ECE590- Mobile Systems and Applications

Dr. Xiang Chen

Name: Saurabh Deshpande

G01036463

Lab-2

Introduction

In this lab, I have developed a game named ColorMatch. The Lab demands 2 activities, Main Activity and the Target activity. On main activity there are nine buttons with no color in default. On clicking any button among those nine, the color of that specific button must change to blue and on repetitive clicking the color toggles between blue, green and red. This applies to all the nine buttons. On the Target activity there is a fixed color pattern and idea behind these two activities is to match the color pattern on Main activity to that of Target.

Users can toggle between the two activities and the main activity should preserve its state whenever user comes back to main activity from the Target activity. As the pattern of both activity matches, a success message should be displayed on the main activity screen.

Main

The lab-2 requirements can is achieved by creating two xml files and modifying related java files in Android Studio. One is the main activity xml file and other is the Target xml. The main xml consists of a TextView for displaying the activity name and a button to switch to Target activity. Then there is a structure of 9 buttons, clicking on which changes the background color. And also a TextView for displaying success message. The Target xml has the almost same layout as that of main layout except that the color of buttons is fixed and there is no TextView to display the success message.

The main activity java file is modified so that the buttons can change color and to jump to the target activity. Also, it compares the color pattern and displays success message.

Description & Discussion

The layout of main activity is designed in such a way that it can be deployed on any screen size. The main activity TextView, Target button and the button structure is in the relative layout but the structure of buttons are arranged with combinations of linear layouts. A layout with orientation vertical comprises of three linear layouts with orientation as horizontal which weights equal to one. Each linear layout with horizontal orientation has three buttons and each button has weight equal to one. This arrangement ensures that the

button structure has proportionate screen occupancy which takes care of the different screen sizes and different device orientation (screen rotation). The same layout is used for target activity.

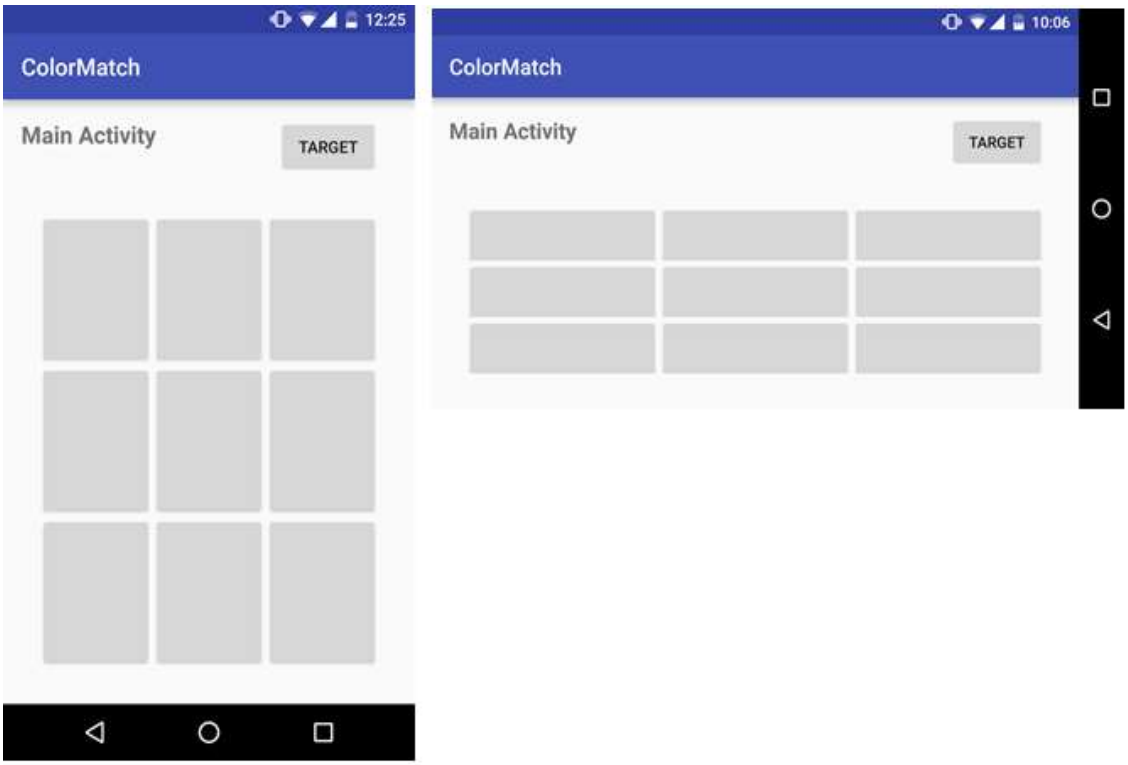

Figure 1: Main Activity

As there are two activities, transition from main activity to target activity and target to main was a challenge. For transitioning from main activity to target activity an xml attribute ‘ onClick’ was used for the TARGET button , which invokes the method with the same name in the main.java file. After reading some forums, I came to know that Intents can be used to start a new activity. Then I studied about Intents, there are two types of Intents, -IMPLICIT Intents and EXPLICIT Intents. Explicit contents can be used for starting many internal activities, when a component is provided i.e from same app, while for Implicit Intents we need provide enough information for the system to determine which of the available components is best to run for that intent. (eg. Gallery app has share icon which starts activity in another app- Gmail, Instagram,facebook).

Now whenever I switched from main activity to target activity and back and forth, the back button (of device) used to navigate through all the layers of activities. For solving this , intent flags were set in main.java (FLAG_ACTIVITY_REORDER_TO_FRONT the launched activity to be brought to the front of its task's history stack if it is already running and in target.java ,(FLAG_ACTIVITY_CLEAR_TOP) the activity being launched is already running in the current task, then instead of launching a new instance of that activity, all of the other

activities on top of it will be closed and this Intent will be delivered to the (now on top) old activity as a new Intent.

Now the task was to change the color of the buttons every time we click them, in the domain of blue, green and red. This was pretty easy as I knew about the onClick attribute. I created 9 methods for 9 different buttons. On clicking each button, respective method is invoked which contains the logic for changing the background color and the current background color is stored in the String variable.

The main challenge was to retain the current state of the main activity when user jumps to target activity and comes back. I studied about onCreate and onSavedInstanceState function and the hints given in the Lab 2 requirement document. While searching through the developer.android.com, I came across the launchMode attribute used in the AndroidManifest.xml file. Setting this attribute to singleTask I came to know that the system creates the activity at the root of a new task and routes the intent to it. However, if an instance of the activity already exists, the system routes the intent to existing instance, rather than creating a new one. The change in the device orientation, starts new activity and to retain the current state after change in orientation configChanges attribute in the AndroidManifest.xml was set to 'orientation|screenSize'.

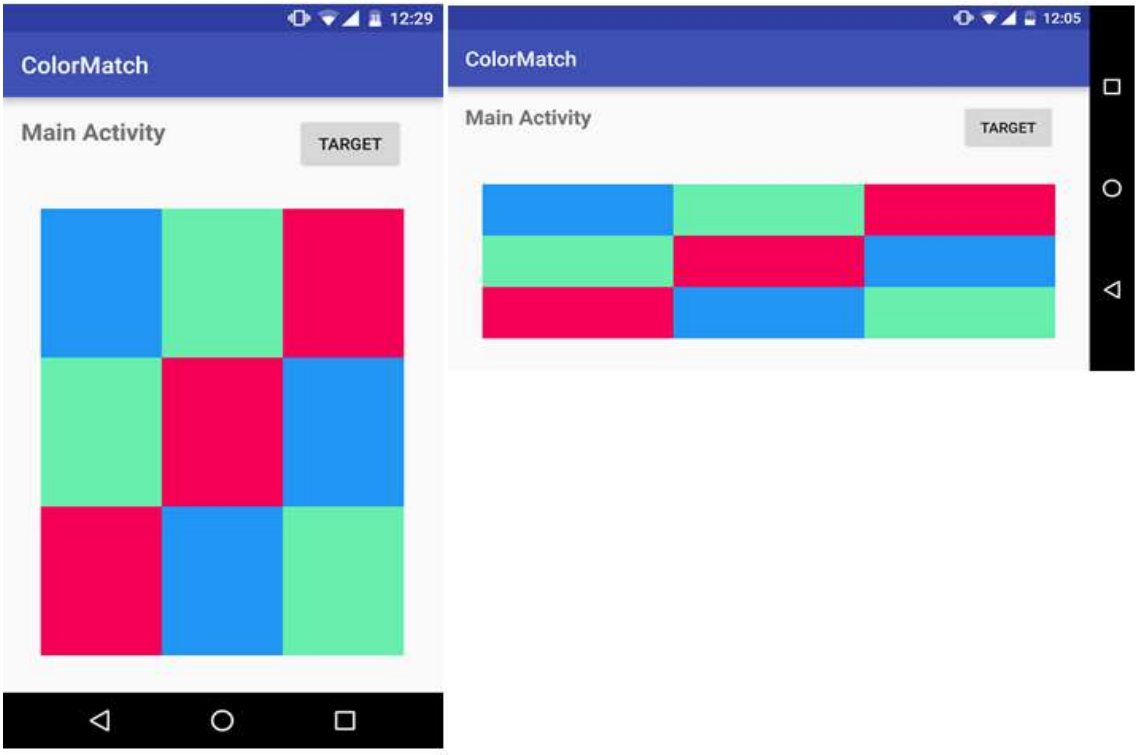

Figure 2: Retaining current state after orientation change

The last part was to display a 'Success!' message in the main activity if the color pattern of the buttons matches with the target activity color pattern. For that checking logic was developed and that check method was called in every button method. Usually in modern apps , if they have to display a message they usually flash that message. Keeping this in mind I tried doing the same with Toast. Toast- A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive. With the help of this, I managed to pop up the Success! Message every time the pattern matches.

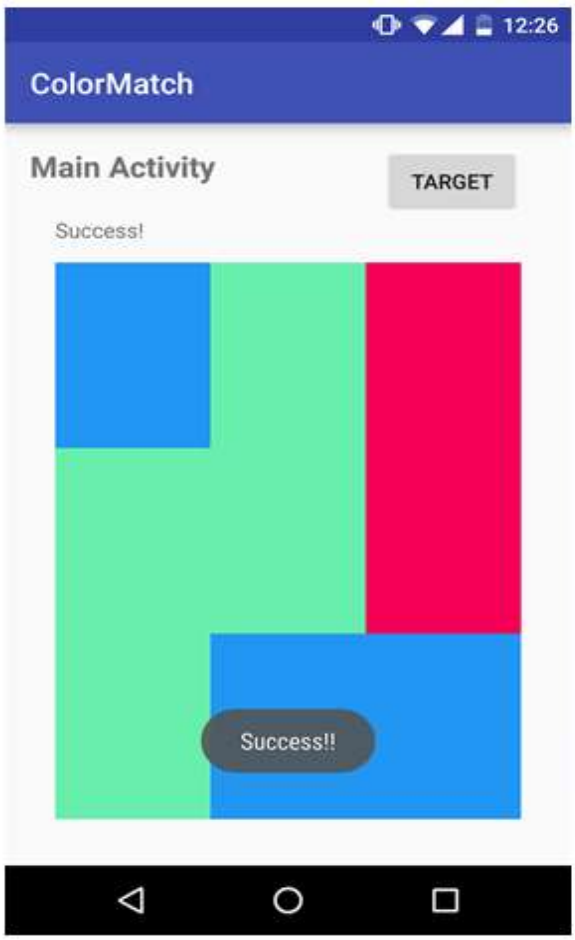

Figure 3: Success!!

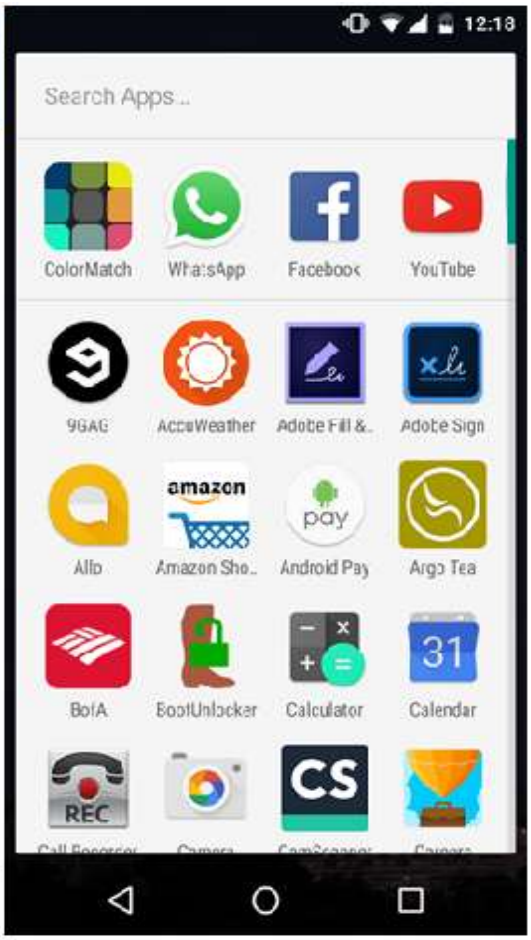

Figure 4: ColorMatch (Icon)

To make the app presentable I gave a nice Icon to the application suitable to its name and function.(AndroidManifest.xml icon attribute)

Conclusion

As per the lab requirements, the application is created and also few more additional functionalities has been added. I learned about the life cycle of the Activity and the communication method between the activities. The lab provided insight to the workflow of onclicking function, activity transition and retention of its current value. The application was successfully developed and installed on an android phone (Nexus 5).

Output

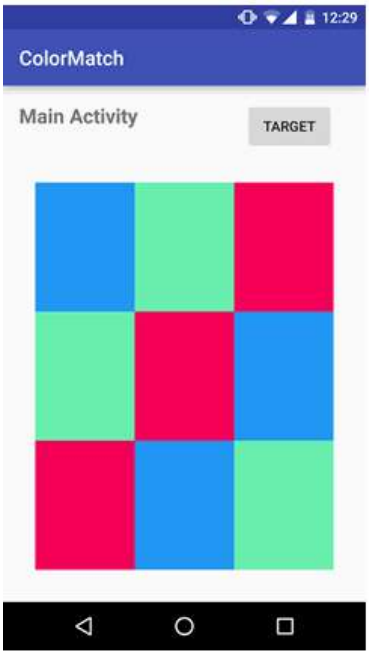

Figure 5: Main Activity

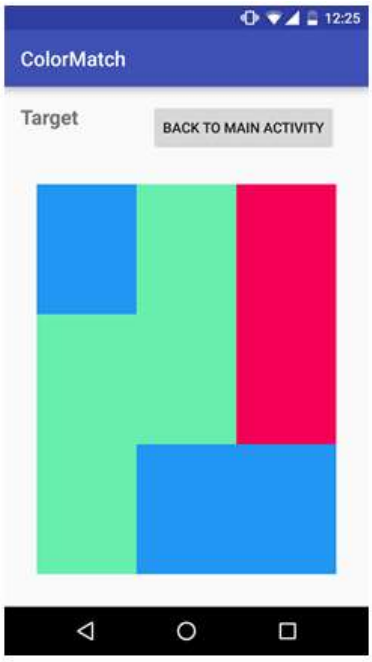

Figure 6: Target Activity

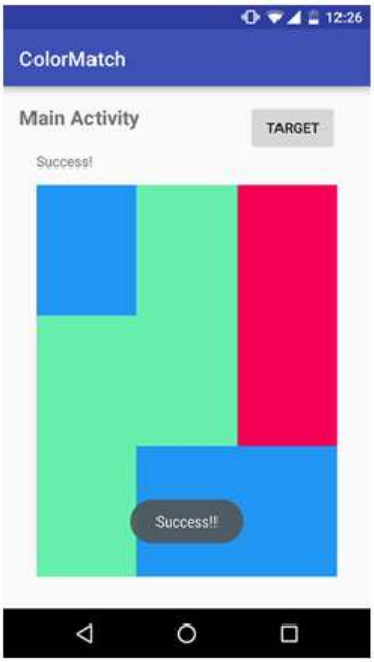

Figure 7: Success!!

Code

1.activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.d27sa.colormatch.MainActivity">
<!--TextView to display the Activity-->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:text="Main Activity"
        android:textSize="20dp"
        android:textStyle="bold"/>
<!--Target button to switch to the TARGET activity
    Onclick is used to call the Terget_Button method in mainactivity .java -->
    <Button
        android:text="Target"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="16dp"
        android:id="@+id/button"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:onClick="Target_Button"/>
<!--This TextView displays message as 'Success!' as the colour pattern on
mainactivity matches with the colour pattern in terget activity-->
    <TextView
        android:text=" "
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:id="@+id/textView"
        android:layout_below="@+id/button"
        android:layout_alignParentStart="true" />
<!--This code is for the structure of 9 buttons. Linear Layouts are used , one
with the vertical
 orientation and each linear layout within it is oriented as horizontal.-->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="80dp"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginBottom="16dp"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:orientation="horizontal">

            <Button
                android:layout_width="0dp"
                android:layout_height="130dp"
                android:layout_weight="1"
                android:id="@+id/button1"
                android:onClick="ColourChangeb1"/>
            <Button
                android:layout_width="0dp"
                android:layout_height="130dp"
                android:layout_weight="1"
                android:onClick="ColourChangeb2"/>
            <Button
                android:layout_width="0dp"
                android:layout_height="130dp"
                android:layout_weight="1"
                android:onClick="ColourChangeb3"/>

        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="0dp"
```

```
            android:layout_weight="1"
            android:orientation="horizontal">

            <Button
                android:layout_width="0dp"
                android:layout_height="130dp"
                android:layout_weight="1"
                android:onClick="ColourChangeb4"/>
            <Button
                android:layout_width="0dp"
                android:layout_height="130dp"
                android:layout_weight="1"
                android:onClick="ColourChangeb5"/>
            <Button
                android:layout_width="0dp"
                android:layout_height="130dp"
                android:layout_weight="1"
                android:onClick="ColourChangeb6"/>

        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:orientation="horizontal">

            <Button
                android:layout_width="0dp"
                android:layout_height="130dp"
                android:layout_weight="1"
                android:onClick="ColourChangeb7"/>
            <Button
                android:layout_width="0dp"
                android:layout_height="130dp"
                android:layout_weight="1"
                android:onClick="ColourChangeb8"/>
            <Button
                android:layout_width="0dp"
                android:layout_height="130dp"
                android:layout_weight="1"
                android:onClick="ColourChangeb9"/>

        </LinearLayout>

    </LinearLayout>


</RelativeLayout>
```

2.MainaActivity.java

```
package com.example.d27sa.colormatch;

import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
```

```
public class MainActivity extends AppCompatActivity {

    //Blue ,Red, Green defined
    String blue = "#2196F3";
    String green = "#68EFAD";
    String red = "#F50057";
    // b1 to b9 for 9 buttons to store the current colour of the button.
Initially defined to default colour of the button.
    String b1 = "#E0E0E0";
    String b2 = "#E0E0E0";
    String b3 = "#E0E0E0";
    String b4 = "#E0E0E0";
    String b5 = "#E0E0E0";
    String b6 = "#E0E0E0";
    String b7 = "#E0E0E0";
    String b8 = "#E0E0E0";
    String b9 = "#E0E0E0";


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (savedInstanceState!=null){
            this.onRestoreInstanceState(savedInstanceState);
        }

    }

    @Override
    protected void onSaveInstanceState(Bundle outState){
        super.onSaveInstanceState(outState);
    }

    /* Method Terget_Button is called when Target button is clicked to shift
from main activity to target activity.
 FLAG_ACTIVITY_REORDER_TO_FRONT this flag will cause the launched activity to
be brought to the front
 of its task's history stack if it is already running.
 */
    public void Target_Button(View view){
        Intent i = new Intent(this, Target.class);
        i.setFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);
        startActivity(i);
    }

    /*ColourChangeb1 method is set on the OnClick attribute of the Button 1.
    Onclicking the button1 , it sets the background colour of the button to
blue
    and further clicks changes to green to red to blue.
     */
    public void ColourChangeb1(View btn) {
        if ((b1 == "#E0E0E0") || (b1 == red)) {
            btn.setBackgroundColor(Color.parseColor(blue));
            b1 = blue;
            check();
            return;
        }
        if ((b1 == blue)) {
            btn.setBackgroundColor(Color.parseColor(green));
            b1 = green;
            check();
            return;
        }
```

```
    if ((b1 == green)) {
        btn.setBackgroundColor(Color.parseColor(red));
        b1 = red;
        check();
        return;
    }

}

public void ColourChangeb2(View btn) {
    if ((b2 == "#E0E0E0") || (b2 == red)) {
        btn.setBackgroundColor(Color.parseColor(blue));
        b2 = blue;
        check();
        return;
    }
    if ((b2 == blue)) {
        btn.setBackgroundColor(Color.parseColor(green));
        b2 = green;
        check();
        return;
    }

    if ((b2 == green)) {
        btn.setBackgroundColor(Color.parseColor(red));
        b2 = red;
        check();
        return;
    }
}

public void ColourChangeb3(View btn) {
    if ((b3 == "#E0E0E0") || (b3 == red)) {
        btn.setBackgroundColor(Color.parseColor(blue));
        b3 = blue;
        check();
        return;
    }
    if ((b3 == blue)) {
        btn.setBackgroundColor(Color.parseColor(green));
        b3 = green;
        check();
        return;
    }

    if ((b3 == green)) {
        btn.setBackgroundColor(Color.parseColor(red));
        b3 = red;
        check();
        return;
    }


}

public void ColourChangeb4(View btn) {
    if ((b4 == "#E0E0E0") || (b4 == red)) {
        btn.setBackgroundColor(Color.parseColor(blue));
        b4 = blue;
        check();
        return;
    }
    if ((b4 == blue)) {
```

```
            btn.setBackgroundColor(Color.parseColor(green));
            b4 = green;
            check();
            return;
        }

        if ((b4 == green)) {
            btn.setBackgroundColor(Color.parseColor(red));
            b4 = red;
            check();
            return;
        }
    }

    public void ColourChangeb5(View btn) {
        if ((b5 == "#E0E0E0") || (b5 == red)) {
            btn.setBackgroundColor(Color.parseColor(blue));
            b5 = blue;
            check();
            return;
        }
        if ((b5 == blue)) {
            btn.setBackgroundColor(Color.parseColor(green));
            b5 = green;
            check();
            return;
        }

        if ((b5 == green)) {
            btn.setBackgroundColor(Color.parseColor(red));
            b5 = red;
            check();
            return;
        }
    }

    public void ColourChangeb6(View btn) {
        if ((b6 == "#E0E0E0") || (b6 == red)) {
            btn.setBackgroundColor(Color.parseColor(blue));
            b6 = blue;
            check();
            return;
        }
        if ((b6 == blue)) {
            btn.setBackgroundColor(Color.parseColor(green));
            b6 = green;
            check();
            return;
        }

        if ((b6 == green)) {
            btn.setBackgroundColor(Color.parseColor(red));
            b6 = red;
            check();
            return;
        }
    }

    public void ColourChangeb7(View btn) {
        if ((b7 == "#E0E0E0") || (b7 == red)) {
            btn.setBackgroundColor(Color.parseColor(blue));
            b7 = blue;
            check();
            return;
```

```
        }
        if ((b7 == blue)) {
            btn.setBackgroundColor(Color.parseColor(green));
            b7 = green;
            check();
            return;
        }

        if ((b7 == green)) {
            btn.setBackgroundColor(Color.parseColor(red));
            b7 = red;
            check();
            return;
        }
    }

    public void ColourChangeb8(View btn) {
        if ((b8 == "#E0E0E0") || (b8 == red)) {
            btn.setBackgroundColor(Color.parseColor(blue));
            b8 = blue;
            check();
            return;
        }
        if ((b8 == blue)) {
            btn.setBackgroundColor(Color.parseColor(green));
            b8 = green;
            check();
            return;
        }

        if ((b8 == green)) {
            btn.setBackgroundColor(Color.parseColor(red));
            b8 = red;
            check();
            return;
        }
    }

    public void ColourChangeb9(View btn) {
        if ((b9 == "#E0E0E0") || (b9 == red)) {
            btn.setBackgroundColor(Color.parseColor(blue));
            b9 = blue;
            check();
            return;
        }
        if ((b9 == blue)) {
            btn.setBackgroundColor(Color.parseColor(green));
            b9 = green;
            check();
            return;
        }

        if ((b9 == green)) {
            btn.setBackgroundColor(Color.parseColor(red));
            b9 = red;
            check();
            return;
        }
    }
    /*This is to check the mainactivity colour pattern matches with the target
colour pattern.
    If the colour pattern is matched , a success message is displayed in the
TextView as well as a flash message is
    displayed using the Toast.
```

```
     */
     public void check(){

if((b1==blue)&&(b2==green)&&(b3==red)&&(b4==green)&&(b5==green)&&(b6==red)&&(b7
==green)&&(b8==blue)&&(b9==blue)){
        displayMessage("Success!");
            Toast.makeText(getApplicationContext(), "Success!",
                    Toast.LENGTH_SHORT).show();
        }
         else {
            displayMessage(" ");
        }
    }

    public void displayMessage(String message){
        TextView statusTextView = (TextView) findViewById(R.id.textView);
        statusTextView.setText(message);
    }
}
```

3.activity_target.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_target"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.d27sa.colormatch.Target">
<!--TextView to dislppay the Target activity-->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Target"
        android:textSize="20dp"
        android:textStyle="bold"/>
<!--button used to switch back to the main activity screen-->
    <Button
        android:text="BACK TO MAIN ACTIVITY"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="16dp"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:onClick="BacktoMA"/>
<!-- Linear Layouts used to structure the 9 buttons with fixed colours-->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="80dp"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginBottom="16dp"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
```

```
        android:orientation="horizontal">

        <Button
            android:layout_width="0dp"
            android:layout_height="130dp"
            android:layout_weight="1"
            android:background="#2196F3"/>
        <Button
            android:layout_width="0dp"
            android:layout_height="130dp"
            android:layout_weight="1"
            android:background="#68EFAD"/>
        <Button
            android:layout_width="0dp"
            android:layout_height="130dp"
            android:layout_weight="1"
            android:background="#F50057"/>

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:orientation="horizontal">

        <Button
            android:layout_width="0dp"
            android:layout_height="130dp"
            android:layout_weight="1"
            android:background="#68EFAD"/>
        <Button
            android:layout_width="0dp"
            android:layout_height="130dp"
            android:layout_weight="1"
            android:background="#68EFAD"/>
        <Button
            android:layout_width="0dp"
            android:layout_height="130dp"
            android:layout_weight="1"
            android:background="#F50057"/>

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:orientation="horizontal"
        >

        <Button
            android:layout_width="0dp"
            android:layout_height="130dp"
            android:layout_weight="1"
            android:background="#68EFAD"/>
        <Button
            android:layout_width="0dp"
            android:layout_height="130dp"
            android:layout_weight="1"
            android:background="#2196F3"/>
        <Button
            android:layout_width="0dp"
            android:layout_height="130dp"
```

```
                android:layout_weight="1"
                android:background="#2196F3"/>

        </LinearLayout>

    </LinearLayout>

</RelativeLayout>
```

4.Target.java

```
package com.example.d27sa.colormatch;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class Target extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_target);
    }
/*Method to switch from target activity to the main activity by generating an
Intent
    Flag_activity_clear_top used as the activity being launched is already
running in the current task,
     then instead of launching a new instance of that activity, all of the
other activities on top of it
      will be closed and this Intent will be delivered to the (now on top) old
activity as a new Intent.*/
    public void BacktoMA(View view){
        Intent i = new Intent(this, MainActivity.class);
        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(i);
    }
}
```

5.AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.d27sa.colormatch">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/icon"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity"
            android:launchMode="singleTask"
            android:configChanges="orientation|screenSize">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Target"></activity><!-- ATTENTION: This was
```

```
auto-generated to add Google Play services to your project for
     App Indexing.  See https://g.co/AppIndexing/AndroidStudio for more
information. -->
        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />
    </application>

</manifest>
```