Android程式設計

整存整付計算機

王昱景 Brian Wang brian.wang.frontline@gmail.com

1. 題目說明:

請開啓 COM.TQC.GDD01 專案,設計「整存整付計算機」應用程式。在輸入本金、年利率與存款期數後,按下「計算本利和」按鈕,即可在下方顯示本利和,請依下列題意完成作答。

Pa	□ □ 上午 2:14
整存整付計算機 整存整付試算:	A STATE OF THE PARTY OF THE PAR
本金(NT\$):	
年利率(%):	
存款期数:	
811	算本利和

<參考圖>

2. 設計說明:

- (1) 請開啓 main.xml 檔案,依參考圖利用 LinearLayout 製作出整存整付計 算機的主要畫面,並開啓 strings.xml 檔案,將字串變數值設定於此。
- (2) 在 strings.xml 檔案中,找到 app_name 將值修改為【整存整付計算機】。
- (3) 請設計一個 string 名稱為 title 的 TextView, string 值設定為【整存整付 試算:】。

- (4)請設計一個 string 名稱為 lend 的 TextView, string 值設定為【本金(NT\$):】。
- (5) 請設計一個變數名稱為 input_lend 的 EditText, 且限制輸入數值(整數)。
- (6) 請設計一個 string 名稱為 rate 的 TextView, string 值設定為【年利率(%):]。
- (7) 請設計一個變數名稱為 input_rate 的 EditText, 且限制輸入數值(整數)。
- (8) 請設計一個 string 名稱為 number 的 TextView, string 值設定為【存款期數:】。
- (9) 請設計一個變數名稱為 input_number 的 EditText, 且限制輸入數值(整數)。
- (10) 請設計一個 string 名稱為 btn 和變數名稱為 submit 的 Button, string 值設定為【計算本利和】。
- (11) 請設計一個變數名稱為 amount 的 TextView, text 值預設為空。
- (12) 於點選「計算本利和」的按鈕時,計算出本利和總數,顯示 【本利和為:xx】,將計算出的金額代入 xx 内。
 - ❖ 提示:整存整付公式:本利和=本金*(月利率 + 1)^存款期數。
- 3. 執行結果參考畫面:



4. 自行測試程式是否達成下列結果:

- (1) 如參考圖所示,畫面元件依 res/layout/main.xml 的定義正確顯示。
- (2) 開啟 strings.xml 檔案,將字串變數值設定於此。
- (3) 變數名稱為 input_lend、input_rate 及 input_number 的 EditText, 需限 制輸入數值,且數值為整數。
- (4) 變數名稱為 amount 的 TextView, text 值預設為空。
- (5) 點選「計算本利和」的按鈕,依公式計算出本利和總數。

5. 評分項目:

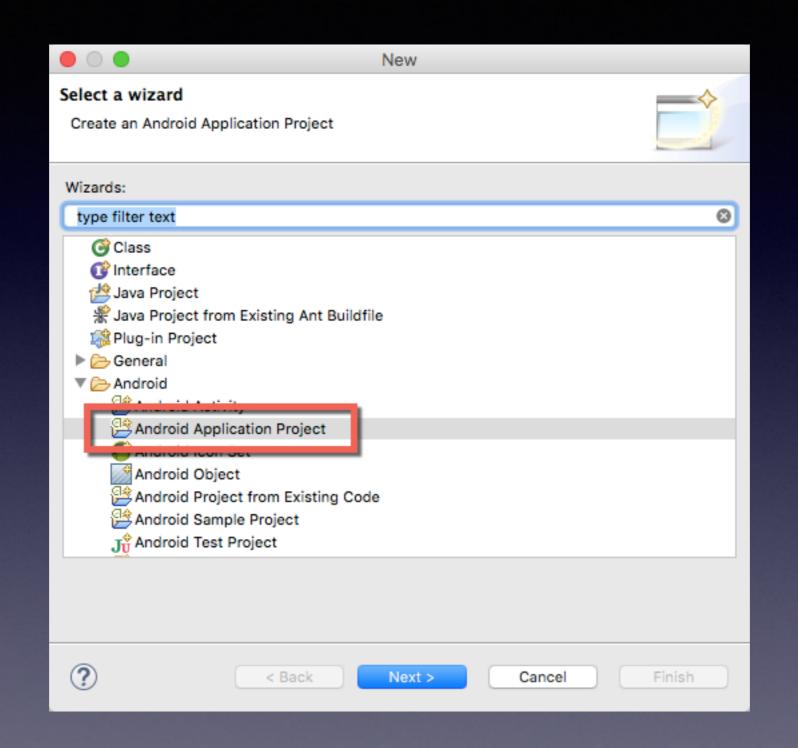
項		配	分	得	分
(1) 畫面元件依 res/layout/main.xml 的定義 示,各個物件及其屬性設定與參考圖相同		5			
(2) 字串變數設定於 strings.xml 檔案內		5			
(3) 變數名稱為 input_lend input_rate 及 input_的 EditText,需限制輸入數值,且數值為到		2			
(4) 變數名稱為 amount 的 TextView, text 值預	設為空	2	2		
(5) 點選「計算本利和」的按鈕,依公式正確 本利和總數	計算出	6			
總		2	0		

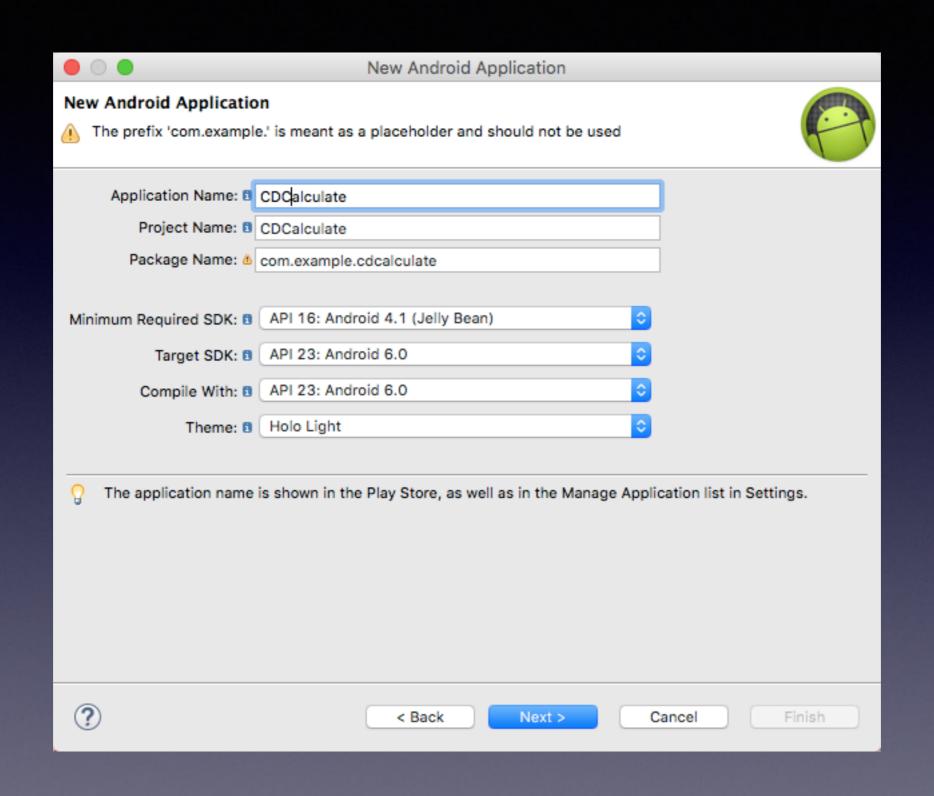
CDCalculate

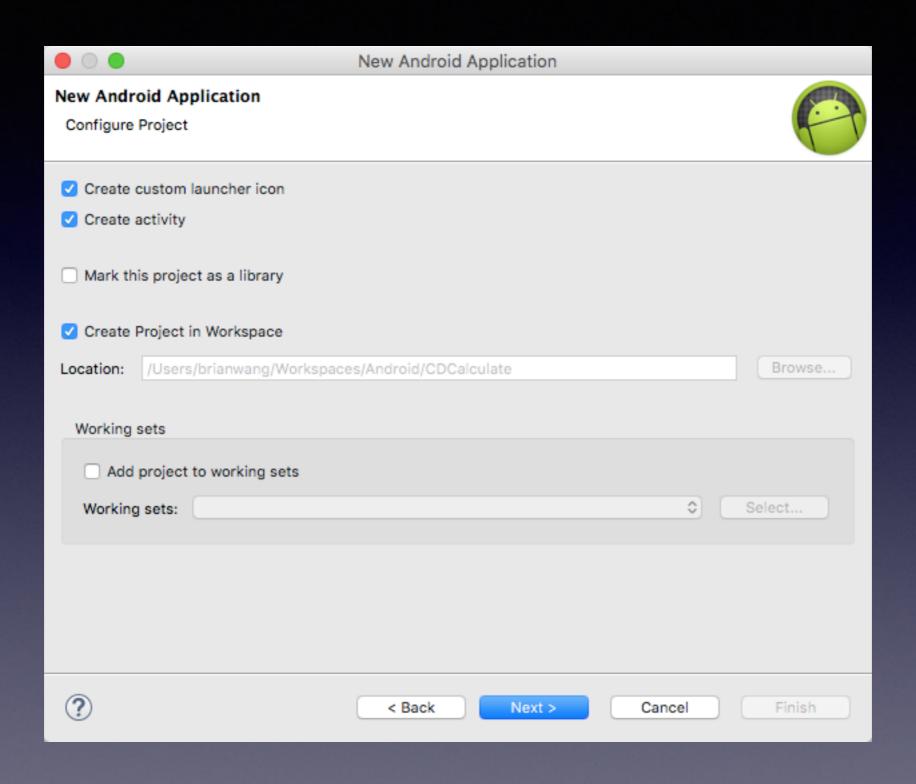
- 開啟和執行 Android 專案
- 建立元件的版面配置
- 建立 Activity 活動類別

開啟和執行 Android 專案

- 請啟動 Eclipse IDE
- 建立 Android 專案
 - Project Name: CDCalculate
 - Build Target: Android 4
 - Package Name: tw.edu.vnu.cdcalculate





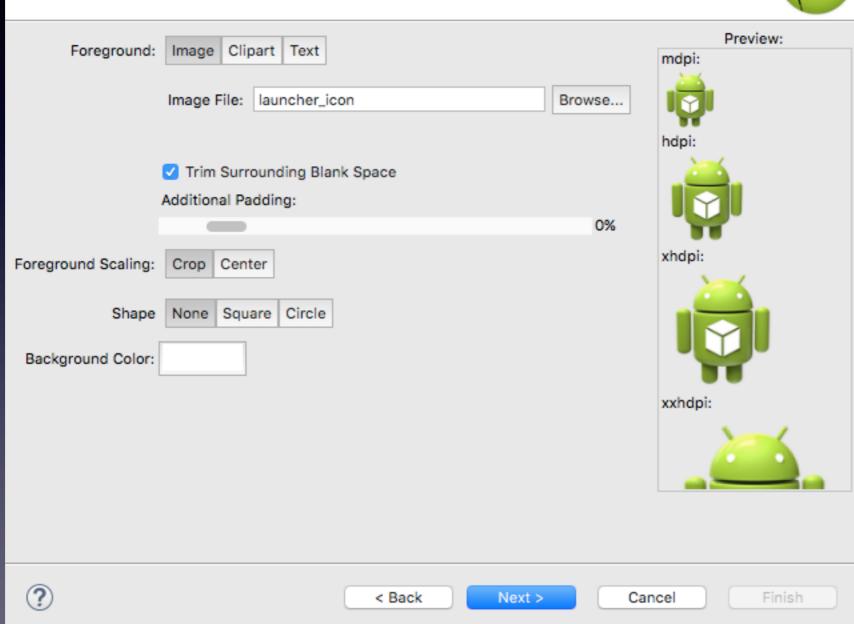


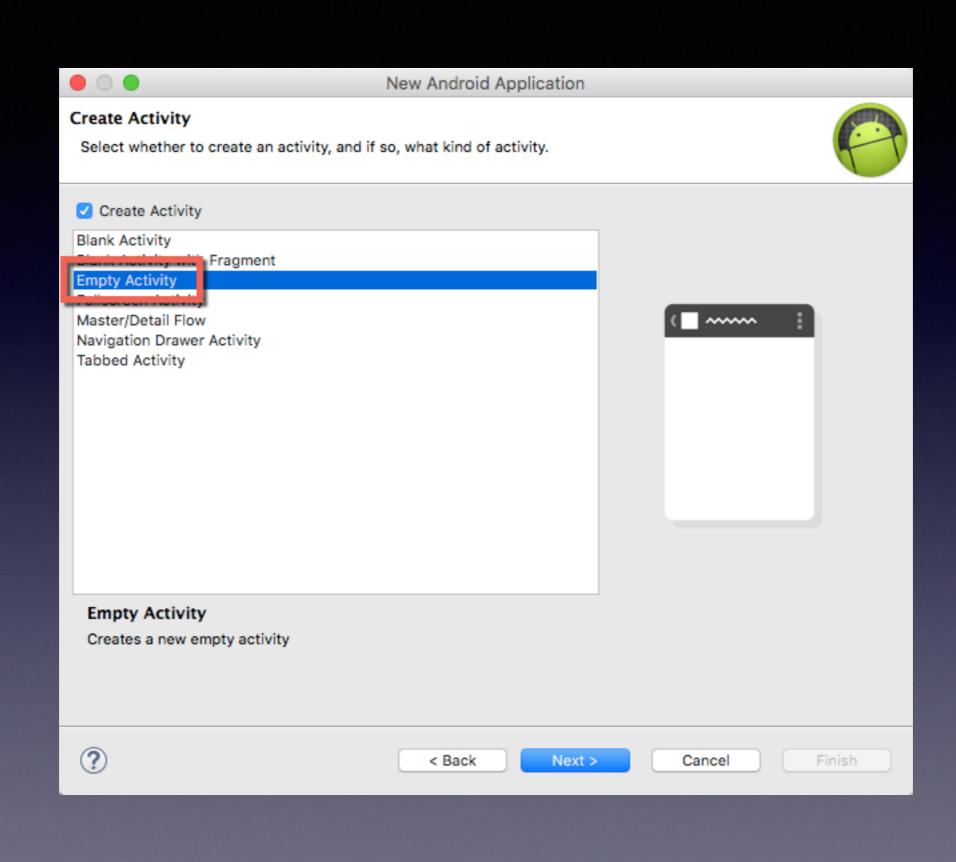


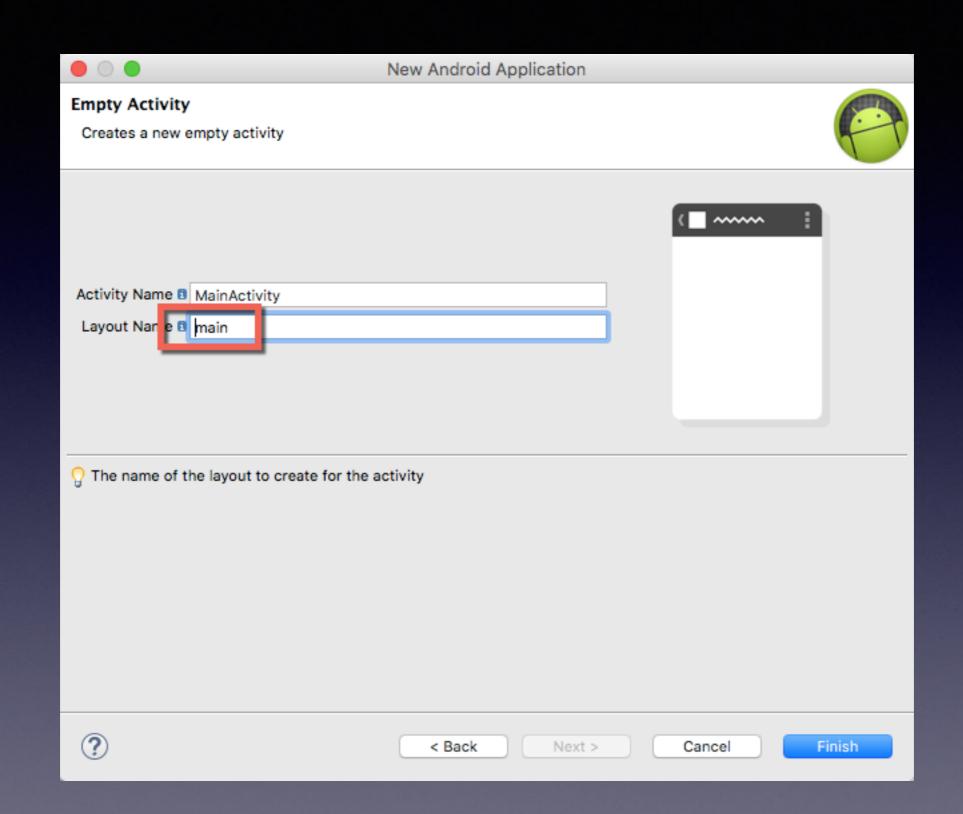
Configure Launcher Icon

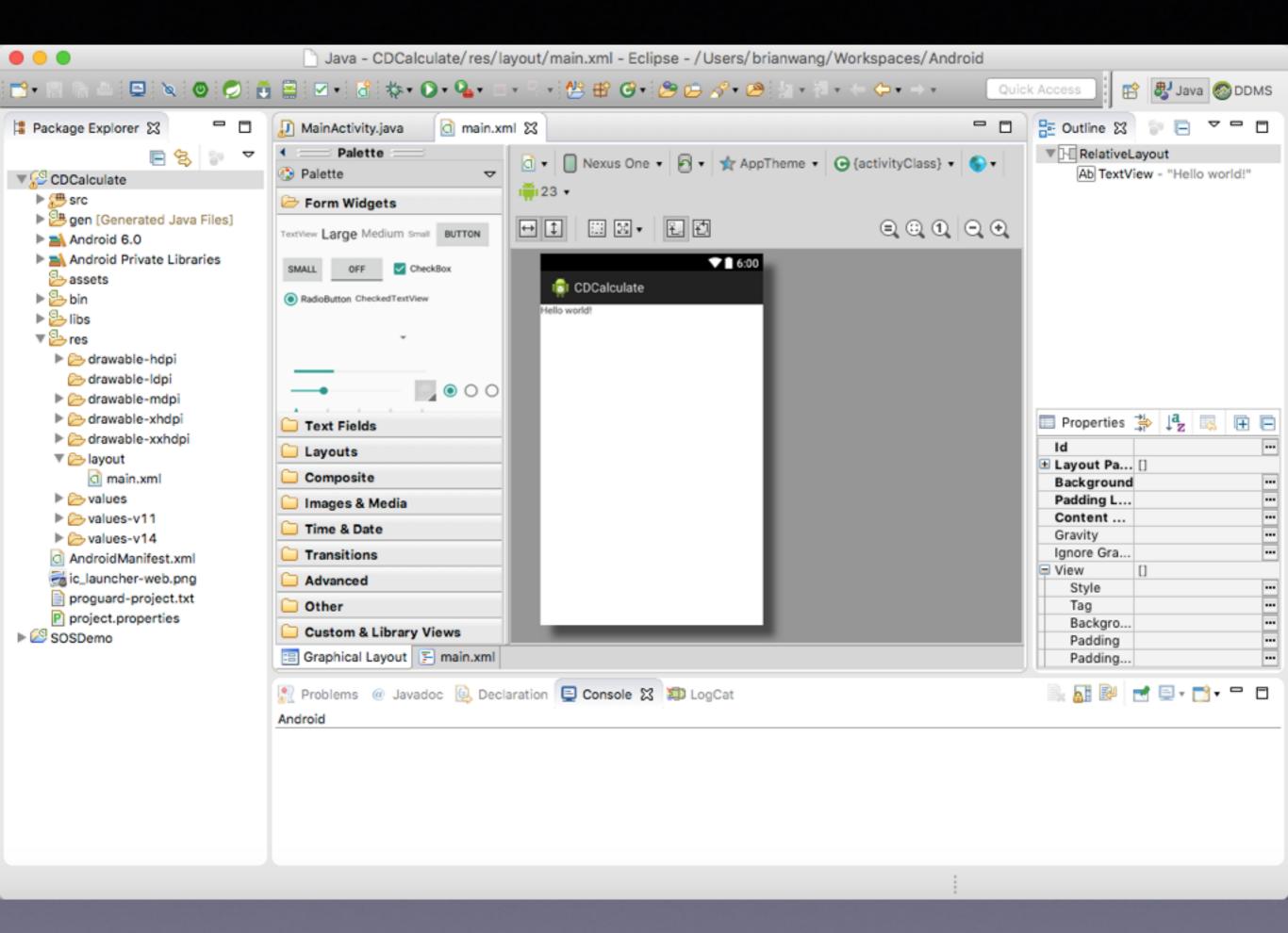
Configure the attributes of the icon set





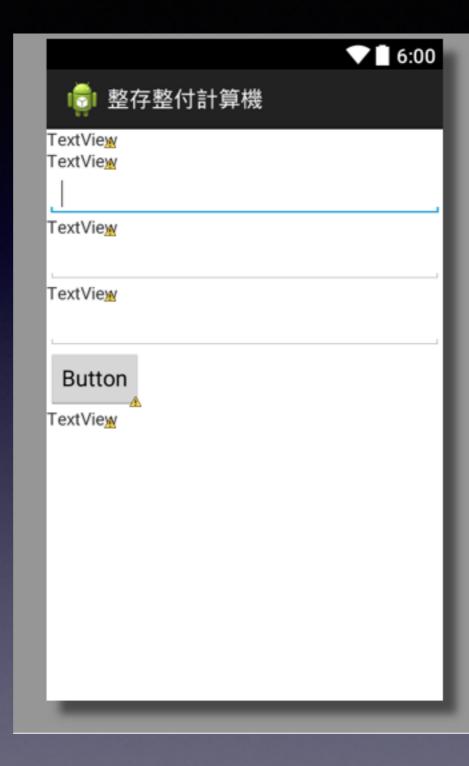




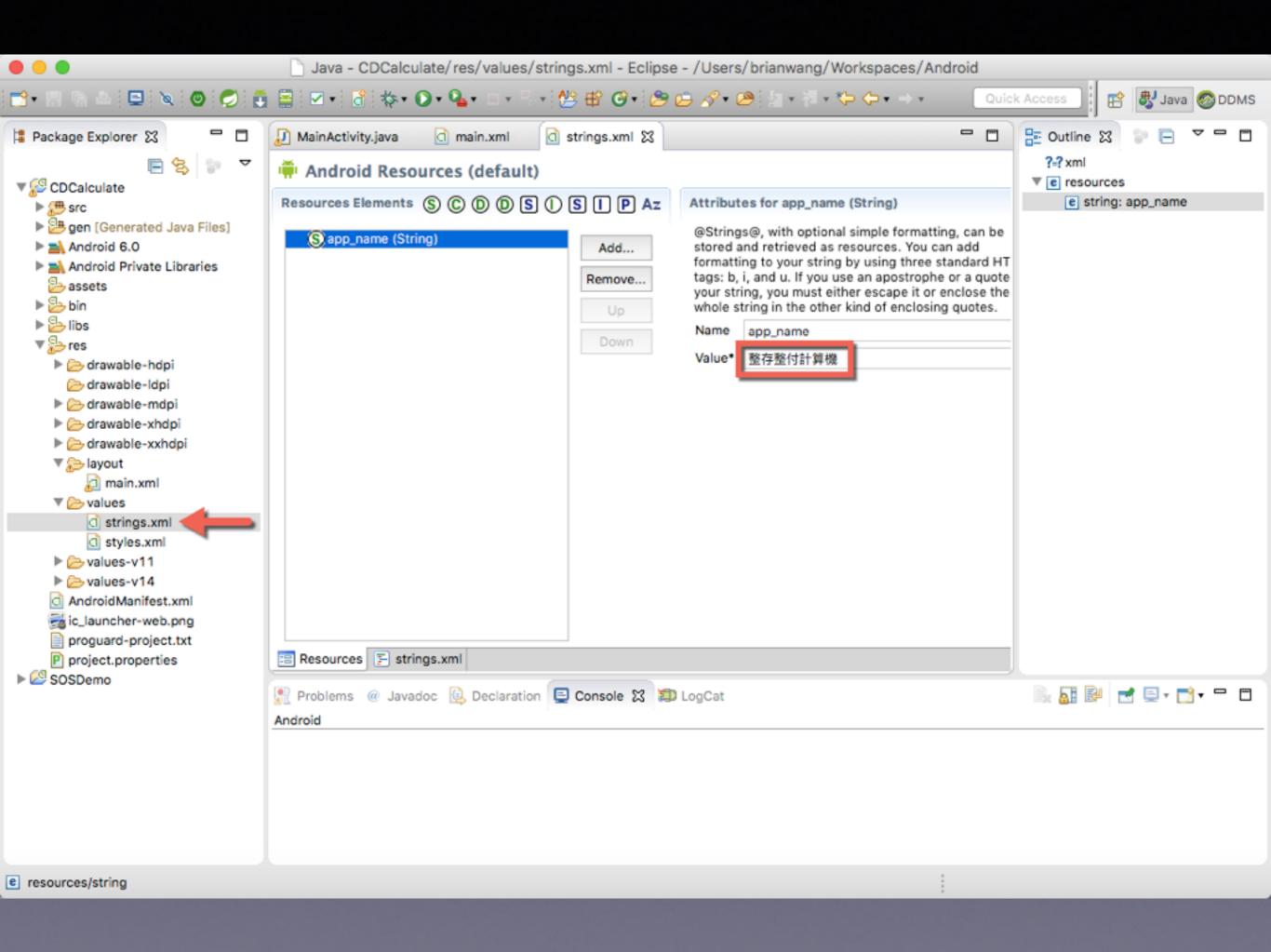


建立元件的版面配置

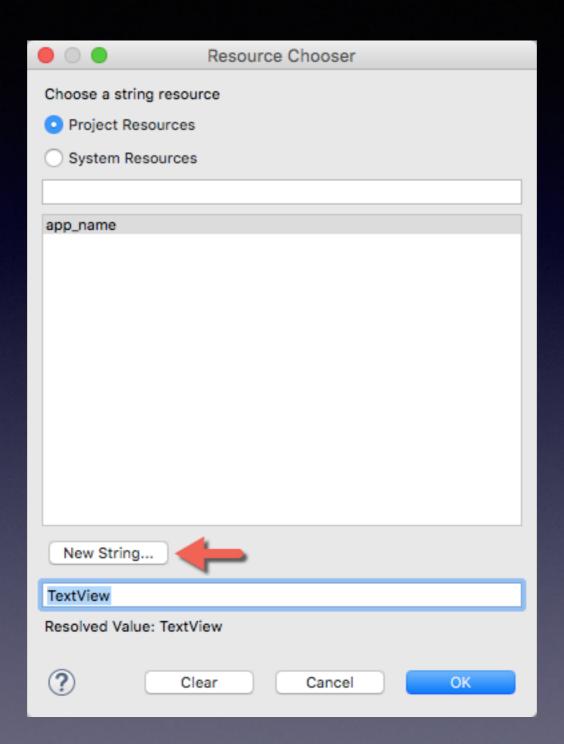
- 刪除 "Hello World" TextView
- 將 RelativeLayout 改為 LinearLayout (Vertical)
- 在 LinearLayout (Vertical) 依序垂直編排
 TextViwe、TextViwe、EditText (Number) 、
 TextViwe、EditText (Number /
 NumberDecimal) 、TextViwe 、
 EditText (Number) 和 Button 元件

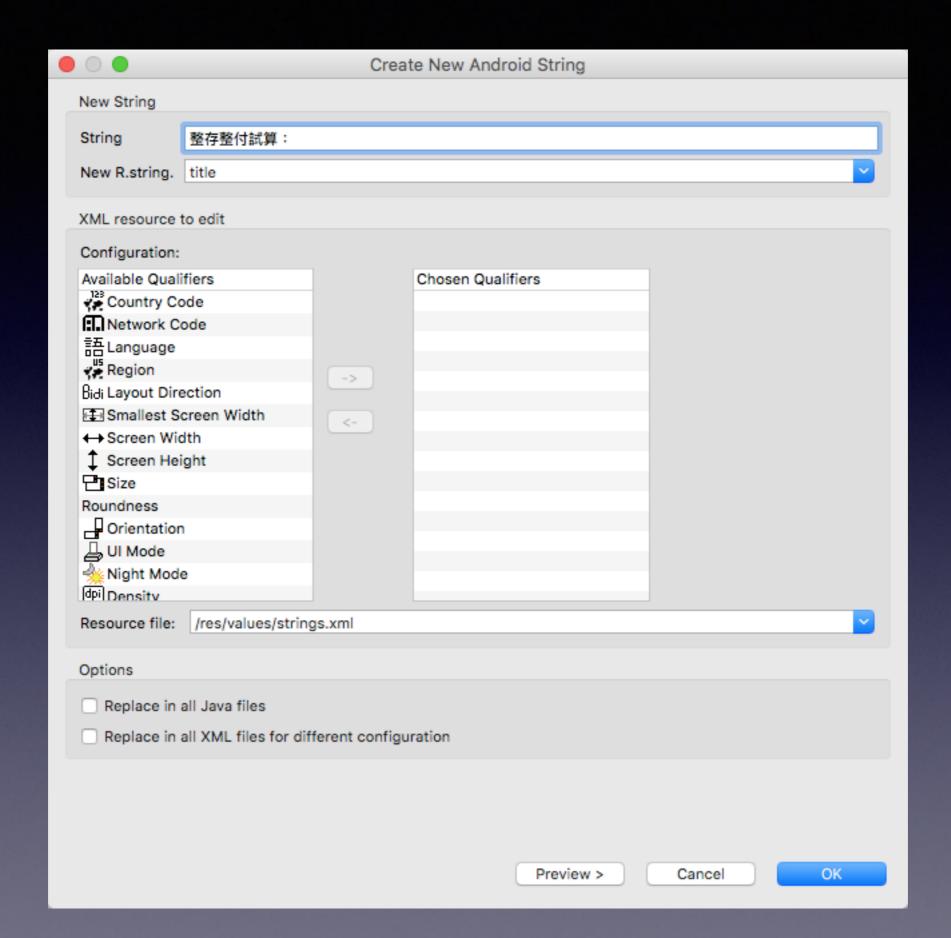


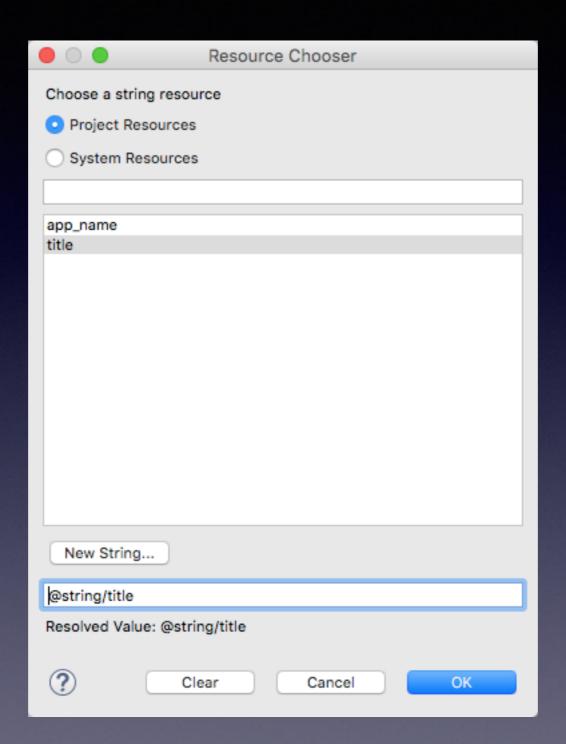
- 刪除 string.xml 的 hello_world,修改 app_name
 為"整存整付計算機"
- 將所有的 TextViwe 和 Button 的 Layout Width 改 為 Match Parent



- 將第一個 TextView 新增 title 的 string 值為 "整存整付試算:"
- 第二個 TextView 新增 lend 的 string 值為 "本金 (NT\$):"
- 第三個 TextView 新增 rate 的 string 值為 "年利率 (%):"
- 第四個 TextView 新增 number 的 string 值為 "存款期數:"

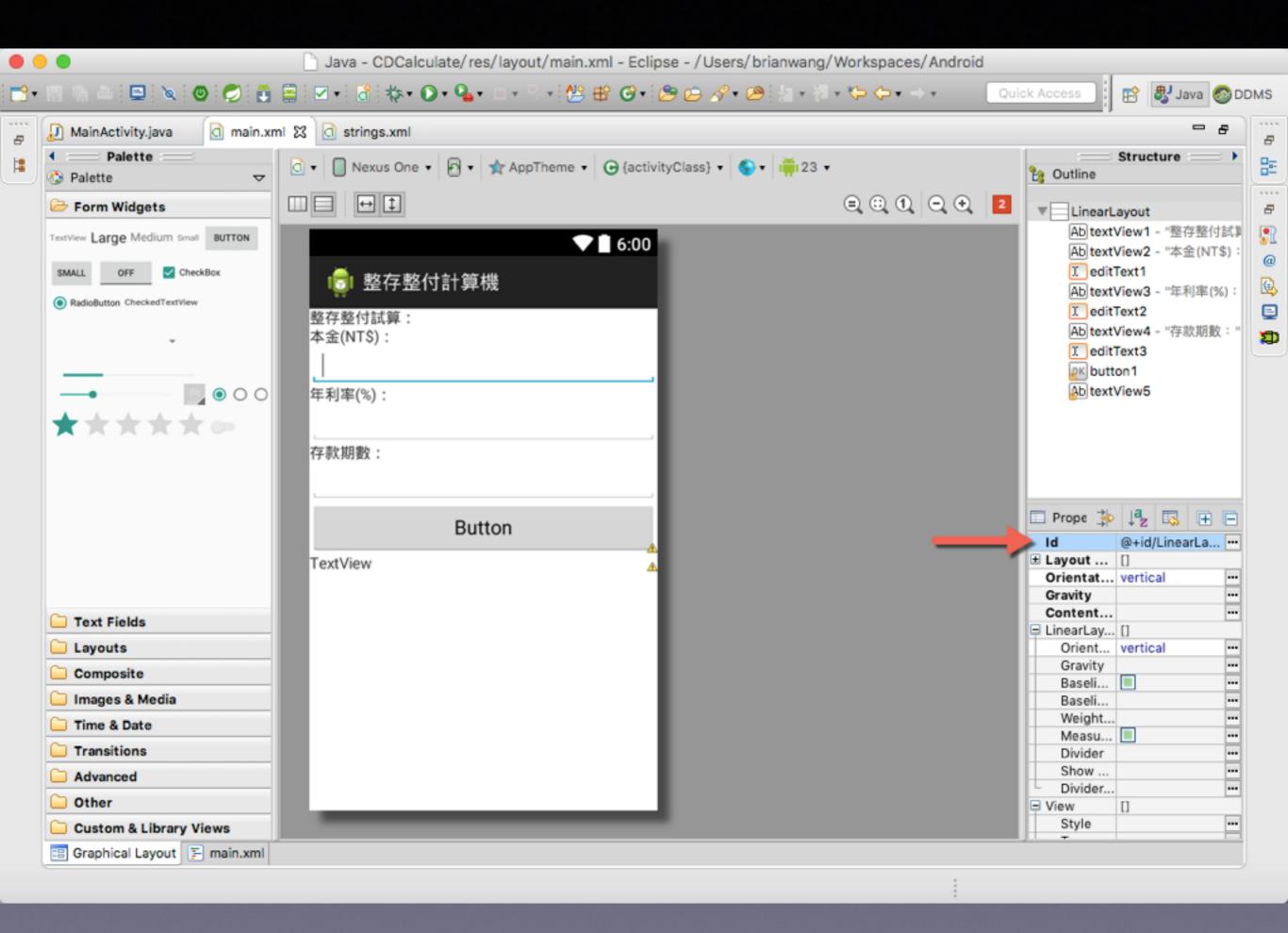


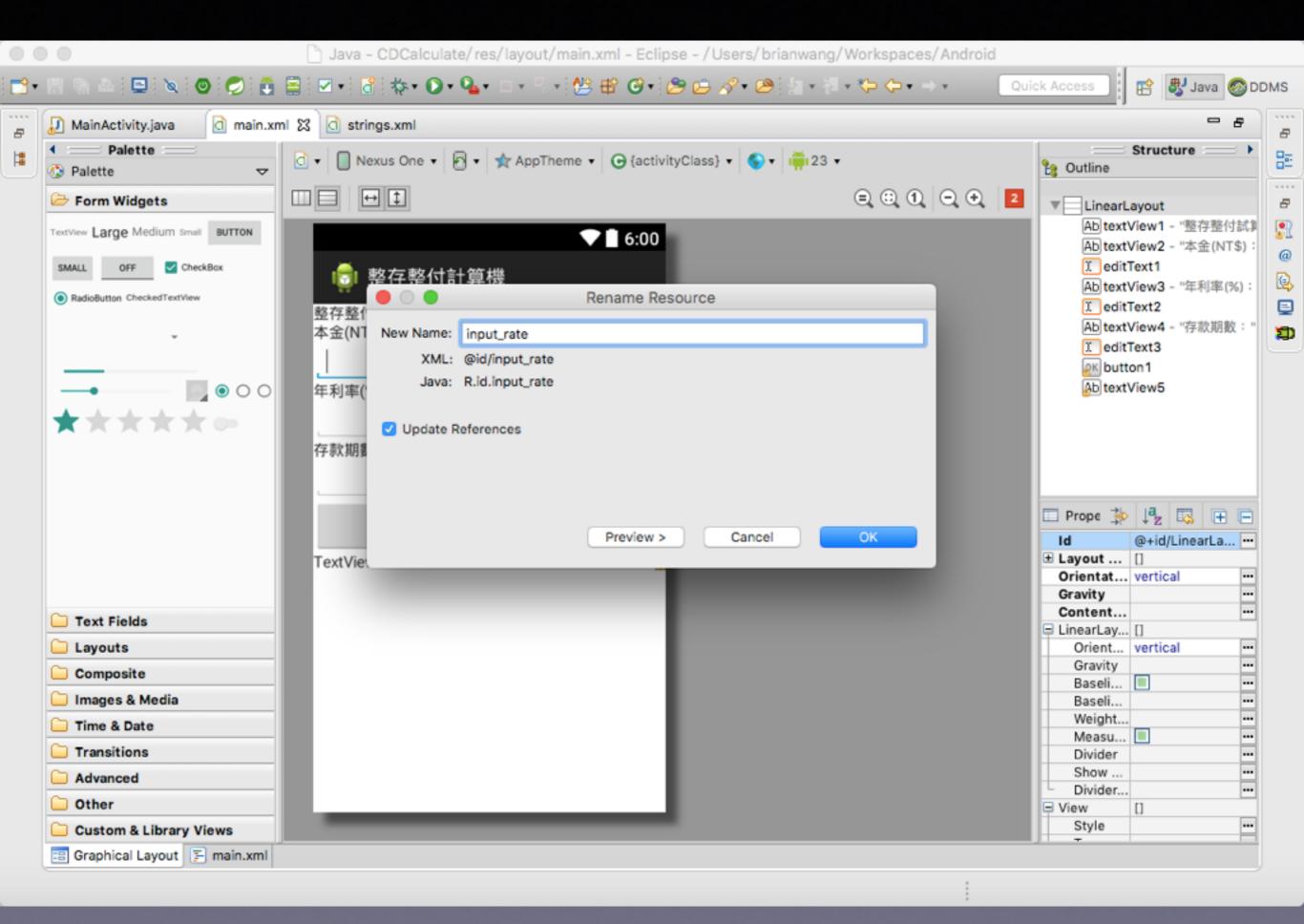


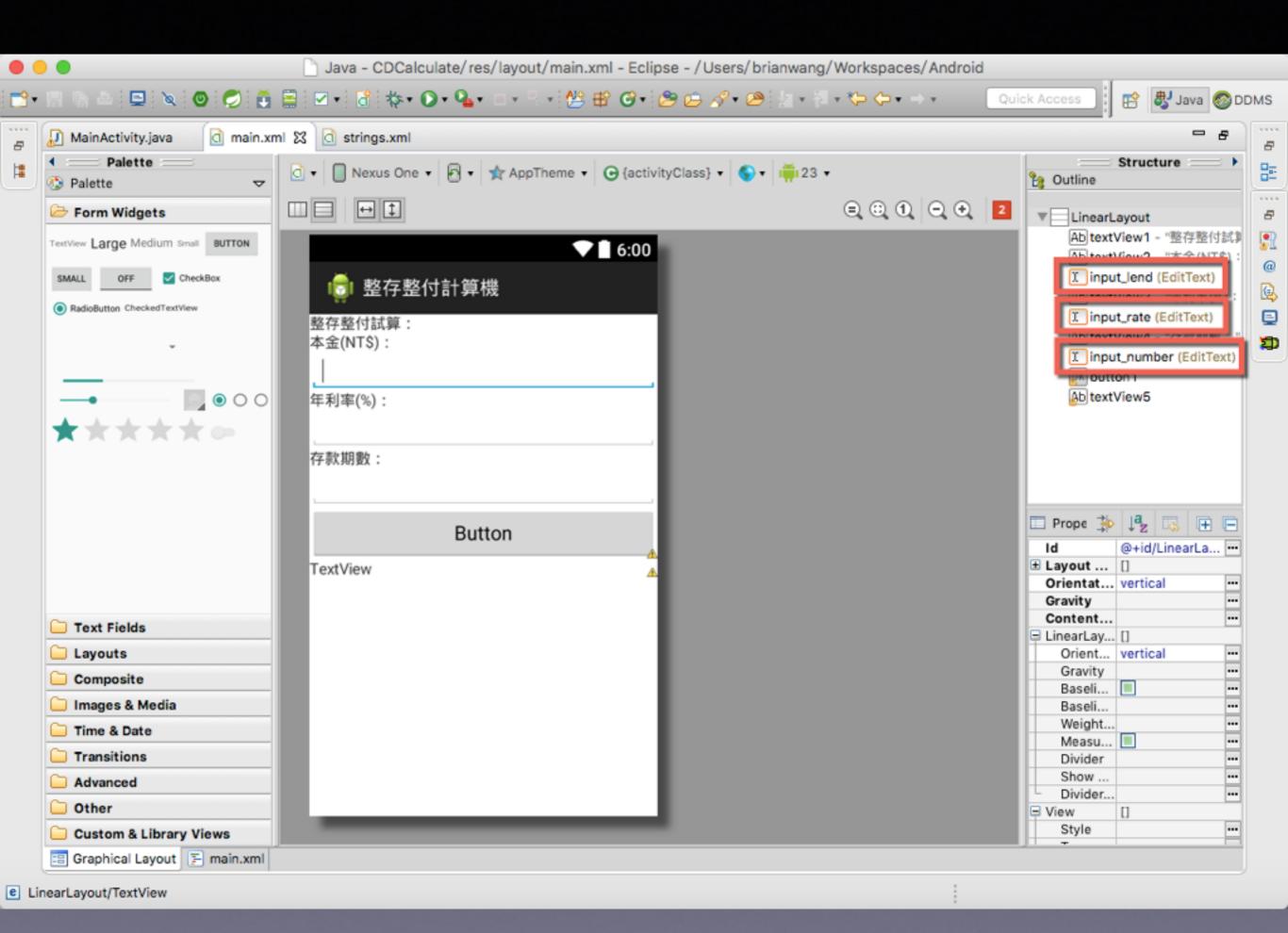


		V	6:00
👘 整存整	付計算機		
整存整付試算: 本金(NT\$): 			
_ <u> </u> 年利率(%):			
存款期數:			
	Button		A
TextView			<u>A</u>

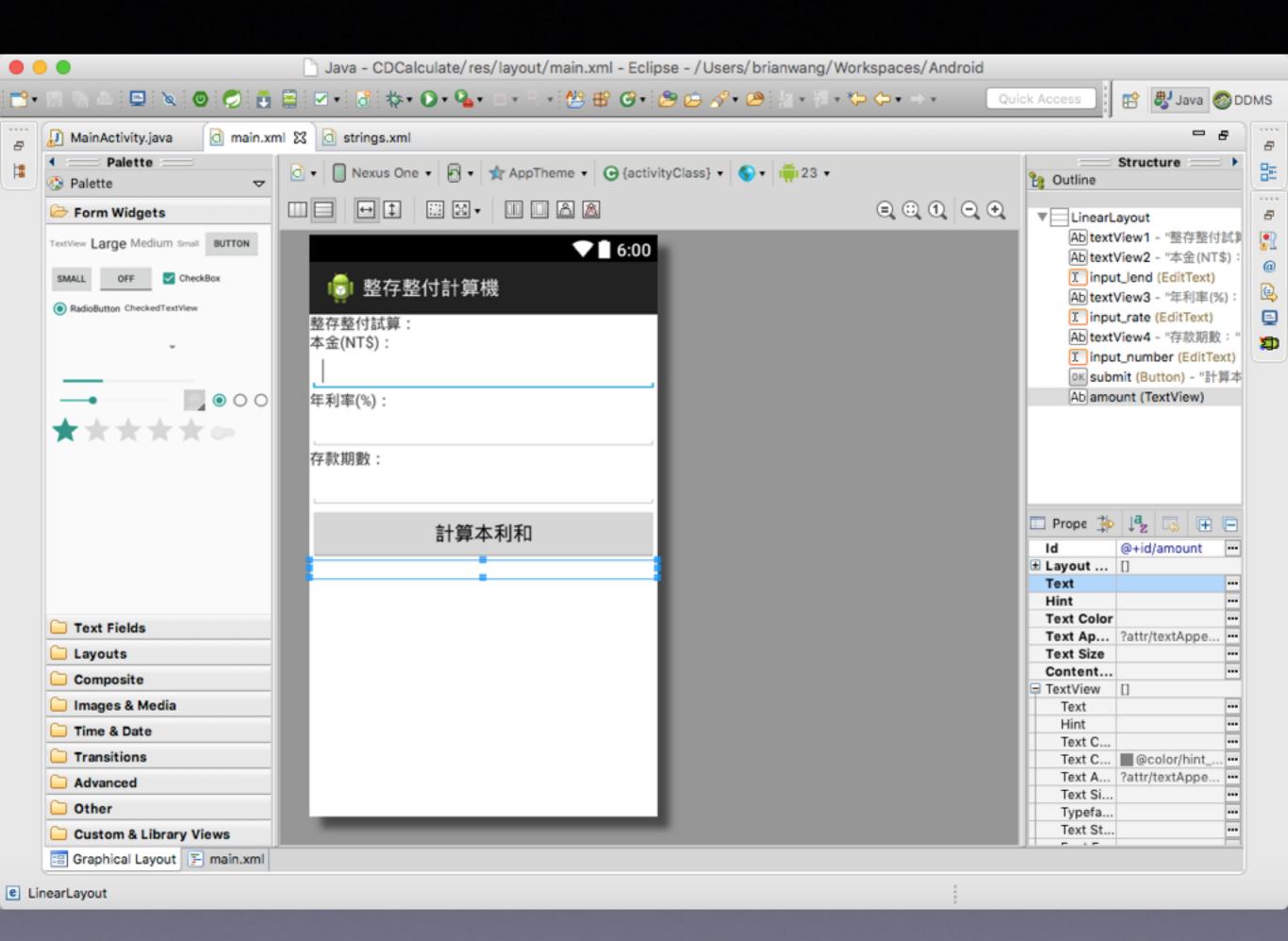
- 將第一個 EditText 的 id 設定為 input_lend
- 第二個 EditText 的 id 設定為 input_rate
- 第三個 EditText 的 id 設定為 input_number







- 將 Button 的 id 設定為 submit, on Click 設定為 calculate_Click, 並新增 btn 的 string 值為 "計算本利和"
- 將第五個 TextView 的id 設定為 amount,並清空 text



```
inearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/LinearLayout"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
   tools:context="${relativePackage}.${activityClass}" >
  <TextView
       android:id="@+id/textView1"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:text="@string/title" />
   <TextView
       android:id="@+id/textView2"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:text="@string/lend" />
  <EditText
       android:id="@+id/input_lend"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:ems="10"
       android:inputType="number" >
       <requestFocus />
  </EditText>
  <TextView
       android:id="@+id/textView3"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:text="@string/rate" />
  <EditText
       android:id="@+id/input_rate"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:ems="10"
       android:inputType="numberDecimal" />
  <TextView
       android:id="@+id/textView4"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:text="@string/number" />
  <EditText
       android:id="@+id/input_number"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:ems="10"
       android:inputType="number" />
  <Button
       android:id="@+id/submit"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:onClick="calculate_Click"
       android:text="@string/btn" />
  <TextView
       android:id="@+id/amount"
       android:layout_width="match_parent"
       android:layout_height="wrap_content" />
/LinearLayout>
```

建立 Activity 活動類別

- 宣告成員變數
- onCreate() 方法
- calculate_Click() 事件處理方法

宣告成員變數

```
package com.example.cdcalculate;
import android.app.Activity;

public class MainActivity extends Activity {

    public EditText input_lend;
    public EditText input_rate;
    public EditText input_number;
    public TextView amount;

@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

onCreate() 方法

```
package com.example.cdcalculate;
import android.app.Activity;

public class MainActivity extends Activity {

   public EditText input_lend;
   public EditText input_rate;
   public EditText input_number;
   public TextView amount;

@Override
   protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        input_lend = (EditText) findViewById(R.id.input_lend);
        input_rate = (EditText) findViewById(R.id.input_rate);
        input_number = (EditText) findViewById(R.id.input_number);
        amount = (TextView) findViewById(R.id.amount);
}
```

本利和公式

- 本利和=本金*(1+年利率/12/)%^存款期數
- lend * Math.pow((rate / 12) / 100 + 1, number)

calculate_Click() 事件處理方法

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    input_lend = (EditText) findViewById(R.id.input_lend);
    input_rate = (EditText) findViewById(R.id.input_rate);
    input_number = (EditText) findViewById(R.id.input_number);
   amount = (TextView) findViewById(R.id.amount);
public void calculate_Click(View view) {
   if (input_lend.getText().toString().isEmpty()) {
       Toast.makeText(this, "請輸入本金!", Toast.LENGTH_SHORT).show();
        return;
   }
   if (input_rate.getText().toString().isEmpty()) {
       Toast.makeText(this, "請輸入年利率!", Toast.LENGTH_SHORT).show();
        return;
   if (input_number.getText().toString().isEmpty()) {
       Toast.makeText(this, "請輸入存款期數!", Toast.LENGTH_SHORT).show();
        return;
   int lend = Integer.parseInt(input_lend.getText().toString());
   double rate = Double.parseDouble(input_rate.getText().toString());
   int number = Integer.parseInt(input_number.getText().toString());
   int result = (int) (lend * Math.pow((rate / 12) / 100 + 1, number));
   amount.setText("本利和為:" + result);
```