Coding showing the following:

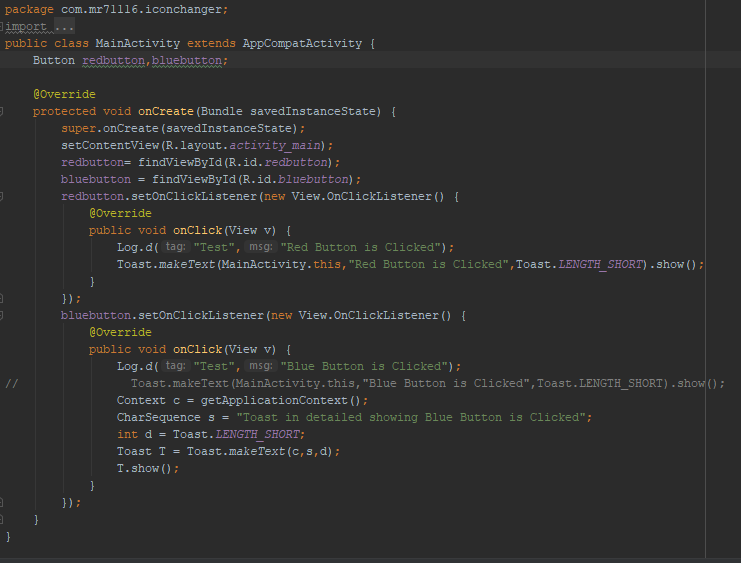
.setOnclickListener

Log.d

Toast in single line code

Toast in multiple line code

Button declaration without casting (new feature in Android, previously it need to type casting like (Button)



**Customer Toast**

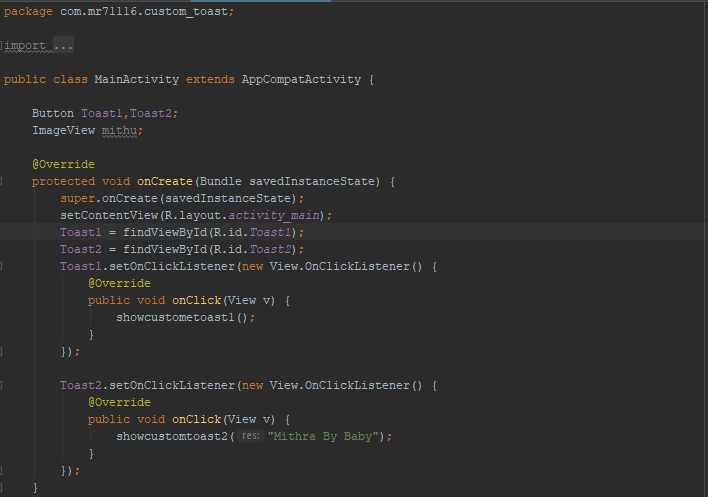
Toast 1 with Image as Toast

Toast 2 with Textview as Toast

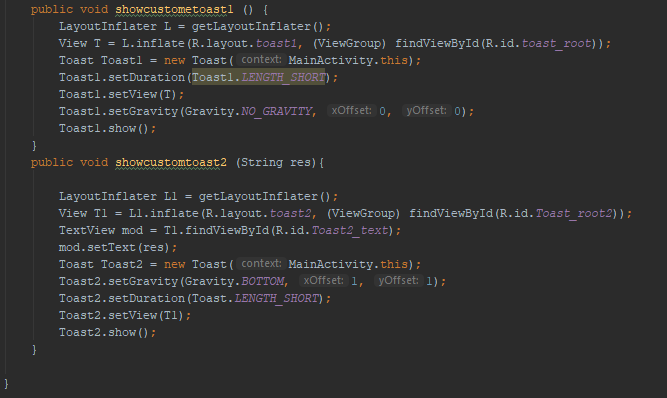
Screenshot 1:

Oncreate method with Two buttons Onclick Listeners.

Toast methods are called here in Oncreate method and Toast methods are written separately in Screenshot 2.



Screenshot 2: Methods for Toast:



LayoutInflater L1 = getLayoutInflater();

View T1 = L1.inflate(R.layout.*toast2*, (ViewGroup) findViewById(R.id.*Toast\_root2*));

The above two lines are fixed to be written for all customer toasts.

Two parameters –

1. R.layout.toast2, is name of the layout created for the toast.

2. ((ViewGroup) findViewById(R.id.*Toast\_root2*));

This is the Layout ID.

Toast Toast2 = new Toast(MainActivity.this);  
Toast2.setGravity(Gravity.*BOTTOM*, 1, 1);  
Toast2.setDuration(Toast.*LENGTH\_SHORT*);  
Toast2.setView(T1);  
Toast2.show();

The above line of codes are for Toast and we are making it customized with Gravity (location of toast), duration etc

We can also pass values to Toast instead of hard coding it and it is shown below:

Line 1 below: Initialing the String res as a parameter

public void showcustomtoast2 (String res){

Line 2 and 3 are used to assign the Textview and assign Text to the Textview specified.

TextView mod = T1.findViewById(R.id.*Toast2\_text*);  
mod.setText(res);

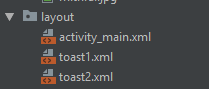
Now below code is written within the onClickListener for any button and so when button is cliekd, the string will be passed to the Textview etc.

showcustomtoast2("Mithra By Baby");

The above method is called with a single parameter.

Hence making the customer toast with a parameter.

Separately, a layout is created for the Custom Toast



Toast1.xml

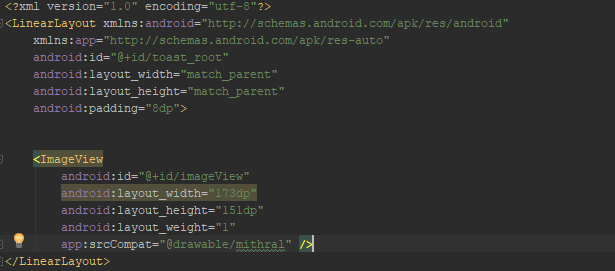
Toast2.xml in the above screen is the layout created for Customer toasts.

How to create layout as shown above:

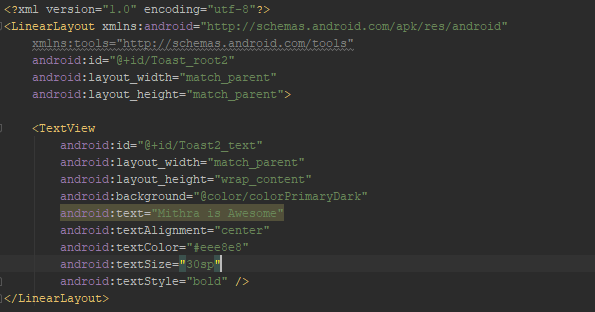
Right click on the layout, new -> Layout Resource file

Layouts created and customized for Toasts:

Layout 1:

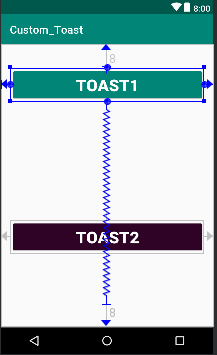


Layout 2:

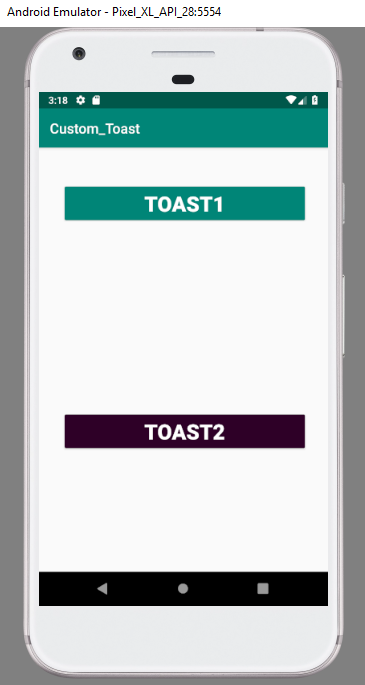


This is how, out main activity looked in constraint layout:

Two buttons

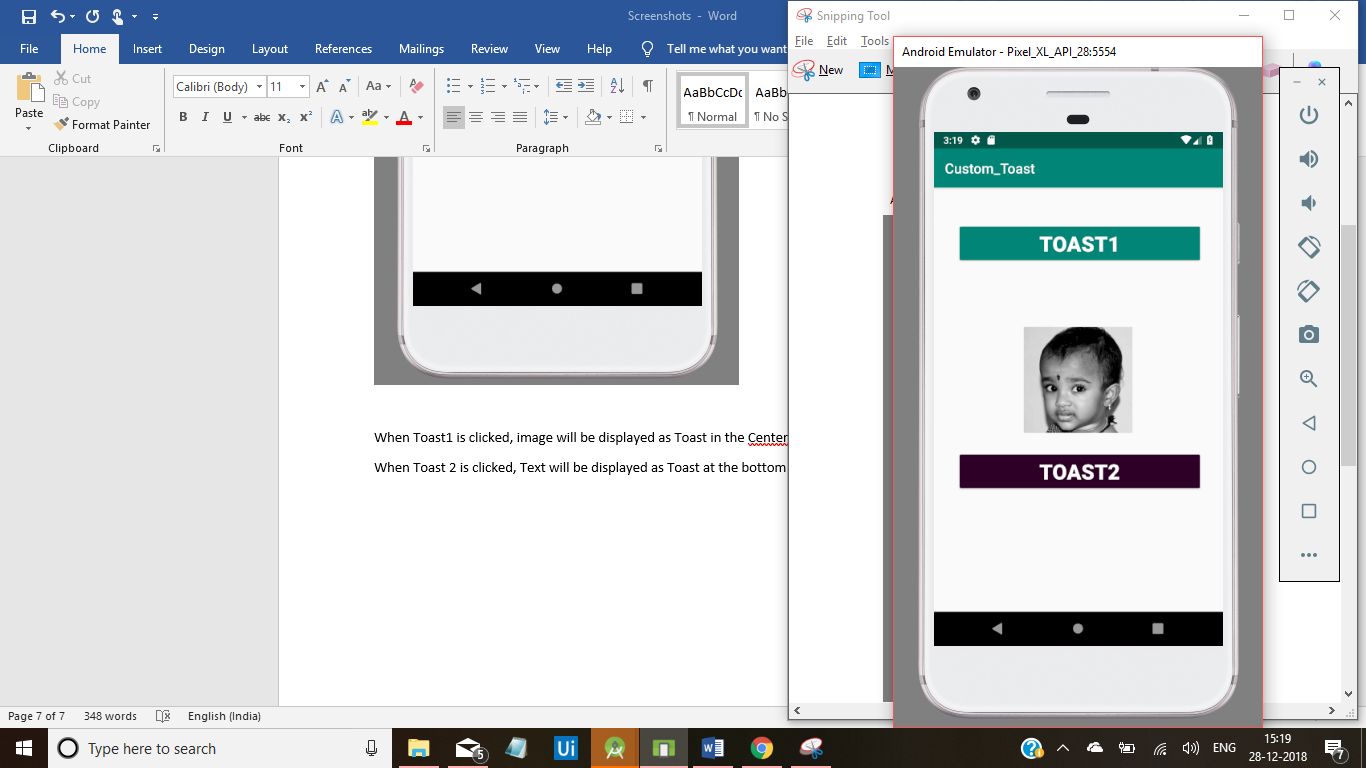


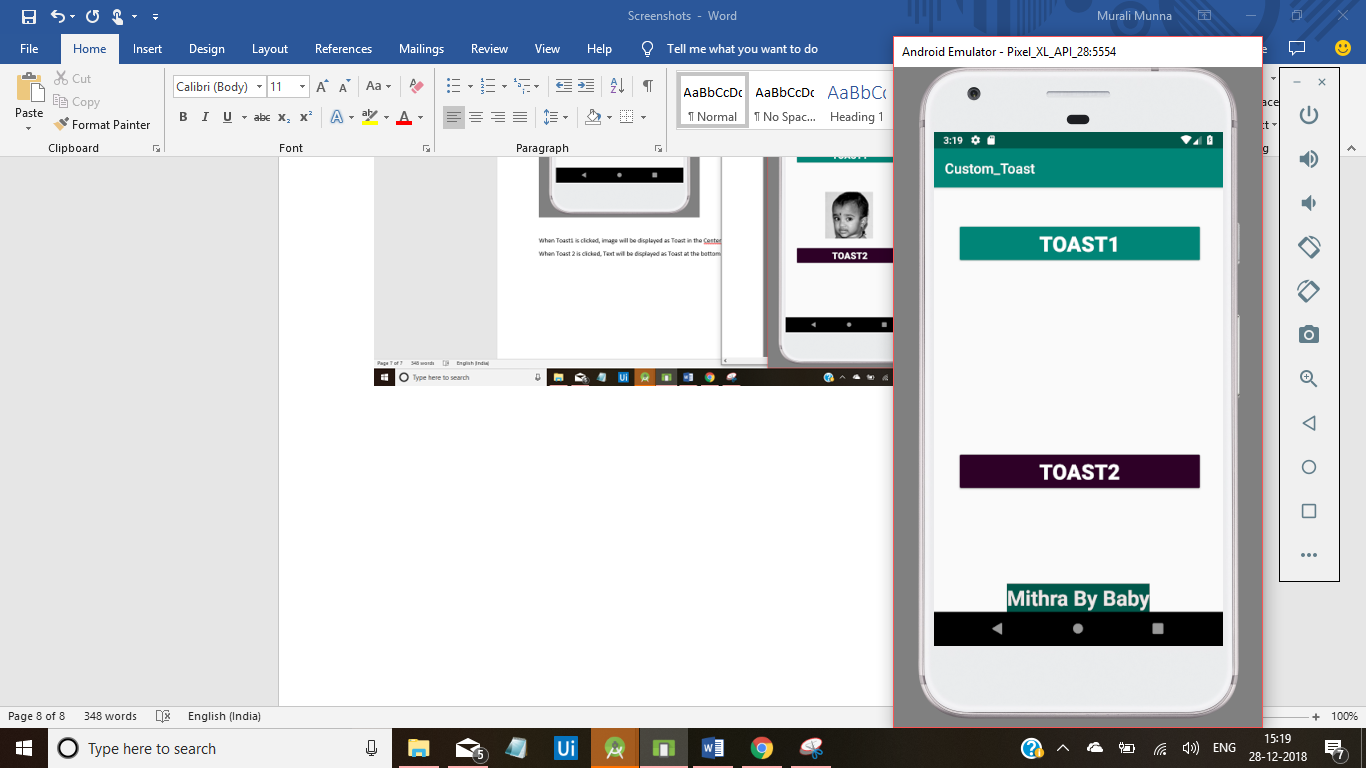
This is how it looked in emulator:



When Toast1 is clicked, image will be displayed as Toast in the Center.

When Toast 2 is clicked, Text will be displayed as Toast at the bottom





Well done, Custom Toast!!!

Gradle:

Button customizations again:

We can add the following in the layout customizations:

Example showed for Linear layout

To display the different views vertically (single colum, multi rows)

android:orientation="vertical"

To give background to the layout:

android:background="#EAF0F1"

Button Customization code:

<Button  
 android:id="@+id/button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginHorizontal="5dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginBottom="10dp"  
 android:text="Error Button"  
 android:textSize="18dp"  
 android:background="#E44236"  
 android:textColor="#FFFFFF"  
 tools:layout\_editor\_absoluteX="148dp"  
 tools:layout\_editor\_absoluteY="103dp" />

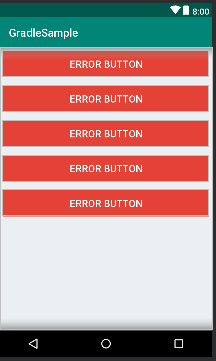
New things:

marginTop

marginBottom

marginHorizontal – give same difference in both the sides

I copy pasted these codes 5 times to get five buttons as below:



Now we can customize these buttons with different background color, text etc, different onclick methods etc

In General,

Error are shown in Red

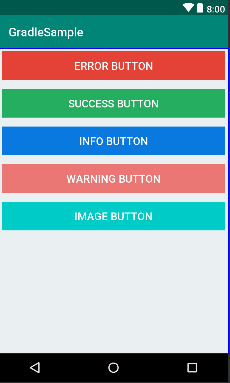
Success is shown in Green

Info is shown in Blue

Warning is shown somewhat orange

Image is shown is light green

Sample buttons view shown below:

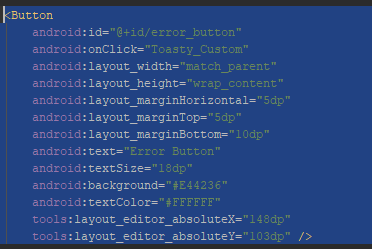


Add one more attritube –

android:onClick="Toasty\_Custom"

Add this method for all the Button (We are not going to create one method for one button)

Final attributes below:



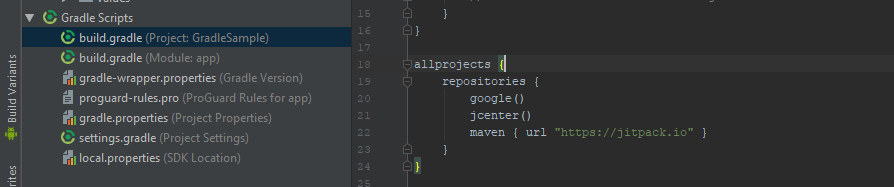
First time in the course, we are using github for taking few codes from there. Looks awesome and we are surely exploring more day by day:

<https://github.com/GrenderG/Toasty>



Added as below:

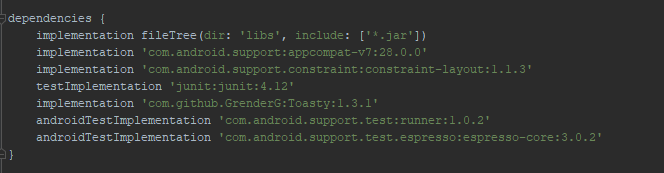
Added in build.grade (Project: -----)



Add the next code in build.grade(Module: app)



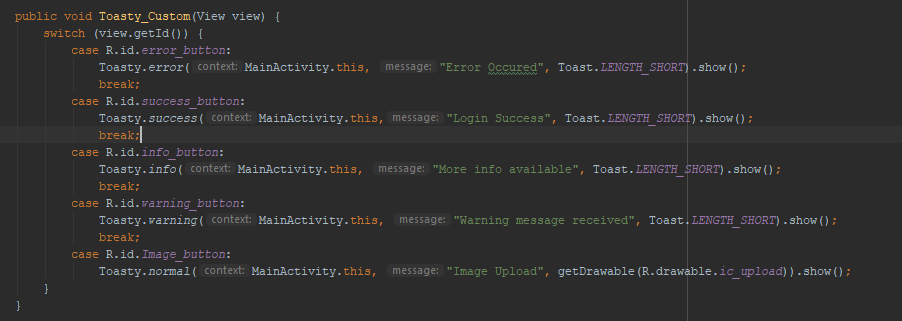
Added as below:



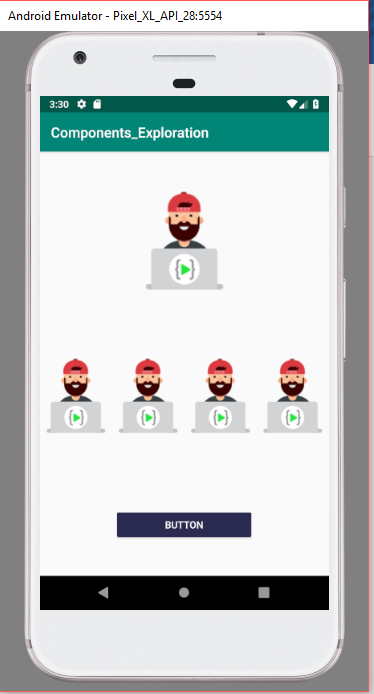
Once these two things are added, click on Sync now

Onclick method for Customer toast:

We are using switch case for each button and we are providing specific **Toasty (not toast)** methods for each buttons.



Layouts:



Lets discuss in detail how to create as above:

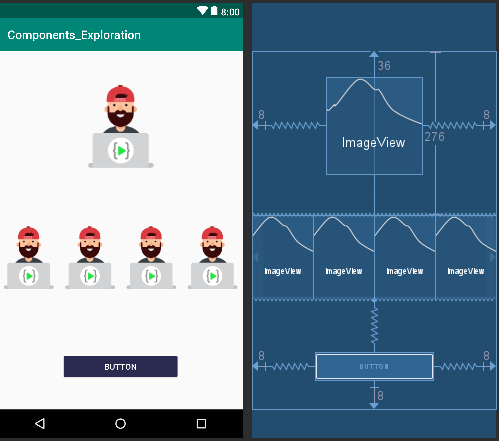
We have Constraint layout with one Image view (First page)

We have Linear Layout below the Constraint layout with 4 images

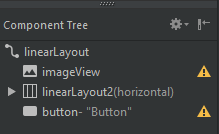
We have one Button below the Linear Layout (this button is tagged to primary constraint layout).

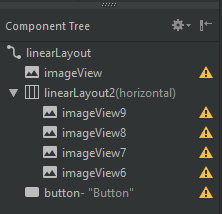
We have applied constrains for each of these components.

Below showing the constraints:



Below showing the component tree:

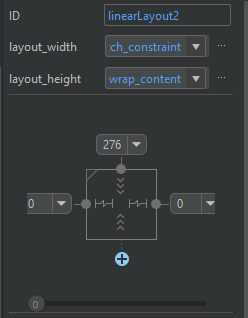




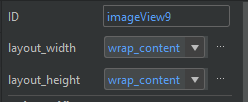
Full Text view on the activity\_main.xml:

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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:id="@+id/linearLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="162dp"  
 android:layout\_height="163dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="36dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/linearLayout2"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.125"  
 app:srcCompat="@drawable/email" />  
  
 <LinearLayout  
 android:id="@+id/linearLayout2"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="276dp"  
 android:orientation="horizontal"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <ImageView  
 android:id="@+id/imageView9"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 app:srcCompat="@drawable/email" />  
  
 <ImageView  
 android:id="@+id/imageView8"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 app:srcCompat="@drawable/email" />  
  
 <ImageView  
 android:id="@+id/imageView7"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 app:srcCompat="@drawable/email" />  
  
 <ImageView  
 android:id="@+id/imageView6"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 app:srcCompat="@drawable/email" />  
 </LinearLayout>  
  
 <Button  
 android:id="@+id/button"  
 style="@style/Widget.AppCompat.Button.Colored"  
 android:layout\_width="199dp"  
 android:layout\_height="47dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="40dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:fadingEdge="horizontal"  
 android:text="Button"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/linearLayout2"  
 app:layout\_constraintVertical\_bias="0.544" />  
</android.support.constraint.ConstraintLayout>

Linear layout positioning:



Positioning of the Linear Layout images (a sample). Same was followed for other 3 as well.



This page is saved in Assets folder:

C:\Users\Mithra\AndroidStudioProjects\Android P Projects\Assets

Dice Roller App:

Dice was designed in the designer web application called Abode XD:

<https://www.adobe.com/in/products/xd.html>

For this app, we are creating 6 images – 1st image Dice with one dot, 2nd image Dice with 2 dot so on.

Once all the 6 images are created we are exporting those images one by one from Abode XD to our Assets folders.

For our app, we downloaded these images from the assets website provided in this course.

<https://learncodeonline.in/androidp/>

In the course assets we were also provided with the Adobe xd image also, so we should them to change the color of dice to RED in Abode xd and then exported them to our assets for creating the app.

Now, 6 dice images are available for us and we copied them and pasted in the drawable folder in Res.

Now next is the UI design for the Dice Roller app:

Requirements:

We want to Two Red dices images to be displayed

One button with onclick listener which will created as a roller to roll the dice

One Textview to display the app name

Red is the theme required.

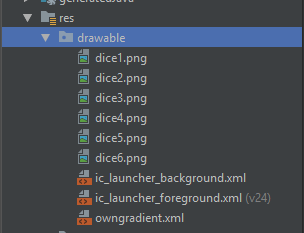
Our design coding below:

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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">  
  
 <Button  
 android:id="@+id/button"  
 style="@style/Widget.AppCompat.Button.Colored"  
 android:layout\_width="345dp"  
 android:layout\_height="54dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:layout\_weight="1"  
 android:background="@drawable/owngradient"  
 android:text=" Roll Your Dice"  
 android:textAlignment="center"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/linearLayout" />  
  
 <LinearLayout  
 android:id="@+id/linearLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="310dp"  
 android:layout\_marginTop="116dp"  
 android:orientation="horizontal"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:layout\_editor\_absoluteX="0dp">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 app:srcCompat="@drawable/dice1" />  
  
 <ImageView  
 android:id="@+id/imageView1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 app:srcCompat="@drawable/dice1" />  
 </LinearLayout>  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="396dp"  
 android:layout\_height="39dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:background="@drawable/owngradient"  
 android:text="DiceRoller App"  
 android:textAlignment="center"  
 android:textAppearance="@style/TextAppearance.AppCompat.Caption"  
 android:textColor="#FFFFFF"  
 android:textSize="30sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toTopOf="@+id/linearLayout"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
</android.support.constraint.ConstraintLayout>

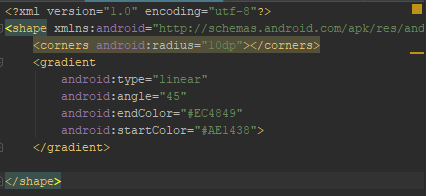
Colors looked as below:

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="colorPrimary">#FF3E4D</color>  
 <color name="colorPrimaryDark">#192A56</color>  
 <color name="colorAccent">#D81B60</color>  
</resources>

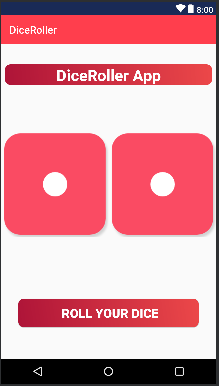
Our Res folder looked as below:



Our owngradient looked as below:



Our full UI looked as below:



Coding part:

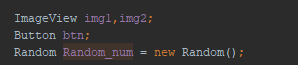
Now, we are creating a Onclick listener for the button and calling a method we created for the roller.

Roller method should generate random number between 1 to 6 and based on the number generated, dice image should be changed and displayed for both the dice images.

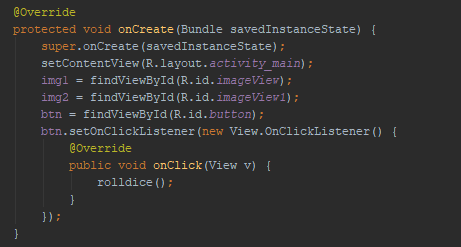
Random is the class used for the Random number between 1 to 6

Method used is obj.nextInt(int)

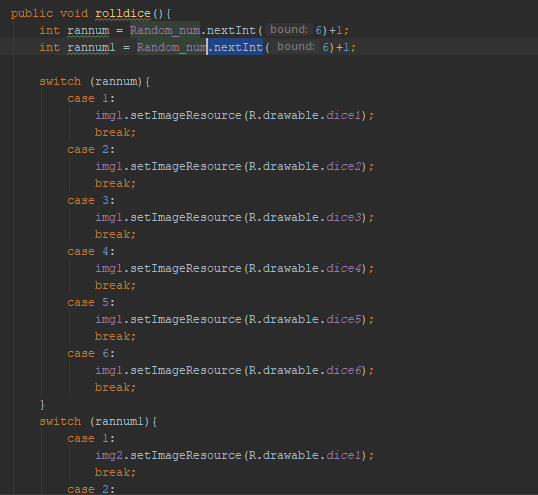
Variables and Class initialization:



Oncreate method with Onclick Listener for the button:



Method we created for Roller:



Full code below:

package com.mr71116.diceroller;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
  
import java.util.Random;  
  
public class MainActivity extends AppCompatActivity {  
  
 ImageView img1,img2;  
 Button btn;  
 Random Random\_num = new Random();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 img1 = findViewById(R.id.*imageView*);  
 img2 = findViewById(R.id.*imageView1*);  
 btn = findViewById(R.id.*button*);  
 btn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 rolldice();  
 }  
 });  
 }  
  
 public void rolldice(){  
 int rannum = Random\_num.nextInt(6)+1;  
 int rannum1 = Random\_num.nextInt(6)+1;  
  
 switch (rannum){  
 case 1:  
 img1.setImageResource(R.drawable.*dice1*);  
 break;  
 case 2:  
 img1.setImageResource(R.drawable.*dice2*);  
 break;  
 case 3:  
 img1.setImageResource(R.drawable.*dice3*);  
 break;  
 case 4:  
 img1.setImageResource(R.drawable.*dice4*);  
 break;  
 case 5:  
 img1.setImageResource(R.drawable.*dice5*);  
 break;  
 case 6:  
 img1.setImageResource(R.drawable.*dice6*);  
 break;  
 }  
 switch (rannum1){  
 case 1:  
 img2.setImageResource(R.drawable.*dice1*);  
 break;  
 case 2:  
 img2.setImageResource(R.drawable.*dice2*);  
 break;  
 case 3:  
 img2.setImageResource(R.drawable.*dice3*);  
 break;  
 case 4:  
 img2.setImageResource(R.drawable.*dice4*);  
 break;  
 case 5:  
 img2.setImageResource(R.drawable.*dice5*);  
 break;  
 case 6:  
 img2.setImageResource(R.drawable.*dice6*);  
 break;  
 }  
 }  
}

Fun Background App:

I have designed App background changes in two ways:

1. Via Context Options Menu and while clicking on the options – app background color will change.
2. Next is when a Button is clicked, app background color will change automatically

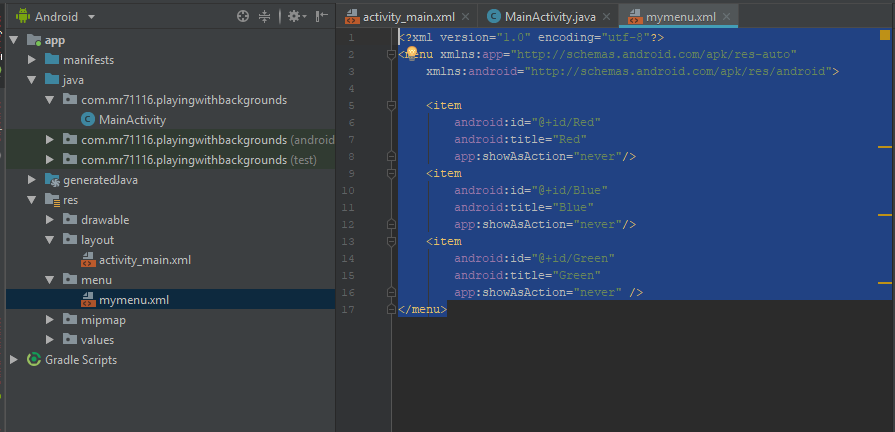
Method1:

Created a new folder in layout and named it as mymenu and draped and dropped 3 menu items

Gave their attirubutes values like

Id, title, showasaction etc

Now, Options menu is ready and using java code we can inflate this menu bar in the main activity.



Java code:

package com.mr71116.playingwithbackgrounds;  
  
  
  
import android.graphics.Color;  
import android.support.constraint.ConstraintLayout;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
  
import java.util.Random;  
  
  
public class MainActivity extends AppCompatActivity {  
 ConstraintLayout la;  
 Button B;  
  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 la = findViewById(R.id.*activity\_mainlayout*);  
 B=findViewById(R.id.*button*);  
 B.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Random R= new Random();  
 int mycolor = Color.*argb*(255,R.nextInt(255),R.nextInt(255),R.nextInt(255));  
 la.setBackgroundColor(mycolor);  
 }  
 });  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*mymenu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 switch (item.getItemId()) {  
 case R.id.*Red*:  
 changebackgroundRed();  
 break;  
 case R.id.*Blue*:  
 changebackgroundBlue();  
 break;  
 case R.id.*Green*:  
 changebackgroundGreen();  
 break;  
 }  
 return true;  
  
 }  
  
 public void changebackgroundRed ()  
 {  
 la.setBackgroundColor(Color.*RED*);  
 }  
 public void changebackgroundBlue ()  
 {  
 la.setBackgroundColor(Color.*BLUE*);  
 }  
 public void changebackgroundGreen ()  
 {  
 la.setBackgroundColor(Color.*GREEN*);  
 }  
  
  
 }

We declared the Constraint layout “la”

ConstraintLayout la;

We used the two methods related to Options menu

One to display and one to handle on click

public boolean onCreateOptionsMenu(Menu menu) {

public boolean onOptionsItemSelected(MenuItem item) {

Also, we are creating a new method for changing the background color in the layout and calling these methods in OnOptions item selected method

Code below:

@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*mymenu*, menu);  
 return true;  
}  
  
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
 switch (item.getItemId()) {  
 case R.id.*Red*:  
 changebackgroundRed();  
 break;  
 case R.id.*Blue*:  
 changebackgroundBlue();  
 break;  
 case R.id.*Green*:  
 changebackgroundGreen();  
 break;  
 }  
 return true;  
  
}  
  
 public void changebackgroundRed ()  
 {  
 la.setBackgroundColor(Color.*RED*);  
 }  
 public void changebackgroundBlue ()  
 {  
 la.setBackgroundColor(Color.*BLUE*);  
 }  
 public void changebackgroundGreen ()  
 {  
 la.setBackgroundColor(Color.*GREEN*);  
 }

Method 2:

Random background color change:

One button designed in the layout and we are going to change the background color randomly while we click on the button.

Declaring the layout and the button

public class MainActivity extends AppCompatActivity {  
 ConstraintLayout la;  
 Button B;

New methods used:

Random

Color.nextInt

SetBackground(int)

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 la = findViewById(R.id.*activity\_mainlayout*);  
 B=findViewById(R.id.*button*);  
 B.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Random R= new Random();  
 int mycolor = Color.*argb*(255,R.nextInt(255),R.nextInt(255),R.nextInt(255));  
 la.setBackgroundColor(mycolor);  
 }  
 });  
}

Full code: package com.mr71116.playingwithbackgrounds;  
  
  
  
import android.graphics.Color;  
import android.support.constraint.ConstraintLayout;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
  
import java.util.Random;  
  
  
public class MainActivity extends AppCompatActivity {  
 ConstraintLayout la;  
 Button B;  
  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 la = findViewById(R.id.*activity\_mainlayout*);  
 B=findViewById(R.id.*button*);  
 B.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Random R= new Random();  
 int mycolor = Color.*argb*(255,R.nextInt(255),R.nextInt(255),R.nextInt(255));  
 la.setBackgroundColor(mycolor);  
 }  
 });  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*mymenu*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 switch (item.getItemId()) {  
 case R.id.*Red*:  
 changebackgroundRed();  
 break;  
 case R.id.*Blue*:  
 changebackgroundBlue();  
 break;  
 case R.id.*Green*:  
 changebackgroundGreen();  
 break;  
 }  
 return true;  
  
 }  
  
 public void changebackgroundRed ()  
 {  
 la.setBackgroundColor(Color.*RED*);  
 }  
 public void changebackgroundBlue ()  
 {  
 la.setBackgroundColor(Color.*BLUE*);  
 }  
 public void changebackgroundGreen ()  
 {  
 la.setBackgroundColor(Color.*GREEN*);  
 }  
  
  
 }

Activity main xml:

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 android:id="@+id/activity\_mainlayout"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 android:textAlignment="center"  
 android:textSize="30sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/button"  
 style="@style/Widget.AppCompat.Button.Colored"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="152dp"  
 android:layout\_marginEnd="11dp"  
 android:layout\_marginBottom="8dp"  
 android:text="Button"  
 android:textAlignment="center"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView" />  
  
</android.support.constraint.ConstraintLayout>

Menu xml:

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:app="http://schemas.android.com/apk/res-auto"  
xmlns:android="http://schemas.android.com/apk/res/android">  
  
<item  
 android:id="@+id/Red"  
 android:title="Red"  
 app:showAsAction="never"/>  
<item  
 android:id="@+id/Blue"  
 android:title="Blue"  
 app:showAsAction="never"/>  
<item  
 android:id="@+id/Green"  
 android:title="Green"  
 app:showAsAction="never" />  
</menu>

Well done!

Animations:

This section we are going to cover how we can add images to our app as background and how we can make them look animated (images keep changing at the background).

We are going to take 3 images for this example.

These images are downloaded from Assets files given for this course and saved in

C:\Users\Mithra\AndroidStudioProjects\Android P Projects\Assets\Animated app\AnimatedBGAssets

1. Copy these images and paste them under drawable folder under Res
2. Create a new Drawable Resource folder and name it as animatedbackground.xml
3. In that animatedbackground.xml, change the first tag to have it has animated-list as below:

<animation-list xmlns:android="http://schemas.android.com/apk/res/android">

Add your 3 images here as one items each and fill their attributes 1. Drawable and 2. Duration

<item  
 android:drawable="@drawable/one"  
 android:duration="4000"/>  
<item  
 android:drawable="@drawable/two"  
 android:duration="4000"/>  
<item  
 android:drawable="@drawable/three"  
 android:duration="4000"/>

Full code of the animatedbackground.xml below:

<?xml version="1.0" encoding="utf-8"?>  
<animation-list xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item  
 android:drawable="@drawable/one"  
 android:duration="4000"/>  
 <item  
 android:drawable="@drawable/two"  
 android:duration="4000"/>  
 <item  
 android:drawable="@drawable/three"  
 android:duration="4000"/>  
</animation-list>

Now, go to activity\_main.xml file

In the Text view, add the below two

1. Background and have it as the name of the xml file newly created ie. Animatedbackground.xml
2. Provide a id to the layout so that we can use it in the java file

android:background="@drawable/animatedbackground"  
android:id="@+id/layout"

we also made few customizations on the button.

Full code of the activity\_main.xml below:

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 android:background="@drawable/animatedbackground"  
 android:id="@+id/layout"  
 tools:context=".MainActivity">  
  
 <Button  
 style="@style/Widget.AppCompat.Button.Colored"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:text="Login"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.909" />  
  
</android.support.constraint.ConstraintLayout>

Now, java coding for background images animation:

First, declare the constraint layout

ConstraintLayout l;

Then,

Assign the layout details to the declared constraint layout

l=findViewById(R.id.*layout*);

Animation coding below:

First line is assigning the animation object Ani to the layout(l) background.

.setEnterDuration

.setExitDuration

.Start();

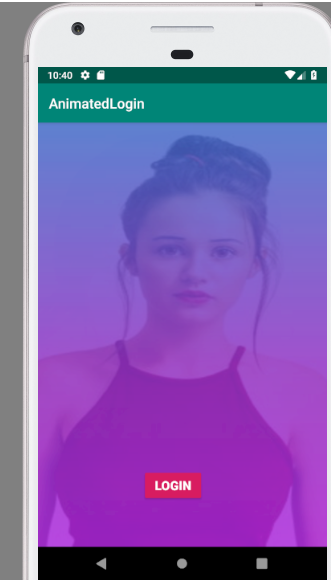
Above are three animations attributes available in AnimationDrawable class.

AnimationDrawable Ani = (AnimationDrawable) l.getBackground();  
Ani.setEnterFadeDuration(2000);  
Ani.setExitFadeDuration(3000);  
Ani.start();

Full code of the MainActivity.java below:

package com.mr71116.animatedlogin;  
  
import android.graphics.drawable.AnimationDrawable;  
import android.support.constraint.ConstraintLayout;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
  
public class MainActivity extends AppCompatActivity {  
 ConstraintLayout l;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 l=findViewById(R.id.*layout*);  
 AnimationDrawable Ani = (AnimationDrawable) l.getBackground();  
 Ani.setEnterFadeDuration(2000);  
 Ani.setExitFadeDuration(3000);  
 Ani.start();  
 }  
}

Our App looks like below and background changes as per the duration given in the animatedbackground.xml and it fades while entering and faces while exit based on the duration given in the Mainactivity.java



Button customization:

Create a new drawable folder and write this coding:

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <corners  
 android:radius="10dp"/>  
  
 <padding  
 android:left="2dp"  
 android:top="2dp"  
 android:bottom="2dp"  
 android:right="2dp"/>  
 <solid  
 android:color="#FFFFFFFF"/>  
 <size  
 android:width="270dp"  
 android:height="60dp"/>  
  
</shape>

Attributes for button are

Padding

Solid/Gradient

Corners

Size

Button looks like below for the above code



Button Customization and Button cycle (clicked, default, disabled):

From many learnings till now, button can be designed in following ways:

1. Add a button in activity.main.xml design view and write coding in the text view

Example: button style, text color, textstyle=bold, text size, padding, height, width, background, background tint etc.

1. Second one is create a new drawable folder and input all design things there and post design this drawable folder, go to the activity\_main.xml folder and in the buttonview, mention background as the @drawable/filename of the drawable xml file created for button design.

Example of the drawable folder created for button design and how it is used in the activity\_main.xml is below:

Button attributes are:

Corners – android:radius

Padding – top, left, right, bottom

Gradient – angle, startcolor, endcolor

Stroke – width, color

Size – width, height

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <corners  
 android:radius="10dp"/>  
  
 <padding  
 android:left="2dp"  
 android:top="2dp"  
 android:bottom="2dp"  
 android:right="2dp"/>  
 <gradient  
 android:angle="360"  
 android:startColor="@color/colorPrimaryDark"  
 android:endColor="@color/colorPrimary"/>  
 <stroke  
 android:width="2dp"  
 android:color="@color/colorAccent"/>  
  
 <size  
 android:width="270dp"  
 android:height="60dp"/>  
  
</shape>

This is how this drawabe button xml file was used in the activity\_main.xml file under buttonview

android:background="@drawable/buttonlist"

1. Last way is, create a style for button in the style folder so that any time you create a button, button will be created in the said style

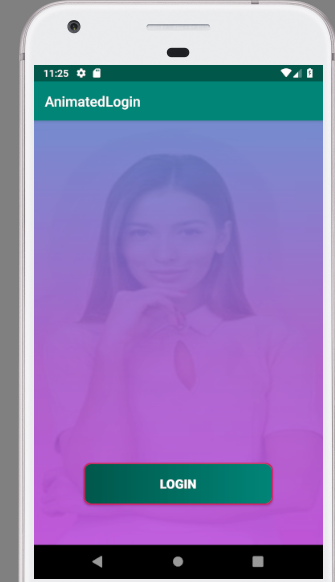
Steps are create a new style in the style.xml file

Add this style name in the main style coding as button style.

Style name: Button\_theme is the new style we created and we used that style in the main code of the style as below:

<resources>  
  
 <!-- Base application theme. -->  
 <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
 <!-- Customize your theme here. -->  
 <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="colorAccent">@color/colorAccent</item>  
 <item name="buttonStyle">@style/Button\_theme</item>  
 </style>  
  
  
 <style name="Button\_theme" parent="Widget.AppCompat.Button">  
 <item name="android:background">@drawable/button\_shape</item>  
 <item name="android:textColor">#FFFFFFFF</item>  
 <item name="android:textStyle">bold</item>  
 <item name="android:layout\_width">wrap\_content</item>  
 <item name="android:layout\_height">wrap\_content</item>  
 <item name="android:textSize">18sp</item>  
 </style>  
 </resources>

This is how our app looked like:



Next section is about the Button life cycle:

1. Clicked
2. Disabled
3. Default etc

In this example we are creating drawable folders for each stage of button

Button\_shape created for default look of the button

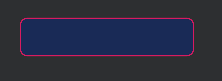
<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <corners  
 android:radius="10dp"/>  
  
 <padding  
 android:left="2dp"  
 android:top="2dp"  
 android:bottom="2dp"  
 android:right="2dp"/>  
 <gradient  
 android:angle="360"  
 android:startColor="@color/colorPrimaryDark"  
 android:endColor="@color/colorPrimary"/>  
 <stroke  
 android:width="2dp"  
 android:color="@color/colorAccent"/>  
  
 <size  
 android:width="270dp"  
 android:height="60dp"/>  
  
</shape>



Button\_clicked.xml created for look of a button when clicked:

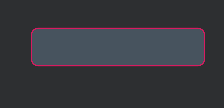
We have changed only the background from gradient to solid

<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <corners  
 android:radius="10dp"/>  
  
 <padding  
 android:left="2dp"  
 android:top="2dp"  
 android:bottom="2dp"  
 android:right="2dp"/>  
 <solid  
 android:color="#192A56"/>  
 <stroke  
 android:width="2dp"  
 android:color="@color/colorAccent"/>  
  
 <size  
 android:width="270dp"  
 android:height="60dp"/>  
  
</shape>



Next is Button\_disabled.xml created when button is in disabled state:

<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <corners  
 android:radius="10dp"/>  
  
 <padding  
 android:left="2dp"  
 android:top="2dp"  
 android:bottom="2dp"  
 android:right="2dp"/>  
 <solid  
 android:color="#47535E"/>  
 <stroke  
 android:width="2dp"  
 android:color="@color/colorAccent"/>  
  
 <size  
 android:width="270dp"  
 android:height="60dp"/>  
  
</shape>



Once these 3 xml files are created, we created another xml file for listing these states of button

Xml file created with name buttonlist.xml

<?xml version="1.0" encoding="utf-8"?>  
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item android:state\_pressed="true"  
 android:drawable="@drawable/button\_clicked"/>  
  
 <item android:state\_enabled="false"  
 android:drawable="@drawable/button\_disabled"/>  
  
 <item  
 android:drawable="@drawable/button\_shape"/>  
  
</selector>

After creating the above, we go to activity\_main.xml and for background of the button we use this buttonlist.xml

android:background="@drawable/buttonlist"

so when button is clicked, we will see a change in the color of the background etc.

Full code of this project below:

Activity\_main.xml:

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 android:background="@drawable/animatedbackground"  
 android:id="@+id/layout"  
 tools:context=".MainActivity">  
  
 <Button  
 style="@style/Widget.AppCompat.Button.Colored"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:text="Login"  
 android:textSize="18sp"  
 android:textColor="#FFFFFFFF"  
 android:textStyle="bold"  
 android:background="@drawable/buttonlist"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.909" />  
  
</android.support.constraint.ConstraintLayout>

Mainactivity.java

With Animations codes:

package com.mr71116.animatedlogin;  
  
import android.graphics.drawable.AnimationDrawable;  
import android.support.constraint.ConstraintLayout;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
  
public class MainActivity extends AppCompatActivity {  
 ConstraintLayout l;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 l=findViewById(R.id.*layout*);  
 AnimationDrawable Ani = (AnimationDrawable) l.getBackground();  
 Ani.setEnterFadeDuration(2000);  
 Ani.setExitFadeDuration(3000);  
 Ani.start();  
 }  
}

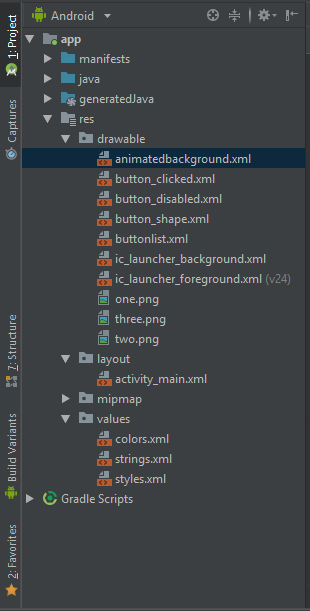
animatedbackground.xml

<?xml version="1.0" encoding="utf-8"?>  
<animation-list xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item  
 android:drawable="@drawable/one"  
 android:duration="4000"/>  
 <item  
 android:drawable="@drawable/two"  
 android:duration="4000"/>  
 <item  
 android:drawable="@drawable/three"  
 android:duration="4000"/>  
</animation-list>

Style.xml file:

<resources>  
  
 <!-- Base application theme. -->  
 <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
 <!-- Customize your theme here. -->  
 <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="colorAccent">@color/colorAccent</item>  
 <item name="buttonStyle">@style/Button\_theme</item>  
 </style>  
  
  
 <style name="Button\_theme" parent="Widget.AppCompat.Button">  
 <item name="android:background">@drawable/button\_shape</item>  
 <item name="android:textColor">#FFFFFFFF</item>  
 <item name="android:textStyle">bold</item>  
 <item name="android:layout\_width">wrap\_content</item>  
 <item name="android:layout\_height">wrap\_content</item>  
 <item name="android:textSize">18sp</item>  
 </style>  
 </resources>

Res:



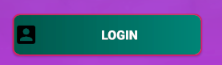
I have written to view the disabled state of the button in the Mainactivity.java

package com.mr71116.animatedlogin;  
  
import android.graphics.drawable.AnimationDrawable;  
import android.support.constraint.ConstraintLayout;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
  
import static java.lang.Thread.*sleep*;  
  
public class MainActivity extends AppCompatActivity {  
 ConstraintLayout l;  
 Button b;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 l=findViewById(R.id.*layout*);  
 AnimationDrawable Ani = (AnimationDrawable) l.getBackground();  
 Ani.setEnterFadeDuration(2000);  
 Ani.setExitFadeDuration(3000);  
 Ani.start();  
 b=findViewById(R.id.*button*);  
 ((View) b).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 ((Button) b).setEnabled(false);  
  
  
 }  
 });  
 }  
}

Post clicking the button, button will go to disabled state based on the below code:

ublic void onClick(View v) {  
 ((Button) b).setEnabled(false);

We can also add a image to the button, ie both text and image can be shown in a button as below:



Code for that should be added in the buttonview

android:drawableLeft="@drawable/ic\_account\_box\_black\_24dp"

I have used the vector asset and assigned the above button with image and text.

If we want to create button without borders:

style="?android:attr/borderlessButtonStyle"

We can use android color for any purpose like background color of button etc

Here is an example of android colors

"?android:attr/colorPressedHighlight" />

Truth or Dare game:

UI Design:

Download the images from the course asset links and have it stored in the laptop assets folder for android p

Two resources: 1. Bottle image and 2. Button customized code.

Copy and paste the bottle in the drawable folder

Drag and drop the button.xml document to sublime application to see the code of it.

Now create a new drawable folder for button\_custom and paste the full code from sublime (button design).

In the activity\_main.xml

First drag and drop a imageview and select bottle for this.

Next drag and drop a button.

Few botton attributes specifications:

Text: Spin

Background: @drawable/customer\_button

Textsize 18sp

Etc

UI is ready.

Full button\_customer code:

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle" >  
 <corners  
 android:topLeftRadius="35dp"  
 android:topRightRadius="35dp"  
 android:bottomLeftRadius="35dp"  
 android:bottomRightRadius="35dp"  
 />  
 <solid  
 android:color="#0A77DB"  
 />  
 <padding  
 android:left="0dp"  
 android:top="0dp"  
 android:right="0dp"  
 android:bottom="0dp"  
 />  
 <size  
 android:width="270dp"  
 android:height="60dp"  
 />  
 <stroke  
 android:width="3dp"  
 android:color="#FCFCFC"  
 />  
</shape>

Also change the colors of the action bar and back values -> colors of the app in colors.xml

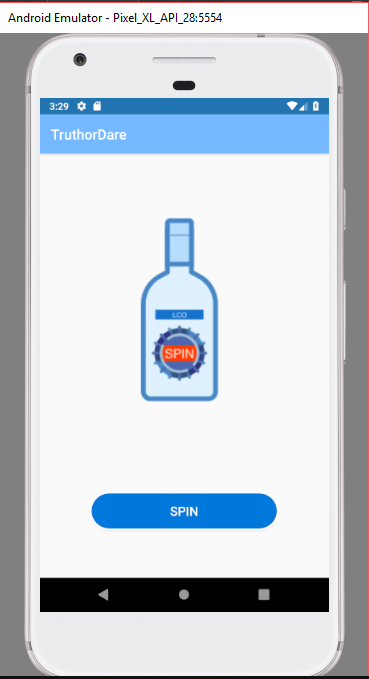
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="colorPrimary">#74B9FF</color>  
 <color name="colorPrimaryDark">#2475B0</color>  
 <color name="colorAccent">#D81B60</color>  
</resources>

Activity\_main.xml code below:

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="185dp"  
 android:layout\_height="300dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="76dp"  
 android:layout\_marginEnd="8dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/spin" />  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="56dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:background="@drawable/custom\_button"  
 android:text="Spin"  
 android:textColor="#FFFFFFFF"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView"  
 app:layout\_constraintVertical\_bias="0.619" />  
</android.support.constraint.ConstraintLayout>

In this app, while designing the button, it didn’t show any color here in the activity\_main.xml but while we run the app, in the emulator it showed the actual color and other designs there.

This is how our UI of the Truth or Dare looked:



Awesome!

Next is the code of this:

True or dare code:

We are using Random class again here.

RotateAnimation method takes 4 parameters.

1. LastDirection
2. NewDirection
3. Pivot x
4. Pivot y

Pivot x and y determines the position of the image from which it should rotate and in our example we are rotating from the center of the image.

We are going to use Random number for new direction. This new direction helps to determine how much times the bottle should spin. We can make sure bottle spins atleast once by adding 360 to the random number generated.

Pivot x and pivot y we are getting values by the below code.

Imageview.Getwidth/2

Imageview. Getheight/2

Code for Rotation:

Animation Ani = new RotateAnimation(lastdir,newdir,pivotx,pivoty);  
Ani.setDuration(5000);  
Ani.setFillAfter(true);

First line is for initiatlizing object for the RotateAnimation and we are passing the 4 paramters.

Second line is for setting the duration of the rotation.

Thirdline is for showing the image after animation.

We also have animation listencer to handle the animation efficiently. Like during the animation we can disable the button click once again etc.

We also changed the button color during the animation using the drawable folder like button\_disabled and design the button attributes for this drawable folder and then creating another new drawable folder called buttonlist.xml and listing these drawable folders as item – button state pressed and button state disabled etc.

Refer the previous lesson for more details about the button life cycle.

3 methods are available for animation listener:

public void onAnimationStart(Animation animation) {

ublic void onAnimationEnd(Animation animation) {

public void onAnimationRepeat(Animation animation) {

IN our example, we have used first 2 methods;

Full java code below:

package com.mr71116.truthordare;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.view.animation.Animation;  
import android.view.animation.RotateAnimation;  
import android.widget.Button;  
import android.widget.ImageView;  
  
import java.util.Random;  
  
public class MainActivity extends AppCompatActivity {  
 private Button B;  
 private ImageView I;  
 private Random Ran = new Random();  
 private int lastdir;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 B=findViewById(R.id.*button*);  
 I= findViewById(R.id.*imageView*);  
 }  
  
 public void Spin(View view) {  
 int newdir = Ran.nextInt(3600)+2000;  
 float pivotx = I.getWidth()/2;  
 float pivoty = I.getHeight()/2;  
 Animation Ani = new RotateAnimation(lastdir,newdir,pivotx,pivoty);  
 Ani.setDuration(5000);  
 Ani.setFillAfter(true);  
 Ani.setAnimationListener(new Animation.AnimationListener() {  
 @Override  
 public void onAnimationStart(Animation animation) {  
 B.setEnabled(false);  
 }  
  
 @Override  
 public void onAnimationEnd(Animation animation) {  
 B.setEnabled(true);  
 }  
  
 @Override  
 public void onAnimationRepeat(Animation animation) {  
  
 }  
 });  
  
 I.startAnimation(Ani);  
  
 }  
}

Activity\_main.xml:

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="185dp"  
 android:layout\_height="300dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="76dp"  
 android:layout\_marginEnd="8dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/spin" />  
  
 <Button  
 android:id="@+id/button"  
 android:onClick="Spin"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="56dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:background="@drawable/buttonlist"  
 android:text="Spin"  
 android:textColor="#FFFFFFFF"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView"  
 app:layout\_constraintVertical\_bias="0.619" />  
</android.support.constraint.ConstraintLayout>

Button:

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle" >  
 <corners  
 android:topLeftRadius="35dp"  
 android:topRightRadius="35dp"  
 android:bottomLeftRadius="35dp"  
 android:bottomRightRadius="35dp"  
 />  
 <solid  
 android:color="#0A77DB"  
 />  
 <padding  
 android:left="0dp"  
 android:top="0dp"  
 android:right="0dp"  
 android:bottom="0dp"  
 />  
 <size  
 android:width="270dp"  
 android:height="60dp"  
 />  
 <stroke  
 android:width="3dp"  
 android:color="#FCFCFC"  
 />  
</shape>

Color.xml:

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="colorPrimary">#74B9FF</color>  
 <color name="colorPrimaryDark">#2475B0</color>  
 <color name="colorAccent">#D81B60</color>  
</resources>

Button\_clicked.xml

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle" >  
 <corners  
 android:topLeftRadius="35dp"  
 android:topRightRadius="35dp"  
 android:bottomLeftRadius="35dp"  
 android:bottomRightRadius="35dp"  
 />  
 <solid  
 android:color="#218F76"  
 />  
 <padding  
 android:left="0dp"  
 android:top="0dp"  
 android:right="0dp"  
 android:bottom="0dp"  
 />  
 <size  
 android:width="270dp"  
 android:height="60dp"  
 />  
 <stroke  
 android:width="3dp"  
 android:color="#FCFCFC"  
 />  
</shape>

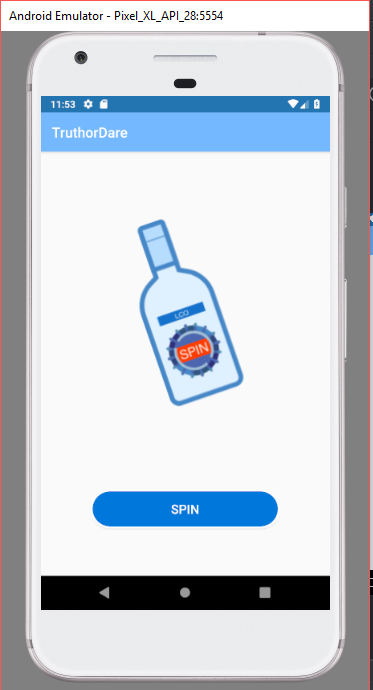
Button\_disabled:

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle" >  
 <corners  
 android:topLeftRadius="35dp"  
 android:topRightRadius="35dp"  
 android:bottomLeftRadius="35dp"  
 android:bottomRightRadius="35dp"  
 />  
 <solid  
 android:color="#616C6F"  
 />  
 <padding  
 android:left="0dp"  
 android:top="0dp"  
 android:right="0dp"  
 android:bottom="0dp"  
 />  
 <size  
 android:width="270dp"  
 android:height="60dp"  
 />  
 <stroke  
 android:width="3dp"  
 android:color="#FCFCFC"  
 />  
</shape>

Button\_list.xml:

<?xml version="1.0" encoding="utf-8"?>  
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item android:state\_pressed="true"  
 android:drawable="@drawable/button\_clicked"/>  
  
 <item android:state\_enabled="false"  
 android:drawable="@drawable/button\_disabled"/>  
  
 <item  
 android:drawable="@drawable/custom\_button"/>  
  
  
</selector>

Our app looked like below:



Country selector, autocomplete text view:

Resources downloaded from the assets provided in the course.

Only one resource: full countries list as array.

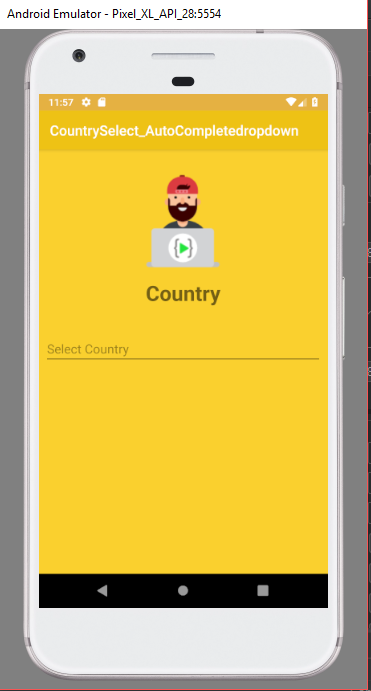
Another resource is the email.png.

We can use the sublime to drag and drop the countries list xml resource file to get the code.

Autocomplete Text view is used here.

Yellowish theme

Our app looked like below:



Activity\_main.xml code:

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 android:background="#FAD02E"  
 tools:context=".MainActivity">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="170dp"  
 android:layout\_height="154dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="24dp"  
 android:layout\_marginEnd="8dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/email" />  
  
 <AutoCompleteTextView  
 android:id="@+id/autoCompleteTextView"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="32dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="300dp"  
 android:completionHint="Select Country"  
 android:hint="Select Country"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView" />  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:text="Country"  
 android:textAlignment="center"  
 android:textSize="30sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toTopOf="@+id/autoCompleteTextView"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView"  
 app:layout\_constraintVertical\_bias="0.6" />  
  
</android.support.constraint.ConstraintLayout>

Java code below:

package com.mr71116.countryselect\_autocompletedropdown;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.widget.Adapter;  
import android.widget.ArrayAdapter;  
import android.widget.AutoCompleteTextView;  
  
public class MainActivity extends AppCompatActivity {  
 String[] countries = new String[]{"Afghanistan", "Albania", "Algeria", "American Samoa", "Andorra", "Angola", "Anguilla", "Antarctica", "Antigua and Barbuda", "Argentina", "Armenia", "Aruba", "Australia", "Austria", "Azerbaijan", "Bahamas", "Bahrain", "Bangladesh", "Barbados", "Belarus", "Belgium", "Belize", "Benin", "Bermuda", "Bhutan", "Bolivia", "Bosnia and Herzegowina", "Botswana", "Bouvet Island", "Brazil", "British Indian Ocean Territory", "Brunei Darussalam", "Bulgaria", "Burkina Faso", "Burundi", "Cambodia", "Cameroon", "Canada", "Cape Verde", "Cayman Islands", "Central African Republic", "Chad", "Chile", "China", "Christmas Island", "Cocos (Keeling) Islands", "Colombia", "Comoros", "Congo", "Congo, the Democratic Republic of the", "Cook Islands", "Costa Rica", "Cote d'Ivoire", "Croatia (Hrvatska)", "Cuba", "Cyprus", "Czech Republic", "Denmark", "Djibouti", "Dominica", "Dominican Republic", "East Timor", "Ecuador", "Egypt", "El Salvador", "Equatorial Guinea", "Eritrea", "Estonia", "Ethiopia", "Falkland Islands (Malvinas)", "Faroe Islands", "Fiji", "Finland", "France", "France Metropolitan", "French Guiana", "French Polynesia", "French Southern Territories", "Gabon", "Gambia", "Georgia", "Germany", "Ghana", "Gibraltar", "Greece", "Greenland", "Grenada", "Guadeloupe", "Guam", "Guatemala", "Guinea", "Guinea-Bissau", "Guyana", "Haiti", "Heard and Mc Donald Islands", "Holy See (Vatican City State)", "Honduras", "Hong Kong", "Hungary", "Iceland", "India", "Indonesia", "Iran (Islamic Republic of)", "Iraq", "Ireland", "Israel", "Italy", "Jamaica", "Japan", "Jordan", "Kazakhstan", "Kenya", "Kiribati", "Korea, Democratic People's Republic of", "Korea, Republic of", "Kuwait", "Kyrgyzstan", "Lao, People's Democratic Republic", "Latvia", "Lebanon", "Lesotho", "Liberia", "Libyan Arab Jamahiriya", "Liechtenstein", "Lithuania", "Luxembourg", "Macau", "Macedonia, The Former Yugoslav Republic of", "Madagascar", "Malawi", "Malaysia", "Maldives", "Mali", "Malta", "Marshall Islands", "Martinique", "Mauritania", "Mauritius", "Mayotte", "Mexico", "Micronesia, Federated States of", "Moldova, Republic of", "Monaco", "Mongolia", "Montserrat", "Morocco", "Mozambique", "Myanmar", "Namibia", "Nauru", "Nepal", "Netherlands", "Netherlands Antilles", "New Caledonia", "New Zealand", "Nicaragua", "Niger", "Nigeria", "Niue", "Norfolk Island", "Northern Mariana Islands", "Norway", "Oman", "Pakistan", "Palau", "Panama", "Papua New Guinea", "Paraguay", "Peru", "Philippines", "Pitcairn", "Poland", "Portugal", "Puerto Rico", "Qatar", "Reunion", "Romania", "Russian Federation", "Rwanda", "Saint Kitts and Nevis", "Saint Lucia", "Saint Vincent and the Grenadines", "Samoa", "San Marino", "Sao Tome and Principe", "Saudi Arabia", "Senegal", "Seychelles", "Sierra Leone", "Singapore", "Slovakia (Slovak Republic)", "Slovenia", "Solomon Islands", "Somalia", "South Africa", "South Georgia and the South Sandwich Islands", "Spain", "Sri Lanka", "St. Helena", "St. Pierre and Miquelon", "Sudan", "Suriname", "Svalbard and Jan Mayen Islands", "Swaziland", "Sweden", "Switzerland", "Syrian Arab Republic", "Taiwan, Province of China", "Tajikistan", "Tanzania, United Republic of", "Thailand", "Togo", "Tokelau", "Tonga", "Trinidad and Tobago", "Tunisia", "Turkey", "Turkmenistan", "Turks and Caicos Islands", "Tuvalu", "Uganda", "Ukraine", "United Arab Emirates", "United Kingdom", "United States", "United States Minor Outlying Islands", "Uruguay", "Uzbekistan", "Vanuatu", "Venezuela", "Vietnam", "Virgin Islands (British)", "Virgin Islands (U.S.)", "Wallis and Futuna Islands", "Western Sahara", "Yemen", "Yugoslavia", "Zambia", "Zimbabwe"};  
 AutoCompleteTextView a;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 a= findViewById(R.id.*autoCompleteTextView*);  
 ArrayAdapter<String> adop = new ArrayAdapter<>(MainActivity.this, android.R.layout.*simple\_list\_item\_1*, countries);  
 a.setAdapter(adop);  
 }  
}

Resource of the countries list in array code:

String[] countries = new String[]{"Afghanistan", "Albania", "Algeria", "American Samoa", "Andorra", "Angola", "Anguilla", "Antarctica", "Antigua and Barbuda", "Argentina", "Armenia", "Aruba", "Australia", "Austria", "Azerbaijan", "Bahamas", "Bahrain", "Bangladesh", "Barbados", "Belarus", "Belgium", "Belize", "Benin", "Bermuda", "Bhutan", "Bolivia", "Bosnia and Herzegowina", "Botswana", "Bouvet Island", "Brazil", "British Indian Ocean Territory", "Brunei Darussalam", "Bulgaria", "Burkina Faso", "Burundi", "Cambodia", "Cameroon", "Canada", "Cape Verde", "Cayman Islands", "Central African Republic", "Chad", "Chile", "China", "Christmas Island", "Cocos (Keeling) Islands", "Colombia", "Comoros", "Congo", "Congo, the Democratic Republic of the", "Cook Islands", "Costa Rica", "Cote d'Ivoire", "Croatia (Hrvatska)", "Cuba", "Cyprus", "Czech Republic", "Denmark", "Djibouti", "Dominica", "Dominican Republic", "East Timor", "Ecuador", "Egypt", "El Salvador", "Equatorial Guinea", "Eritrea", "Estonia", "Ethiopia", "Falkland Islands (Malvinas)", "Faroe Islands", "Fiji", "Finland", "France", "France Metropolitan", "French Guiana", "French Polynesia", "French Southern Territories", "Gabon", "Gambia", "Georgia", "Germany", "Ghana", "Gibraltar", "Greece", "Greenland", "Grenada", "Guadeloupe", "Guam", "Guatemala", "Guinea", "Guinea-Bissau", "Guyana", "Haiti", "Heard and Mc Donald Islands", "Holy See (Vatican City State)", "Honduras", "Hong Kong", "Hungary", "Iceland", "India", "Indonesia", "Iran (Islamic Republic of)", "Iraq", "Ireland", "Israel", "Italy", "Jamaica", "Japan", "Jordan", "Kazakhstan", "Kenya", "Kiribati", "Korea, Democratic People's Republic of", "Korea, Republic of", "Kuwait", "Kyrgyzstan", "Lao, People's Democratic Republic", "Latvia", "Lebanon", "Lesotho", "Liberia", "Libyan Arab Jamahiriya", "Liechtenstein", "Lithuania", "Luxembourg", "Macau", "Macedonia, The Former Yugoslav Republic of", "Madagascar", "Malawi", "Malaysia", "Maldives", "Mali", "Malta", "Marshall Islands", "Martinique", "Mauritania", "Mauritius", "Mayotte", "Mexico", "Micronesia, Federated States of", "Moldova, Republic of", "Monaco", "Mongolia", "Montserrat", "Morocco", "Mozambique", "Myanmar", "Namibia", "Nauru", "Nepal", "Netherlands", "Netherlands Antilles", "New Caledonia", "New Zealand", "Nicaragua", "Niger", "Nigeria", "Niue", "Norfolk Island", "Northern Mariana Islands", "Norway", "Oman", "Pakistan", "Palau", "Panama", "Papua New Guinea", "Paraguay", "Peru", "Philippines", "Pitcairn", "Poland", "Portugal", "Puerto Rico", "Qatar", "Reunion", "Romania", "Russian Federation", "Rwanda", "Saint Kitts and Nevis", "Saint Lucia", "Saint Vincent and the Grenadines", "Samoa", "San Marino", "Sao Tome and Principe", "Saudi Arabia", "Senegal", "Seychelles", "Sierra Leone", "Singapore", "Slovakia (Slovak Republic)", "Slovenia", "Solomon Islands", "Somalia", "South Africa", "South Georgia and the South Sandwich Islands", "Spain", "Sri Lanka", "St. Helena", "St. Pierre and Miquelon", "Sudan", "Suriname", "Svalbard and Jan Mayen Islands", "Swaziland", "Sweden", "Switzerland", "Syrian Arab Republic", "Taiwan, Province of China", "Tajikistan", "Tanzania, United Republic of", "Thailand", "Togo", "Tokelau", "Tonga", "Trinidad and Tobago", "Tunisia", "Turkey", "Turkmenistan", "Turks and Caicos Islands", "Tuvalu", "Uganda", "Ukraine", "United Arab Emirates", "United Kingdom", "United States", "United States Minor Outlying Islands", "Uruguay", "Uzbekistan", "Vanuatu", "Venezuela", "Vietnam", "Virgin Islands (British)", "Virgin Islands (U.S.)", "Wallis and Futuna Islands", "Western Sahara", "Yemen", "Yugoslavia", "Zambia", "Zimbabwe"};

ArrayAdopter is used here for listing these values and providing autocomplete suggestion

ArrayAdapter<String> adop = new ArrayAdapter<>(MainActivity.this, android.R.layout.*simple\_list\_item\_1*, countries);  
a.setAdapter(adop);

Arrayadopter has 3 parameters

1. Context – Mainactivity.this
2. Type of the list (here we have the android list) android.R.layout.*simple\_list\_item\_1*
3. Actual list (the values) – countries

Final line is autocomplete Textview.setadopter(adopter obj)

a.setAdapter(adop);

As assignment we are going to use the same concept and create a autocomplete text view for list of games.

All same, instead of countries, games should be listed:

Java code for the assignment is:

package com.mr71116.gameselector\_autocomplete;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.widget.Adapter;  
import android.widget.ArrayAdapter;  
import android.widget.AutoCompleteTextView;  
  
public class MainActivity extends AppCompatActivity {  
 String[] games = new String[]{"Game of Thornes", "PUBG", "Candy Crush", "Mario"};  
 AutoCompleteTextView a;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 a= findViewById(R.id.*autoCompleteTextView*);  
 ArrayAdapter<String> adop = new ArrayAdapter<String>(MainActivity.this,android.R.layout.*simple\_list\_item\_1*,games);  
 a.setAdapter(adop);  
 }  
}

Color best practice:

Dark color for back of the app:

<color name="colorPrimaryDark">#2C3335</color>

Greyish background

android:background="#777E8B"

another grey background:

android:background="#99AAAB"

for blueish color theme:

<color name="colorPrimaryDark">#74B9FF</color>  
<color name="colorAccent">#3C40C6</color>

Customized Rating bar:

Color When normal

<color name="lightgrey">#DAE0E2</color>

Color When rating is provided

<color name="duskyyellow">#FAD02E</color>

<style name="RatingBar" parent="Theme.AppCompat">  
 <item name="colorControlNormal">@color/lightgrey</item>  
 <item name="colorControlActivated">@color/duskyyellow</item>  
  
</style>

In Rating bar button add this theme:

android:theme="@style/RatingBar"

Burger rating bar:

When rating is provided we are using the if condition to display the content for the text view based on the rating provided.

Java code below:

package com.mr71116.burger\_rating;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.RatingBar;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity {  
  
 RatingBar Rate;  
 TextView T;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 Rate = findViewById(R.id.*ratingBar*);  
 T = findViewById(R.id.*textView*);  
 }  
  
 public void Submit(View view) {  
 float ratevalue = Rate.getRating();  
 if(ratevalue <=2){  
 T.setText("The Rating is " +ratevalue+ "\nWe will try better next time");  
 }  
 else if (ratevalue <=3 && ratevalue >2){  
 T.setText("The Rating is " +ratevalue+ "\nWe are Constantly improving");  
 }  
 else if(ratevalue <=4 && ratevalue >3){  
 T.setText("The Rating is " +ratevalue+ "\nThanks and next time sure 5");  
 }  
 else if(ratevalue <=5 && ratevalue >4){  
 T.setText("The Rating is " +ratevalue+ "\nWe are delighted to hear that you are happy with the Service");  
 }  
 }  
}

New things:

Declaration:

RatingBar Rate;

New method for rating:

float ratevalue = Rate.getRating();

if condition:

if(ratevalue <=2){  
 T.setText("The Rating is " +ratevalue+ "\nWe will try better next time");  
}

to move the wordings to the next line

\n

Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 android:background="#777E8B"  
 tools:context=".MainActivity">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="218dp"  
 android:layout\_height="215dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="48dp"  
 android:layout\_marginEnd="8dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/burgerrating" />  
  
 <RatingBar  
 android:id="@+id/ratingBar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="48dp"  
 android:layout\_marginEnd="8dp"  
 android:theme="@style/RatingBar"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.496"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView" />  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="52dp"  
 android:layout\_marginEnd="8dp"  
 android:text=""  
 android:textSize="18dp"  
 android:textColor="#FFFFFFFF"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/ratingBar" />  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="40dp"  
 android:text="Submit"  
 android:onClick="Submit"  
 android:background="@drawable/burger\_button"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent" />  
</android.support.constraint.ConstraintLayout>

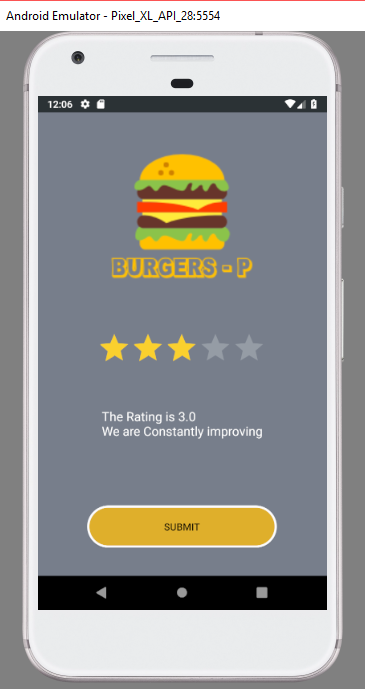
Styles:

New style defined for the rating bar color, when normal and when rating is provided.

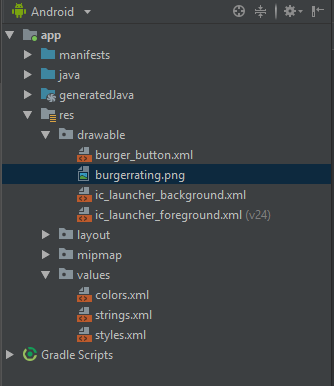
<resources>  
  
 <!-- Base application theme. -->  
 <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">  
 <!-- Customize your theme here. -->  
 <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="colorAccent">@color/colorAccent</item>  
 </style>  
  
 <style name="RatingBar" parent="Theme.AppCompat">  
 <item name="colorControlNormal">@color/lightgrey</item>  
 <item name="colorControlActivated">@color/duskyyellow</item>  
  
 </style>  
  
  
</resources>

Color.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="colorPrimary">#008577</color>  
 <color name="colorPrimaryDark">#2C3335</color>  
 <color name="colorAccent">#D81B60</color>  
 <color name="duskyyellow">#FAD02E</color>  
 <color name="lightgrey">#DAE0E2</color>  
  
</resources>



Folder structure:



Seek bar project:

Seek bar is like volume bar in a music app which allows us to change the volume by dragging left to right.

In this project, we are having a text view with value “Mithra” and customized background for the Textview with button\_custom.xml file.

Xml code below:

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 android:background="#2F363F"  
 tools:context=".MainActivity">  
  
 <SeekBar  
 android:id="@+id/seekBar"  
 android:layout\_width="0dp"  
 android:layout\_height="42dp"  
 android:progress="45"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="240dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:textAlignment="center"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="112dp"  
 android:layout\_marginEnd="8dp"  
 android:text="Mithra"  
 android:layout\_marginBottom="250dp"  
 android:background="@drawable/button\_custom"  
 android:textColor="#FF3031"  
 app:layout\_constraintBottom\_toTopOf="@+id/seekBar"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0" />  
</android.support.constraint.ConstraintLayout>

New thing is on the seek bar, we have an attribute called

android:progress="45"

This progess attribute helpts to set the default seek bar position.

Java code:

package com.mr71116.seekbar;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.widget.SeekBar;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity {  
  
 TextView T;  
 SeekBar SK;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 T = findViewById(R.id.*textView*);  
 SK = findViewById(R.id.*seekBar*);  
 SK.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int value, boolean fromUser) {  
 T.setTextSize(value);  
 }  
  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
  
 }  
  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {  
  
 }  
 });  
 }  
}

New thing is:

Declaration of seekbar and assigning the variable to it.

SeekBar SK;

SK = findViewById(R.id.*seekBar*);

Onclicklistener for seekbar:

SK.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int value, boolean fromUser) {  
 T.setTextSize(value);  
 }

Int value is the value of the seek bar 1 to 100

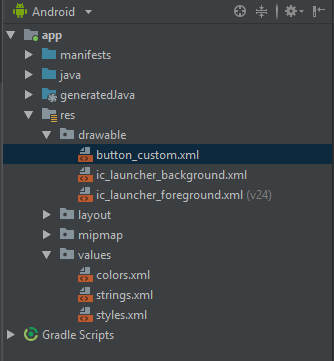
T.setTextSize(value) helps to change the Textsize based on the seek bar position.

Our app:





Project structure:



We also have another two methods for seek bar

@Override  
public void onStartTrackingTouch(SeekBar seekBar) {  
  
}  
  
@Override  
public void onStopTrackingTouch(SeekBar seekBar) {  
  
}

In addition to the project details which was shown in the course, I myself added few codes to make sure that background changes when we move the seek bar.

Requirements:

If seekbar value is less than or equal to 25, apply black

If seekbar value is less than or equal to 50 and greater than 25, apply Blue

If seekbar value is less than or equal to 75 and greater than 50, apply Green

If seekbar value is less than or equal to 100 and greater than 75, apply Green

Code

if(value <=25){  
 la.setBackgroundColor(Color.*BLACK*);  
}  
if(value >25 && value <=50){  
 la.setBackgroundColor(Color.*BLUE*);  
}  
if(value >50 && value <=75){  
 la.setBackgroundColor(Color.*GREEN*);  
}  
if(value >75 && value <=100){  
 la.setBackgroundColor(Color.*RED*);  
}

full java Code for that below:

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 T = findViewById(R.id.*textView*);  
 la = findViewById(R.id.*layout*);  
 SK = findViewById(R.id.*seekBar*);  
 SK.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int value, boolean fromUser) {  
 T.setTextSize(value);  
 if(value <=25){  
 la.setBackgroundColor(Color.*BLACK*);  
 }  
 if(value >25 && value <=50){  
 la.setBackgroundColor(Color.*BLUE*);  
 }  
 if(value >50 && value <=75){  
 la.setBackgroundColor(Color.*GREEN*);  
 }  
 if(value >75 && value <=100){  
 la.setBackgroundColor(Color.*RED*);  
 }  
 }

Progress bar:

To show the progress of an activity.

Two types shown in the course

1. Circle progress bar
2. Horizontal progress bar

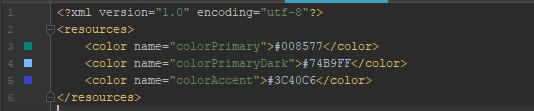
First setup the UI:

Resource given in the course is the customized circle button.



Change the colors in colors.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="colorPrimary">#008577</color>  
 <color name="colorPrimaryDark">#74B9FF</color>  
 <color name="colorAccent">#3C40C6</color>  
</resources>



In styles.xml, change the theme to no action bar:

<resources>  
  
 <!-- Base application theme. -->  
 <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">  
 <!-- Customize your theme here. -->  
 <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="colorAccent">@color/colorAccent</item>  
 </style>  
  
</resources>

Set the layout background color:

android:background="#99AAAB"

New attributes on progress bar (horizontal progress bar):

android:max="100"  
android:progress="30"

Max refer here to the maximum value for the bar

Progress refers here to the default value on which progress should be shown/started.

Full code of activity.main.xml:

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 android:background="#99AAAB"  
 tools:context=".MainActivity">  
  
 <ProgressBar  
 android:id="@+id/progressBar"  
 style="?android:attr/progressBarStyle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="112dp"  
 android:layout\_marginEnd="8dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/start"  
 android:onClick="start"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="52dp"  
 android:layout\_marginTop="236dp"  
 android:background="@drawable/button\_custom"  
 android:text="Start"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/end"  
 android:onClick="end"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="236dp"  
 android:layout\_marginEnd="36dp"  
 android:background="@drawable/button\_custom"  
 android:text="End"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.895"  
 app:layout\_constraintStart\_toEndOf="@+id/start"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/upload"  
 android:onClick="upload"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="120dp"  
 android:background="@drawable/custom\_uploadbutton"  
 android:text="Upload"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.568"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <ProgressBar  
 android:id="@+id/progressBar2"  
 style="?android:attr/progressBarStyleHorizontal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:max="100"  
 android:progress="30"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="60dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/upload"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.501"  
 app:layout\_constraintStart\_toStartOf="parent" />  
</android.support.constraint.ConstraintLayout>

Now, lets see the java coding below:

New things are:

Progress bar declaration:

ProgressBar Progress1, Progress2;

Assigning a variable prog as 0;

int prog = 0;

Below progressbar.setvisibility(view.gone) helps to hide the progress bar

Progress1.setVisibility(View.*GONE*);

Below progressbar.setvisibility(view. *VISIBLE*) helps to show the progress bar

Progress1.setVisibility(View.*VISIBLE*);

New method is written called fakeprogress bar with a parameter (prog)

Thread is used here to sleep for 1 sec and add value 10 to variable prog.

Then we are using **Recursion** technique here to call the function within the function.

Finally we are starting the thread using thread.start;

New method here:

public void fakeprogess(final int prog){  
 Progress2.setProgress(prog);  
 Thread thread = new Thread(new Runnable() {  
 @Override  
 public void run() {  
 try {  
 Thread.*sleep*(1000);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 fakeprogess(prog+10);  
  
 }  
  
 });  
 thread.start();  
}

Thread thread = new Thread (new Runnable(){

In the above code, don’t change any other variable name for this, just use “thread” as variable. Don’t know why.

We have used the android studio feature in the code tool at the top tools menu:

Code -> Surround With and then click on try and catch for the line thread.sleep.

New method related to progress bar:

Progress2.setProgress(prog);

This sets the progress of the progress bar;

fakeprogess(prog+10);

The above is a recursive method which calls the function within the function and here it is incrementing the prog value by 10 and this function is ran again until it reaches 100 as we have the maximum limit of the progress bar as 100 in activity\_main.xml under the horizontal progress bar.

thread.start();

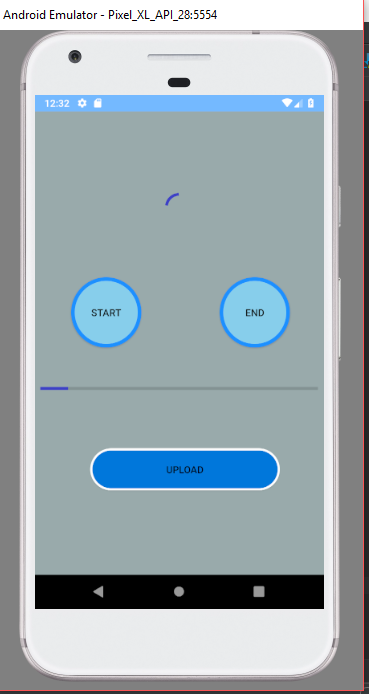
the above code helps to start the thread.

Our app looks like below:

The circle progress bar is visible when start button is clicked and gets hided when end button is clicked.

It will be keep on circling if end button is not clicked.

The horizontal progress bar will progress once in 1 sec by 10 value when upload button is clicked and it will stop post reaching the full bar/100



Time and Date Picker:

First is the Time picker:

No resources provided from course.

We are going to show the clock in the layout and while we change the time in the clock we can see the same time getting shown in the text view we have below:

New thing:

TimePicker tp;

tp = findViewById(R.id.*timepick*);

tp.setOnTimeChangedListener(new TimePicker.OnTimeChangedListener() {  
 @Override  
 public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {  
 tv.setText(+hourOfDay+ ":"+minute);  
 }  
});

Timepicker is the component for displaying time

First we are declaring the variable for time picker

Then we are assigning the object using findviewby id

Next is the method of the time picker

.setonTimeChangedListener

This method will create another method ontimechange and in that we can make our textview to display the changed time by using the below code:

tp.setOnTimeChangedListener(new TimePicker.OnTimeChangedListener() {  
 @Override  
 public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {  
 tv.setText(+hourOfDay+ ":"+minute);  
 }  
});

hourofDay and minute are the parameter in this method.

We are setText to display the time and minute

Full java code:

package com.mr71116.timeanddatepicker;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.widget.TextView;  
import android.widget.TimePicker;  
  
public class MainActivity extends AppCompatActivity {  
  
 TimePicker tp;  
 TextView tv;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 tp = findViewById(R.id.*timepick*);  
 tv = findViewById(R.id.*textView*);  
 tp.setOnTimeChangedListener(new TimePicker.OnTimeChangedListener() {  
 @Override  
 public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {  
 tv.setText(+hourOfDay+ ":"+minute);  
 }  
 });  
 }  
}

Full activity\_main.xml code below:

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 android:background="#777E8B"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 android:textColor="#FFFFFF"  
 android:textSize="18dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.919" />  
  
 <TimePicker  
 android:id="@+id/timepick"  
 android:layout\_width="384dp"  
 android:layout\_height="470dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/textView"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.615">  
  
 </TimePicker>  
  
</android.support.constraint.ConstraintLayout>

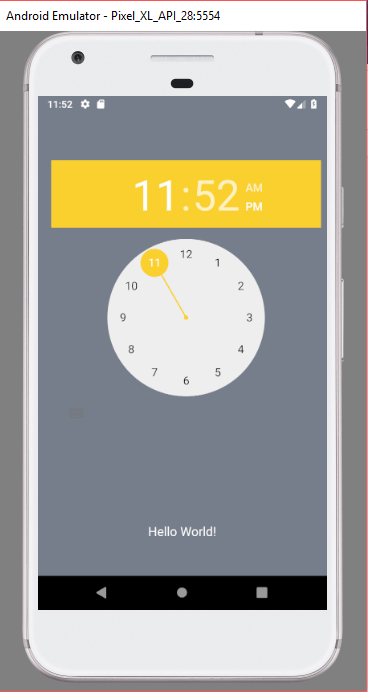
Style.xml:

<resources>  
  
 <!-- Base application theme. -->  
 <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">  
 <!-- Customize your theme here. -->  
 <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="colorAccent">@color/colorAccent</item>  
 </style>  
  
</resources>

Color.xml:

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="colorPrimary">#008577</color>  
 <color name="colorPrimaryDark">#777E8B</color>  
 <color name="colorAccent">#FAD02E</color>  
</resources>

Our Time picker app looks like below:



Now, lets see the date picker:

We are going to bring a date picker into the screen and display the date changed in the text view below as we did for time picker.

New thing is:

Datepicker component

We are using Calendar class but not as usual object creation but as below:

Calendar cal = Calendar.*getInstance*();

We are getting instance of it.

Then we are using the date picker view object and using init method and using the below parameters in it we are getting the year, month and date:

And then we are using the Ondatechange listener to handle the change, ie once the date is changed we are displaying it in Textview with below code;

public void onDateChanged(DatePicker view, int year, int monthOfYear, int dayOfMonth) {  
 monthOfYear = monthOfYear+1;  
 tv.setText(+year+"/"+monthOfYear+"/"+dayOfMonth);

we are adding +1 to the monthofYear because month here starts with 0;

Full java code:

package com.mr71116.datepicker;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.widget.DatePicker;  
import android.widget.TextView;  
  
import java.util.Calendar;  
  
public class MainActivity extends AppCompatActivity {  
  
 DatePicker dp;  
 TextView tv;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 dp= findViewById(R.id.*datepicker*);  
 tv=findViewById(R.id.*textView*);  
 Calendar cal = Calendar.*getInstance*();  
 dp.init(  
 cal.get(Calendar.*YEAR*),  
 cal.get(Calendar.*MONTH*),  
 cal.get(Calendar.*DAY\_OF\_MONTH*), new DatePicker.OnDateChangedListener() {  
 @Override  
 public void onDateChanged(DatePicker view, int year, int monthOfYear, int dayOfMonth) {  
 monthOfYear = monthOfYear+1;  
 tv.setText(+year+"/"+monthOfYear+"/"+dayOfMonth);  
 }  
 }  
 );  
 }  
}

full activity.main.xml:

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 android:background="#777E8B"  
 tools:context=".MainActivity">  
  
 <DatePicker  
 android:id="@+id/datepicker"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/textView"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.461">  
  
 </DatePicker>  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text=""  
 android:textColor="#FFFF"  
 android:textSize="18dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.923" />  
  
</android.support.constraint.ConstraintLayout>

Color and styles are same as Time picker project.

Our Date picker app looks like below:

