# Lab 2 part 1

In this part, students will develop a mobile application for changing background color of text view. The layout included will be shown below:

* ConstraintLayout
* LinearLayout
* TextView
* Button
* Spinner
* EditText

Here are the screen captures of the application:

Graphical user interface, application

Description automatically generatedGraphical user interface, application, Teams

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Step 1:

In Android Studio, create a new project named Lab 2 Part 1 with following project setting:

* Choose your project : **Empty Activity**
* Application Name : **Lab 2 Part 1**
* Project location : use the default setting
* Language : **Java**
* Minimum API level: **API 21: Android 5.0 (Lollipop)**
* Click **Finish**

Step 2:

Define string values in **res/values/strings.xml**

<resources>  
 <string name="app\_name">Lab 2 Part 1</string>  
 <string name="first\_color">First Color:</string>  
 <string name="second\_color">Second Color:</string>  
 <string name="hint">Red,Blue or Green</string>  
 <string name="execute">Enter</string>  
 <string name="warning">You need to type something.</string>  
 <string name="format">You typed the worse format.</string>  
 <string-array name="color\_list">  
 <item>Red</item>  
 <item>Blue</item>  
 <item>Green</item>  
 </string-array>  
</resources>

Step 3:

Define string values in **res/values/colors.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<resources>  
 <color name="purple\_200">#FFBB86FC</color>  
 <color name="purple\_500">#FF6200EE</color>  
 <color name="purple\_700">#FF3700B3</color>  
 <color name="teal\_200">#FF03DAC5</color>  
 <color name="teal\_700">#FF018786</color>  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
 <color name="red">#ffaa0000</color>  
 <color name="blue">#0000aa</color>  
 <color name="green">#ff00aa00</color>  
</resources>

Step 4:

Open the **res/layout/activity\_main.xml** file and insert the following:

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout 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"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:text="@string/first\_color"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Spinner  
 android:id="@+id/color\_spinner"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginEnd="16dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView"  
 android:entries="@array/color\_list"/>  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:text="@string/second\_color"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/color\_spinner" />  
  
 <EditText  
 android:id="@+id/color\_editText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginEnd="16dp"  
 android:ems="10"  
 android:hint="@string/hint"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView2" />

<Button  
 android:id="@+id/color\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:text="@string/execute"  
 android:onClick="onClick"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/color\_editText" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginEnd="16dp"  
 android:orientation="horizontal"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/color\_button">  
  
 <TextView  
 android:id="@+id/colorTextView1"  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:gravity="center\_horizontal" />  
  
 <TextView  
 android:id="@+id/colorTextView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:gravity="center\_horizontal" />  
  
 </LinearLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Graphical user interface

Description automatically generated

ConstraintLayout : blue frame

LinearLayout : orange frame

You will see there is a warning for android:onClick="onClick" at the moment. It will be fixed later.

Step 5:

Open **MainActivity.java** and insert the following code for the MainActivity class:

public class MainActivity extends AppCompatActivity {  
  
 private Spinner colorSpinner;  
 private EditText colorEditText;  
 private TextView colorTextView1, colorTextView2;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 colorSpinner = findViewById(R.id.*color\_spinner*);  
 colorTextView1 = findViewById(R.id.*colorTextView1*);  
 colorTextView2 = findViewById(R.id.*colorTextView2*);  
 colorEditText = findViewById(R.id.*color\_editText*);  
  
 }

}

You will there are the warning that highlighted the class just like the following:

Graphical user interface, text, application

Description automatically generated

In this case, click option + return (Mac) / ALT + Enter (Window) to import the related elements.

You can also go to **File(Window) / Preference (Mac) -> Settings -> Editor -> General -> Auto Import** and make the following changes:

* Insert imports on paste value : **All**
* Add unambigious imports on the fly option : **Checked**

Step 6:

Add onClick() method outside the onCreate() method. Then define the related attributes for spinner and edit text.

public void onClick(View v){  
 *//call specific array from strings.xml* String[] color = getResources().getStringArray(R.array.*color\_list*);

*//get the position from array* int colorId = colorSpinner.getSelectedItemPosition();  
  
 *//get edittext input* String input =colorEditText.getText().toString();  
  
}

Step 7:

Add the logic of spinner selection into onClick() method. and get string from the edit text. getResources().get… is used to link the elements from the xml file. setText is used to display the text.

switch(colorId) {  
 case 0:  
 colorTextView1.setBackgroundColor(getResources().getColor(R.color.*red*));  
 break;  
 case 1:  
 colorTextView1.setBackgroundColor(getResources().getColor(R.color.*blue*));  
 break;  
 case 2:  
 colorTextView1.setBackgroundColor(getResources().getColor(R.color.*green*));  
 break;  
 default:  
 break;  
}  
colorTextView1.setText(color[colorId]);

Step 8:

Add the logic of getting string from the edit text into onClick() method. setError is used to show warning. Toast is used to show a short alert. isEmpty() is used to check edit text is nothing or not.

*//For EditText*if(input.equals("Red")){  
 colorTextView2.setBackgroundResource(R.color.*red*);  
 colorTextView2.setText(input);  
}else if(input.equals("Blue")){  
 colorTextView2.setBackgroundColor(getResources().getColor(R.color.*blue*));  
 colorTextView2.setText(input);  
}else if(input.equals("Green")){  
 colorTextView2.setBackgroundColor(getResources().getColor(R.color.*green*));  
 colorTextView2.setText(input);  
} else if(input.isEmpty()) {  
 colorEditText.setError(getResources().getString(R.string.*warning*));  
}else {  
 Toast.*makeText*(getApplicationContext(),getResources().getString(R.string.*format*),Toast.*LENGTH\_SHORT*).show();  
}

Remark: if you use if else to set the condition, please remember use .equals() for String, = = for int/double.

Step 9 (extra):

We have created an application with a vertical layout, how about horizontal?

To ensure the app can run smoothly in different rotation, the horizontal layout should also be created.

Open the activity\_main.xml and click “Design”, then select this button.

Icon

Description automatically generated

Choose “Create Landscape Variation”. Modify the code if necessary. For example, change the size of colour text view to touch the bottom.

Chart, bar chart

Description automatically generated