - 在**Android**佈局檔(Layout)中必須在根結點上定義
 - **android** 字串為 **namespace-prefix** 命名空間前綴字元
 - **"http://schemas.android.com/apk/res/android"**使用一個**URL**字串作為統一資源識別項(**URI Uniform Resource Identifier**)
 - 使用**URL(Uniform Resource Locators)**字串因為它的名字通常獨一無二，可以避免元素命名衝突
 - **android**命名空間內的屬性編譯後會作用在顯示元件上



ANDROID 介面元件

- 命名空間：**tools**
- **xmlns:tools="http://schemas.android.com/tools"**
- 在Android佈局檔(Layout)中必須在根結點上定義
 - **tools** 字串為 **namespace-prefix** 命名空間前綴字元
 - **"http://schemas.android.com/tools"** 使用一個**URL**字串作為統一資源識別項(**URI Uniform Resource Identifier**)
 - 提供開發工具顯示設計頁面用，幫助人員進行視覺開發，只作用於開發環境
 - **App**被編譯打包成**APK**時，**tools**命名空間下的屬性將會被捨棄



ANDROID 基礎介面元件

- Text View 顯示文字
- 開啟 activity_textview.xml ，並於其中加入以下程式碼
 - <TextView
 - android:layout_width="200dp"
 - android:layout_height="200dp"
 - android:text="Hello World 我的第一個 Android APP!"
 - android:textColor="#000000"
 - android:textSize="24sp"
 - app:layout_constraintBottom_toBottomOf="parent"
 - app:layout_constraintLeft_toLeftOf="parent"
 - app:layout_constraintRight_toRightOf="parent"
 - app:layout_constraintTop_toTopOf="parent"
 - app:layout_constraintHorizontal_bias="0.133"
 - app:layout_constraintVertical_bias="0.063" />



ANDROID 基礎介面元件

- Text View 顯示文字
 - `android:layout_width="200dp"`
 - 設定寬度
 - `android:layout_height="200dp"`
 - 設定高度
 - `android:text="Hello World 我的第一個 Android APP!"`
 - 設定文字內容
 - `android:textColor="#000000"`
 - 設定顏色
 - `android:textSize="24sp"`
 - 設定字型大小



ANDROID 基礎介面元件

- Text View 顯示文字
 - `app:layout_constraintBottom_toBottomOf="parent"`
 - `app:layout_constraintLeft_toLeftOf="parent"`
 - `app:layout_constraintRight_toRightOf="parent"`
 - `app:layout_constraintTop_toTopOf="parent"`
 - `app:layout_constraintHorizontal_bias="0.133"`
 - `app:layout_constraintVertical_bias="0.063"`
 - 設定位置座標



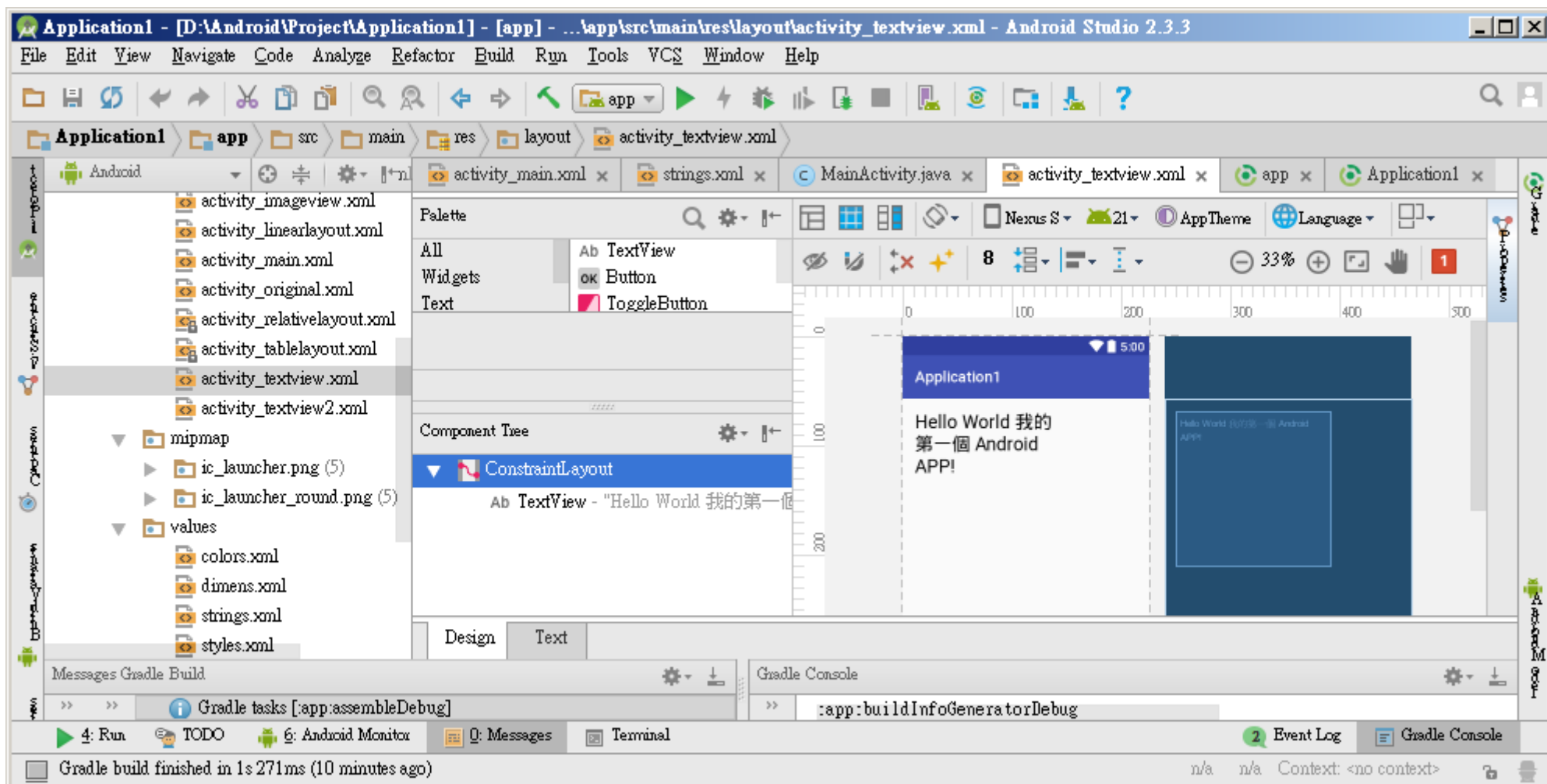
ANDROID 基礎介面元件

- Text View 顯示文字
 - `android:textStyle="bold"`
 - 文字粗體
 - `android:textStyle="italic"`
 - 文字斜體
 - `android:gravity="center_horizontal"`
 - 文字置中
 - `android:gravity="right"`
 - 文字靠右
 - `android:gravity="left"`
 - 文字靠左



ANDROID 基礎介面元件

- Text View 顯示文字-視覺化工具





The screenshot shows the Android Studio 2.3.3 interface. The top toolbar contains icons for file operations, navigation, and execution. The left sidebar displays the project structure for 'Application1', with 'activity_textview.xml' selected under the 'res/layout' directory. The main editor window shows the XML code for the TextView, which is currently in the 'Text' tab. The XML code defines a TextView with a width and height of 200dp, containing the text 'Hello World 我的第一个 Android APP!'. It also sets the text color to black, text size to 24sp, and various layout constraints. The bottom status bar shows the build progress, with a message indicating the build is finished in 1s 271ms (11 minutes ago).

Application1 - [D:\Android\Project\Application1] - [app] - ...app\src\main\res\layout\activity_textview.xml - Android Studio 2.3.3

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Application1 app src main res layout activity_textview.xml

Android activity_imageview.xml activity_linearlayout.xml activity_main.xml activity_original.xml activity_relativelayout.xml activity_tablelayout.xml activity_textview.xml activity_textview2.xml mipmap ic_launcher.png (5) ic_launcher_round.png (5) values colors.xml dimens.xml strings.xml styles.xml

9 <TextView
10 android:layout_width="200dp"
11 android:layout_height="200dp"
12 android:text="Hello World 我的第一个 Android APP!"
13 android:textColor="#000000"
14 android:textSize="24sp"
15 app:layout_constraintBottom_toBottomOf="parent"
16 app:layout_constraintLeft_toLeftOf="parent"
17 app:layout_constraintRight_toRightOf="parent"
18 app:layout_constraintTop_toTopOf="parent"
19 app:layout_constraintHorizontal_bias="0.133"
20 app:layout_constraintVertical_bias="0.063" />
21

Design Text

Messages Gradle Build

Gradle tasks [:app:assembleDebug]

4: Run 6: Android Monitor 0: Messages Terminal

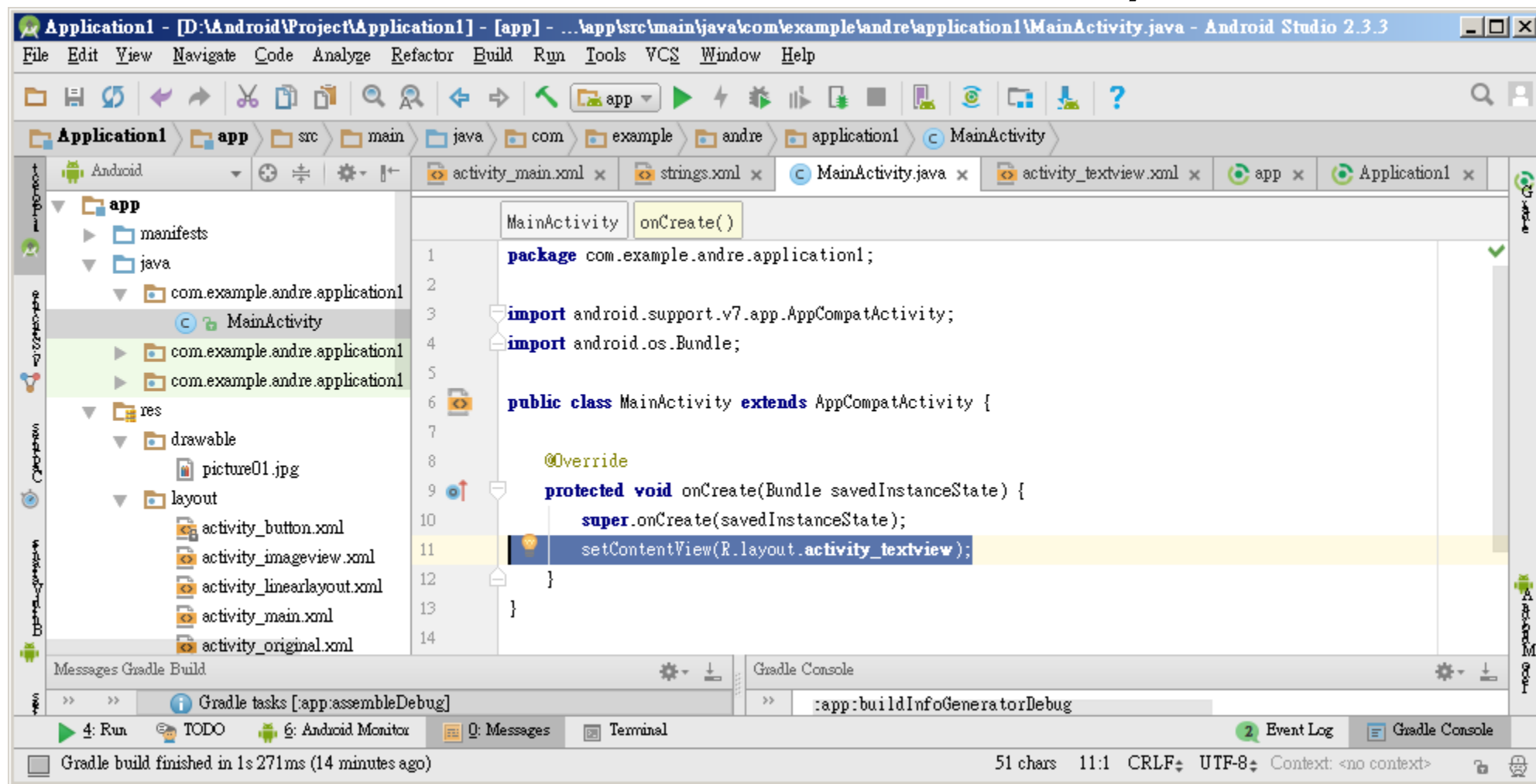
Gradle build finished in 1s 271ms (11 minutes ago)

1:1 CRLF UTF-8 Context: <no context>



ANDROID 基礎介面元件

- Text View 顯示文字-切換 MainActivity顯示





ANDROID 基礎介面元件

- Text View 顯示文字-切換 MainActivity顯示
- 開啟 MainActivity.java ，並於其中加入以下程式碼
 - package com.example.andre.application I;
 - import android.support.v7.app.AppCompatActivity;
 - import android.os.Bundle;
 - public class MainActivity extends AppCompatActivity {
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_textview);
 - }
 - }



ANDROID 基礎介面元件

- Text View 顯示文字-成果





ANDROID 基礎介面元件

- Image View 顯示影像
- 開啟 `activity_imageview.xml` ，並於其中加入以下程式碼
 - `<ImageView`
 - `xmlns:android="http://schemas.android.com/apk/res/android"`
 - `android:layout_width="fill_parent"`
 - `android:layout_height="wrap_content"`
 - `android:scaleType="center"`
 - `android:src="@drawable/picture01"`
 - `/>`



ANDROID 基礎介面元件

- Image View 顯示影像
 - android:scaleType 屬性: 設定顯示方式
 - CENTER 置中
 - 圖片較ImageView大時，只顯示圖片中央的部分，當圖片較ImageView小時，將圖片以置中的方式顯示
 - CENTER_CROP 按比例填滿
 - 當圖片大小不同於ImageView 時，依圖片等比例縮放並填滿，即大圖縮小，小圖放大
 - CENTER_INSIDE 縮小
 - 當圖片較ImageView大時，依圖片等比例縮小，將圖片以置中的方式顯示，圖片較ImageView小時，不縮放



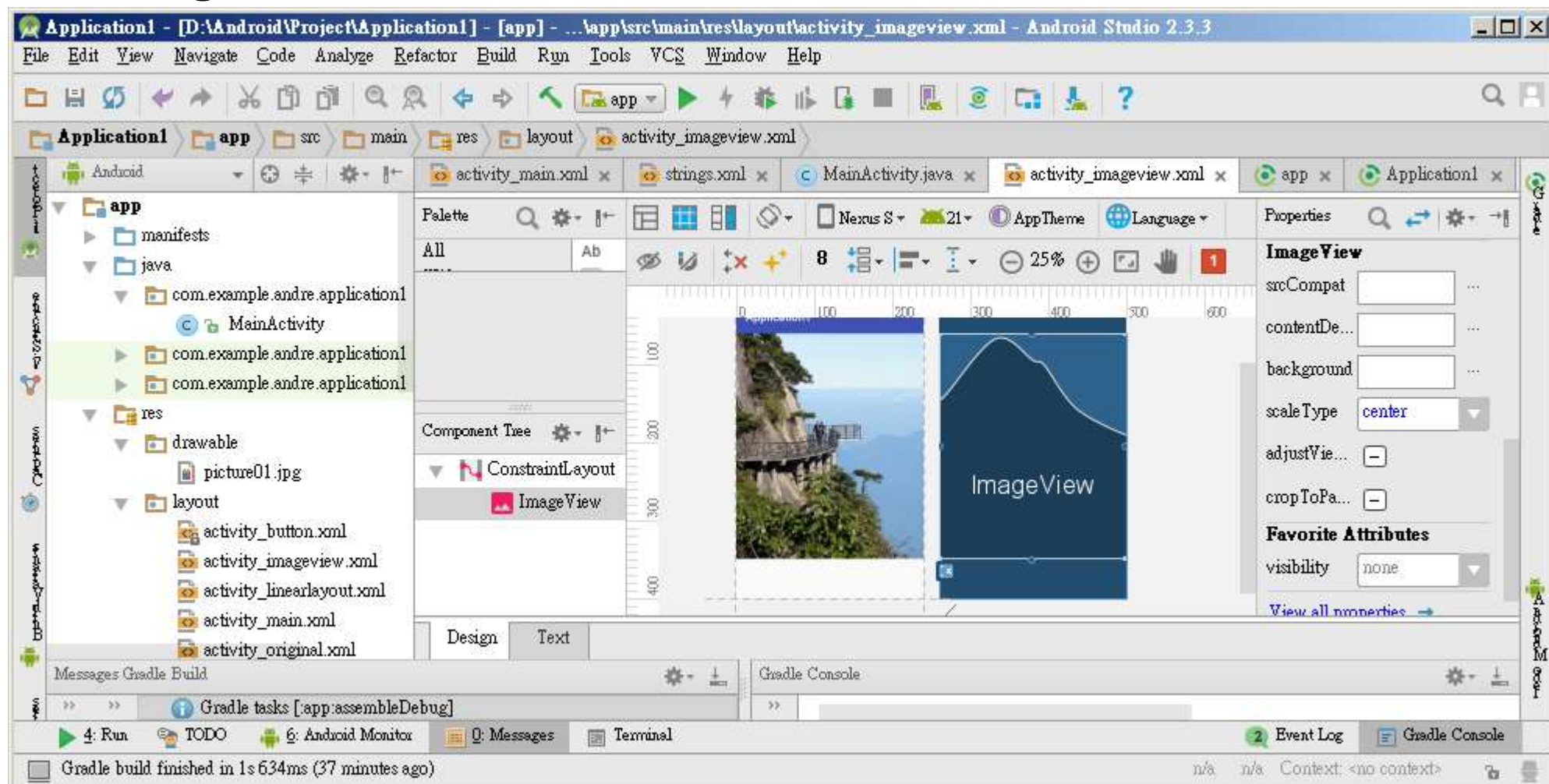
ANDROID 基礎介面元件

- Image View 顯示影像
 - android:scaleType 屬性: 設定顯示方式
 - FitStart 靠左顯示
 - 當圖片小於ImageView 時，右方留白，靠左顯示
 - FitEnd 靠右顯示
 - 當圖片小於ImageView 時，左方留白，靠右顯示
 - FIT_XY 不按比例填滿
 - 按照ImageView大小不按比例縮放，以填滿顯示區



ANDROID 基礎介面元件

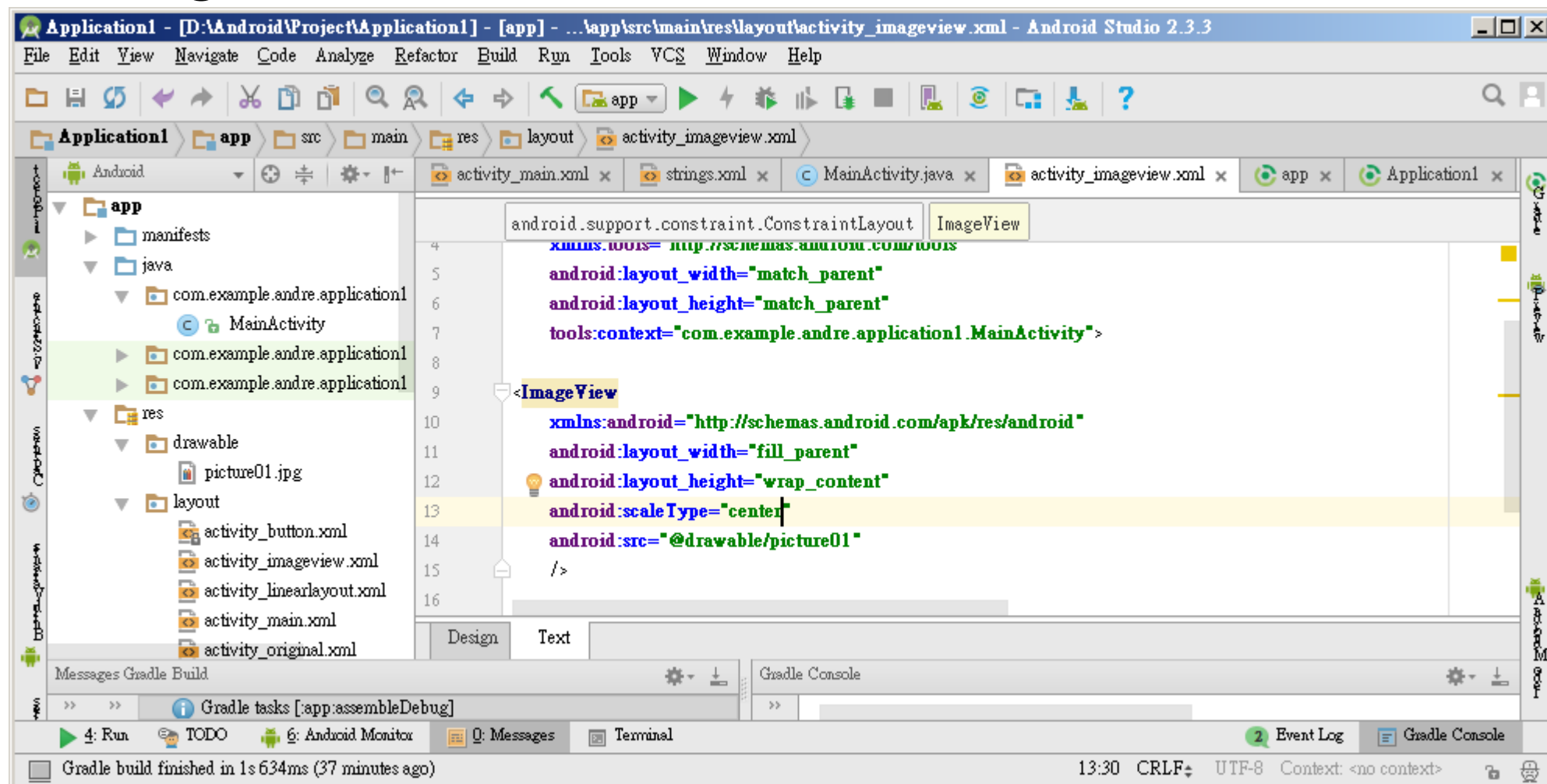
- Image View 顯示影像-視覺化工具





ANDROID 基礎介面元件

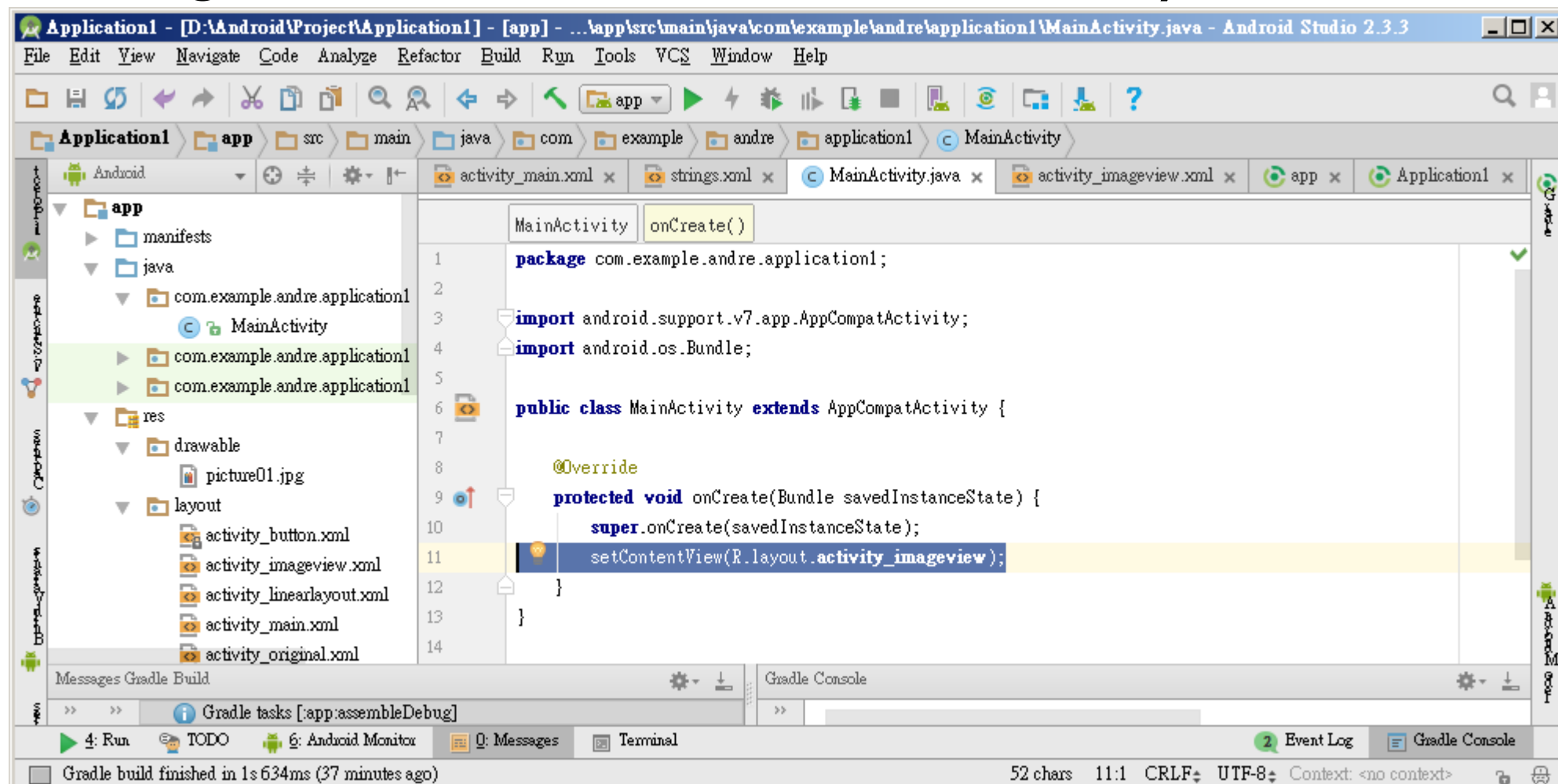
- Image View 顯示影像-文字編輯器





ANDROID 基礎介面元件

- Image View 顯示影像-切換 MainActivity顯示





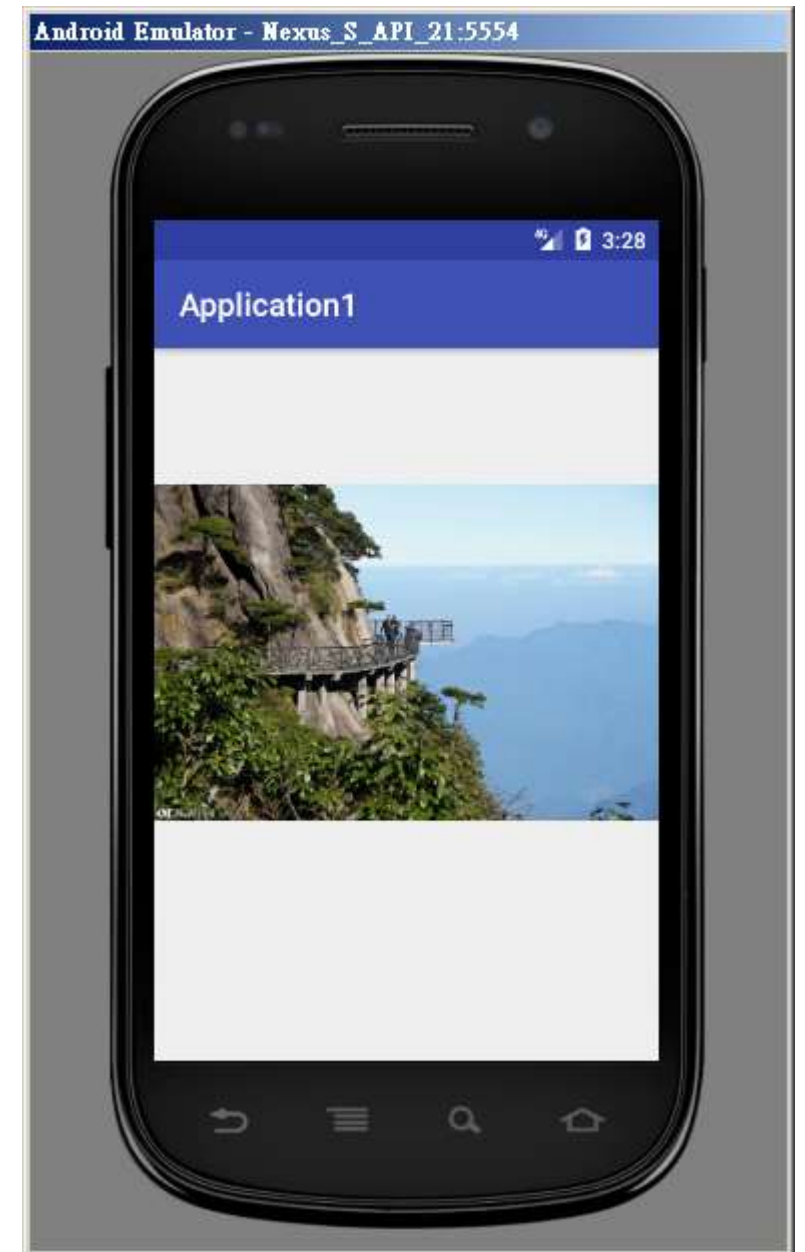
ANDROID 基礎介面元件

- Image View 顯示影像-切換 MainActivity顯示
- 開啟 MainActivity.java ，並於其中加入以下程式碼
 - package com.example.andre.application I;
 - import android.support.v7.app.AppCompatActivity;
 - import android.os.Bundle;
 - public class MainActivity extends AppCompatActivity {
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_ imageview);
 - }
 - }



ANDROID 基礎介面元件

- Image View 顯示影像-成果





ANDROID 基礎介面元件

- Button 按鈕元件
- 開啟 activity_button.xml ，並於其中加入以下程式碼
 - `<Button xmlns:android="http://schemas.android.com/apk/res/android"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:background="#0000FF"`
 - `android:text="我是按鈕請按我"`
 - `android:textColor="#000000"`
 - `app:layout_constraintBottom_toBottomOf="parent"`
 - `app:layout_constraintLeft_toLeftOf="parent"`
 - `app:layout_constraintRight_toRightOf="parent"`
 - `app:layout_constraintTop_toTopOf="parent"`
 - `app:layout_constraintHorizontal_bias="0.5"`
 - `app:layout_constraintVertical_bias="0.6" />`



ANDROID 基礎介面元件

- Button 按鈕元件
 - `android:layout_width="wrap_content"`
 - 設定寬度-符合文字寬
 - `android:layout_height="wrap_content"`
 - 設定高度-符合文字高
 - `android:text="我是按鈕請按我"`
 - 設定按鈕顯示文字
 - `android:textColor="#000000"`
 - 設定顯示文字顏色
 - `android:background="#0000FF"`
 - 設定按鈕背景色



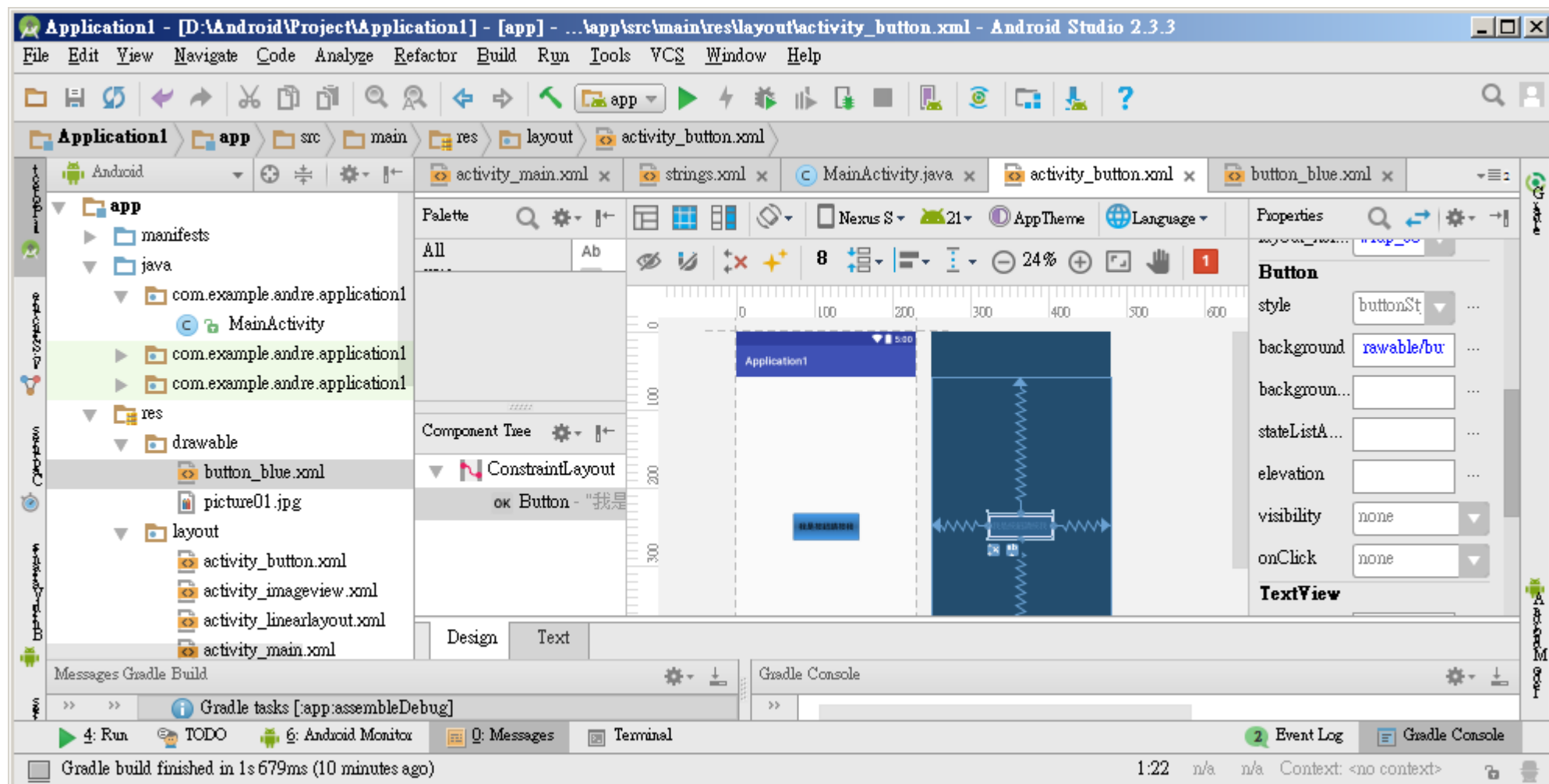
ANDROID 基礎介面元件

- Button 按鈕元件
 - `app:layout_constraintBottom_toBottomOf="parent"`
 - `app:layout_constraintLeft_toLeftOf="parent"`
 - `app:layout_constraintRight_toRightOf="parent"`
 - `app:layout_constraintTop_toTopOf="parent"`
 - `app:layout_constraintHorizontal_bias="0.5"`
 - `app:layout_constraintVertical_bias="0.6"`
 - 設定按鈕顯示位置



ANDROID 基礎介面元件

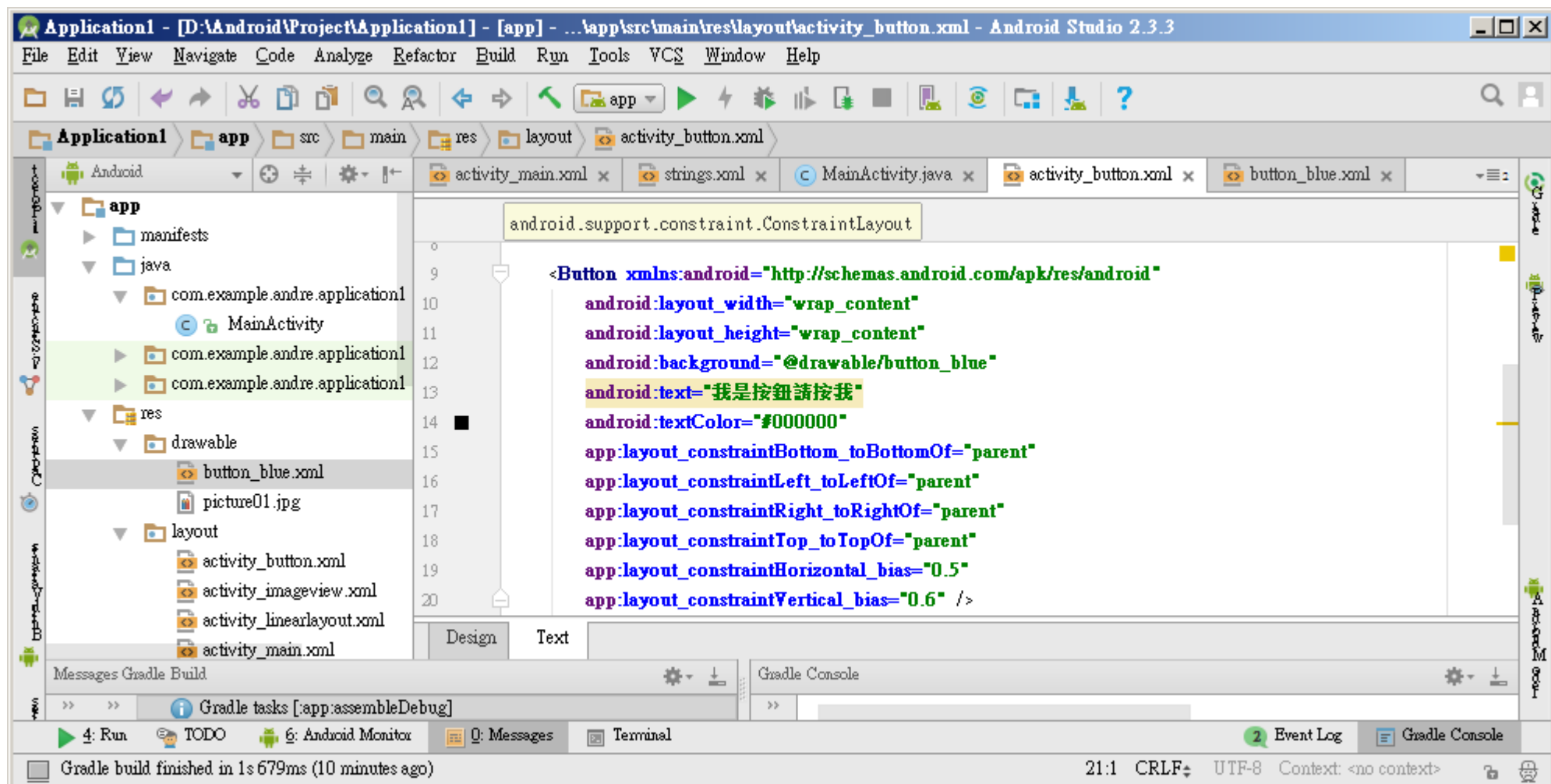
• Button 按鈕元件-視覺化工具





ANDROID 基礎介面元件

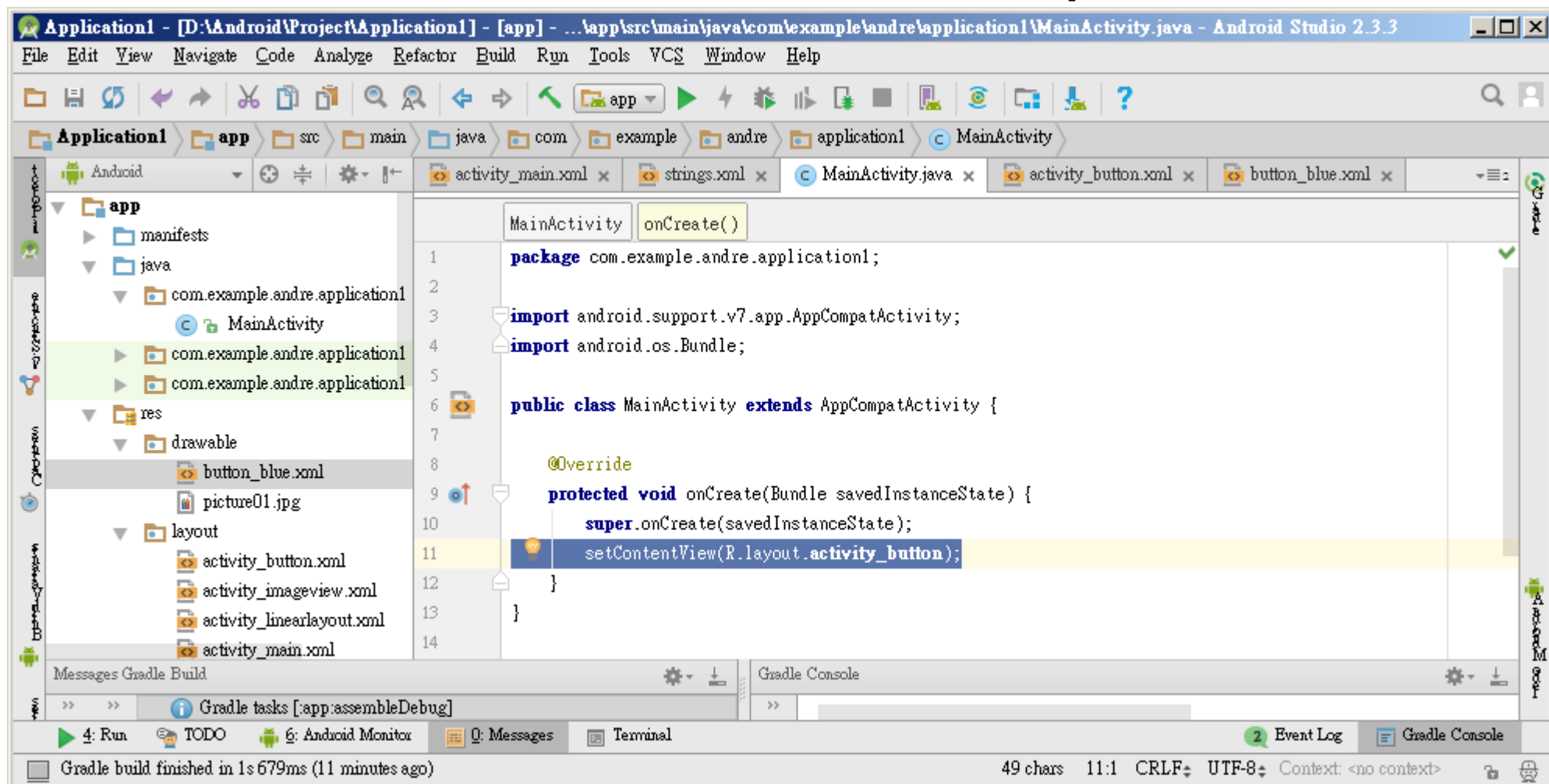
• Button 按鈕元件-文字編輯器





ANDROID 基礎介面元件

- Button 按鈕元件-切換 MainActivity顯示





ANDROID 基礎介面元件

- **Button** 按鈕元件-切換 MainActivity顯示
- 開啟 MainActivity.java ，並於其中加入以下程式碼
 - package com.example.andre.application I;
 - import android.support.v7.app.AppCompatActivity;
 - import android.os.Bundle;
 - public class MainActivity extends AppCompatActivity {
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity.Activity_button);
 - }
 - }



ANDROID 基礎介面元件

- Button 按鈕元件
- 新增 /drawable/button_blue.xml ，並於其中加入以下程式碼
 - `<selector xmlns:android="http://schemas.android.com/apk/res/android">`
 - `<item>`
 - `<shape>`
 - `<gradient`
 - `android:startColor="#449def"`
 - `android:centerColor="#2f6699"`
 - `android:endColor="#449def"`
 - `android:angle="270" />`
 - `<stroke`
 - `android:width="1 dp"`
 - `android:color="#2f6699" />`



ANDROID 基礎介面元件

- Button 按鈕元件
- 新增 /drawable/button_blue.xml - 續
 - `<corners`
 - `android:radius="2dp" />`
 - `<padding`
 - `android:left="10dp"`
 - `android:top="10dp"`
 - `android:right="10dp"`
 - `android:bottom="10dp" />`
 - `</shape>`
 - `</item>`



ANDROID 基礎介面元件

- Button 按鈕元件
- gradient 設定漸層色
 - startColor 起始色
 - centerColor 中間色
 - endColor 終止色
 - angle 顏色角度
 - angle=0 時，表起始色在左，終止色在右
 - angle=90，起始色在下，終止色在上
 - angle=180，起始色在右，終止色在左
 - angle=270，起始色在上，終止色在下
 - angle 為 45 的倍數



ANDROID 基礎介面元件

- Button 按鈕元件
- stroke 設定框線厚度
 - `android:width="1dp"`
 - 框線厚度 1dp
 - `android:color="#2f6699"`
 - 框線顏色 #2f6699
- corners 設定四角
 - `android:radius="2dp"`
 - 圓角 2dp



ANDROID 基礎介面元件

- Button 按鈕元件
- padding 設定內部文字邊界
 - `android:left="10dp"`
 - 左邊留 10dp
 - `android:top="10dp"`
 - 上方留 10dp
 - `android:right="10dp"`
 - 右邊留 10dp
 - `android:bottom="10dp"`
 - 下方留 10dp



ANDROID 基礎介面元件

- Button 按鈕元件 - 成果





ANDROID 基礎介面元件

- EditText 文字輸入元件
- 開啟 activity_edittext.xml ，並於其中加入以下程式碼
 - <EditText
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:hint="提示字樣"
 - android:maxLines="1"
 - android:textColor="#ff8c00"
 - android:textColorHighlight="#cccccc"
 - android:textColorHint="#238745"
 - android:textScaleX="1.5"
 - android:textStyle="bold"
 - android:typeface="monospace" />



ANDROID 基礎介面元件

- **EditText** 文字輸入元件
 - `android:hint="提示字樣"`
 - 設定提示文字
 - `android:maxLines="1"`
 - 設定行數
 - `android:textColorHint="#238745"`
 - 設定顯示提示文字顏色
 - `android:textColor="#000000"`
 - 設定顯示文字顏色
 - `android:textColorHighlight="#cccccc"`
 - 設定選取後文字顏色



ANDROID 基礎介面元件

- **EditText** 文字輸入元件
 - `android:textScaleX="1.5"`
 - 設定文字間距
 - `android:textStyle="bold"`
 - 設定字體
 - `android:typeface="monospace"`
 - 設定字型



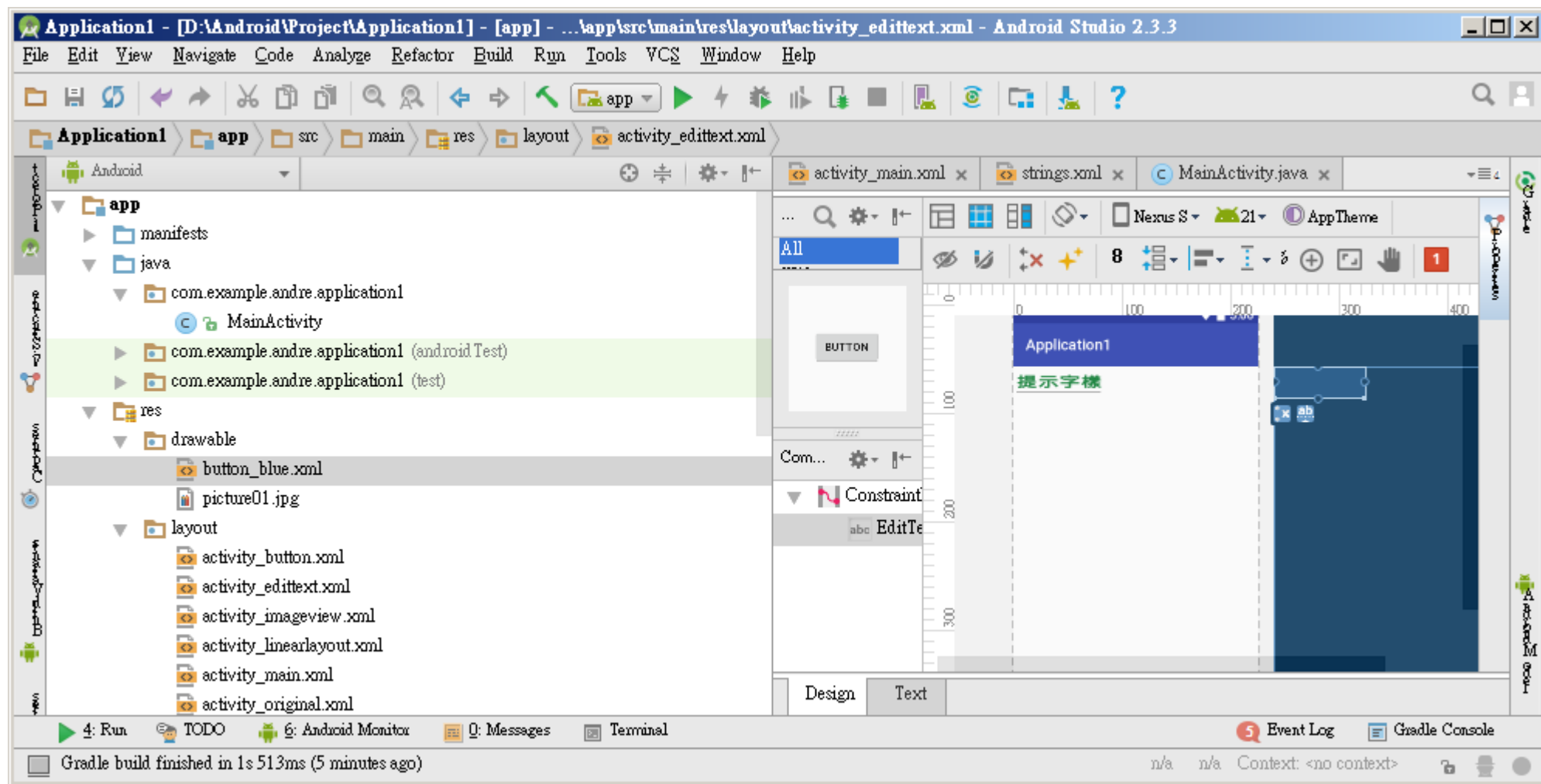
ANDROID 基礎介面元件

- **EditText** 文字輸入元件- 嘗試加入以下內容調整位置
 - `app:layout_constraintBottom_toBottomOf="parent"`
 - `app:layout_constraintLeft_toLeftOf="parent"`
 - `app:layout_constraintRight_toRightOf="parent"`
 - `app:layout_constraintTop_toTopOf="parent"`
 - `app:layout_constraintHorizontal_bias="0.5"`
 - `app:layout_constraintVertical_bias="0.6"`



ANDROID 基礎介面元件

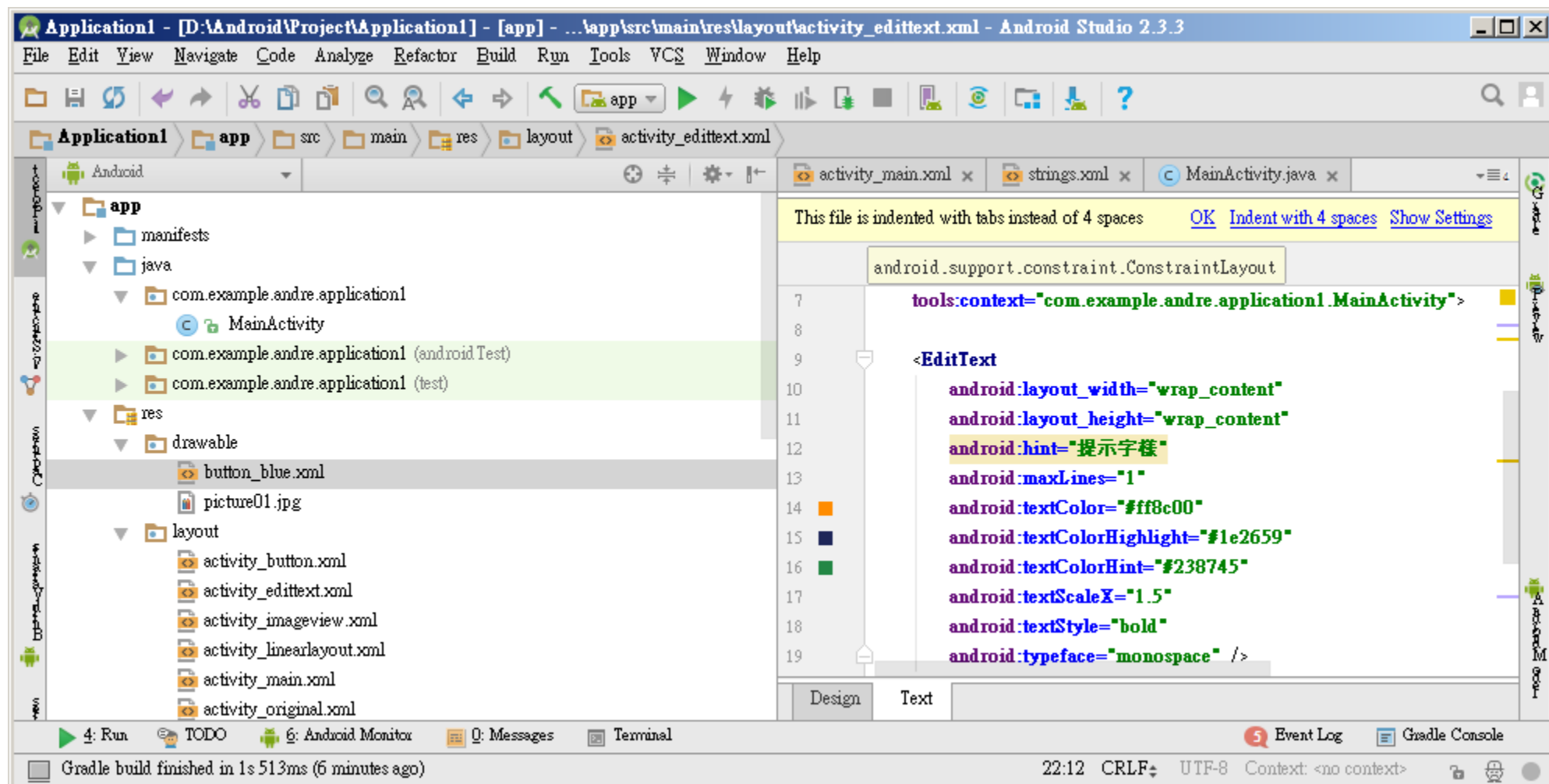
- EditText 文字輸入元件-視覺化工具





ANDROID 基礎介面元件

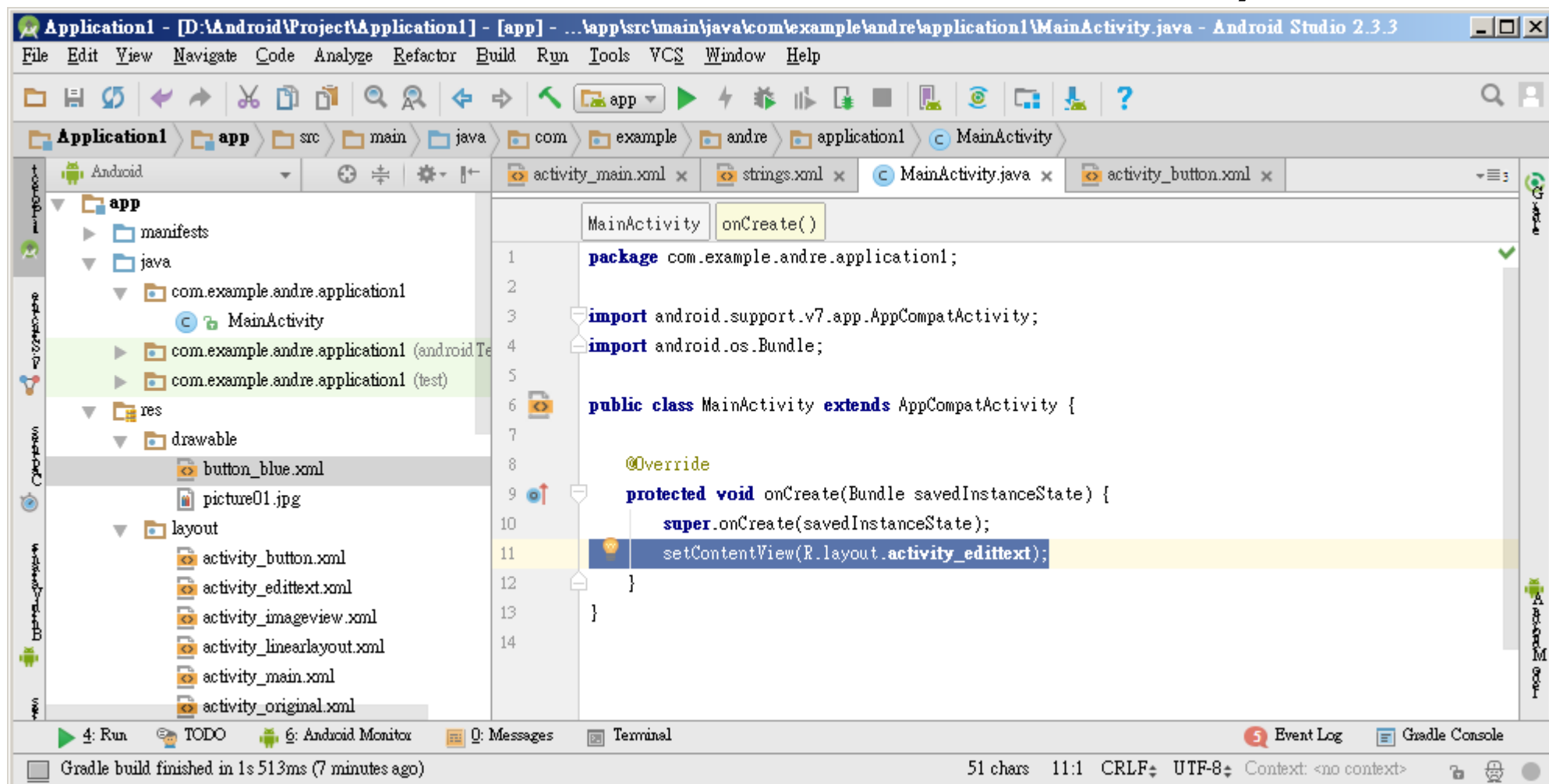
• EditText 文字輸入元件-文字編輯器





ANDROID 基礎介面元件

- EditText 文字輸入元件-切換 MainActivity顯示





ANDROID 基礎介面元件

- **EditText** 文字輸入元件-切換 **MainActivity**顯示
- 開啟 **MainActivity.java** ，並於其中加入以下程式碼
 - `package com.example.andre.application I;`
 - `import android.support.v7.app.AppCompatActivity;`
 - `import android.os.Bundle;`
 - `public class MainActivity extends AppCompatActivity {`
 - `@Override`
 - `protected void onCreate(Bundle savedInstanceState) {`
 - `super.onCreate(savedInstanceState);`
 - `setContentView(R.layout.activity_activity_edittext);`
 - `}`
 - `}`



ANDROID 基礎介面元件

- EditText 文字輸入元件-成果





ANDROID 基礎介面元件

- Checkbox 選取方塊元件
- 開啟 `activity_checkbox.xml`，並於其中加入以下程式碼
 - `<CheckBox`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:text="籃球"`
 - `android:textSize="20sp"`
 - `app:layout_constraintBottom_toBottomOf="parent"`
 - `app:layout_constraintLeft_toLeftOf="parent"`
 - `app:layout_constraintRight_toRightOf="parent"`
 - `app:layout_constraintTop_toTopOf="parent"`
 - `app:layout_constraintHorizontal_bias="0.5"`
 - `app:layout_constraintVertical_bias="0.6"`
 - `android:button="@drawable/checkbox_selector"`
 - `/>`



ANDROID 基礎介面元件

- Checkbox 選取方塊元件
 - `android:layout_width="wrap_content"`
 - 設定寬度-符合文字寬
 - `android:layout_height="wrap_content"`
 - 設定高度-符合文字高
 - `android:text="籃球"`
 - 設定顯示文字
 - `android:textSize="20sp"`
 - 設定顯示文字大小



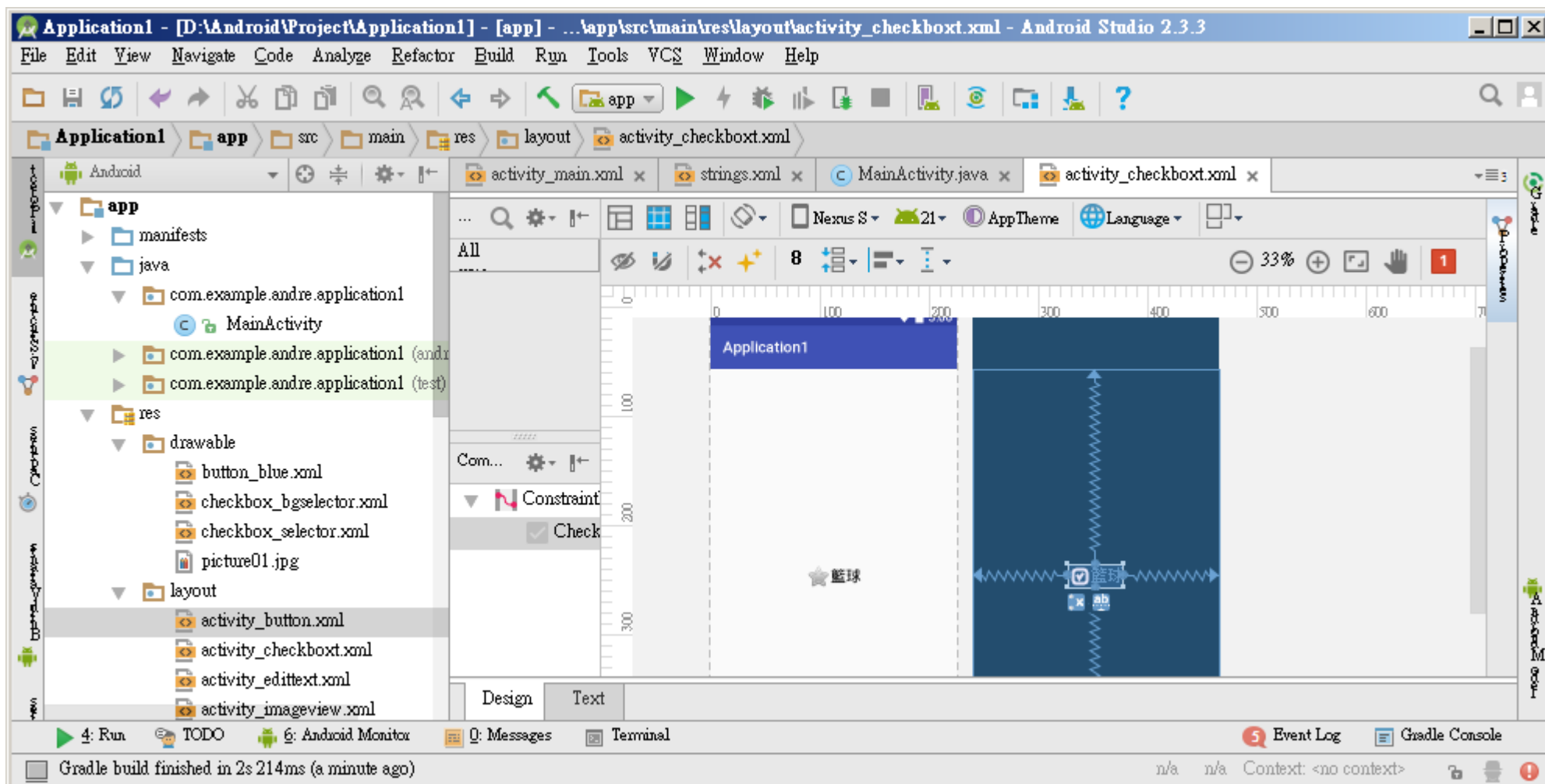
ANDROID 基礎介面元件

- Checkbox 選取方塊元件
 - `app:layout_constraintBottom_toBottomOf="parent"`
 - `app:layout_constraintLeft_toLeftOf="parent"`
 - `app:layout_constraintRight_toRightOf="parent"`
 - `app:layout_constraintTop_toTopOf="parent"`
 - `app:layout_constraintHorizontal_bias="0.5"`
 - `app:layout_constraintVertical_bias="0.6"`
- 設定位置座標



ANDROID 基礎介面元件

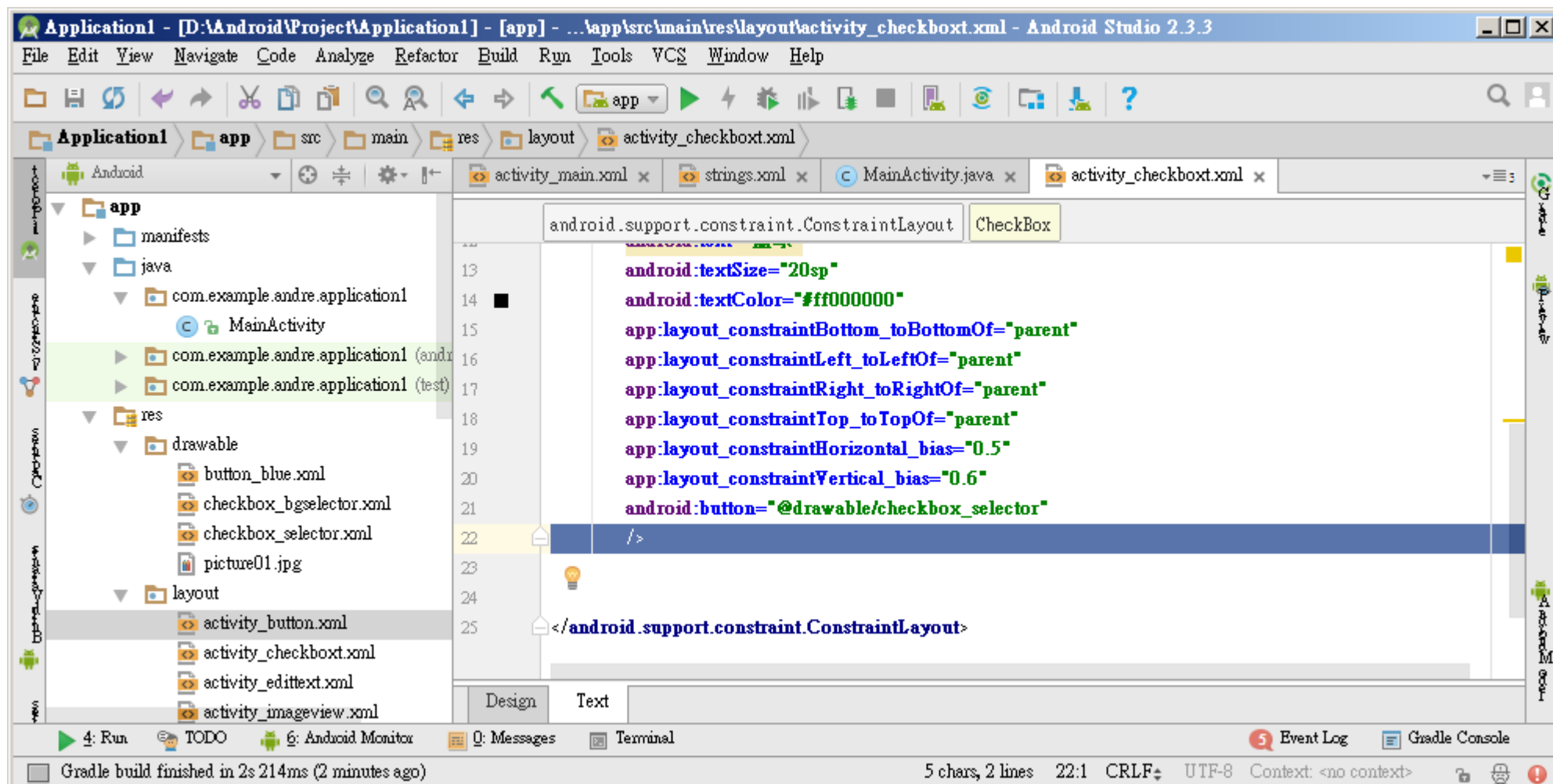
- Checkbox 選取方塊元件-視覺化工具





ANDROID 基礎介面元件

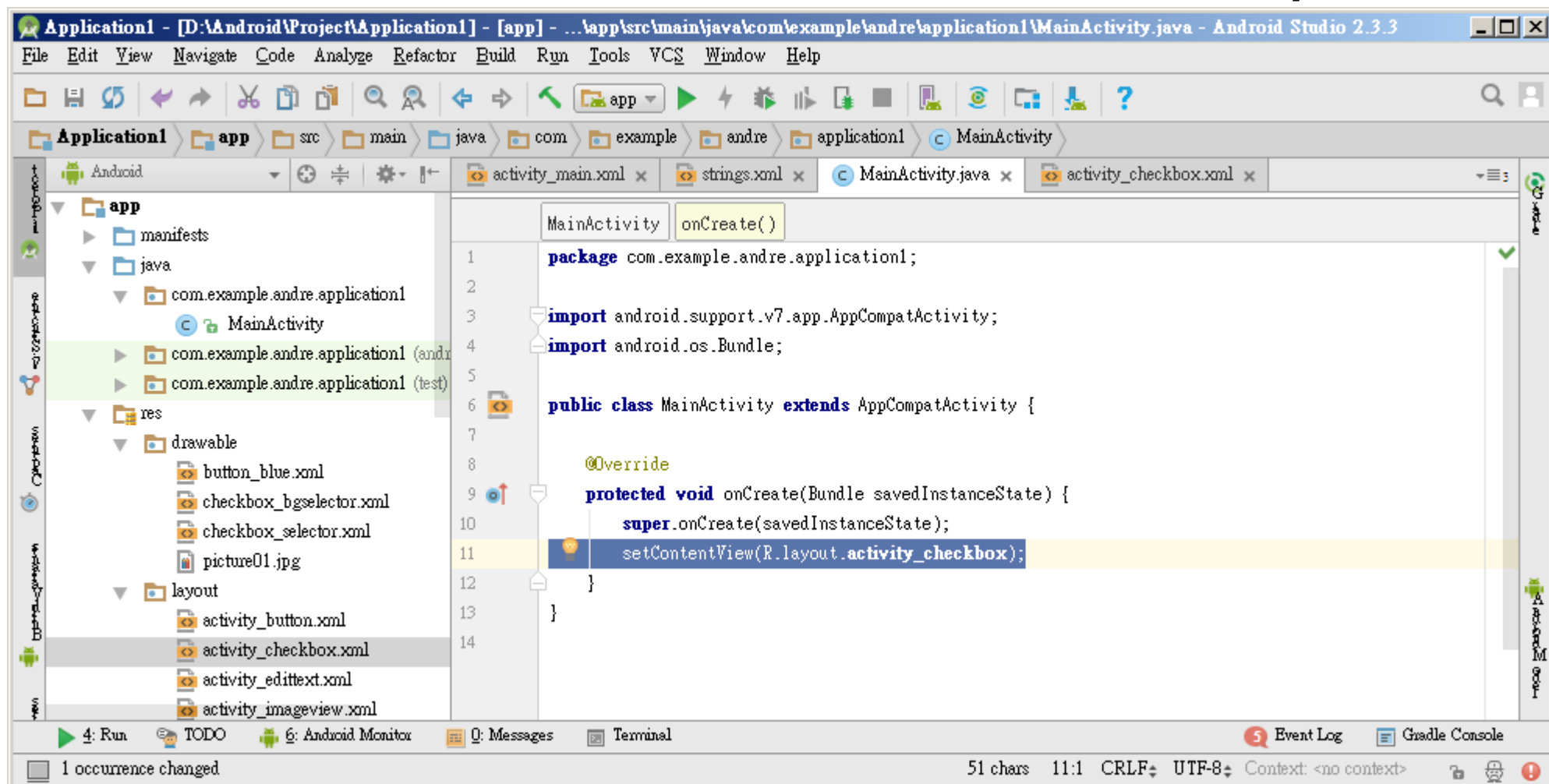
- Checkbox 選取方塊元件-文字編輯器





ANDROID 基礎介面元件

- Checkbox 選取方塊元件-切換 MainActivity顯示





ANDROID 基礎介面元件

- **Checkbox** 選取方塊元件-切換 **MainActivity**顯示
- 開啟 **MainActivity.java** ，並於其中加入以下程式碼
 - `package com.example.andre.application I;`
 - `import android.support.v7.app.AppCompatActivity;`
 - `import android.os.Bundle;`
 - `public class MainActivity extends AppCompatActivity {`
 - `@Override`
 - `protected void onCreate(Bundle savedInstanceState) {`
 - `super.onCreate(savedInstanceState);`
 - `setContentView(R.layout.activity_activity_checkbox);`
 - `}`
 - `}`



ANDROID 基礎介面元件

- Checkbox 選取方塊元件
- 新增 /drawable/checkbox_selector.xml，並於其中加入以下程式碼
 - `<?xml version="1.0" encoding="utf-8"?>`
 - `<selector xmlns:android="http://schemas.android.com/apk/res/android"`
 - `android:constantSize="true" >`
 - `<item android:state_checked="true"`
 - `android:drawable="@android:drawable/btn_star_big_on" />`
 - `<item android:drawable="@android:drawable/btn_star_big_off" />`
 - `</selector>`



ANDROID 基礎介面元件

- Checkbox 選取方塊元件
- `android:state_checked="true"`
- 勾選的時候顯示
 - `android:drawable="@android:drawable/btn_star_big_on"`
- 未選取顯示
 - `android:drawable="@android:drawable/btn_star_big_off"`



ANDROID 基礎介面元件

- Checkbox 選取方塊元件-成果
 - `android:button="@drawable/checkbox_selector"`





ANDROID 基礎介面元件

- **RadioButton** 單選按鈕元件
- 開啟 `activity_radiobutton.xml`，並於其中加入以下程式碼
 - `<RadioGroup`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_marginTop="30dp" >`
 - `<RadioButton`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:text="www.google.com" />`
 - `<RadioButton`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:text="www.yahoo.com" />`
 - `<RadioButton`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:text="www.cnyes.com" />`
 - `</RadioGroup>`



ANDROID 基礎介面元件

- **RadioButton 單選按鈕元件**
 - `android:layout_width="wrap_content"`
 - 設定寬度-符合文字寬
 - `android:layout_height="wrap_content"`
 - 設定高度-符合文字高
 - `android:text="www.google.com"`
 - 設定顯示文字
 - `android:layout_marginTop="30dp"`
 - 設定上方邊界



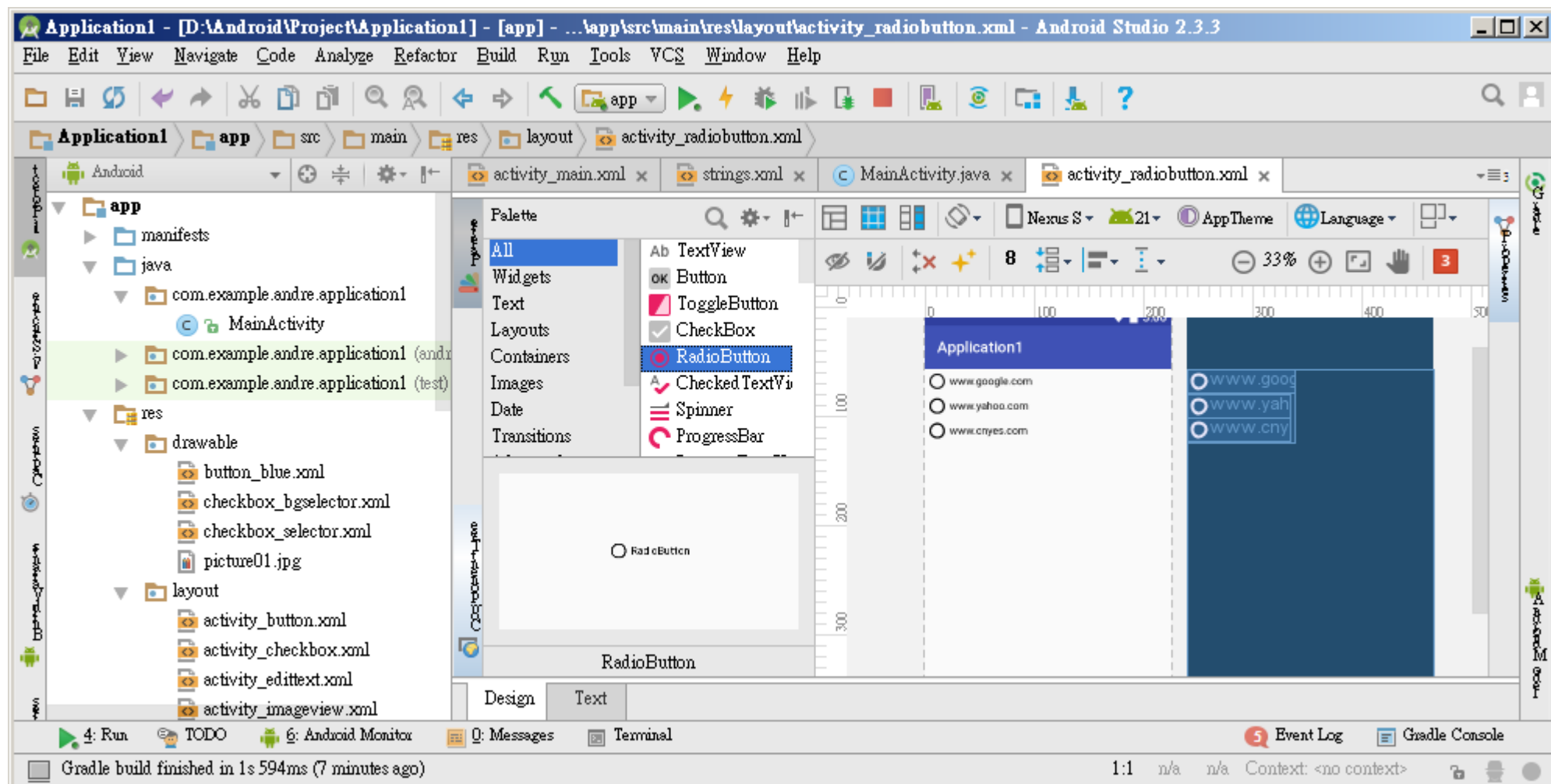
ANDROID 基礎介面元件

- **RadioButton 單選按鈕元件**
- 嘗試在 **RadioGroup** 中設定位置座標
 - `app:layout_constraintBottom_toBottomOf="parent"`
 - `app:layout_constraintLeft_toLeftOf="parent"`
 - `app:layout_constraintRight_toRightOf="parent"`
 - `app:layout_constraintTop_toTopOf="parent"`
 - `app:layout_constraintHorizontal_bias="0.5"`
 - `app:layout_constraintVertical_bias="0.6"`



ANDROID 基礎介面元件

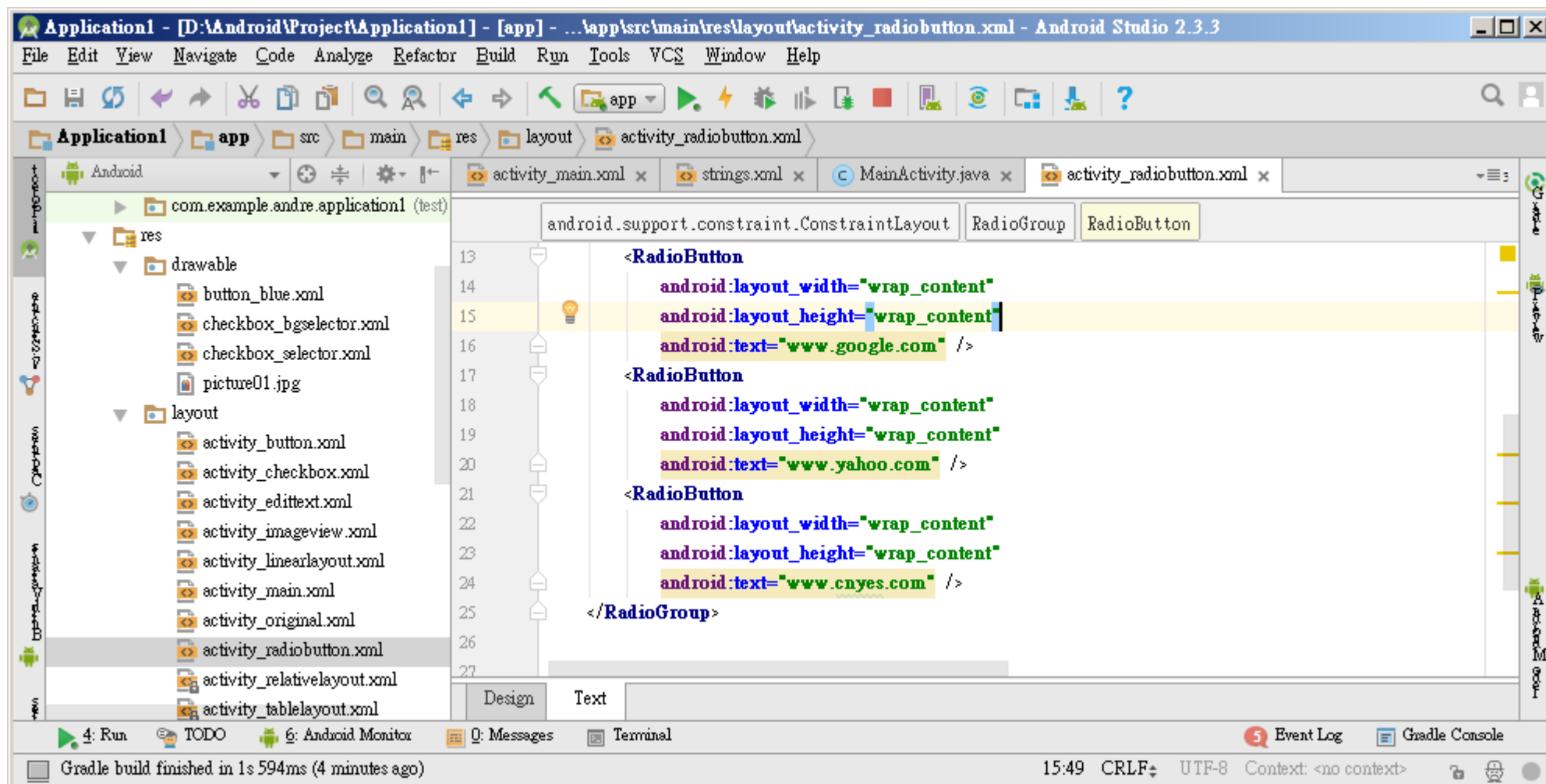
• RadioButton 單選按鈕元件-視覺化工具





ANDROID 基礎介面元件

- RadioButton 單選按鈕元件-文字編輯器





ANDROID 基礎介面元件

- RadioButton 單選按鈕元件-切換 MainActivity顯示

```
Application1 - [D:\Android\Project\Application1] - [app] - ...app\src\main\java\com\example\andre\application1\MainActivity.java - Android Studio 2.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Application1 app src main java com example andre application1 MainActivity
activity_main.xml x strings.xml x MainActivity.java x activity_radiobutton.xml x

com.example.andre.application1 (test)
  res
    drawable
      button_blue.xml
      checkbox_bgselector.xml
      checkbox_selector.xml
      picture01.jpg
    layout
      activity_button.xml
      activity_checkbox.xml
      activity_edittext.xml
      activity_imageview.xml
      activity_linearlayout.xml
      activity_main.xml
      activity_original.xml
      activity_radiobutton.xml
      activity_relativelayout.xml
      activity_tablelayout.xml

MainActivity onCreate()
1 package com.example.andre.application1;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_radiobutton);
12     }
13 }
14

4: Run TODO 6: Android Monitor Q: Messages Terminal Event Log Gradle Console
Gradle build finished in 1s 594ms (3 minutes ago) 54 chars 11:1 CRLF UTF-8 Context: <no context>
```



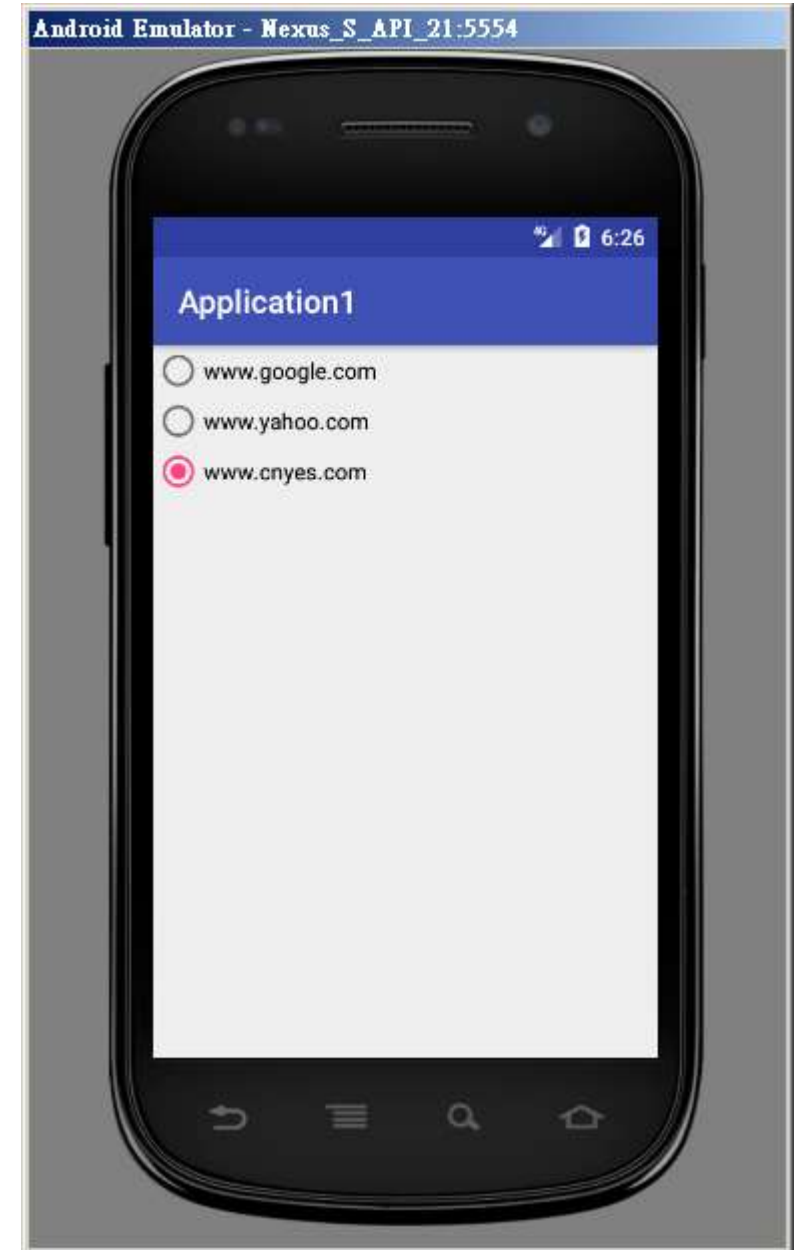
ANDROID 基礎介面元件

- **RadioButton** 單選按鈕元件-切換 MainActivity顯示
- 開啟 MainActivity.java ，並於其中加入以下程式碼
 - package com.example.andre.application I;
 - import android.support.v7.app.AppCompatActivity;
 - import android.os.Bundle;
 - public class MainActivity extends AppCompatActivity {
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_activity_radiobutton);
 - }
 - }



ANDROID 基礎介面元件

- RadioButton 單選按鈕元件-
成果





本節課程內容

- 本節課程內容將包含以下教學內容:

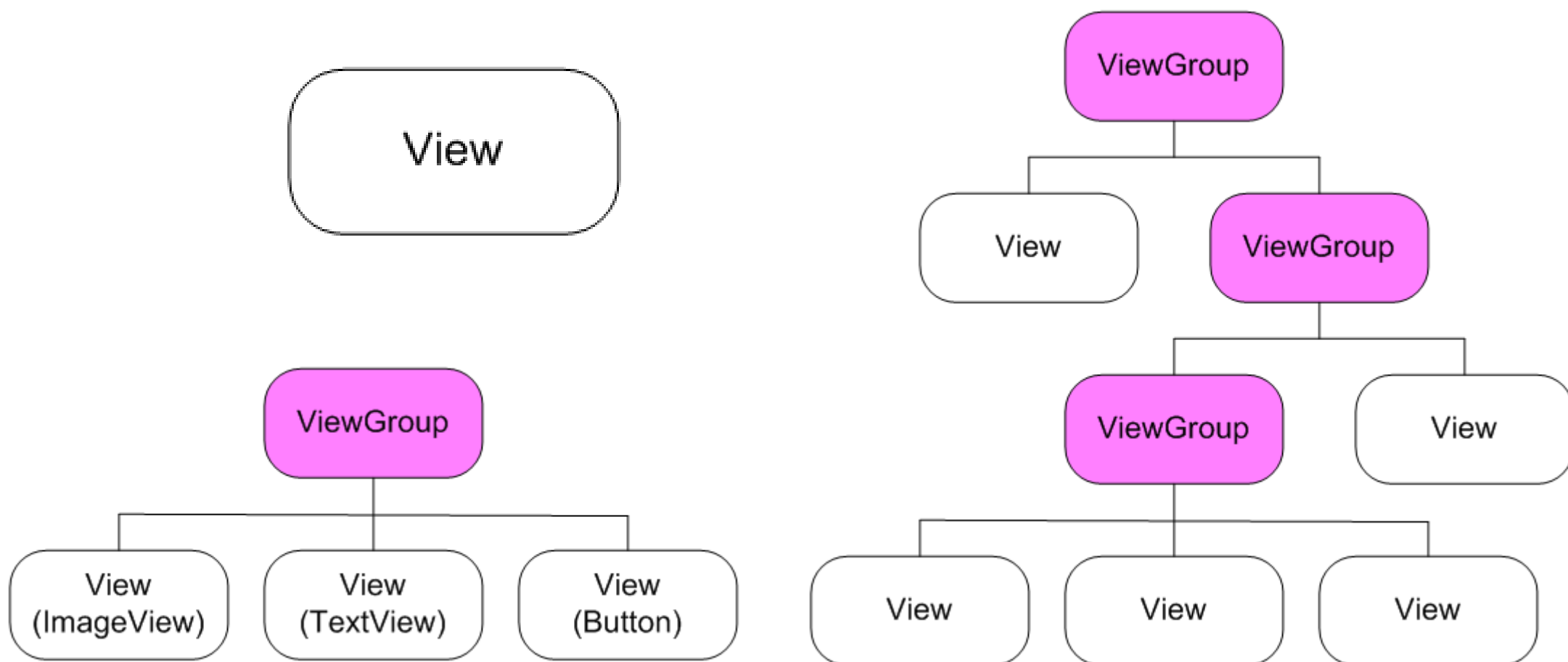
UI 使用者介面設計與規劃

- LinearLayout
- Relative Layout
- Table Layout
- FrameLayout
- GridLayout
- ConstraintLayout



UI 使用者介面設計與規劃

- Layout、ViewGroup、View 組合使用





UI 使用者介面設計與規劃

- LinearLayout
- 開啟 activity_linearlayout.xml ，並於其中加入以下程式碼
 - <ImageView
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:src="@drawable/picture01"
 - />
 - <TextView
 - android:layout_width="match_parent"
 - android:layout_height="wrap_content"
 - android:textSize="30sp"
 - android:text="@string/myname"
 - />



UI 使用者介面設計與規劃

- LinearLayout
- 開啟 activity_linearlayout.xml ， 續
 - <LinearLayout
 - android:layout_width="match_parent"
 - android:layout_height="wrap_content"
 - android:orientation="horizontal"
 - android:gravity="center_horizontal"
 - >
 - <Button
 - android:layout_width="wrap_content"
 - android:layout_height="match_parent"
 - android:textSize="20sp"
 - android:text="@string/website"
 - />



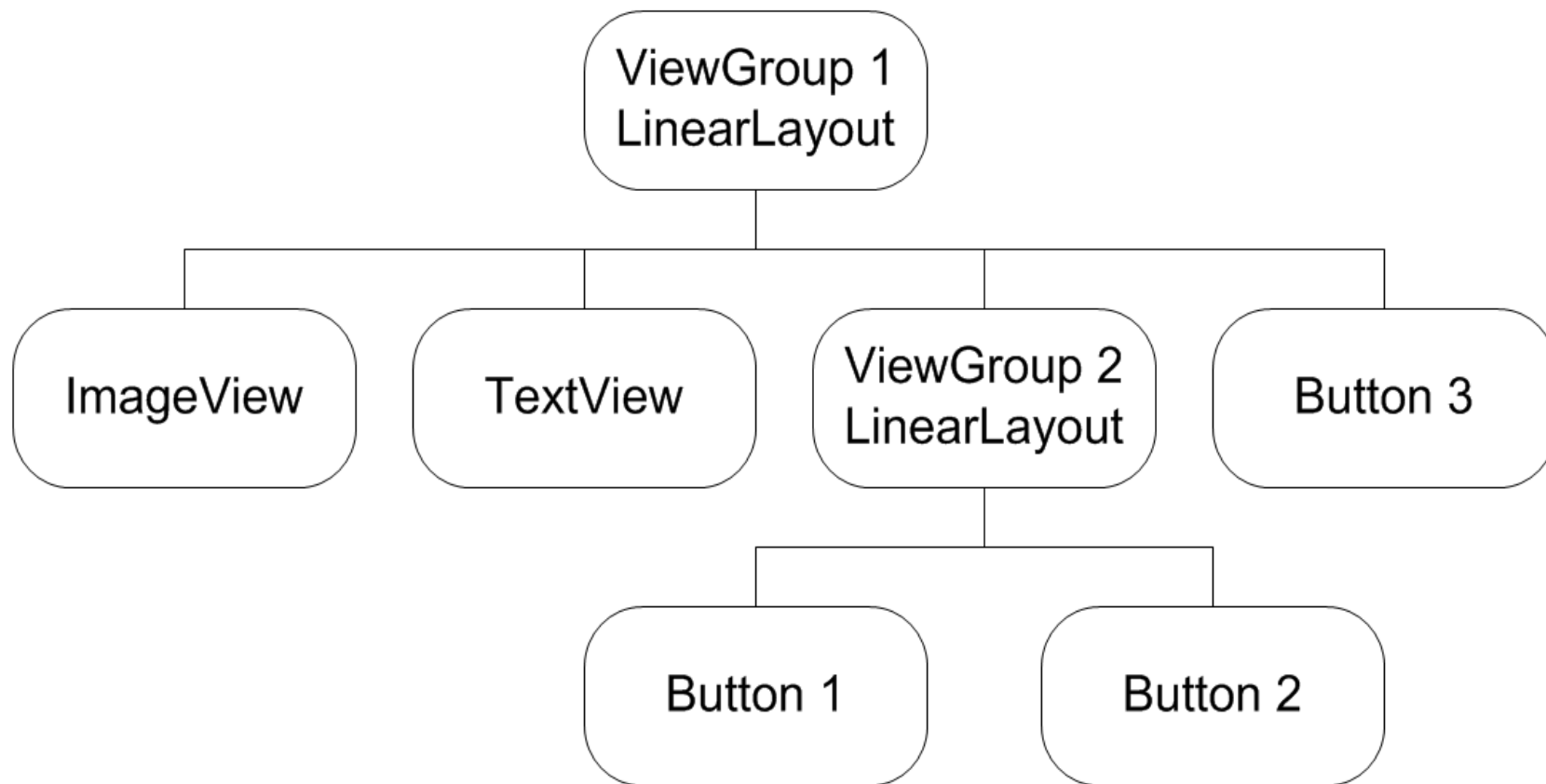
UI 使用者介面設計與規劃

- LinearLayout
- 開啟 activity_linearlayout.xml ， 續
 - <Button
 - android:layout_width="wrap_content"
 - android:layout_height="match_parent"
 - android:textSize="20sp"
 - android:text="@string/myapps"
 - />
 - </LinearLayout>
 - <Button
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:textSize="20sp"
 - android:text="@string/mybutton"
 - />
 - </LinearLayout>



UI 使用者介面設計與規劃

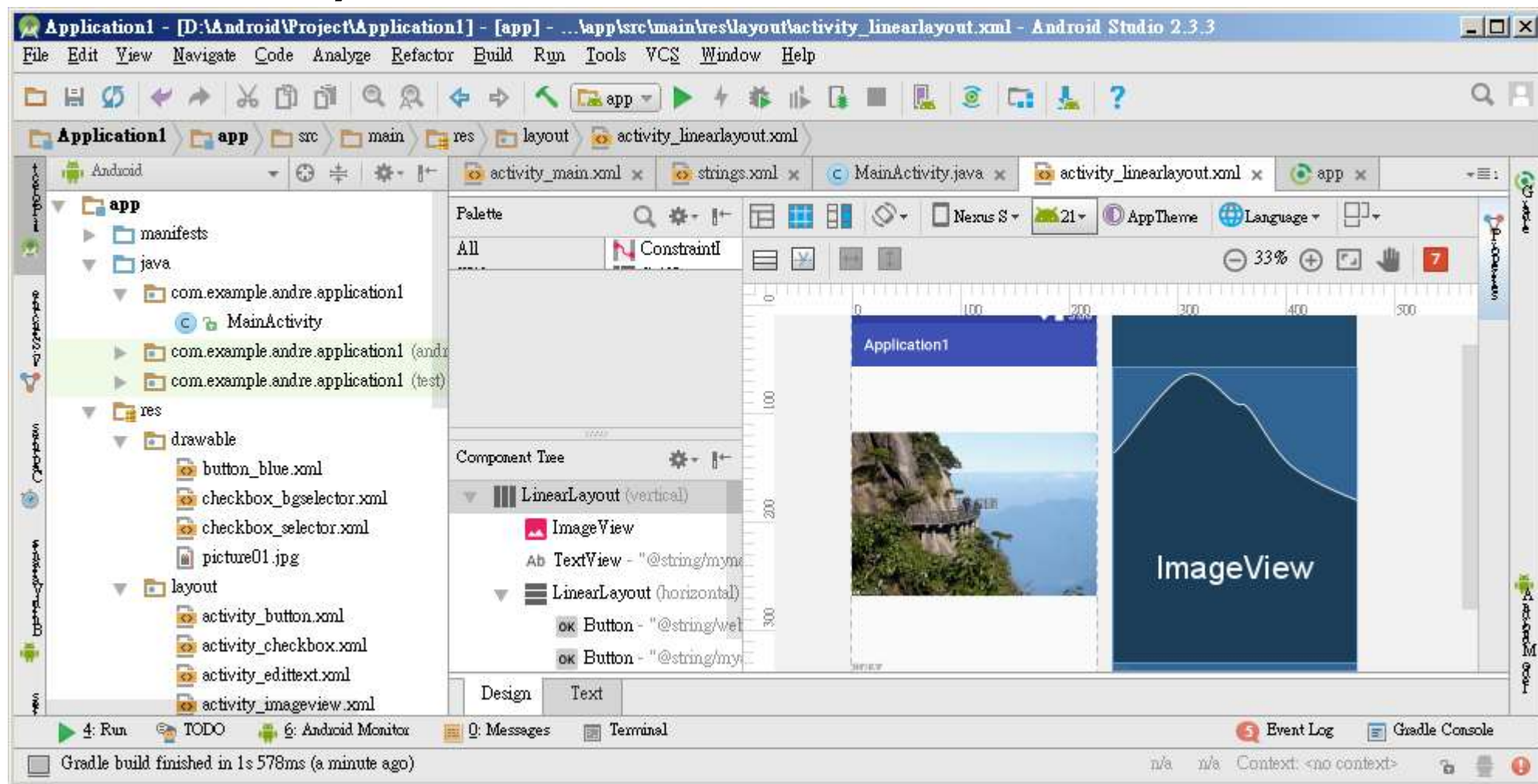
- LinearLayout 樹狀結構





ANDROID 基礎介面元件

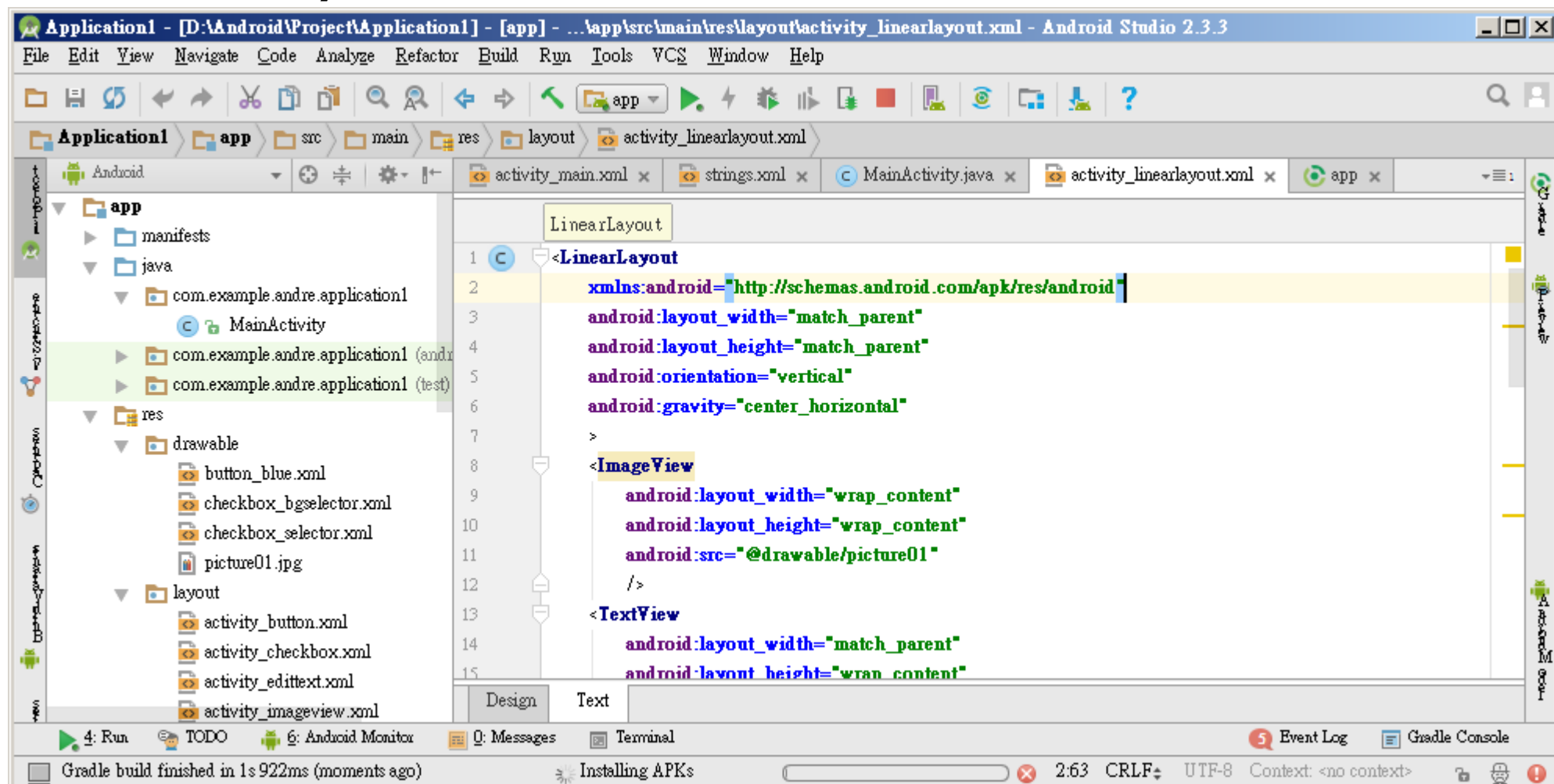
- LinearLayout - 視覺化工具





ANDROID 基礎介面元件

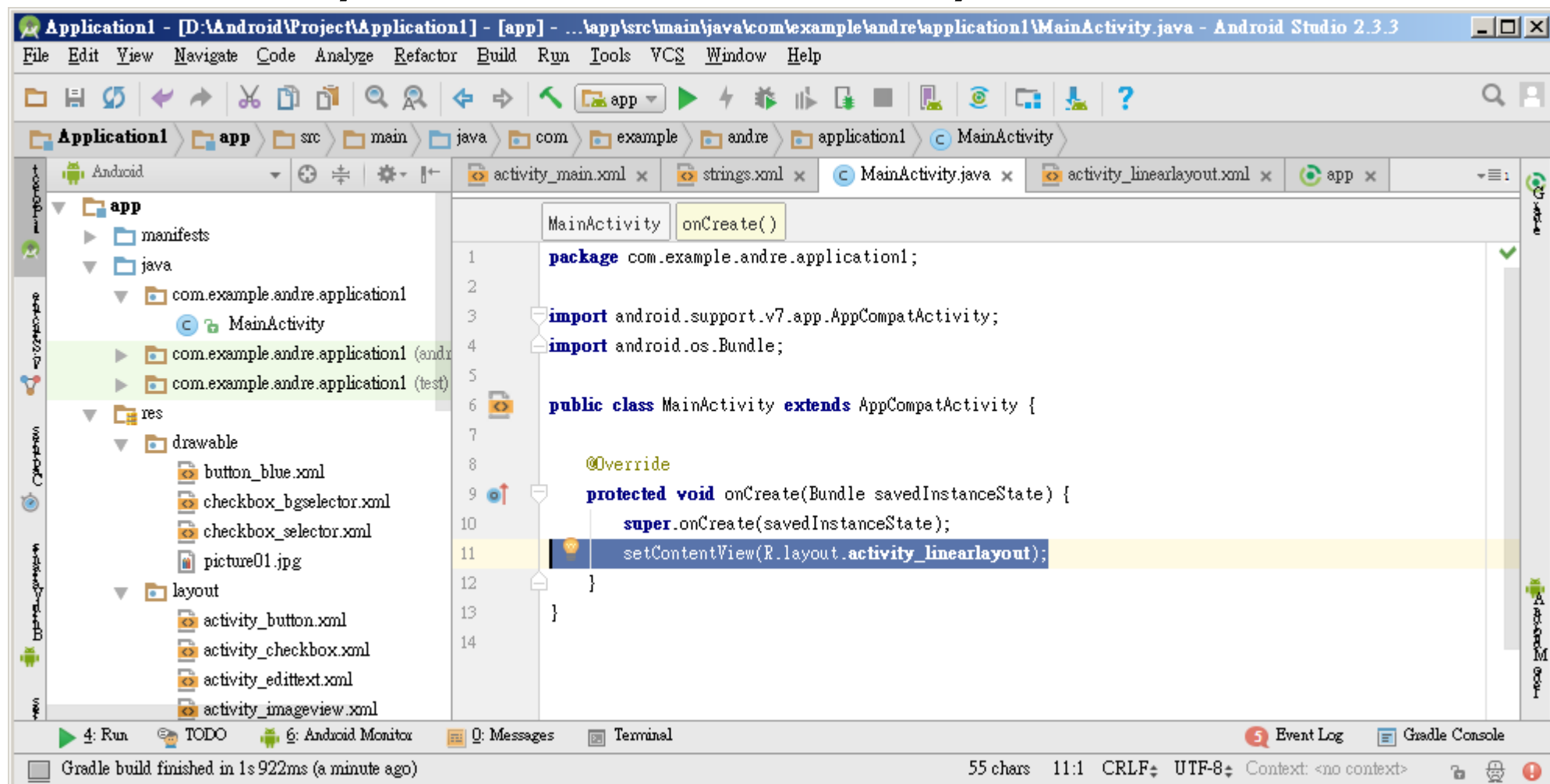
- LinearLayout - 文字編輯器





ANDROID 基礎介面元件

- LinearLayout - 切換 MainActivity 顯示





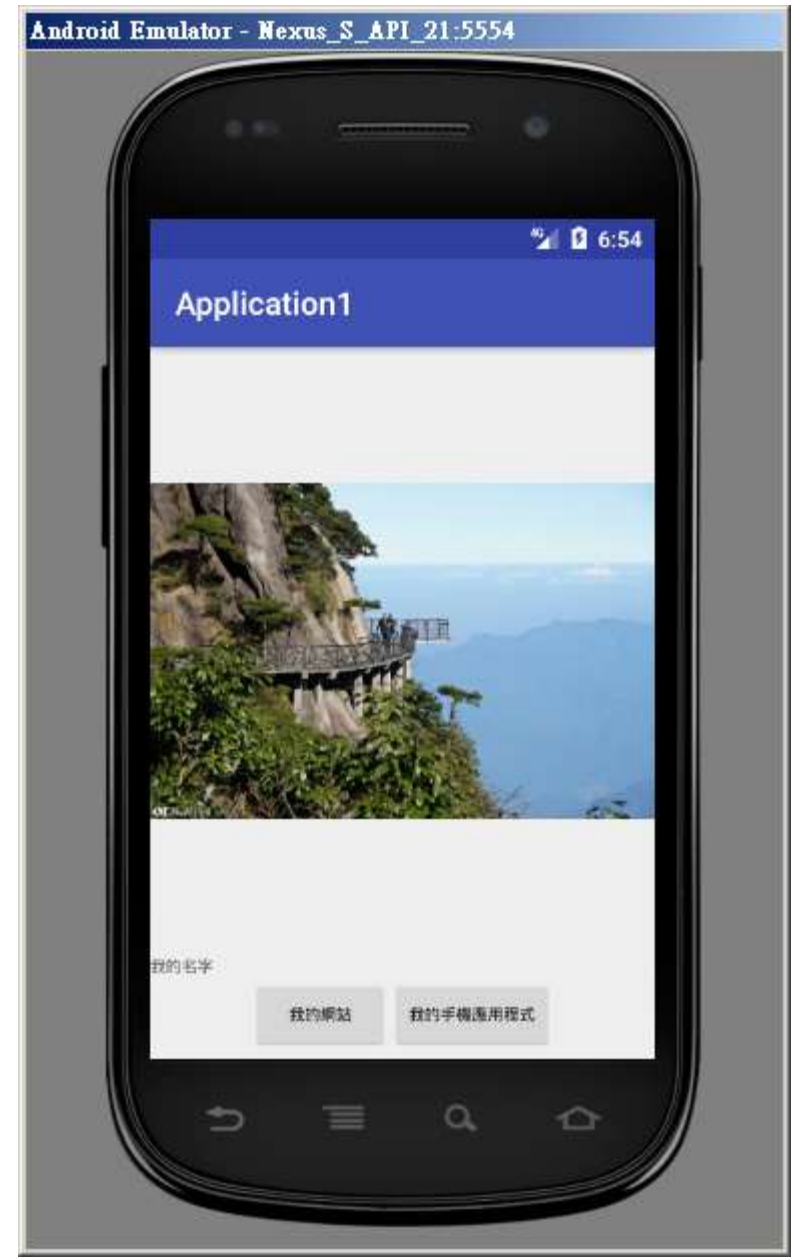
ANDROID 基礎介面元件

- LinearLayout - 切換 MainActivity 顯示
- 開啟 MainActivity.java ，並於其中加入以下程式碼
 - package com.example.andre.application I;
 - import android.support.v7.app.AppCompatActivity;
 - import android.os.Bundle;
 - public class MainActivity extends AppCompatActivity {
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_activity_linearlayout);
 - }
 - }



ANDROID 基礎介面元件

- LinearLayout - 成果





UI 使用者介面設計與規劃

- RelativeLayout
- 開啟 activity_relativelayout.xml ，並於其中加入以下程式碼
 - `<TextView android:id="@+id/tv"`
 - `android:layout_width="fill_parent"`
 - `android:layout_height="wrap_content"`
 - `android:textSize="20sp"`
 - `android:text="@string/myapps"`
 - `/>`
 - `<Button android:id="@+id/btn"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_alignParentBottom="true"`
 - `android:text="Button"`
 - `/>`
 - `<LinearLayout`
 - `android:layout_height="wrap_content"`
 - `android:layout_width="fill_parent"`



UI 使用者介面設計與規劃

- RelativeLayout
- 開啟 activity_relativelayout.xml ， 續
 - `android:layout_above="@id/btn"`
 - `android:orientation="horizontal">`
 - `<ImageView`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:src="@android:drawable/star_big_on"`
 - `/>`
 - `<ImageView`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:src="@android:drawable/star_big_on"`
 - `/>`



UI 使用者介面設計與規劃

- RelativeLayout
- 開啟 activity_relativelayout.xml ， 續
 - <ImageView
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:src="@android:drawable/star_big_on"
 - />
 - </LinearLayout>
 - <EditText
 - android:layout_width="fill_parent"
 - android:layout_height="wrap_content"
 - android:layout_below="@id/tv"
 - />
 - </RelativeLayout>



UI 使用者介面設計與規劃

- RelativeLayout
 - `android:id="@+id/tv"`
 - 利用@[資源類別]/[識別名稱]來讀取資源
 - 「@」符號後面跟著一個「+」符號，是要建立一個資源
- `android:layout_alignParentBottom`
 - Parent指的是RelativeLayout
- `android:layout_above="@id/btn"`
 - LinearLayout會位於Button的上方
 - 定義"@+id/btn"的地方要在使用"@id/btn"之前
- `@android:drawable/star_big_on`
 - 系統所提供的圖檔
 - 參閱SDK文件中的R.drawable類別



UI 使用者介面設計與規劃

- RelativeLayout
- 位置相對屬性對象為父元件(容器) :
 - android:layout_alignParentTop
 - 對齊容器的頂端
 - android:layout_alignParentBottom
 - 對齊容器的底端
 - android:layout_alignParentLeft
 - 對齊容器的左邊緣
 - android:layout_alignParentRight
 - 對齊容器的右邊緣
 - android:layout_alignParentCenter
 - 元件放在容器的中央



UI 使用者介面設計與規劃

- RelativeLayout
- 位置相對屬性對象為同容器內一般元件：
 - android:layout_above
 - 元件放在對象之上
 - android:layout_below
 - 元件放在對象之下
 - android:layout_toLeftOf
 - 元件的右邊緣對齊對象的左邊緣
 - android:layout_toRightOf
 - 元件的左邊緣對齊對象的右邊緣
 - android:layout_alignTop
 - 元件的上緣對齊對象的上緣



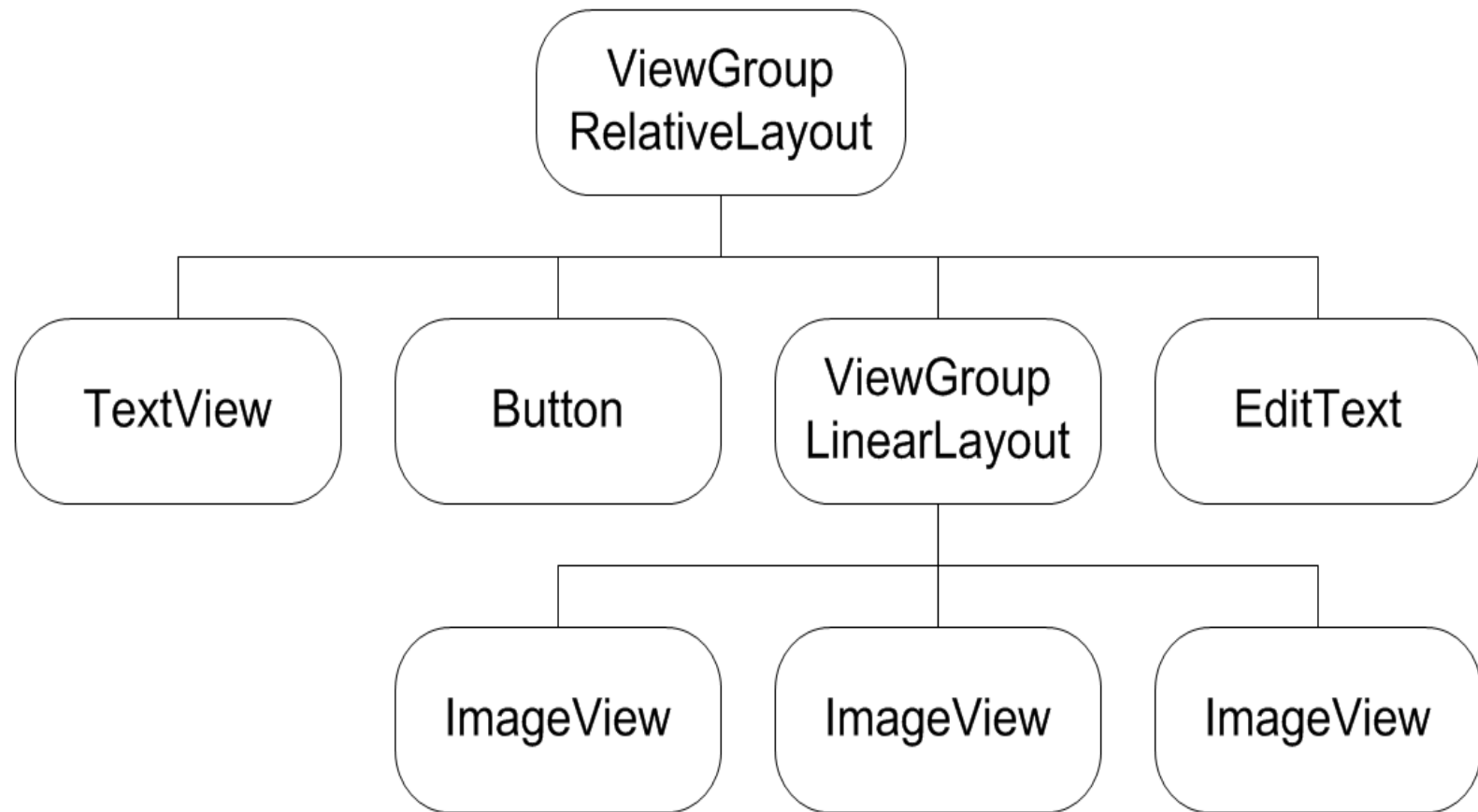
UI 使用者介面設計與規劃

- RelativeLayout
- 位置相對屬性對象為同容器內一般元件：
 - android:layout_alignBottom
 - 元件的下緣對齊對象的下緣
 - android:layout_alignLeft
 - 元件的左邊緣對齊對象的左邊緣
 - android:layout_alignRight
 - 元件的右邊緣對齊對象的右邊緣
 - android:layout_centerHorizontal
 - 元件水平置中
 - android:layout_centerVertical
 - 元件垂直置中



UI 使用者介面設計與規劃

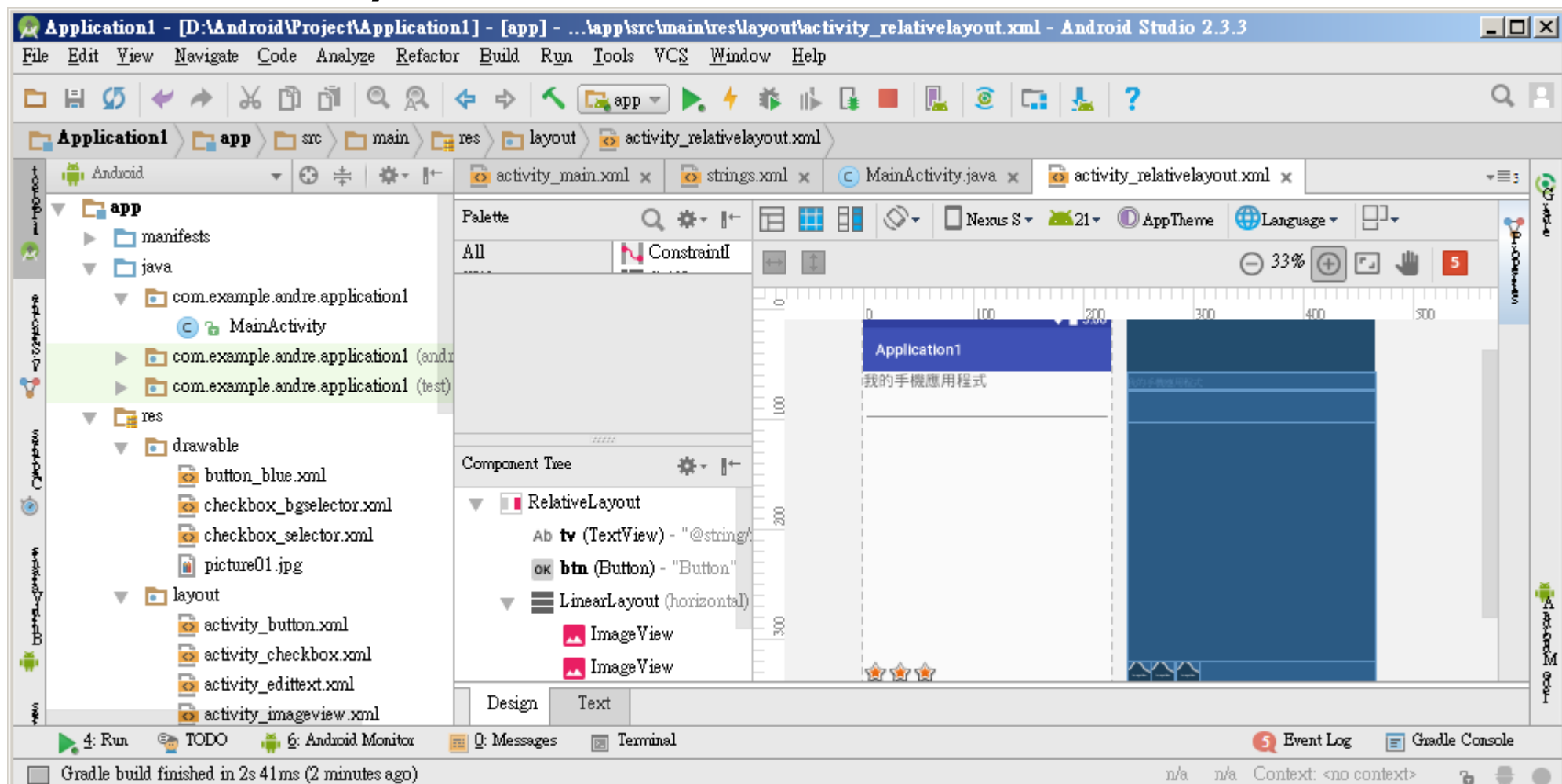
- RelativeLayout 樹狀結構





ANDROID 基礎介面元件

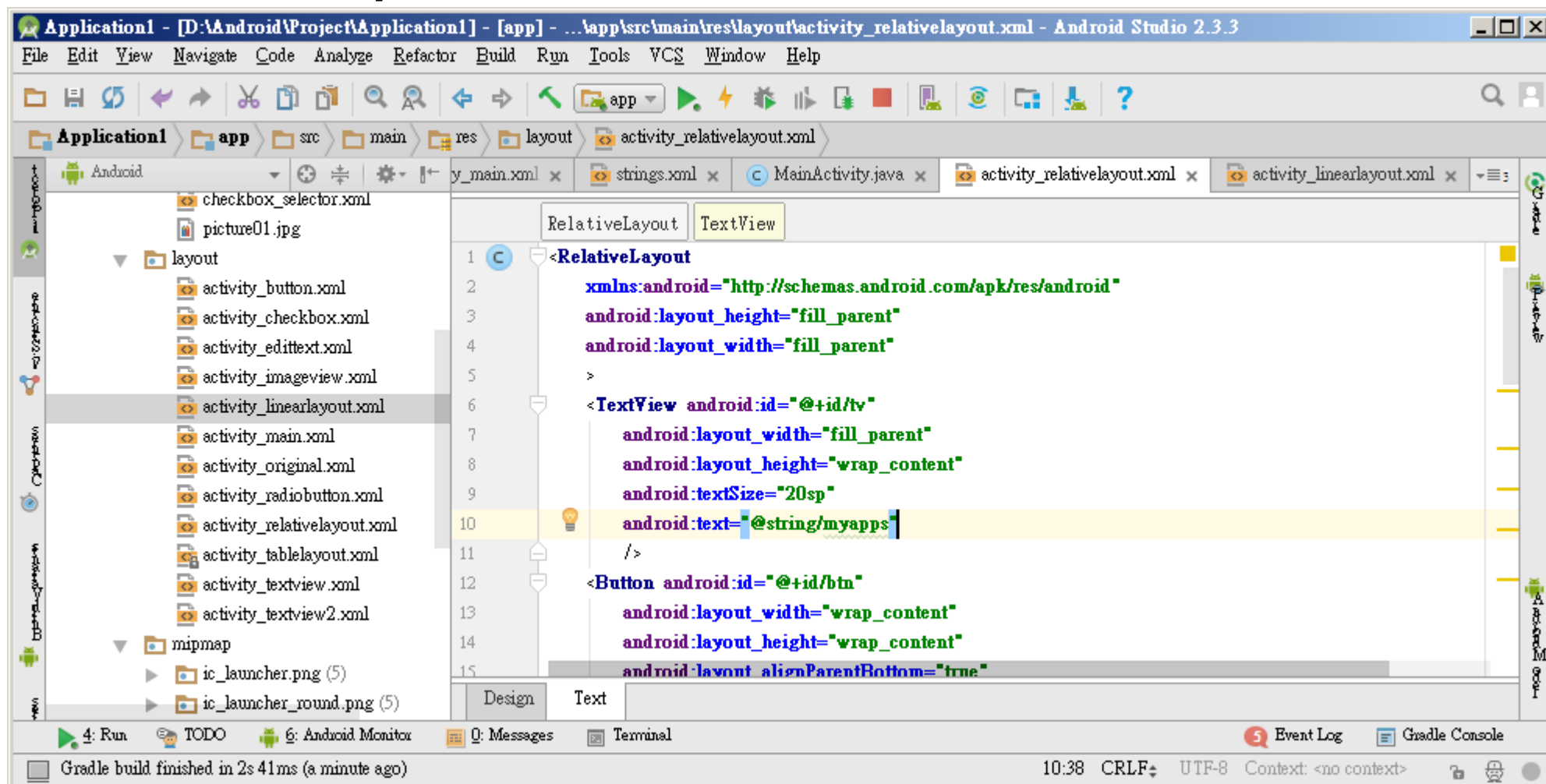
- RelativeLayout - 視覺化工具





ANDROID 基礎介面元件

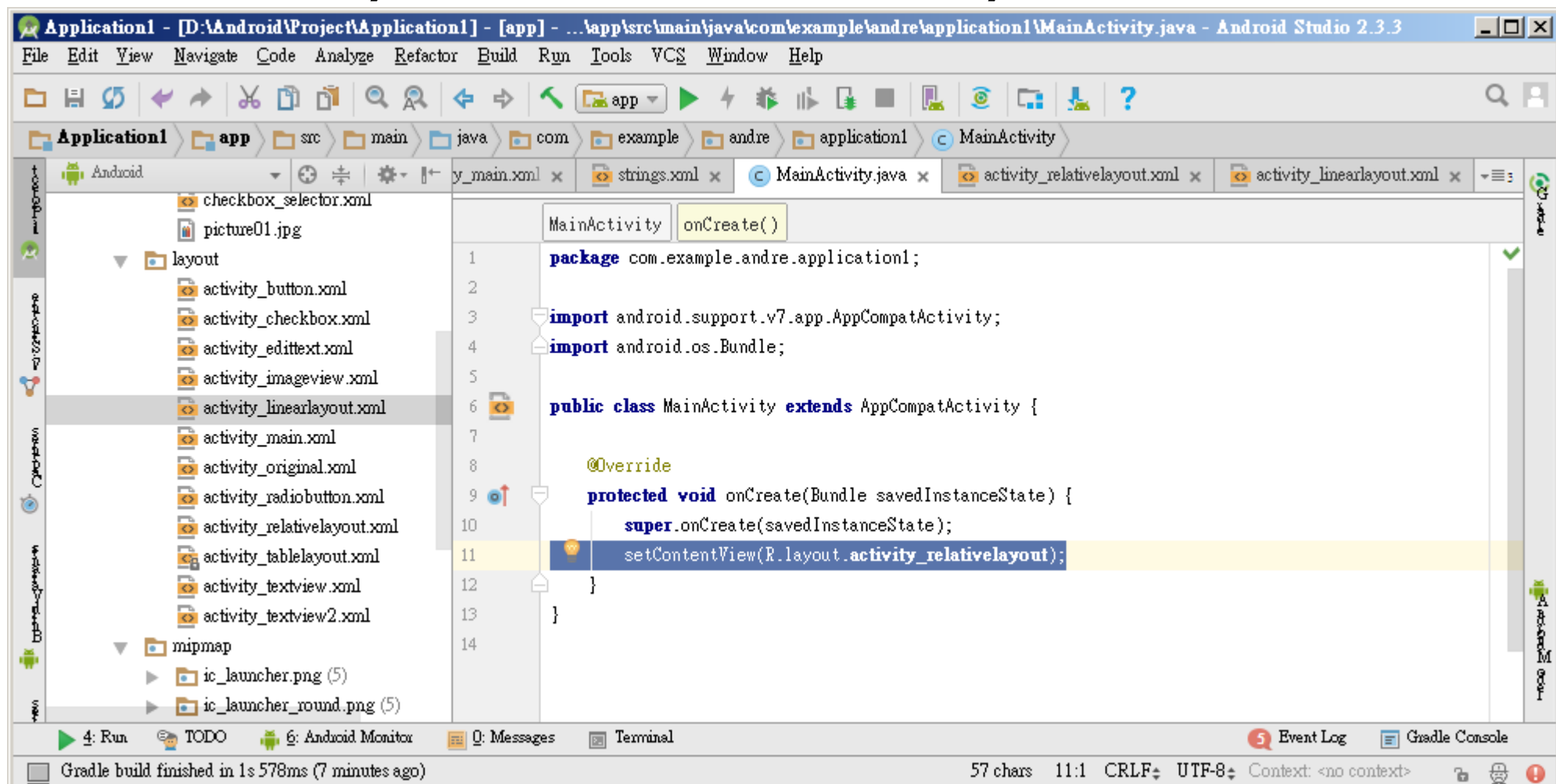
- RelativeLayout - 文字編輯器





ANDROID 基礎介面元件

- RelativeLayout - 切換 MainActivity 顯示





ANDROID 基礎介面元件

- RelativeLayout - 切換 MainActivity 顯示
- 開啟 MainActivity.java ，並於其中加入以下程式碼
 - package com.example.andre.application I;
 - import android.support.v7.app.AppCompatActivity;
 - import android.os.Bundle;
 - public class MainActivity extends AppCompatActivity {
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_activity_relativelayout);
 - }
 - }



ANDROID 基礎介面元件

- RelativeLayout - 成果





UI 使用者介面設計與規劃

- **TableLayout**
- 開啟 `activity_tablelayout.xml`，並於其中加入以下程式碼
 - `<TableLayout`
 - `xmlns:android="http://schemas.android.com/apk/res/android"`
 - `android:layout_height="fill_parent"`
 - `android:layout_width="fill_parent"`
 - `>`
 - `<TableRow>`
 - `<TextView android:text="英雄聯盟"/>`
 - `<TextView android:text="Version 1.6.8"/>`
 - `<TextView android:text="Android 2.2"/>`
 - `</TableRow>`
 - `<ImageView`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`



UI 使用者介面設計與規劃

- TableLayout
- 開啟 activity_tablelayout.xml.xml ， 續
 - `android:layout_centerHorizontal="true"`
 - `android:layout_centerVertical="true"`
 - `android:src="@android:drawable/star_big_on"`
 - `/>`
 - `<TableRow>`
 - `<TextView android:text="傳說對決"/>`
 - `<TextView android:text="Version 1.8"/>`
 - `<TextView android:text="Android 4.0"/>`
 - `</TableRow>`
 - `<ImageView`
 - `android:layout_width="wrap_content"`



UI 使用者介面設計與規劃

- TableLayout
- 開啟 activity_tablelayout.xml.xml ， 續
 - android:layout_height="wrap_content"
 - android:layout_centerHorizontal="true"
 - android:layout_centerVertical="true"
 - android:src="@android:drawable/star_big_on"
 - />
 - <TableRow>
 - <TextView android:text="戰慄時空"/>
 - <TextView android:text="Version 2.1"/>
 - <TextView android:text="Android 2.3"/>
 - </TableRow>
 - </TableLayout>



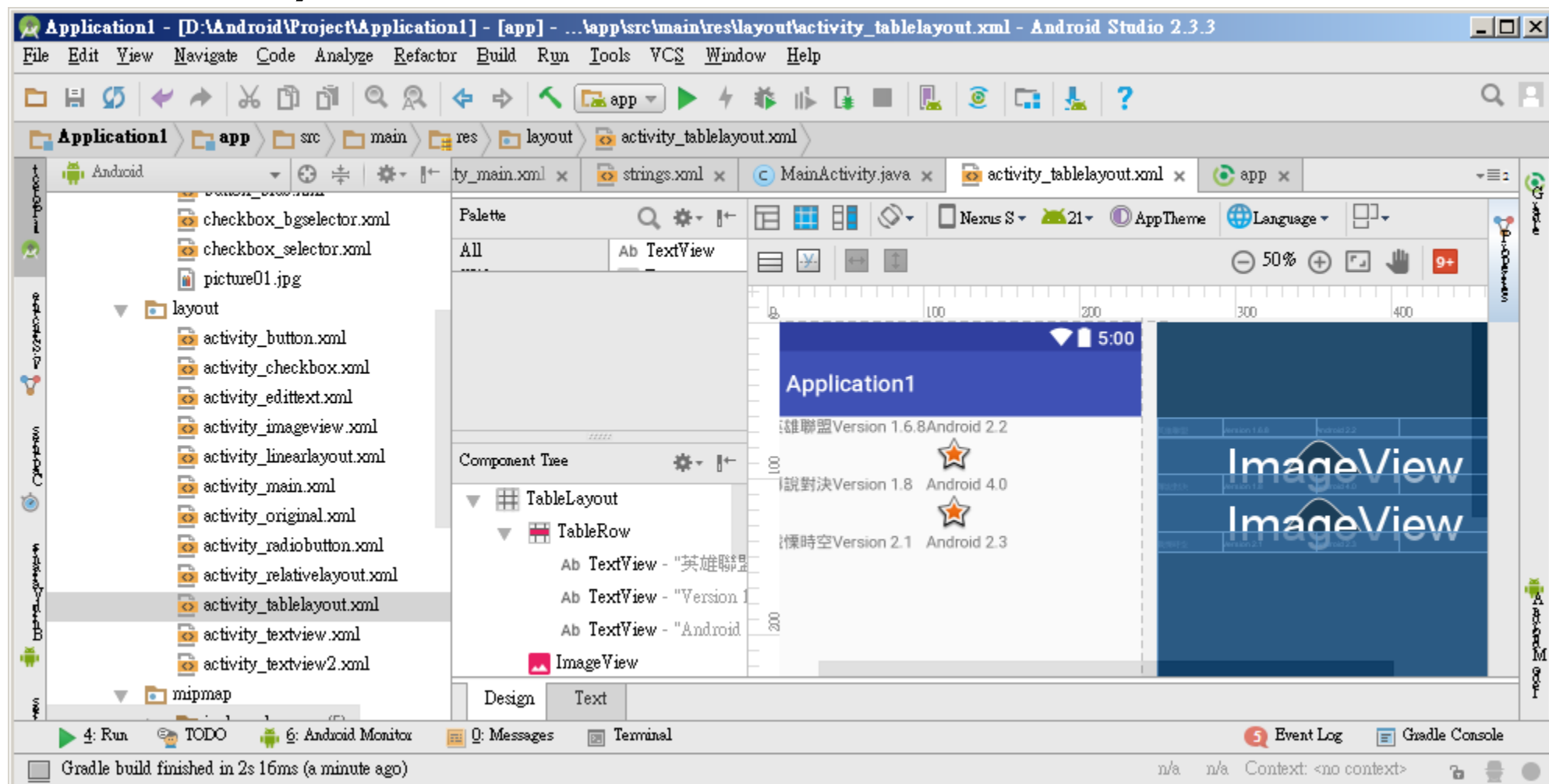
UI 使用者介面設計與規劃

- 請嘗試畫出範例程式的 `TableLayout` 樹狀結構圖



ANDROID 基礎介面元件

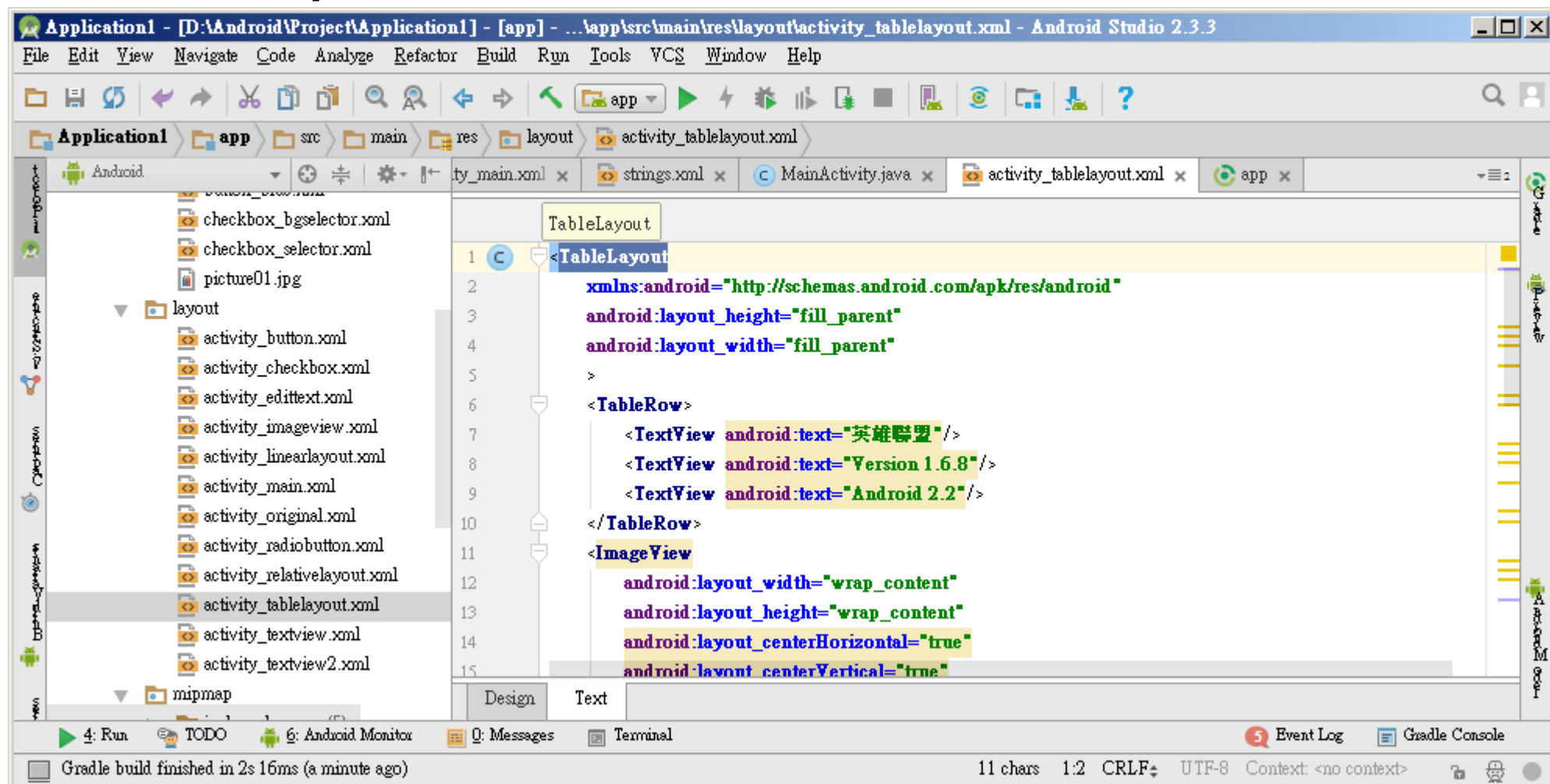
- TableLayout - 視覺化工具





ANDROID 基礎介面元件

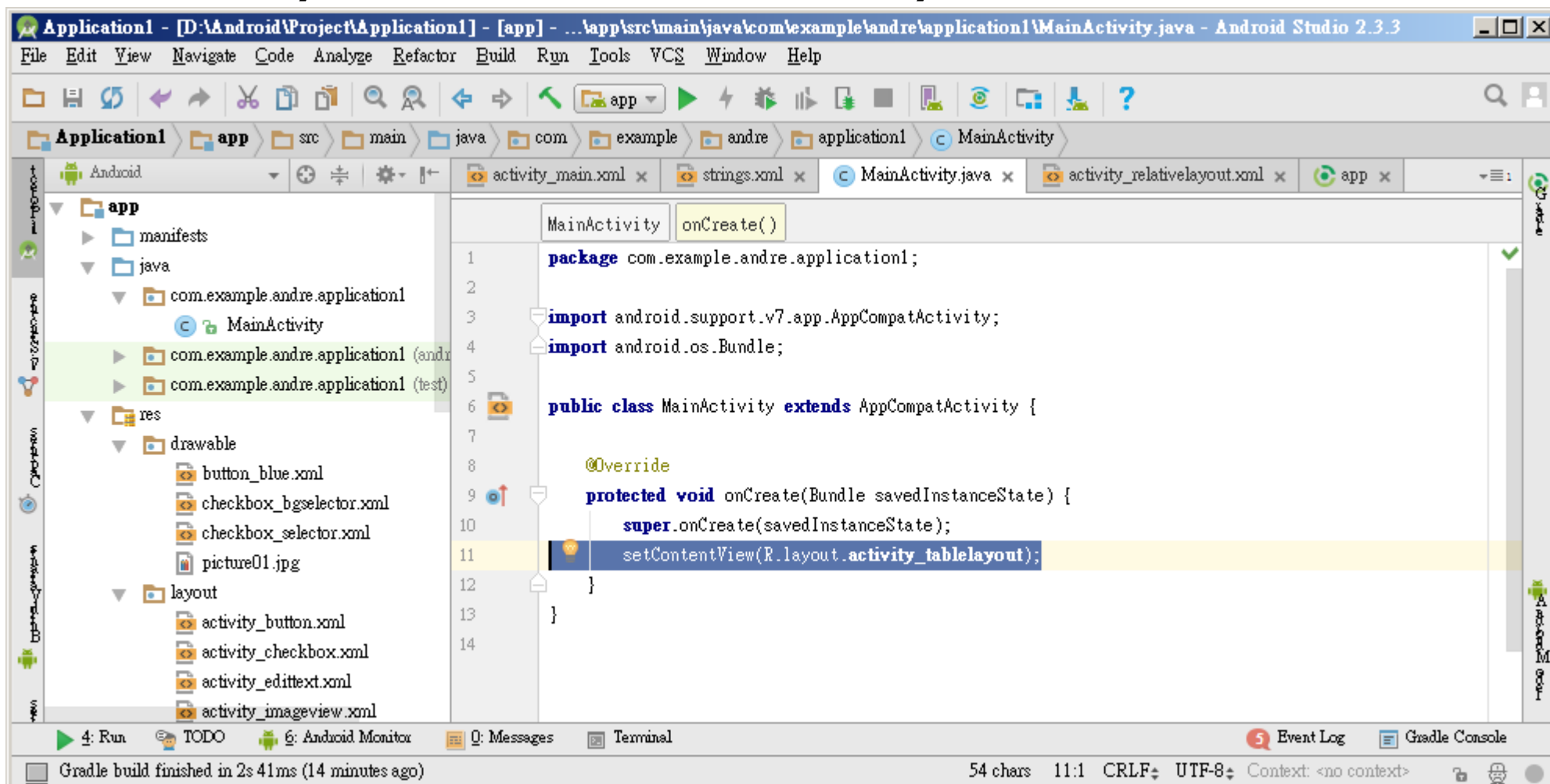
- TableLayout - 文字編輯器





ANDROID 基礎介面元件

- TableLayout - 切換 MainActivity 顯示





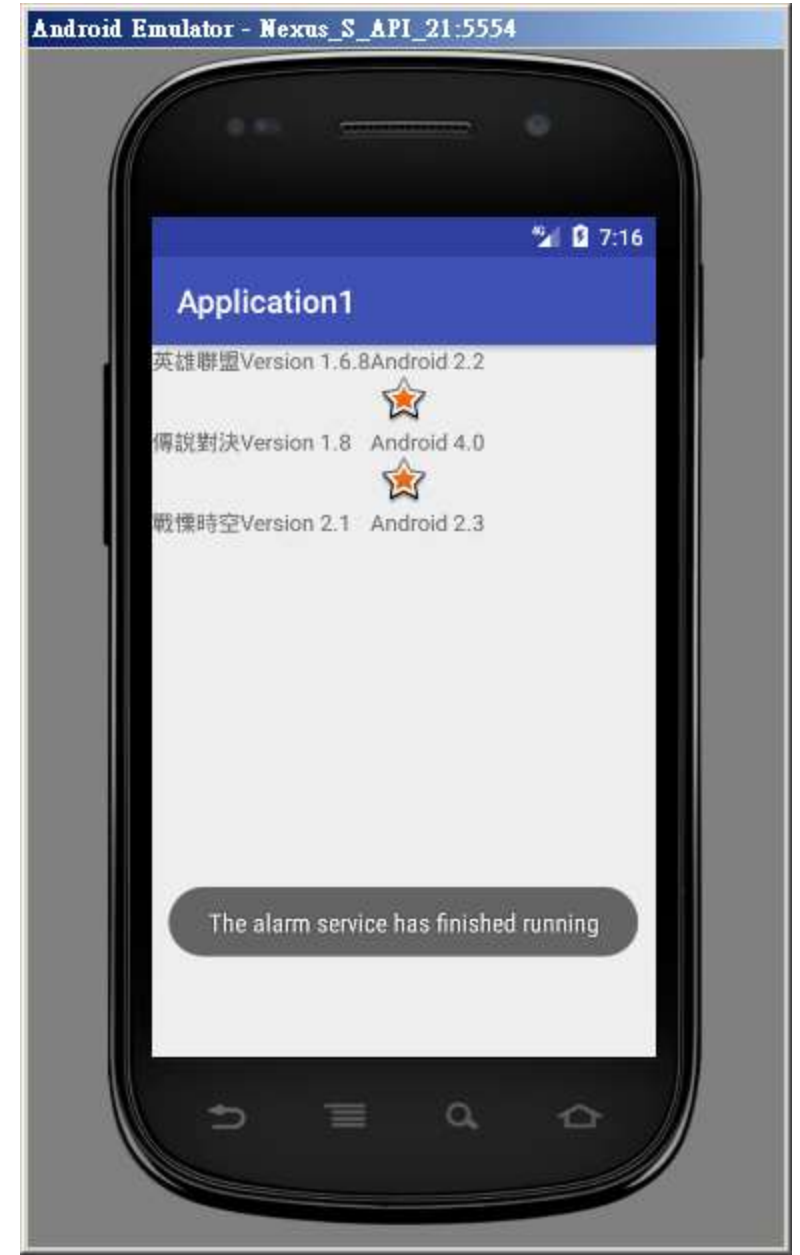
ANDROID 基礎介面元件

- **TableLayout** - 切換 MainActivity 顯示
- 開啟 MainActivity.java ，並於其中加入以下程式碼
 - `package com.example.andre.application I;`
 - `import android.support.v7.app.AppCompatActivity;`
 - `import android.os.Bundle;`
 - `public class MainActivity extends AppCompatActivity {`
 - `@Override`
 - `protected void onCreate(Bundle savedInstanceState) {`
 - `super.onCreate(savedInstanceState);`
 - `setContentView(R.layout.activity_tablelayout);`
 - `}`
 - `}`



ANDROID 基礎介面元件

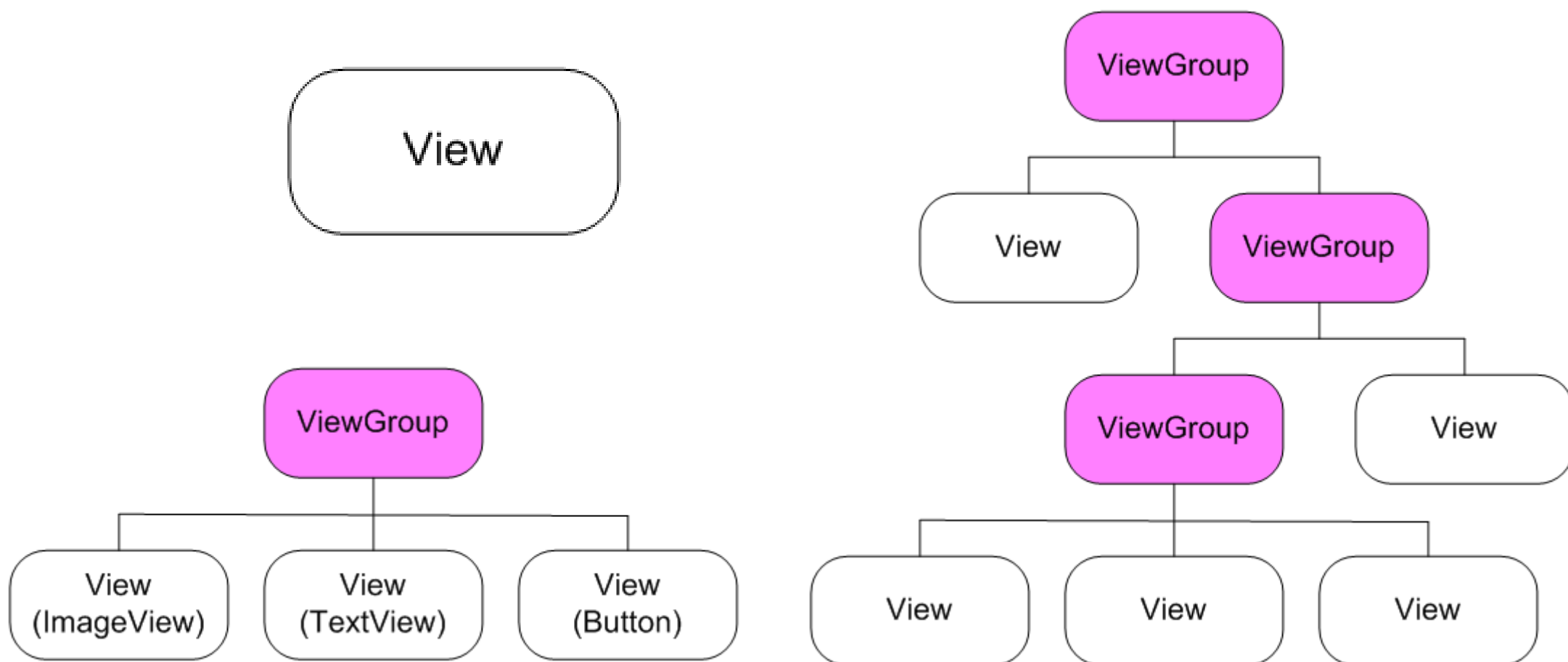
- TableLayout - 成果





UI 使用者介面設計與規劃

- Layout、ViewGroup、View 組合使用





UI 使用者介面設計與規劃

- **FrameLayout 框架佈局**
- 開啟 `activity_framelayout.xml`，並於其中加入以下程式碼
 - `<FrameLayout`
`xmlns:android="http://schemas.android.com/apk/res/android"`
 - `android:layout_width="match_parent"`
 - `android:layout_height="match_parent"`
 - `android:foreground="@drawable/android_logo"`
 - `android:foregroundGravity="right|bottom">`
 - `<TextView`
 - `android:layout_width="280dp"`
 - `android:layout_height="280dp"`
 - `android:layout_gravity="center"`
 - `android:background="#FFB5C5" />`



UI 使用者介面設計與規劃

- **FrameLayout 框架佈局**
- 開啟 `activity_framelayout.xml` ， 續
 - `<TextView`
 - `android:layout_width="220dp"`
 - `android:layout_height="220dp"`
 - `android:layout_gravity="center"`
 - `android:background="#FFD700" />`
 - `<TextView`
 - `android:layout_width="160dp"`
 - `android:layout_height="160dp"`
 - `android:layout_gravity="center"`
 - `android:background="#7BFE00" />`



UI 使用者介面設計與規劃

- **FrameLayout 框架佈局**
- 開啟 `activity_framelayout.xml` .xml ， 續
 - `<TextView`
 - `android:layout_width="100dp"`
 - `android:layout_height="100dp"`
 - `android:layout_gravity="center"`
 - `android:background="#FFFF00" />`
 - `<TextView`
 - `android:layout_width="40dp"`
 - `android:layout_height="40dp"`
 - `android:layout_gravity="center"`
 - `android:background="#000000" />`



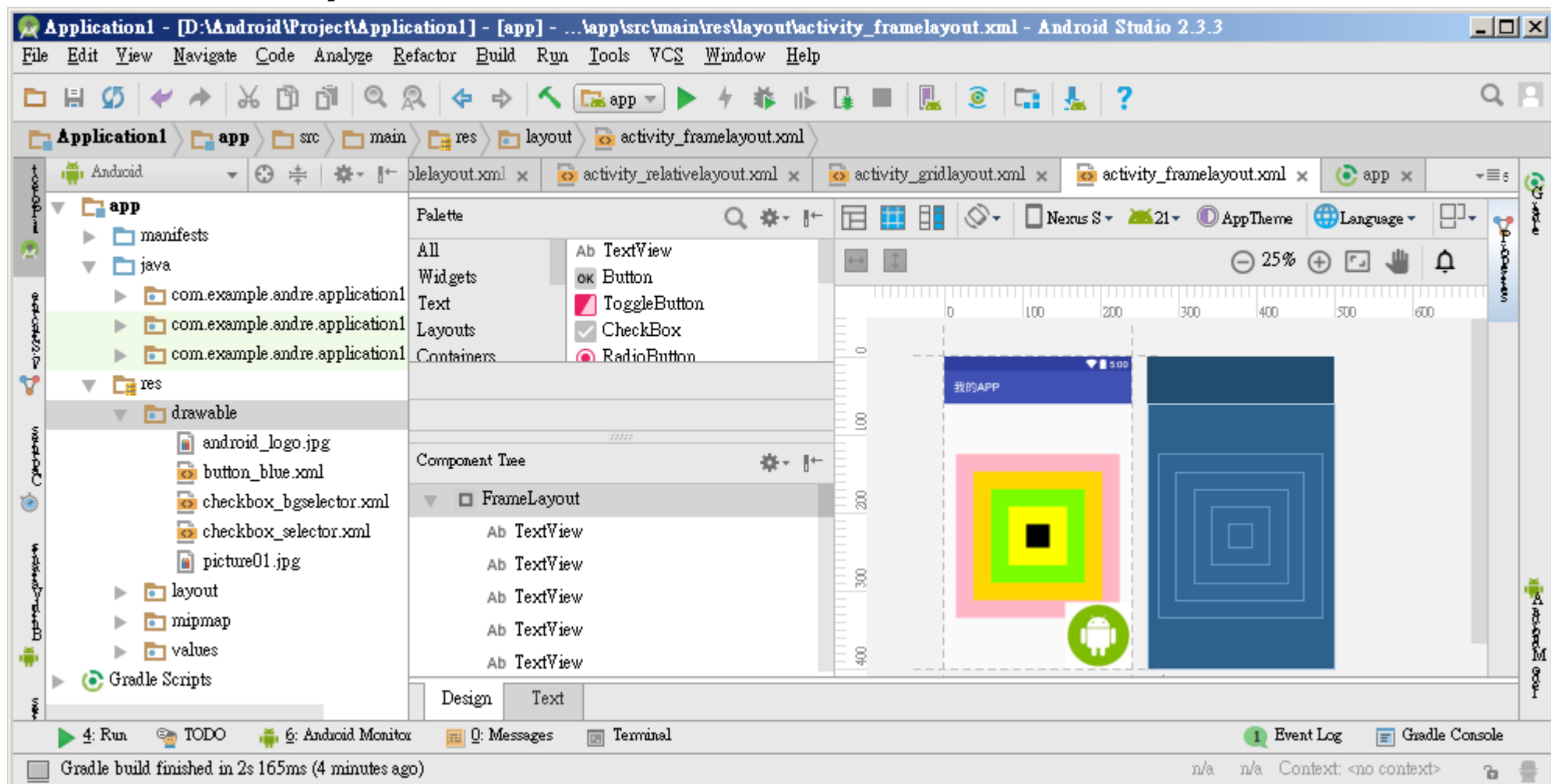
UI 使用者介面設計與規劃

- `FrameLayout` 框架佈局 - 常用屬性
 - `android:foreground`
 - 設置前景圖示
 - `android:foregroundGravity`
 - 設置前景圖示顯示位置



UI 使用者介面設計與規劃

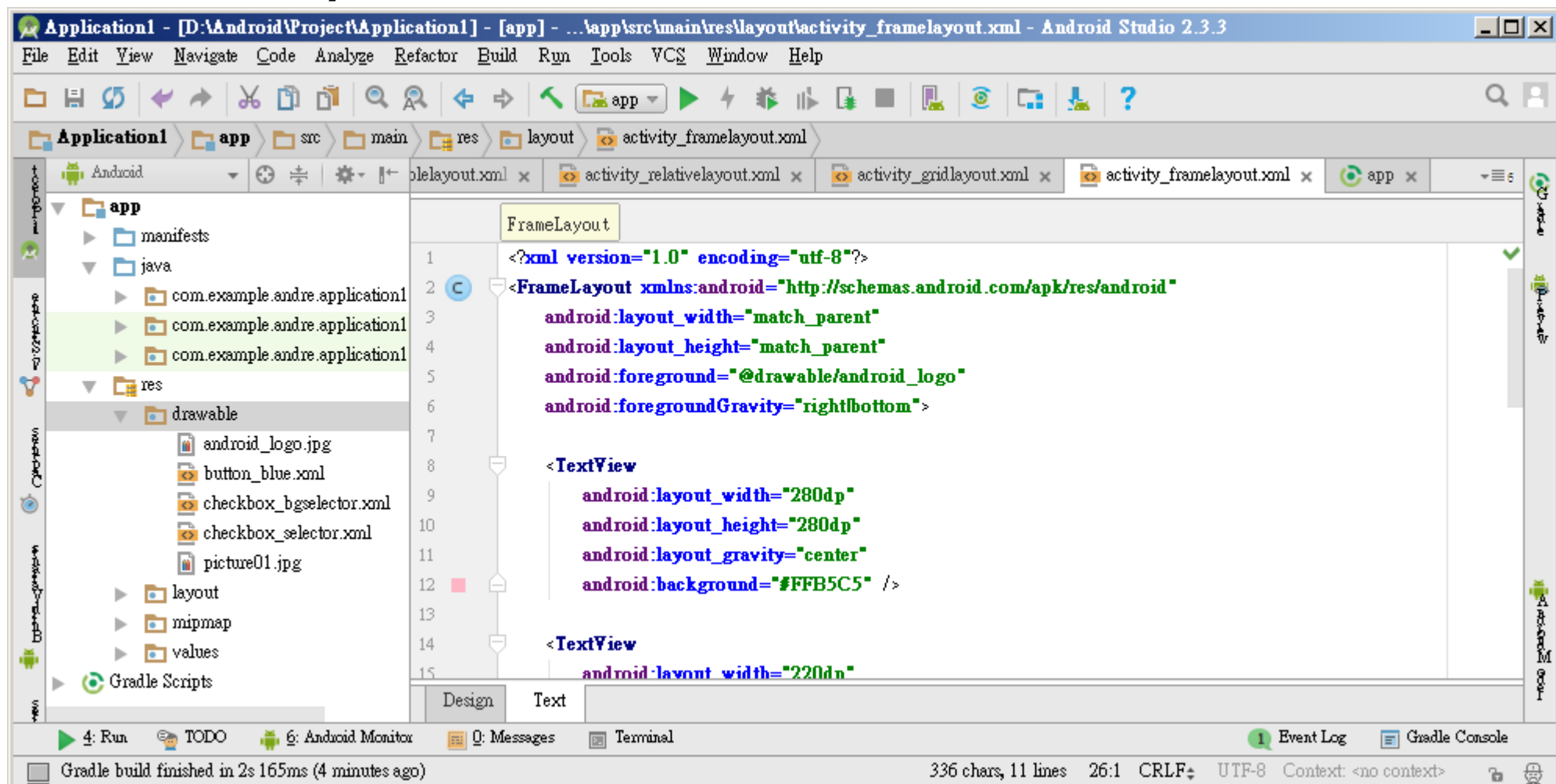
- FrameLayout 框架佈局 - 視覺化工具





UI 使用者介面設計與規劃

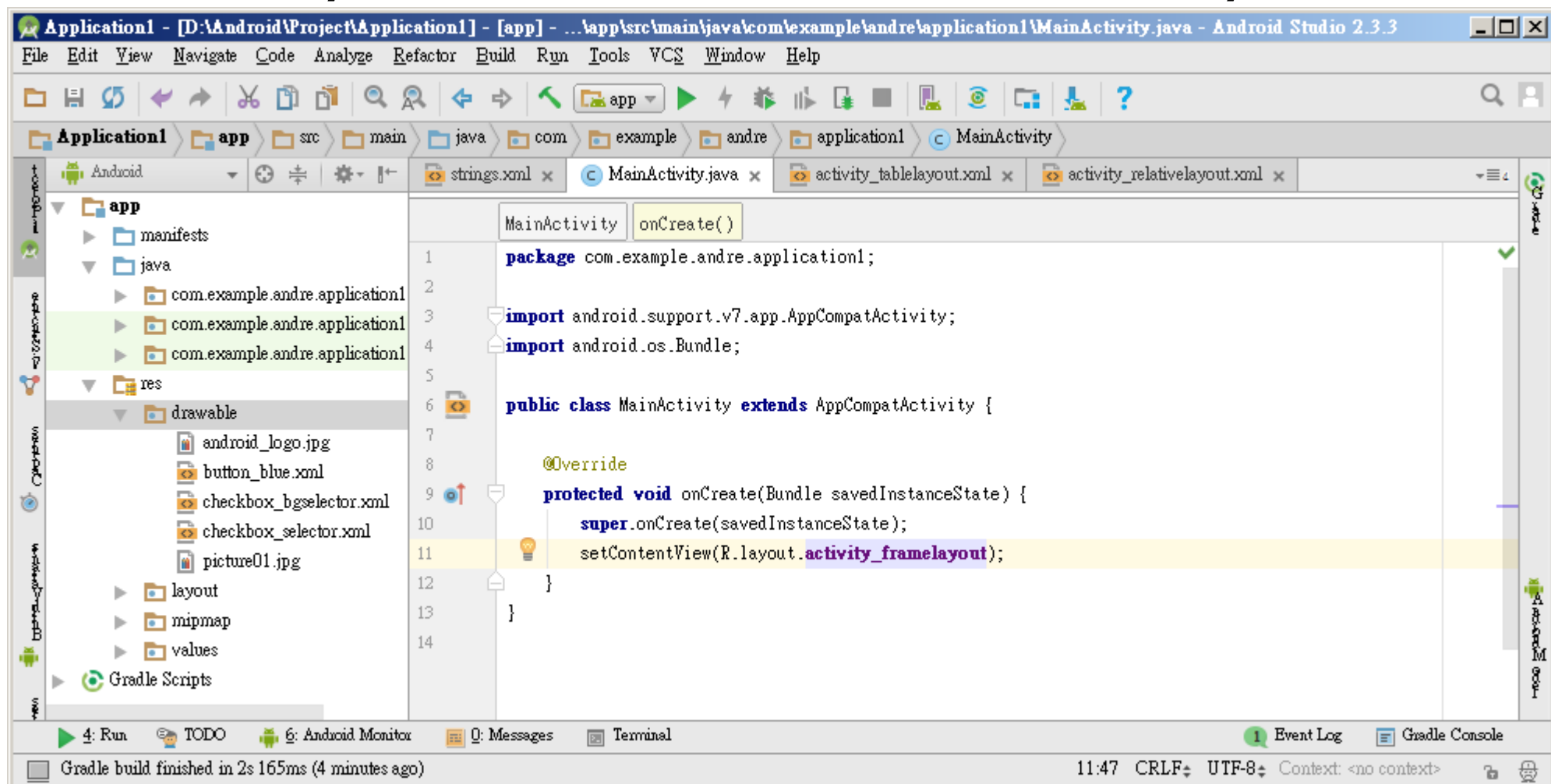
- **FrameLayout** 框架佈局-文字編輯器





UI 使用者介面設計與規劃

- **FrameLayout** 框架佈局-切換 MainActivity顯示





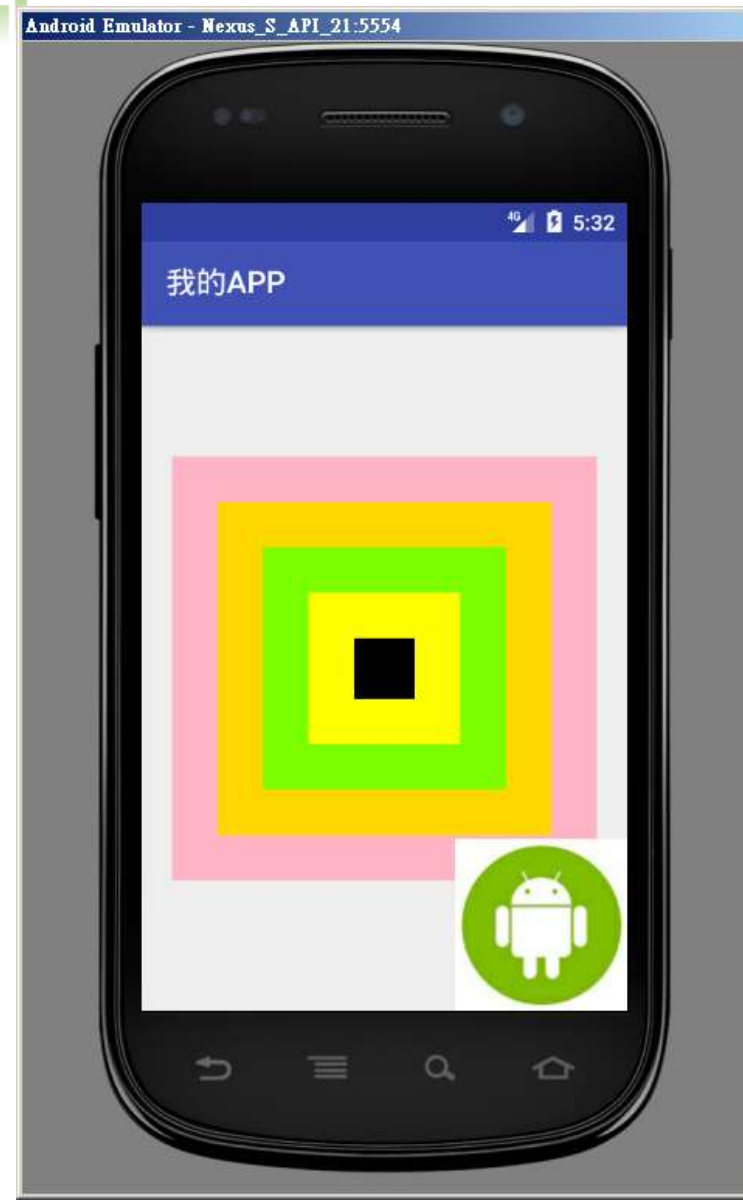
UI 使用者介面設計與規劃

- **FrameLayout** 框架佈局 - 切換 **MainActivity**顯示
- 開啟 **MainActivity.java** ，並於其中加入以下程式碼
 - `package com.example.andre.application I;`
 - `import android.support.v7.app.AppCompatActivity;`
 - `import android.os.Bundle;`
 - `public class MainActivity extends AppCompatActivity {`
 - `@Override`
 - `protected void onCreate(Bundle savedInstanceState) {`
 - `super.onCreate(savedInstanceState);`
 - `setContentView(R.layout.activity_activity_framelayout);`
 - `}`
 - `}`



UI 使用者介面設計與規劃

- FrameLayout - 成果





UI 使用者介面設計與規劃

- 請嘗試以 `FrameLayout` 作出以下畫面，30分鐘





UI 使用者介面設計與規劃

- GridLayout 網格佈局
- 開啟 activity_gridlayout.xml ，並於其中加入以下程式碼
 - `<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"`
 - `xmlns:tools="http://schemas.android.com/tools"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:columnCount="4"`
 - `android:rowCount="3">`
 - `<TextView`
 - `android:id="@+id/tv_show"`
 - `android:layout_width="fill_parent"`



UI 使用者介面設計與規劃

- GridLayout 網格佈局
- 開啟 activity_gridlayout.xml ， 續
 - `android:layout_height="fill_parent"`
 - `android:layout_columnSpan="4"`
 - `android:layout_gravity="fill"`
 - `android:layout_marginLeft="5dp"`
 - `android:layout_marginRight="5dp"`
 - `android:layout_marginTop="8dp"`
 - `android:background="#EEEEEE0"`
 - `android:paddingLeft="8dp"`
 - `android:text="0"`
 - `android:textSize="50sp" />`



UI 使用者介面設計與規劃

- GridLayout 網格佈局
- 開啟 activity_gridlayout.xml ， 續
 - `<Button`
 - `android:id="@+id/btn_cal_1"`
 - `android:layout_width="80dp"`
 - `android:text="1" />`
 - `<Button`
 - `android:id="@+id/btn_cal_2"`
 - `android:layout_width="80dp"`
 - `android:text="2" />`
 - `<Button`
 - `android:id="@+id/btn_cal_3"`
 - `android:layout_width="80dp"`
 - `android:text="3" />`



UI 使用者介面設計與規劃

- GridLayout 網格佈局
- 開啟 activity_gridlayout.xml ， 續
 - <Button
 - android:id="@+id/btn_cal_0"
 - android:layout_columnSpan="2"
 - android:layout_gravity="fill"
 - android:text="0" />
 - <Button
 - android:id="@+id/btn_equal"
 - android:layout_columnSpan="2"
 - android:layout_gravity="fill"
 - android:text="=" />



UI 使用者介面設計與規劃

- GridLayout 網格佈局-常用屬性
 - android:columnCount
 - 最大列數
 - android:rowCount
 - 最大行數
 - android:orientation
 - GridLayout 中，子元素的佈局方向
 - horizontal - 水平
 - Vertical - 垂直佈局



UI 使用者介面設計與規劃

- GridLayout網格佈局-常用屬性
- android:layout_column
 - 顯示於列位置
 - android:layout_column="0"，在第1列顯示
 - android:layout_column="1"，在第2列顯示
- android:layout_columnSpan
 - 跨列顯示，設定跨列數。
 - android:layout_columnSpan="2"，跨2列顯示
- android:layout_row
 - 顯示於行位置
 - android:layout_row="0"，在第1行顯示
 - android:layout_row="1"，在第2行顯示



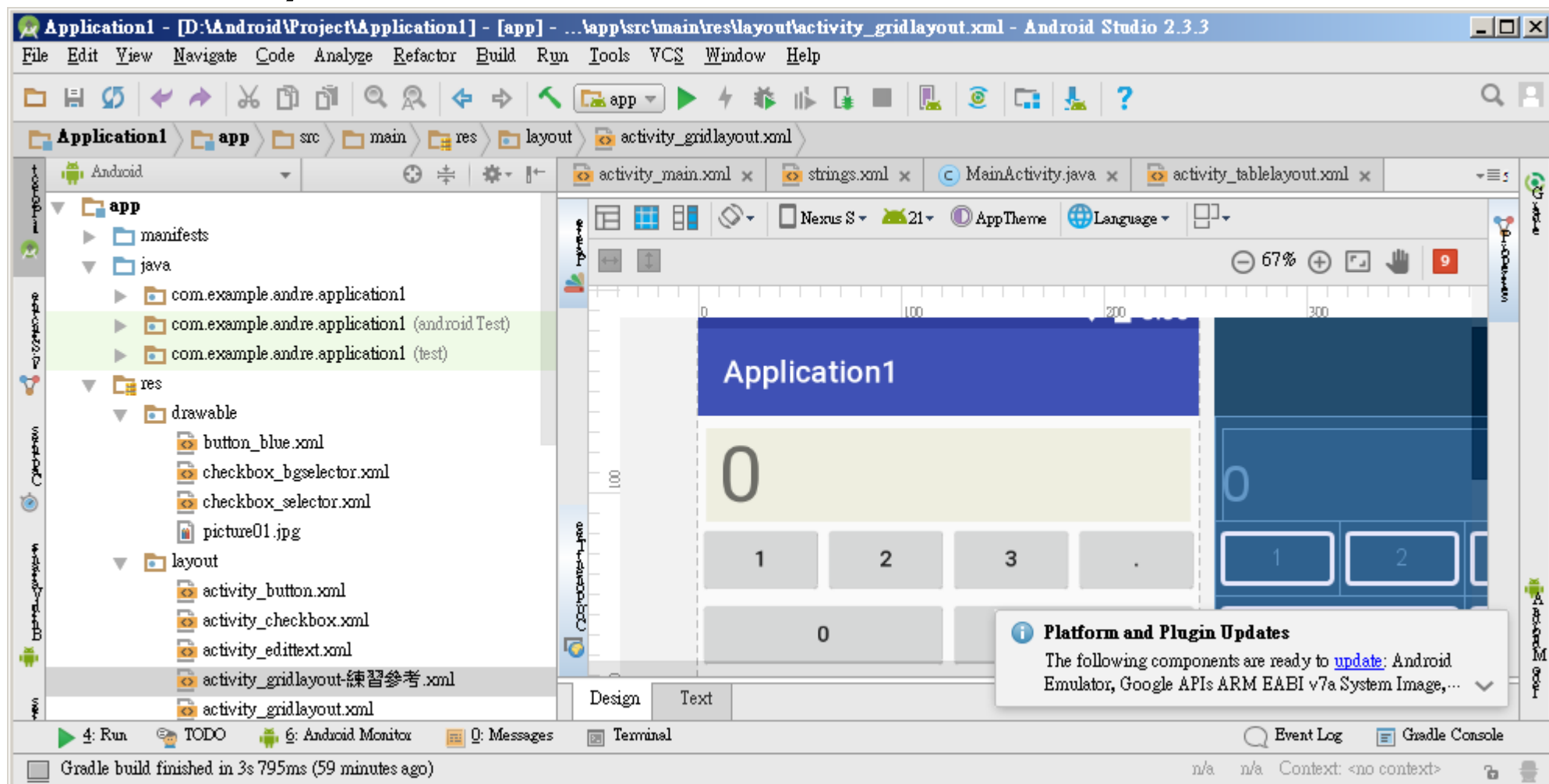
UI 使用者介面設計與規劃

- GridLayout網格佈局-常用屬性
- android:layout_rowSpan
 - 跨行顯示，設定跨行數
 - android:layout_rowSpan="2"，跨2行顯示
- android:layout_gravity
 - top - 置於容器頂部，不改變元件的大小
 - bottom- 置於容器底部，不改變元件的大小
 - left - 置於容器左邊，不改變元件的大小
 - right- 置於容器右邊，不改變元件的大小
 - center - 置於容器中間，不改變元件的大小
 - fill_vertical- 往垂直方向延伸該元件
 - fill_horizontal- 往水平方向延伸該元件
 - fill- 往水平、垂直方向延伸該元件



UI 使用者介面設計與規劃

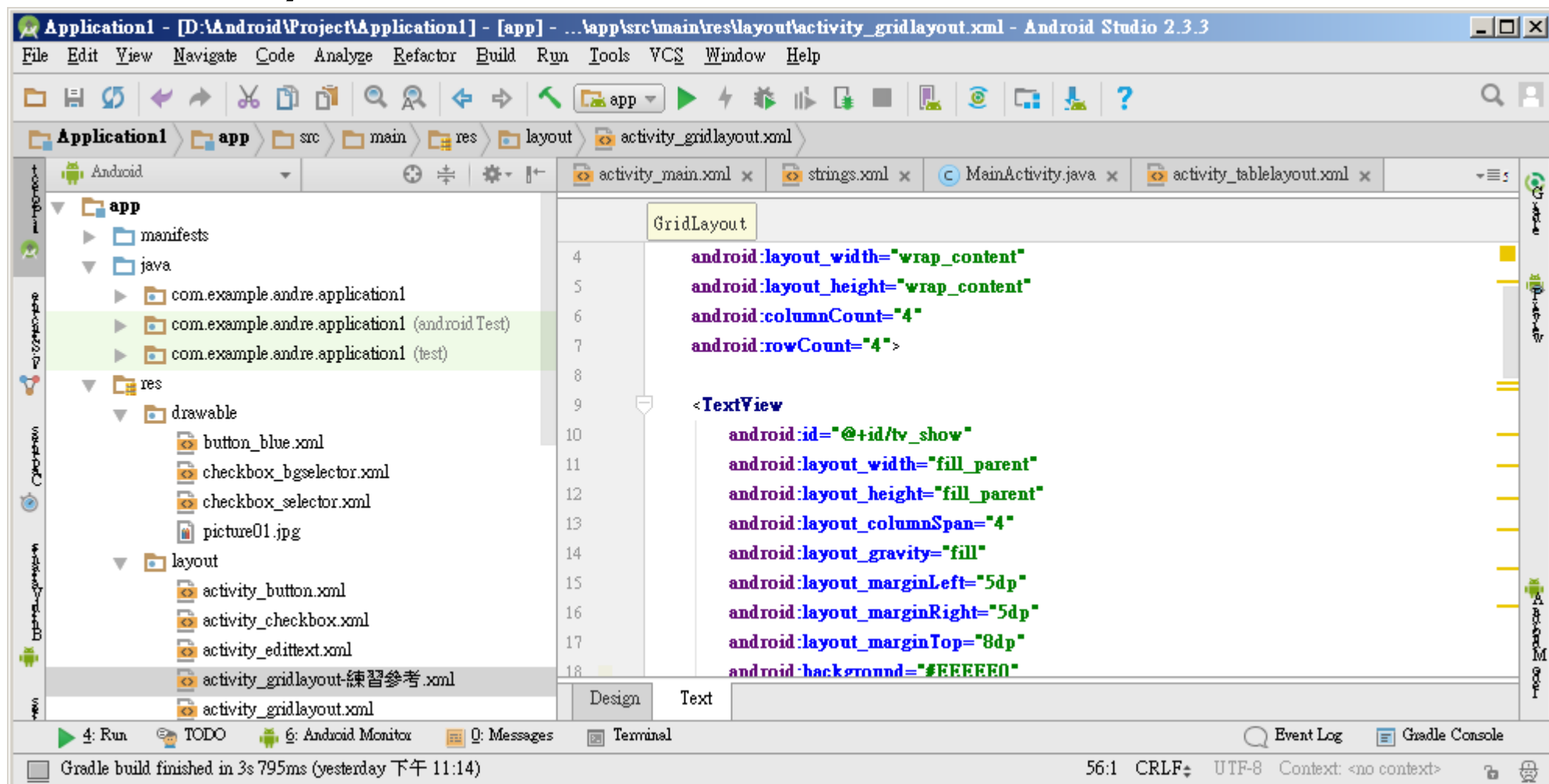
• GridLayout 網格佈局 - 視覺化工具





UI 使用者介面設計與規劃

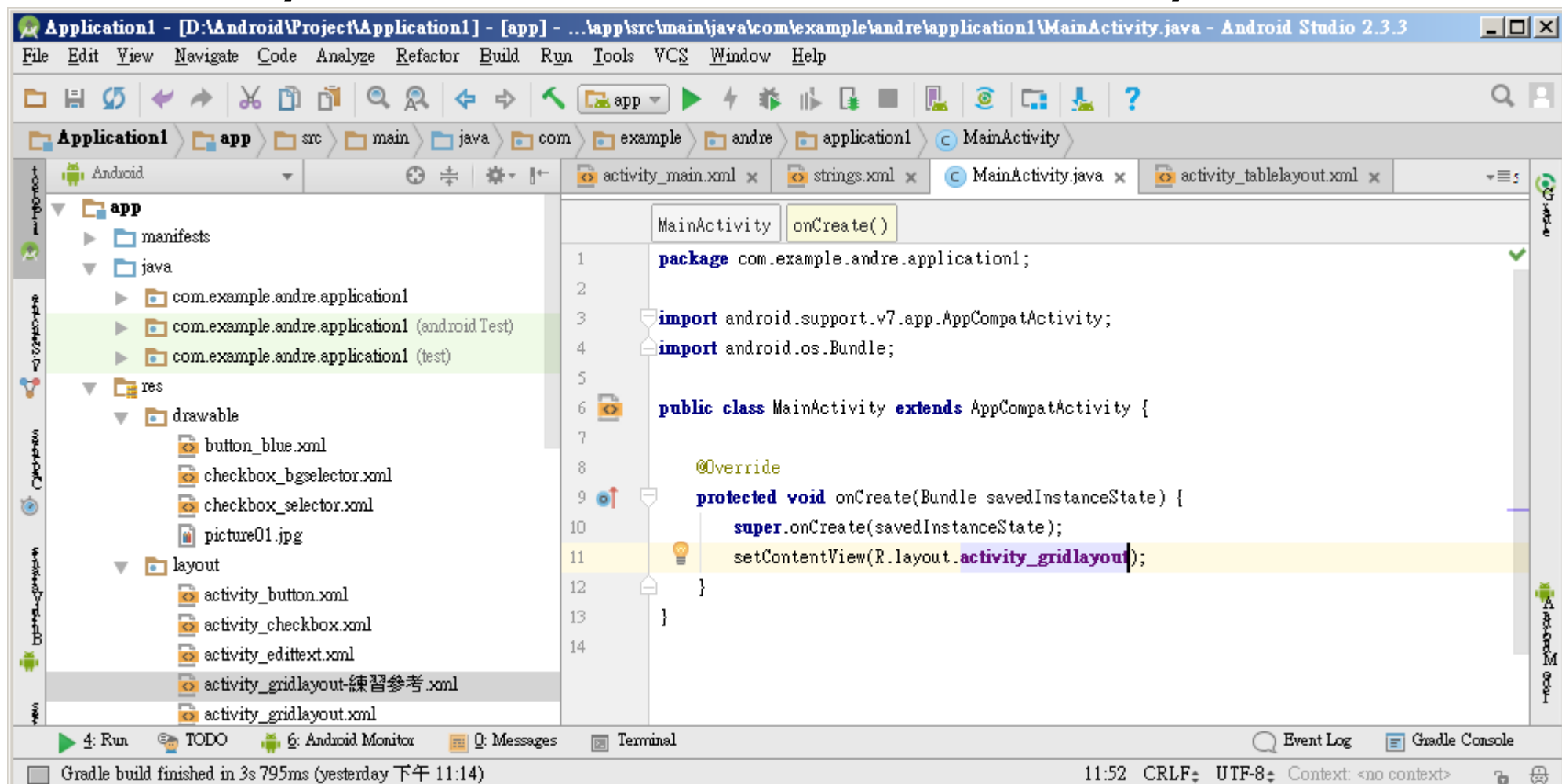
- GridLayout 網格佈局-文字編輯器





UI 使用者介面設計與規劃-

- GridLayout 網格佈局-切換 MainActivity顯示





UI 使用者介面設計與規劃

- GridLayout 網格佈局- 切換 MainActivity顯示
- 開啟 MainActivity.java ，並於其中加入以下程式碼
 - package com.example.andre.application I;
 - import android.support.v7.app.AppCompatActivity;
 - import android.os.Bundle;
 - public class MainActivity extends AppCompatActivity {
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_activity_gridlayout);
 - }
 - }



UI 使用者介面設計與規劃

- GridLayout - 成果





UI 使用者介面設計與規劃

- 請嘗試以 GridLayout 作出以下畫面，30分鐘





UI 使用者介面設計與規劃

- **ConstraintLayout 限制佈局**
 - 2016年的Google I/O 大会中，Google 呈現了全新的佈局元件 ConstraintLayout
 - 使用LinearLayout與RelativeLayout能夠實作出絕大部份的版面，但Layout層級太多會造成較耗費資源，啟動速度慢
 - ConstraintLayout元件能將複雜的版面設計的層級變的較為扁平化





UI 使用者介面設計與規劃

- ConstraintLayout 限制佈局
- 開啟activity_constraintlayout.xml，並於其中加入以下程式碼
 - android:id="@+id/constraintLayout">
 - <Button
 - android:id="@+id/button_cancel"
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:layout_marginBottom="16dp"
 - android:layout_marginStart="16dp"
 - android:text="取消"
 - app:layout_constraintBottom_toBottomOf="@+id/constraintLayout"
 - app:layout_constraintStart_toStartOf="@+id/constraintLayout" />



UI 使用者介面設計與規劃

- ConstraintLayout 限制佈局
- 開啟 activity_constraintlayout.xml ， 續
 - <Button
 - android:id="@+id/button_next"
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:layout_marginBottom="16dp"
 - android:layout_marginStart="16dp"
 - android:text="下一步"
 - app:layout_constraintBottom_toBottomOf="@+id/constraintLayout"
 - app:layout_constraintStart_toEndOf="@+id/button_cancel" />



UI 使用者介面設計與規劃

- ConstraintLayout 限制佈局
- 開啟 activity_constraintlayout.xml ， 續
 - <Button
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:id="@+id/button_center"
 - android:text="置中"
 - app:layout_constraintTop_toTopOf="parent"
 - app:layout_constraintBottom_toBottomOf="parent"
 - app:layout_constraintLeft_toLeftOf="parent"
 - app:layout_constraintRight_toRightOf="parent" />



UI 使用者介面設計與規劃

- ConstraintLayout 限制佈局
- 開啟 activity_constraintlayout.xml ， 續
 - <Button
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:id="@+id/button_bais"
 - android:text="左上置中"
 - app:layout_constraintTop_toTopOf="parent"
 - app:layout_constraintBottom_toBottomOf="parent"
 - app:layout_constraintStart_toStartOf="parent"
 - app:layout_constraintEnd_toEndOf="parent"
 - app:layout_constraintHorizontal_bias="0.25"
 - app:layout_constraintVertical_bias="0.25" />



UI 使用者介面設計與規劃

- ConstraintLayout 限制佈局
- 開啟 activity_constraintlayout.xml ， 續
 - `<android.support.constraint.Guideline`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:id="@+id/guideline"`
 - `android:orientation="vertical"`
 - `app:layout_constraintGuide_begin="200dp" />`



UI 使用者介面設計與規劃

- ConstraintLayout 限制佈局
- 開啟 activity_constraintlayout.xml ， 續
 - <Button
 - android:id="@+id/button_guidline"
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:text="對齊參考"
 - app:layout_constraintStart_toStartOf="@+id/guideline"
 - app:layout_constraintTop_toTopOf="parent"
 - app:layout_constraintBottom_toBottomOf="parent"
 - app:layout_constraintVertical_bias="0.25" />



UI 使用者介面設計與規劃

- **ConstraintLayout 限制佈局 - 常用屬性**
 - `app:layout_constraintBottom_toBottomOf`
 - 相對於底部
 - `app:layout_constraintLeft_toLeftOf`
 - 相對於左方
 - `app:layout_constraintRight_toRightOf`
 - 相對於右方
 - `app:layout_constraintTop_toTopOf`
 - 相對於上方
 - `app:layout_constraintHorizontal_bias`
 - 水平偏移



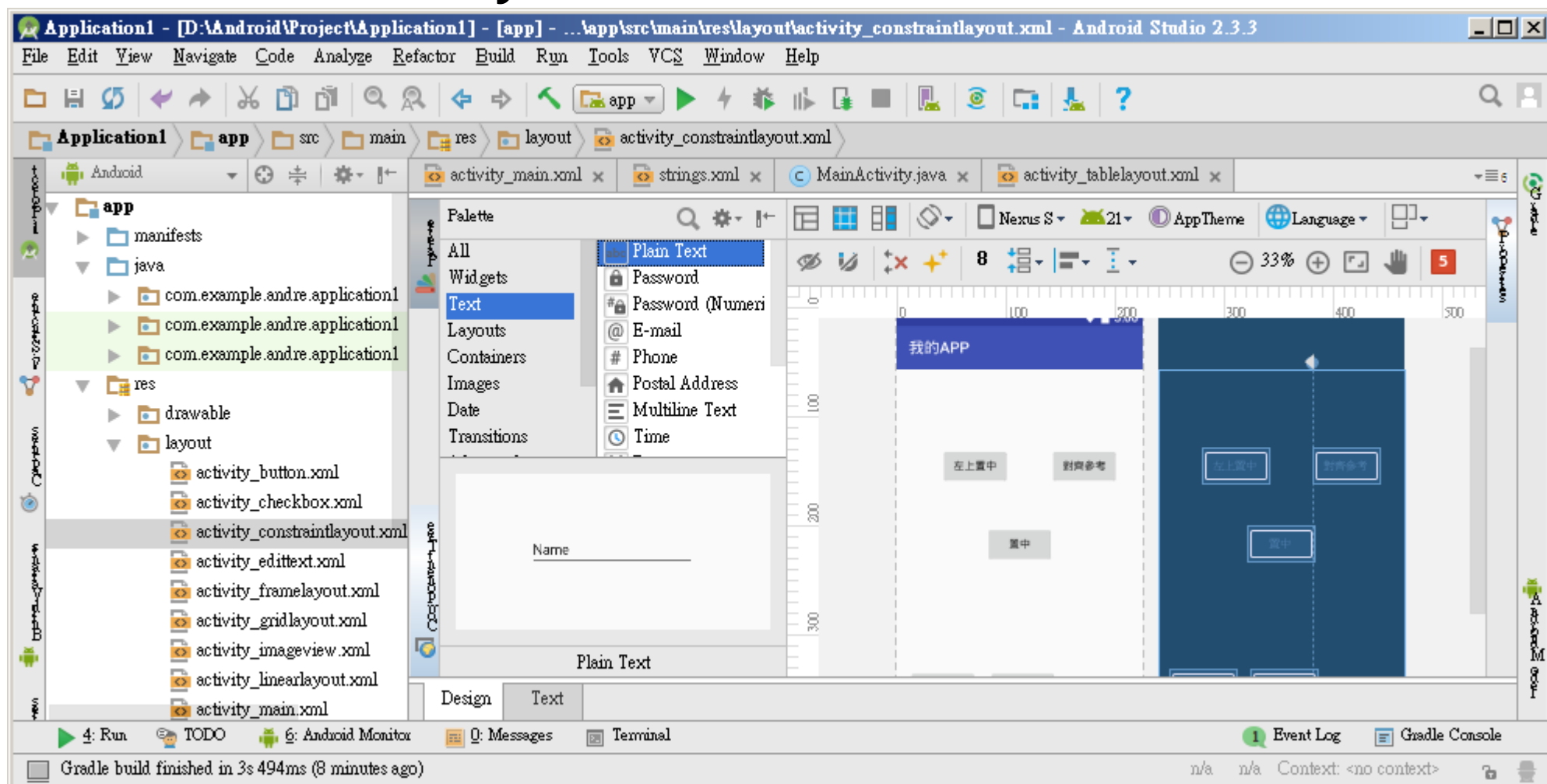
UI 使用者介面設計與規劃

- **ConstraintLayout 限制佈局 - 常用屬性**
 - `app:layout_constraintVertical_bias`
 - 垂直偏移
 - `app:layout_constraintHorizontal_bias`
 - 水平偏移
- **ConstraintLayout **Guideline** - 常用屬性**
 - `android:orientation="vertical"`
 - Vertical/垂直軸
 - Horizontal /水平軸
 - `app:layout_constraintGuide_begin="200dp"`
 - 偏移 200dp



UI 使用者介面設計與規劃

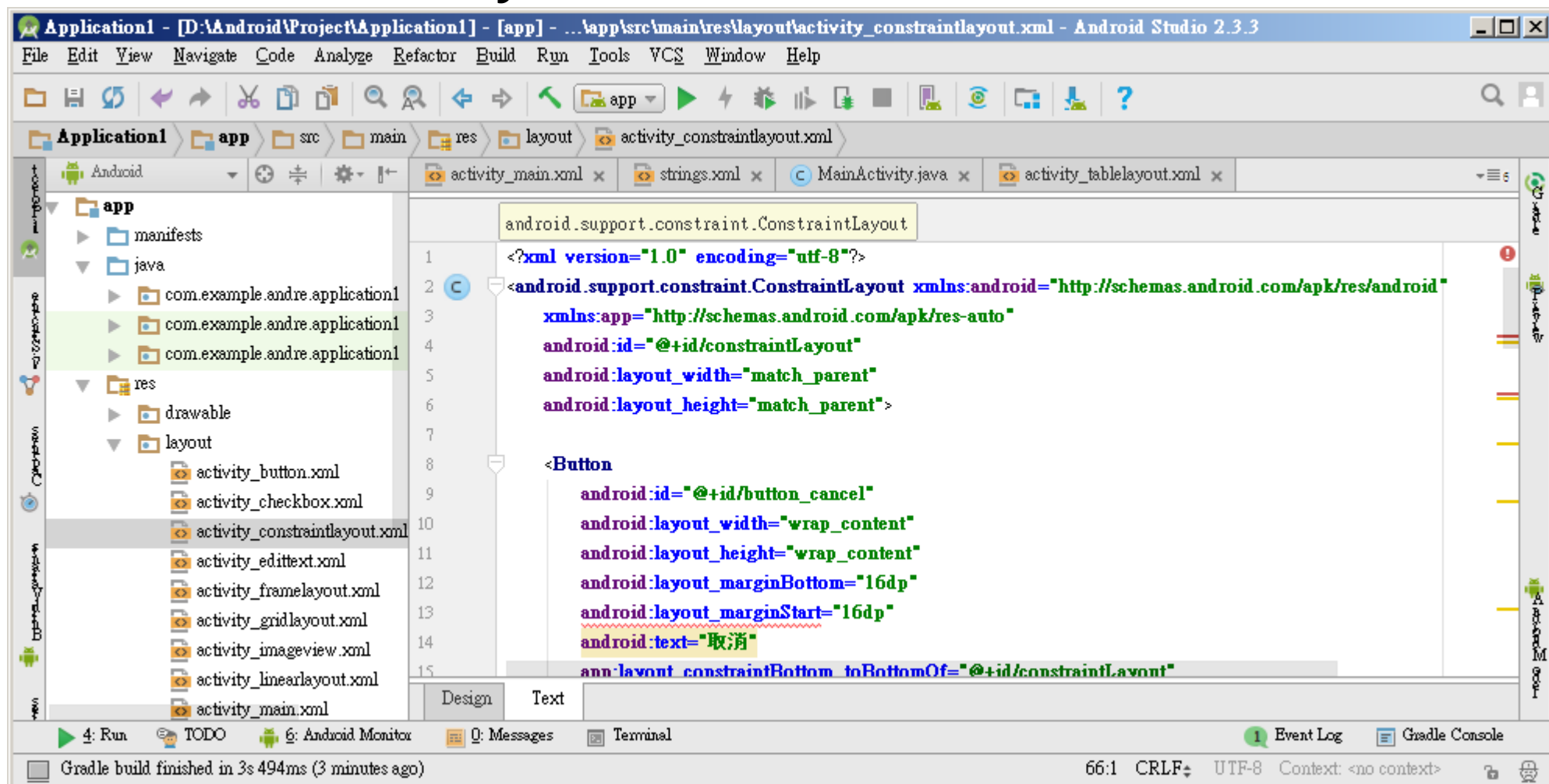
- ConstraintLayout 限制佈局 - 視覺化工具





UI 使用者介面設計與規劃

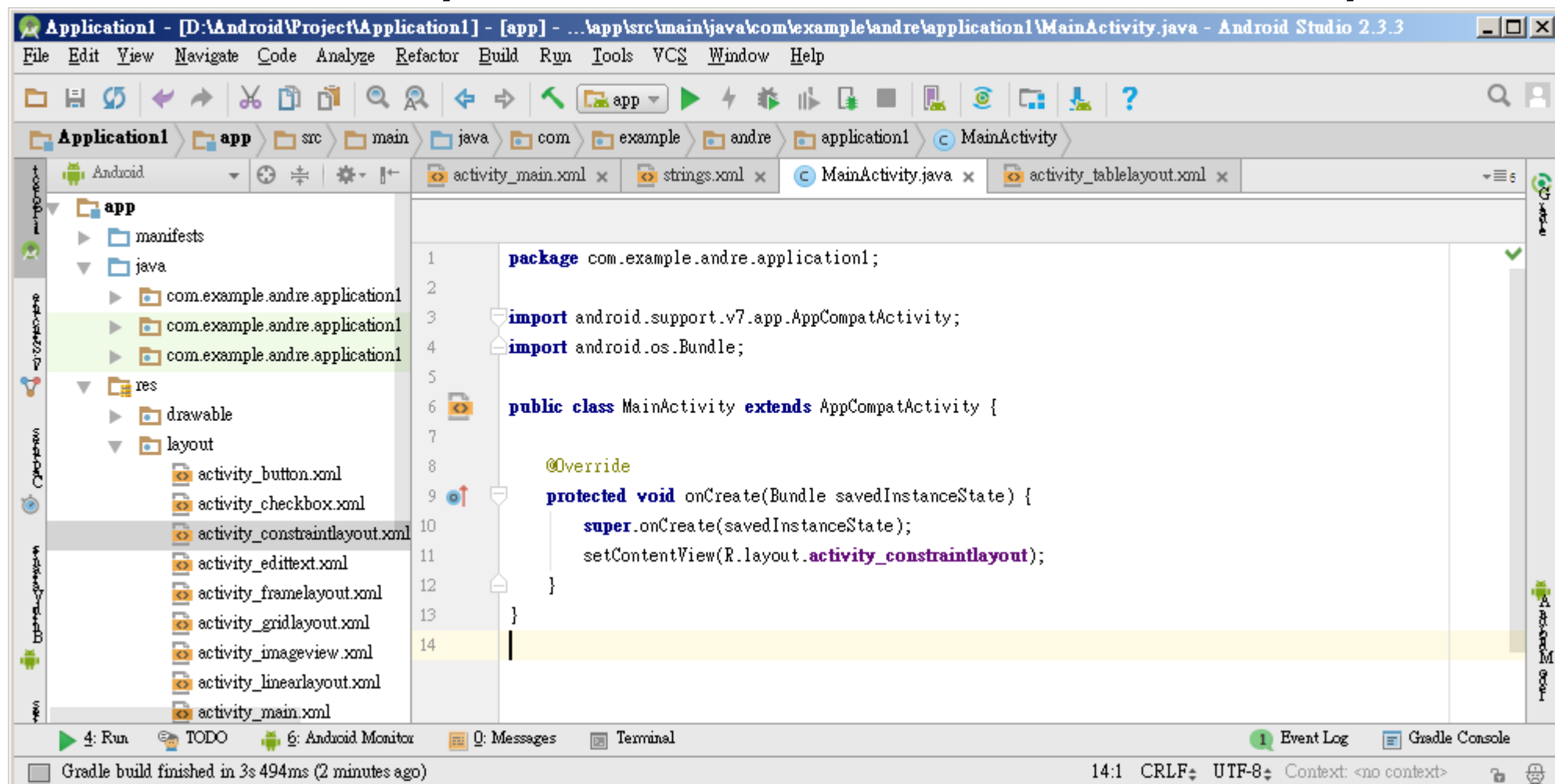
- ConstraintLayout 限制佈局-文字編輯器





UI 使用者介面設計與規劃

- ConstraintLayout 限制佈局-切換 MainActivity顯示





UI 使用者介面設計與規劃

- ConstraintLayout 限制佈局 - 切換 MainActivity 顯示
- 開啟 MainActivity.java ，並於其中加入以下程式碼
 - package com.example.andre.application I;
 - import android.support.v7.app.AppCompatActivity;
 - import android.os.Bundle;
 - public class MainActivity extends AppCompatActivity {
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_activity_constraintlayout);
 - }
 - }



UI 使用者介面設計與規劃

- ConstraintLayout - 成果





 **THE END**