CST2335 – Graphical Interface Programming

Lab 2

Introduction:

The goal of this lab is become familiar with how XML and Java are combined for creating user interfaces. The layouts and widgets are created in XML, and Java then gets references to the objects created in XML to control the behavior of the application

References:

Chapter 4 in Android Programming with Android Studio.

Steps:

1. Create a branch of your software from Lab 1. Select the VCS menu in Android Studio, and click “Git” -> “branches”. From the dialog box, select “New Branch” and call it “Lab 2”. Using a new branch means that all of your software commits will only change this branch, and not the original branch. This means you can go back to the “master” branch at any time.
2. Go to http://flaglane.com/ and look for an image of a Canadian flag. You should download the small PNG file, 200 pixels wide. Once downloaded, rename the file **flag.png**. Copy the file into the drawable folder in AndroidStudio. Next right-click on the “res” folder and select “New” -> “Android Resource Directory”. Set the resource type to be drawable, and set the directory name to be “drawable-XX”, but replace the XX with the two letter language code that you used for your strings file, for example “drawable-es” for Spanish. Next, download a flag of that country from <http://flaglane.com/>, a PNG file 200 pixels wide. Rename that file **flag.png** and copy it into the new drawable-XX folder you created. You will use these flags for the ImageButton later on.
3. The rest of this lab is to create three different layouts that all look like the image on the next page.
   1. One should use only LinearLayout. Call it activity\_main\_linear.xml.
   2. One should use a GridLayout. Call this activity\_main\_grid.xml.
   3. One should use a RelativeLayout. Call this activity\_main\_relative.xml

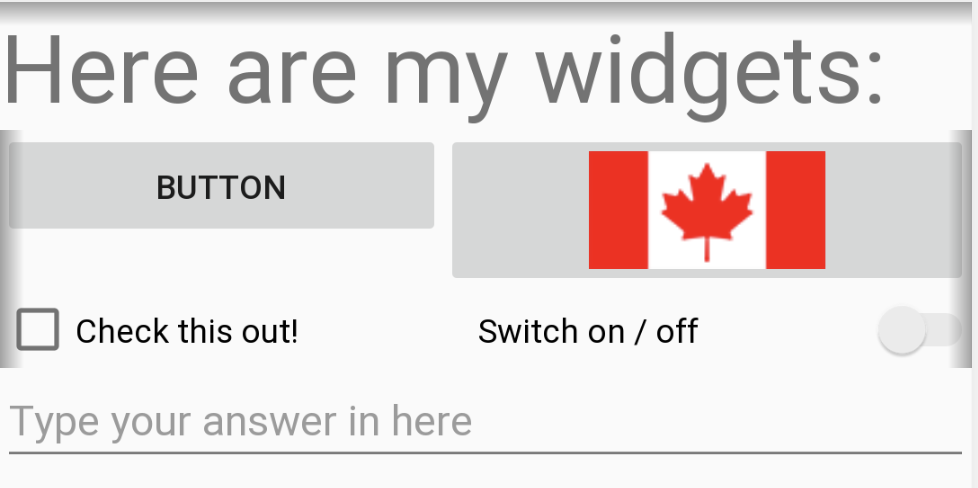


Figure 1 The layout for application. Recreate this LinearLayout, GridLayout, and RelativeLayout

Row 1: The top item is a <TextView> that has the string “Here are my widgets:”. Use the parameter android:textSize=”30dp” to make the text bigger than normal.

Row 2: Below the text view is a row that has a <Button> with text=”Button”, and an <ImageButton> whose src=”@drawable/flag”. They should each take approximately 50% of the width of the row. Remember that @drawable will be either the drawables folder, or drawables-XX folder that you created in step 2. If you change the phone’s language, the flag should switch to the one you downloaded for that country.

Row 3: On the left side is a <CheckBox> with the text=”Check this out!”. On the right side is a <Switch> object with the text=”Switch on / off”.

Row 4: There is only an <EditText> with the hint=”Type your answer here”. Set the inputType=”textPersonName”. This means when you type, every first letter will be capitalized. The EditText should take up the entire row.

1. For the LinearLayout, the root tag should be a vertical LinearLayout, and each row should be a horizontal LinearLayout. For the items in each row, set layout\_weight=”1” and gravity=”fill” if the items should take 50% of the row. If there is only 1 item in a row, then set layout\_width=”match\_parent”.
2. For the GridLayout, use a grid with 4 rows and 2 columns. Set layout\_weight=”1” and gravity=”fill” to make the objects use the entire grid box. Also, use layout\_columnSpan=”2” to make an object use 2 grid boxes.
3. For the RelativeLayout, use a combination of layout\_alignParentTop, layout\_alignParentLeft, layout\_alignParentRight to align widgets to the sides of the screen. Use layout\_below, layout\_toEndOf, layout\_toStartOf, to position object properly.
4. Translate all of the strings used in the layouts to your second language. Anywhere you set text=”something” or hint=”something” should use the “@string/” format.
5. When you are finished, show your lab professor each of the three layouts. Run the project 3 times, each time changing the function setContentView( ) to load each of the 3 layouts. Also, change the language to show that the flag image changes for your second language, and your strings change to the second language.
6. In your project view, right-click on the “App” folder, and select “Git” -> “Commit Directory”. The changes window should show all of the files that were changed in this lab. Put a commit message saying “Added 3 layouts for Lab 2”, and click the “Commit” button. Now your changes have been saved to the “Lab 2” branch of your repository. Right-click again on the “App” folder, and select “Git” -> “Repository” -> “Push”. Now your changes are pushed to either your local repository on your hard drive, or to Github if you configured your account in AndroidStudio.

Marks:

The LinearLayout looks correct. +3

The GridLayout looks correct. +3

The RelativeLayout looks correct. +3

ImageView changes to show the different flag +1

strings.xml file has been updated with text for second language (step 7) +1