Lab1 – Activities

Currency Converter

Create a project

Create a project (fr.android.eurodollar) ;
App name : Currency converter
Activity : CurrencyConverter

XML & Activites

The SDK enables you to create string resources, layout resources, etc.. directly in XML files located in the « res » part of the project. Choose « res/layout/activity_main.xml » and check this out.

The XML format of the manifest file is defined in :
http://developer.android.com/guide/topics/manifest/manifest-intro.html

String and Colour Resources

Android allows to create attributes that can be used in different situations (strings, colours, …) ; they are mainly used in order to adapt the content of the application depending on screen sizes, the system language, …

Select the file « res/values/strings.xml » and add the following strings to it (name, value) :

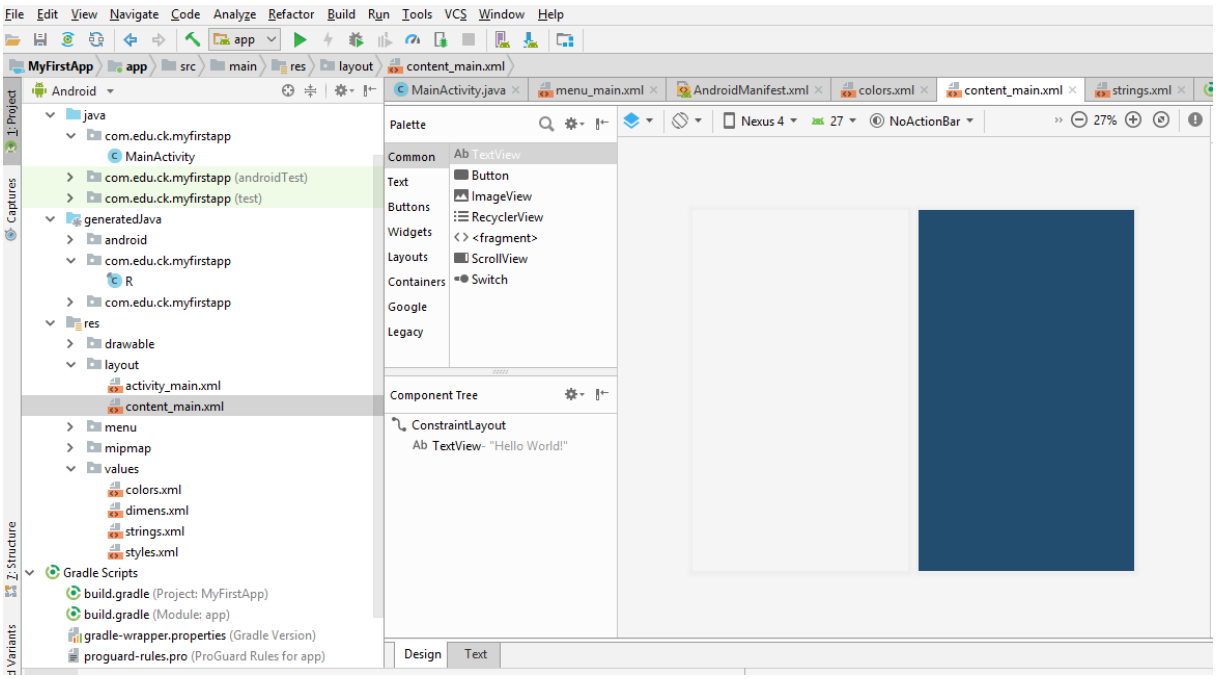
(« euro », «in €»)

(« dollar », «in $»)

Add to the same file a « color » tag, the name being « myColour » and the value being « #3399CC ». (take a look at the colors.xml file and copy the syntax)

Graphical Interface

Select « res/layout/content_main.xml » and open the graphical editor by selecting the « Design » tab. It enables you to drag and drop widgets inside the screen. You can then verify that the corresponding xml tags have been added to the xml file.



On the left side, upon clicking on a component, you will get its list of attributes and event handlers.

1. Open the XML and replace the « ConstraintLayout » component by the following

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context=".MainActivity"
    tools:showIn="@layout/activity_main">
```

2. Go to the attirbute section and set the background to « myCoulour » (we've added it before to our resources)
3. Click on « Hello World » and remove it.
4. Add a « plain text » (EditText component) that we're going to use later to enter the value to convert in euros or dollars.

5. Click on « EditText » and use the attribute section on the right to change the following attributes :
    a. Layout_width to match_parent
    b. Layout_height to wrap_content
    c. Change the text to « value to convert » and add this string as a resource to you project and reference it from here
    d. Set the « inputType » property to « numberSigned » and « numberDecimal »
    e. Set its id to « editText1 »

6. Create a Radio Group and add 2 Radio Buttons to it (with the same attributes as for the EditText)
        i. Change the text for each radio button, the first being @string/euro and the second, @string/dollar
        ii. Select the « checked » property for the 1st Radio Button
7. Add a button to your screen
    a. Name it « Convert » (text value) – add this as a resource first in « strings.xml »
    b. Set the « onClick » property to « myClickHandler »

8. Take a look at the XML content to get used to it.

Handling The Click Event

The Java part contains the « CurrencyConverter » class that we have defined when we first created the project. It inherits either from AppCompatActivity or from Activity.

We want to adapt it in order to handle the click event on the button.

1. Add the following attribute to the class
    a. EditText text ;

2. Add the following
    a. text = (EditText) findViewById(R.id.editText1)
    b. What does it do ?

3. Add the following code for handling the button click and **adapt it**

```
public void myClickHandler(View view) {
    switch (view.getId()) {
```

```
        case R.id.button1:
            RadioButton euroButton = (RadioButton) findViewById(R.id.radio0);
            RadioButton dollarButton = (RadioButton) findViewById(R.id.radio1);
            if (text.getText().length() == 0) {
                Toast.makeText(this, "Please enter a valid number",
                        Toast.LENGTH_LONG).show();
                return;
            }

            float inputValue = Float.parseFloat(text.getText().toString());
            if (EuroButton.isChecked()) {
                text.setText(String
                        .valueOf(convertDollarToEuro(inputValue)));
                euroButton.setChecked(false);
                dollarButton.setChecked(true);
            } else {
                text.setText(String
                        .valueOf(convertEuroToDollar(inputValue)));
                dollarButton.setChecked(false);
                euroButton.setChecked(true);
            }
            break;
        }
    }

    // Convertir Dollar à Euro
    private float convertDollarToEuro(float dollar) {
        return …; // formule à utiliser
    }


    // Convertir Euro à Dollar
    private float convertEuroToDollar(float euro) {
        return …; // formule à utiliser
    }
            }
```

And now, run your first App !!!