|  |  |
| --- | --- |
| Baby’s Drwaing app | Students   1. Dalia Fahad 🡪 202104297 2. Nawal moafa 🡪 202103591 3. Kholod atia 🡪 202108845 4. Sarah Yahya  🡪 202185952   Course   * Applications Development 343 |
|  |  |

**Table**

[**Executive Summary:** 1](#_Toc126374460)

[**Introduction:** 2](#_Toc126374461)

[**Design and Development:** 2](#_Toc126374462)

[**Functionality:** 2](#_Toc126374463)

[**Conclusion:** 3](#_Toc126374464)

[**Screen Shots:** 3](#_Toc126374465)

[ Activity Main 3](#_Toc126374466)

[ Custom Dialog 5](#_Toc126374467)

[ Activity Contact Us 6](#_Toc126374468)

[**XML Codes:** 7](#_Toc126374469)

[ Activity Main 7](#_Toc126374470)

[ Custom Dialog 10](#_Toc126374471)

[ Activity Contact Us 11](#_Toc126374472)

[**Java Codes:** 16](#_Toc126374473)

[ Activity Main 16](#_Toc126374474)

[ Paint View 27](#_Toc126374475)

[ Activity Contact Us 32](#_Toc126374476)

# **Executive Summary:**

The Baby's Drawing application is a highly engaging and interactive tool designed specifically for children who love to draw. This application offers a range of features that make the drawing experience both easy and fun. It features a responsive brush for drawing on the screen, various color options, an eraser button, undo and redo buttons, and the ability to save and restore previous drawings. The user interface is designed to be user-friendly, with all buttons and options clearly labeled and accessible. The Contact Us button provides a way for the user to communicate with the development team for any questions or feedback. To add to the interactive experience, the application features sound effects that are triggered when drawing, erasing, changing colors, undoing, and redoing.

# **Introduction:**

The Baby's Drawing application is a fun and interactive tool for children who love to draw. It has been designed with the user in mind, with the user interface being easy to understand and the buttons and options being clearly labeled and accessible. The focus on user-friendliness and accessibility makes the application a safe and enjoyable experience for children.

# **Design and Development:**

The Baby's Drawing application was developed using Android Studio, which is a widely used development platform for mobile applications. The design of the application has been thoughtfully created to make the drawing experience as easy and fun as possible for children. The brush for drawing on the screen is responsive and provides a smooth and accurate drawing experience. The color options, in the form of a hair game, offer a variety of colors to choose from, allowing the user to add a personal touch to their drawings. The eraser button makes it easy to clean the drawing board, while the undo and redo buttons allow for corrections to be made. The ability to save and restore previous drawings provides the user with the flexibility to continue their creative process even after closing the application.

# **Functionality:**

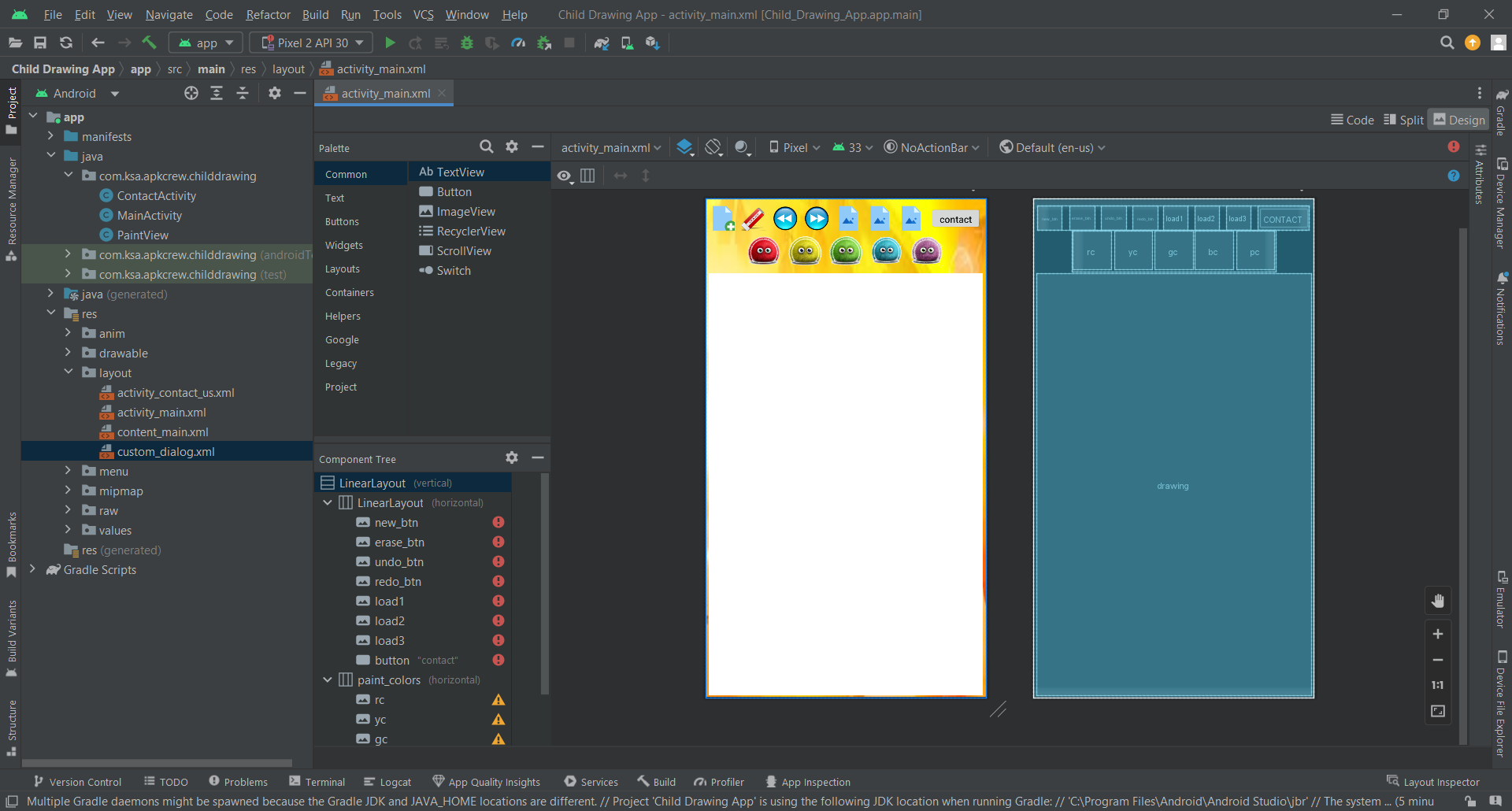
The Baby's Drawing application is packed with functionalities that make the drawing experience enjoyable and interactive. The sound effects, triggered when drawing, erasing, changing colors, undoing, and redoing, add an exciting element to the experience. The drawings created by the user are saved, allowing for continued creativity even after closing the application. This saves the user time and effort as they do not have to start from scratch every time they open the application. The Contact Us button provides a way for the user to communicate with the development team for any questions or feedback, making it easy for the user to get help or provide suggestions for future improvements.

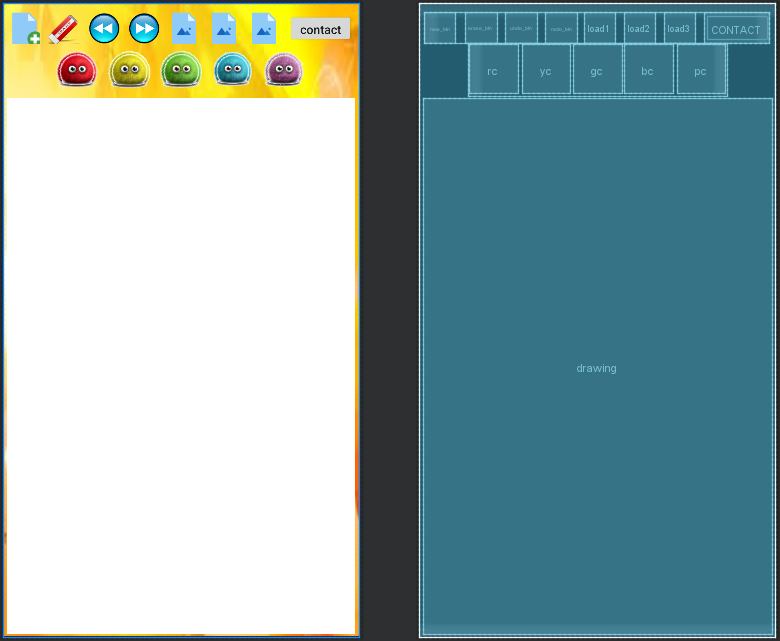
# **Conclusion:**

The Baby's Drawing application is a user-friendly and exciting tool for children who love to draw. With a range of features including a brush, color options, an eraser, an undo and redo button, and the ability to save and restore previous drawings, the application provides a creative outlet for children. The user interface is easy to understand, and the sound effects add an interactive element to the experience. The Contact Us button provides the user with the ability to communicate with the development team for any questions or feedback, making it easy for the user to get help or provide suggestions for future improvements. Overall, the Baby's Drawing application is an excellent tool for children who love to be creative, providing a safe and enjoyable drawing experience.

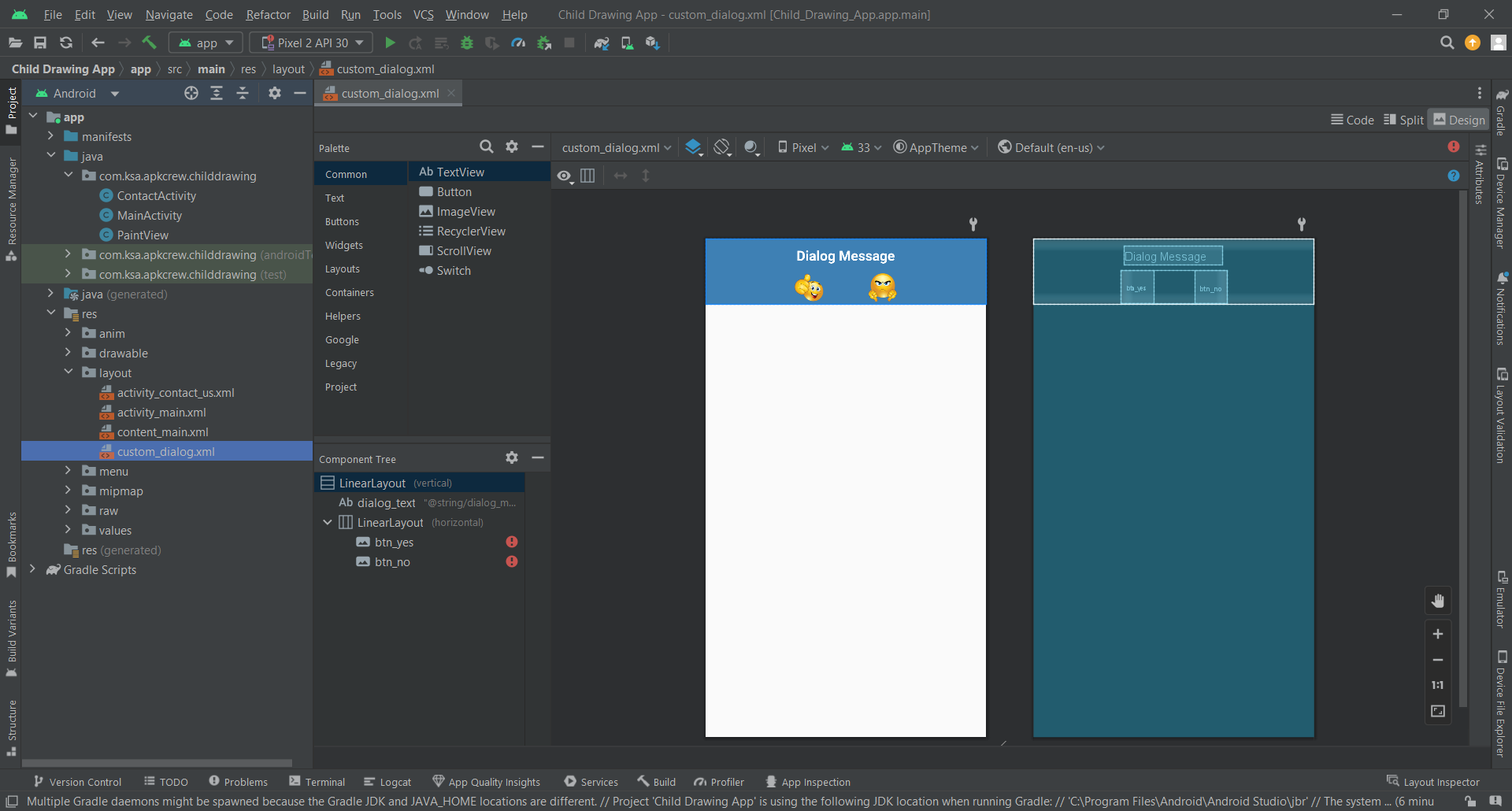
# **Screen Shots:**

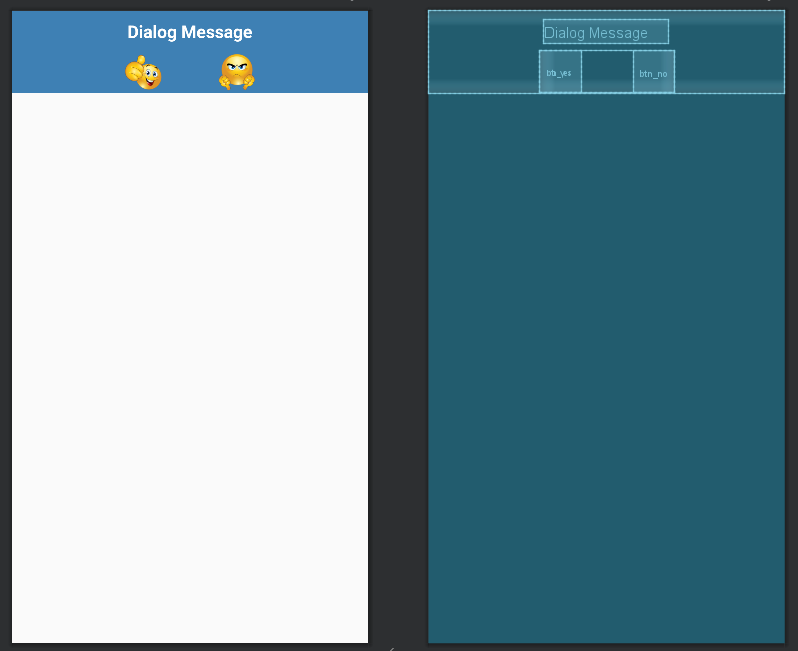
* Activity Main



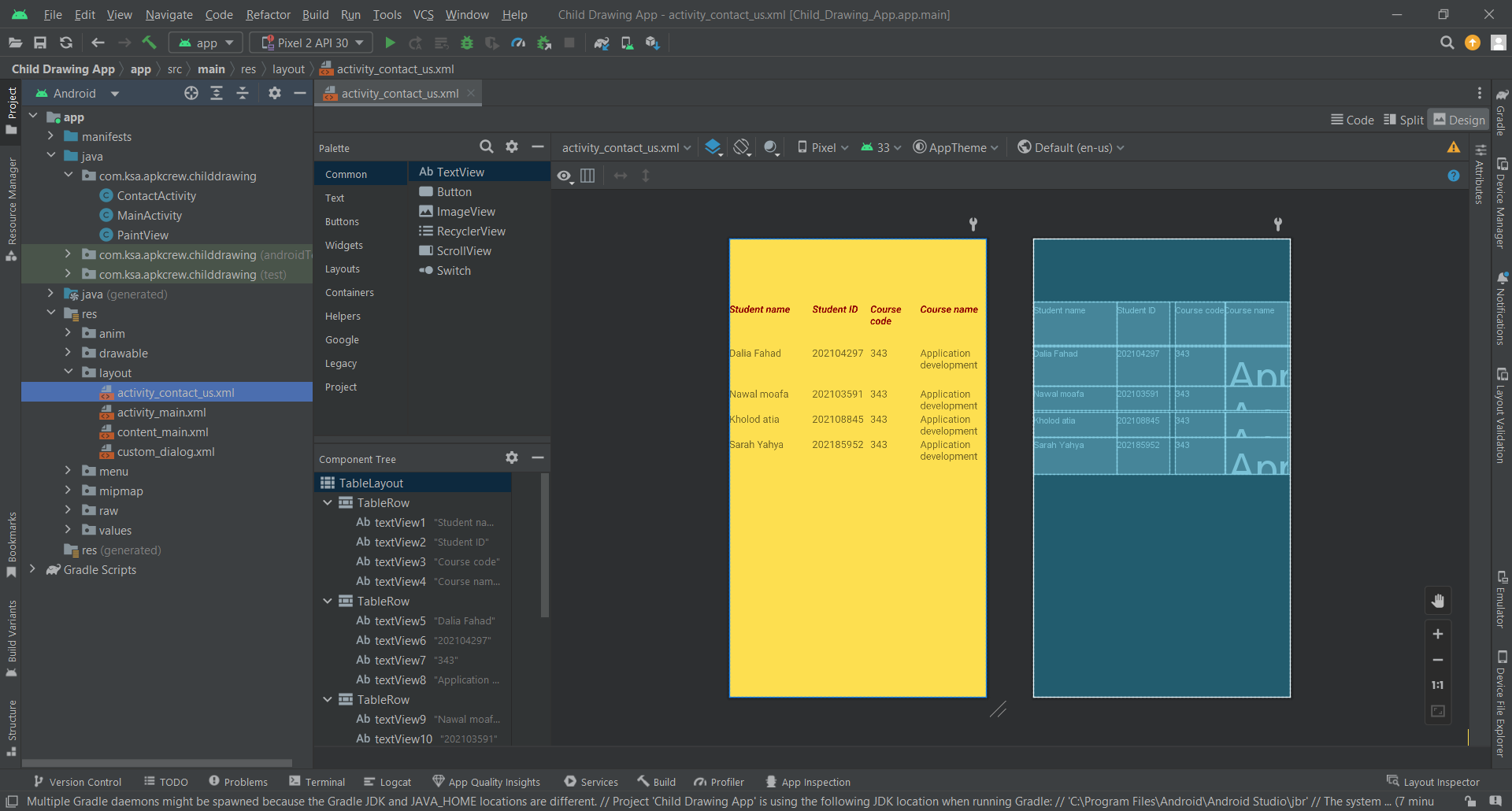


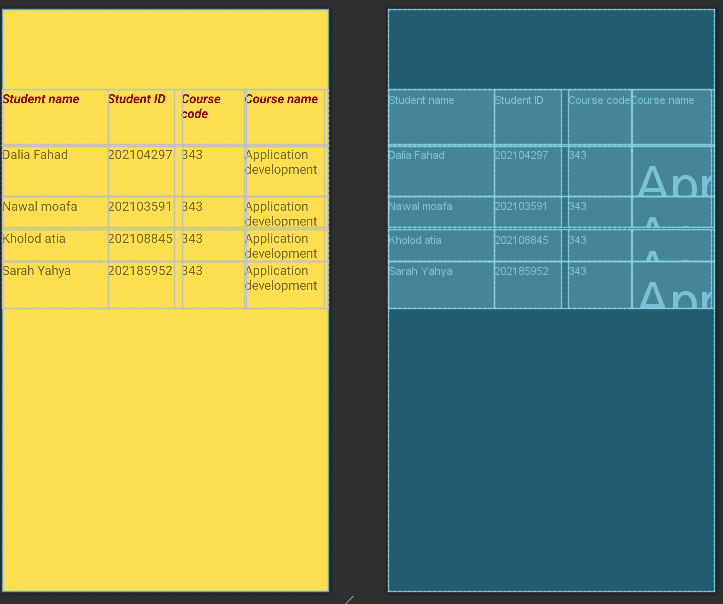
* Custom Dialog





* Activity Contact Us





# **XML Codes:**

* Activity Main

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_height="match\_parent"  
 android:layout\_width="match\_parent"  
 android:background="@drawable/theme"  
 android:orientation="vertical"  
 tools:context=".MainActivity" >  
  
 <LinearLayout  
 android:layout\_width="398dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="10dp"  
 android:orientation="horizontal">  
  
 <ImageButton  
 android:id="@+id/new\_btn"  
 android:layout\_width="36dp"  
 android:layout\_height="36dp"  
 android:layout\_marginEnd="@dimen/endMargin"  
 android:layout\_marginRight="@dimen/endMargin"  
 android:background="#00FFFFFF"  
 android:contentDescription="@string/start\_new"  
 android:src="@mipmap/new\_icon" />  
  
 <ImageButton  
 android:id="@+id/erase\_btn"  
 android:layout\_width="36dp"  
 android:layout\_height="36dp"  
 android:layout\_marginEnd="@dimen/endMargin"  
 android:layout\_marginRight="@dimen/endMargin"  
 android:background="@mipmap/eraser"  
 android:contentDescription="@string/erase"  
 android:src="@mipmap/eraser" />  
  
 <ImageButton  
 android:id="@+id/undo\_btn"  
 android:layout\_width="36dp"  
 android:layout\_height="36dp"  
 android:layout\_marginEnd="@dimen/endMargin"  
 android:layout\_marginRight="@dimen/endMargin"  
 android:background="#00FFFFFF"  
 android:contentDescription="@string/undo"  
 android:src="@mipmap/undo" />  
  
 <ImageButton  
 android:id="@+id/redo\_btn"  
 android:layout\_width="36dp"  
 android:layout\_height="36dp"  
 android:layout\_marginEnd="@dimen/endMargin"  
 android:layout\_marginRight="@dimen/endMargin"  
 android:background="#00FFFFFF"  
 android:contentDescription="@string/redo"  
 android:src="@mipmap/redo" />  
  
 <ImageButton  
 android:id="@+id/load1"  
 android:layout\_width="36dp"  
 android:layout\_height="36dp"  
 android:layout\_marginEnd="@dimen/endMargin"  
 android:layout\_marginRight="@dimen/endMargin"  
 android:background="#00FFFFFF"  
 android:contentDescription="@string/load"  
 app:srcCompat="@mipmap/load" />  
  
 <ImageButton  
 android:id="@+id/load2"  
 android:layout\_width="36dp"  
 android:layout\_height="36dp"  
 android:layout\_marginEnd="@dimen/endMargin"  
 android:layout\_marginRight="@dimen/endMargin"  
 android:background="#00FFFFFF"  
 android:contentDescription="@string/load"  
 app:srcCompat="@mipmap/load" />  
  
 <ImageButton  
 android:id="@+id/load3"  
 android:layout\_width="36dp"  
 android:layout\_height="36dp"  
 android:layout\_marginEnd="@dimen/endMargin"  
 android:layout\_marginRight="@dimen/endMargin"  
 android:background="#00FFFFFF"  
 android:contentDescription="@string/load"  
 app:srcCompat="@mipmap/load" />  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="130dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="5"  
 android:width="130dp"  
 android:capitalize="none"  
 android:maxLength="50"  
 android:text="contact"  
 android:textAllCaps="false" />  
  
 </LinearLayout>  
 <LinearLayout  
 android:id="@+id/paint\_colors"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_gravity="center">  
  
 <ImageButton  
 android:id="@+id/rc"  
 android:layout\_width="@dimen/icon\_size"  
 android:layout\_height="@dimen/icon\_size"  
 android:layout\_margin="2dp"  
 android:background="@mipmap/red"  
 android:contentDescription="@string/paint"  
 android:onClick="paintClicked"  
 android:src="@mipmap/red"  
 android:tag="#FFFF0000" />  
  
 <ImageButton  
 android:layout\_width="@dimen/icon\_size"  
 android:layout\_height="@dimen/icon\_size"  
 android:layout\_margin="2dp"  
 android:background="@mipmap/yellow"  
 android:contentDescription="@string/paint"  
 android:onClick="paintClicked"  
 android:src="@mipmap/yellow"  
 android:tag="#FFFFCC00"  
 android:id="@+id/yc" />  
  
 <ImageButton  
 android:layout\_width="@dimen/icon\_size"  
 android:layout\_height="@dimen/icon\_size"  
 android:layout\_margin="2dp"  
 android:background="@mipmap/green"  
 android:contentDescription="@string/paint"  
 android:onClick="paintClicked"  
 android:src="@mipmap/green"  
 android:tag="#32cd32"  
 android:id="@+id/gc" />  
  
 <ImageButton  
 android:layout\_width="@dimen/icon\_size"  
 android:layout\_height="@dimen/icon\_size"  
 android:layout\_margin="2dp"  
 android:background="@mipmap/blue"  
 android:contentDescription="@string/paint"  
 android:onClick="paintClicked"  
 android:src="@mipmap/blue"  
 android:tag="#00bfff"  
 android:id="@+id/bc" />  
  
 <ImageButton  
 android:layout\_width="@dimen/icon\_size"  
 android:layout\_height="@dimen/icon\_size"  
 android:layout\_margin="2dp"  
 android:background="@mipmap/pink"  
 android:contentDescription="@string/paint"  
 android:onClick="paintClicked"  
 android:src="@mipmap/pink"  
 android:tag="#FFFF00FF"  
 android:id="@+id/pc" />  
 </LinearLayout>  
  
 <com.ksa.apkcrew.childdrawing.PaintView  
 android:id="@+id/drawing"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="0dp"  
 android:layout\_marginBottom="3dp"  
 android:layout\_marginLeft="5dp"  
 android:layout\_marginRight="5dp"  
 android:layout\_marginTop="3dp"  
 android:layout\_weight="1"  
 android:background="#FFFFFFFF" />  
</LinearLayout>

* Custom Dialog

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="#3E80B4"  
 android:orientation="vertical" >  
  
 <TextView  
 android:id="@+id/dialog\_text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp"  
 android:text="@string/dialog\_message"  
 android:textColor="@android:color/white"  
 android:textSize="20sp"  
 android:textStyle="bold"/>  
  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:background="#3E80B4"  
 android:orientation="horizontal" >  
  
 <ImageButton  
 android:id="@+id/btn\_yes"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="#3E80B4"  
 android:layout\_marginRight="30dp"  
 android:layout\_marginEnd="30dp"  
 android:clickable="true"  
 android:src="@mipmap/yes" />  
  
 <ImageButton  
 android:id="@+id/btn\_no"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="#3E80B4"  
 android:layout\_marginLeft="30dp"  
 android:layout\_marginStart="30dp"  
 android:clickable="true"  
 android:src="@mipmap/no" />  
 </LinearLayout>  
  
</LinearLayout>

* Activity Contact Us

*<?*xml version="1.0" encoding="utf-8"*?>*<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
  
 android:background="#FDDF50"  
 android:baselineAligned="true"  
 android:scrollbars="horizontal|vertical">  
  
  
 <TableRow  
  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="100dp">  
  
 <TextView  
 android:id="@+id/textView1"  
 android:layout\_width="133dp"  
 android:layout\_height="match\_parent"  
 android:height="70dp"  
 android:text="Student name"  
 android:textColor="#910505"  
 android:textSize="16sp"  
 android:textStyle="bold|italic" />  
  
 <TextView  
 android:id="@+id/textView2"  
  
  
 android:layout\_width="83dp"  
  
 android:layout\_height="match\_parent"  
 android:height="70dp"  
 android:text="Student ID"  
 android:textColor="#910505"  
 android:textSize="16sp"  
 android:textStyle="bold|italic" />  
  
 <TextView  
 android:id="@+id/textView3"  
  
  
 android:layout\_width="80dp"  
  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginLeft="10dp"  
 android:height="70dp"  
 android:text="Course code"  
 android:textColor="#910505"  
 android:textSize="16sp"  
 android:textStyle="bold|italic" />  
  
 <TextView  
 android:id="@+id/textView4"  
  
  
 android:layout\_width="100dp"  
  
 android:layout\_height="match\_parent"  
 android:height="70dp"  
 android:text="Course name"  
 android:textColor="#910505"  
 android:textSize="16sp"  
 android:textStyle="bold|italic" />  
 </TableRow>  
  
 <TableRow  
  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/textView5"  
  
  
 android:layout\_width="133dp"  
 android:layout\_height="match\_parent"  
  
 android:height="65dp"  
 android:text="Dalia Fahad"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/textView6"  
  
  
 android:layout\_width="83dp"  
 android:layout\_height="match\_parent"  
  
 android:height="65dp"  
 android:text="202104297"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/textView7"  
  
 android:layout\_width="80dp"  
 android:layout\_height="match\_parent"  
  
 android:layout\_marginStart="10dp"  
 android:layout\_marginLeft="10dp"  
 android:height="65dp"  
 android:text="343"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/textView8"  
  
  
 android:layout\_width="100dp"  
  
 android:layout\_height="match\_parent"  
 android:text="Application development"  
 android:textSize="16sp" />  
 </TableRow>  
  
 <TableRow  
  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/textView9"  
  
  
 android:layout\_width="133dp"  
 android:layout\_height="match\_parent"  
 android:text="Nawal moafa"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/textView10"  
  
  
 android:layout\_width="83dp"  
 android:layout\_height="match\_parent"  
 android:text="202103591"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/textView11"  
  
  
 android:layout\_width="80dp"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginLeft="10dp"  
 android:text="343"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/textView12"  
  
  
 android:layout\_width="100dp"  
 android:layout\_height="match\_parent"  
 android:text="Application development"  
 android:textSize="16sp" />  
 </TableRow>  
  
 <TableRow  
  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="65dp">  
  
 <TextView  
 android:id="@+id/textView13"  
  
  
 android:layout\_width="133dp"  
 android:layout\_height="match\_parent"  
 android:text="Kholod atia"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/textView14"  
  
  
 android:layout\_width="83dp"  
 android:layout\_height="match\_parent"  
 android:text="202108845"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/textView15"  
  
  
 android:layout\_width="80dp"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginLeft="10dp"  
 android:text="343"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/textView16"  
  
  
 android:layout\_width="100dp"  
 android:layout\_height="match\_parent"  
 android:text="Application development"  
 android:textSize="16sp" />  
 </TableRow>  
  
 <TableRow  
  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/textView17"  
  
 android:layout\_width="133dp"  
 android:layout\_height="match\_parent"  
  
 android:height="60dp"  
 android:text="Sarah Yahya"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/textView18"  
  
 android:layout\_width="83dp"  
 android:layout\_height="match\_parent"  
  
 android:height="60dp"  
 android:text="202185952"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/textView19"  
 android:layout\_width="80dp"  
 android:layout\_height="match\_parent"  
  
 android:layout\_marginStart="10dp"  
 android:layout\_marginLeft="10dp"  
 android:height="60dp"  
 android:text="343"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/textView20"  
  
 android:layout\_width="100dp"  
 android:layout\_height="match\_parent"  
  
 android:height="60dp"  
 android:text="Application development"  
 android:textSize="16sp" />  
 </TableRow>  
  
  
</TableLayout>

# **Java Codes:**

* Activity Main

package com.ksa.apkcrew.childdrawing;  
  
import android.Manifest;  
import android.app.Dialog;  
import android.content.Context;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.media.MediaPlayer;  
import android.media.ThumbnailUtils;  
import android.os.Bundle;  
import android.os.Environment;  
import android.support.annotation.NonNull;  
import android.support.annotation.Nullable;  
import android.support.v4.app.ActivityCompat;  
import android.support.v4.content.ContextCompat;  
import android.support.v4.content.res.ResourcesCompat;  
import android.support.v7.app.AppCompatActivity;  
import android.util.Log;  
import android.view.View;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.animation.Animation;  
import android.view.animation.AnimationUtils;  
import android.widget.Button;  
import android.widget.ImageButton;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
import android.os.Message;  
import android.os.Handler;  
  
import java.io.File;  
import java.io.FileOutputStream;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Locale;  
  
*/\*\*  
 \* The type Main activity.  
 \*/*public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
 private final int REQUEST\_EXTERNAL\_STORAGE = 10;  
 private PaintView paintView;  
 private ImageButton currPaint, thumbNail1, thumbNail2, thumbNail3;  
 private MediaPlayer fg\_voice;  
 */\*\*  
 \* The Rotate.  
 \*/* Animation rotate, */\*\*  
 \* The Move right.  
 \*/* moveRight, */\*\*  
 \* The Move left.  
 \*/* moveLeft, */\*\*  
 \* The Zoom io.  
 \*/* zoomIO, */\*\*  
 \* The Fade in.  
 \*/* fadeIn, */\*\*  
 \* The Slide ud.  
 \*/* slideUD;  
 private String saveName, savePath, prevFilePath[];  
 private boolean paused = false;  
 private int inActivityTime = 0;  
 */\*\*  
 \* The Context.  
 \*/* final Context context = this;  
 */\*\*  
 \* The Album name.  
 \*/* final String albumName = "Baby's Drawings";  
 final static String *TAG* = "TAG";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 paintView = (PaintView) findViewById(R.id.*drawing*);  
 LinearLayout paintLayout = (LinearLayout) findViewById(R.id.*paint\_colors*);  
 for (int i = 0; i < paintLayout.getChildCount(); ++i) {  
 ImageButton c = (ImageButton) paintLayout.getChildAt(i);  
 if (i == 0) {  
 c.setImageDrawable(ResourcesCompat.*getDrawable*(getResources(), R.drawable.*color\_selected*, null));  
 currPaint = c;  
 continue;  
 }  
 c.setImageDrawable(ResourcesCompat.*getDrawable*(getResources(), R.drawable.*color\_pallete*, null));  
 }  
 ImageButton eraseBtn = (ImageButton) findViewById(R.id.*erase\_btn*);  
 eraseBtn.setOnClickListener(this);  
 ImageButton newBtn = (ImageButton) findViewById(R.id.*new\_btn*);  
 newBtn.setOnClickListener(this);  
  
 Button contactBtn = (Button) findViewById(R.id.*button*);  
 contactBtn.setOnClickListener(this);  
  
 ImageButton undoBtn = (ImageButton) findViewById(R.id.*undo\_btn*);  
 undoBtn.setOnClickListener(this);  
 ImageButton redoBtn = (ImageButton) findViewById(R.id.*redo\_btn*);  
 redoBtn.setOnClickListener(this);  
 thumbNail1 = (ImageButton) findViewById(R.id.*load1*);  
 thumbNail1.setOnClickListener(this);  
 thumbNail2 = (ImageButton) findViewById(R.id.*load2*);  
 thumbNail2.setOnClickListener(this);  
 thumbNail3 = (ImageButton) findViewById(R.id.*load3*);  
 thumbNail3.setOnClickListener(this);  
 rotate = AnimationUtils.*loadAnimation*(this, R.anim.*anim\_rotate*);  
 moveLeft = AnimationUtils.*loadAnimation*(this, R.anim.*move\_left*);  
 moveRight = AnimationUtils.*loadAnimation*(this, R.anim.*move\_right*);  
 fadeIn = AnimationUtils.*loadAnimation*(this, R.anim.*fade\_in*);  
 zoomIO = AnimationUtils.*loadAnimation*(this, R.anim.*zoom\_in\_and\_out*);  
 slideUD = AnimationUtils.*loadAnimation*(this, R.anim.*slide\_up\_and\_down*);  
 setMusic(R.raw.*m1*);  
 saveName = setSaveName();  
 savePath = setSavePath();  
 createThumbNail();  
 Log.*e*(*TAG*, "onCreate: savePath - " + savePath + " saveName" + saveName);  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
 paused = true;  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
*// st.start();* paused = false;  
 Log.*i*("info", "onResume: autsave called");  
 autoSave();  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*menu\_main*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 *// Handle action bar item clicks here. The action bar will  
 // automatically handle clicks on the Home/Up button, so long  
 // as you specify a parent activity in AndroidManifest.xml.* int id = item.getItemId();  
  
 *//noinspection SimplifiableIfStatement* if (id == R.id.*action\_settings*) {  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 */\*\*  
 \* The Handler.  
 \*/* Handler handler = new Handler() {  
 @Override  
 public void handleMessage(Message msg) {  
 if (paintView.drawingChanged) {  
 saveDrawing();  
 paintView.drawingChanged = false;  
 Log.*i*("info", "handleMessage: autosave called after save");  
 autoSave();  
 inActivityTime = 0;  
 } else if (!paintView.drawingChanging) {  
 if (inActivityTime >= 30) {  
 inActivityTime = 0;  
 startVoice(R.raw.*inactivity*);  
 Log.*i*("info", "handleMessage: dialog show ");  
 final Dialog timeoutDialog = new Dialog(context);  
 timeoutDialog.setContentView(R.layout.*custom\_dialog*);  
 TextView dialogText = (TextView) timeoutDialog.findViewById(R.id.*dialog\_text*);  
 dialogText.setText(R.string.*timeout\_msg*);  
 ImageButton yesBtn = (ImageButton) timeoutDialog.findViewById(R.id.*btn\_yes*);  
 ImageButton noBtn = (ImageButton) timeoutDialog.findViewById(R.id.*btn\_no*);  
 yesBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startVoice(R.raw.*yes*);  
 timeoutDialog.dismiss();  
 Log.*i*("info", "onClick: autosave called in yes");  
 autoSave();  
 }  
 });  
 noBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startVoice(R.raw.*no*);  
 timeoutDialog.dismiss();  
 paintView.startNew();  
 Log.*i*("info", "onClick: autosave called in no");  
 autoSave();  
 }  
 });  
 timeoutDialog.getWindow().getAttributes().windowAnimations = R.style.*CustomDialogAnimation*;  
 timeoutDialog.show();  
 } else {  
 inActivityTime += 10;  
 Log.*i*("info", "handleMessage: autosave called");  
 autoSave();  
 }  
 } else {  
 inActivityTime = 0;  
 autoSave();  
 }  
 }  
 };  
  
 */\*\*  
 \* Auto save.  
 \*/* public void autoSave() {  
 if (!paused) {  
 Runnable runnable = new Runnable() {  
 @Override  
 public void run() {  
 Log.*i*("info", "run: runnable started");  
 long start = System.*currentTimeMillis*();  
 long end;  
 while (true) {  
 end = System.*currentTimeMillis*();  
 if (end - start >= 10000)  
 break;  
 }  
 handler.sendEmptyMessage(0);  
 }  
 };  
 Thread myThread = new Thread(runnable);  
 myThread.start();  
 }  
 }  
  
  
 */\*\*  
 \* Paint clicked.  
 \*  
 \* @param view the view  
 \*/* public void paintClicked(View view) {  
 paintView.drawingChanging = true;  
 *//use chosen color* if (view.getId() == R.id.*rc*) {  
 startVoice(R.raw.*red*);  
 setMusic(R.raw.*m1*);  
 view.startAnimation(rotate);  
 } else if (view.getId() == R.id.*yc*) {  
 startVoice(R.raw.*yellow*);  
 setMusic(R.raw.*m2*);  
 view.startAnimation(rotate);  
 } else if (view.getId() == R.id.*gc*) {  
 startVoice(R.raw.*green*);  
 setMusic(R.raw.*m3*);  
 view.startAnimation(rotate);  
 } else if (view.getId() == R.id.*bc*) {  
 startVoice(R.raw.*blue*);  
 setMusic(R.raw.*m4*);  
 view.startAnimation(rotate);  
 } else if (view.getId() == R.id.*pc*) {  
 startVoice(R.raw.*pink*);  
 setMusic(R.raw.*m5*);  
 view.startAnimation(rotate);  
 }  
 ImageButton imgView = (ImageButton) view;  
 String color = view.getTag().toString();  
 paintView.setColor(color);  
 if (view != currPaint) {  
 *//update color* imgView.setImageDrawable(ResourcesCompat.*getDrawable*(getResources(), R.drawable.*color\_selected*, null));  
 currPaint.setImageDrawable(ResourcesCompat.*getDrawable*(getResources(), R.drawable.*color\_pallete*, null));  
 currPaint = (ImageButton) view;  
 }  
 }  
  
 @Override  
 public void onClick(View view) {  
 *//respond to clicks* switch (view.getId()) {  
 case R.id.*button*:  
 openContactPage(view);  
 break;  
 case R.id.*erase\_btn*:  
 eraserClicked(view);  
 break;  
 case R.id.*new\_btn*:  
 newClicked(view);  
 break;  
 case R.id.*undo\_btn*:  
 view.startAnimation(moveLeft);  
 if (paintView.canUndo()) {  
 startVoice(R.raw.*undo*);  
 } else {  
 startVoice(R.raw.*noundo*);  
 }  
 paintView.onClickUndo();  
 break;  
 case R.id.*redo\_btn*:  
 view.startAnimation(moveRight);  
 if (paintView.canRedo()) {  
 startVoice(R.raw.*redo*);  
 } else {  
 startVoice(R.raw.*noredo*);  
 }  
 paintView.onClickRedo();  
 break;  
 case R.id.*load1*:  
 loadClicked(view, 0);  
 break;  
 case R.id.*load2*:  
 loadClicked(view, 1);  
 break;  
 case R.id.*load3*:  
 loadClicked(view, 2);  
 break;  
 }  
 }  
  
 private void loadClicked(View view, final int i) {  
 paintView.drawingChanging = true;  
 view.startAnimation(zoomIO);  
 if (prevFilePath[i] == null) {  
 startVoice(R.raw.*noprev*);  
 Toast.*makeText*(this, "No previous drawings", Toast.*LENGTH\_SHORT*)  
 .show();  
 return;  
 }  
 startVoice(R.raw.*load*);  
 final Dialog loadDialog = new Dialog(context);  
 loadDialog.setContentView(R.layout.*custom\_dialog*);  
 TextView dialogText = (TextView) loadDialog.findViewById(R.id.*dialog\_text*);  
 dialogText.setText(R.string.*load\_msg*);  
 ImageButton yesBtn = (ImageButton) loadDialog.findViewById(R.id.*btn\_yes*);  
 ImageButton noBtn = (ImageButton) loadDialog.findViewById(R.id.*btn\_no*);  
 yesBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startVoice(R.raw.*yes*);  
 loadDialog.dismiss();  
 paintView.setBitMap(prevFilePath[i]);  
 }  
 });  
 noBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startVoice(R.raw.*no*);  
 loadDialog.dismiss();  
 }  
 });  
 loadDialog.getWindow().getAttributes().windowAnimations = R.style.*CustomDialogAnimation*;  
 loadDialog.show();  
 }  
  
 private void newClicked(View view) {  
 startVoice(R.raw.*clear*);  
 view.startAnimation(slideUD);  
 paintView.drawingChanging = true;  
 final Dialog newDialog = new Dialog(context);  
 newDialog.setContentView(R.layout.*custom\_dialog*);  
 TextView dialogText = (TextView) newDialog.findViewById(R.id.*dialog\_text*);  
 ImageButton yesBtn = (ImageButton) newDialog.findViewById(R.id.*btn\_yes*);  
 ImageButton noBtn = (ImageButton) newDialog.findViewById(R.id.*btn\_no*);  
 dialogText.setText(R.string.*new\_msg*);  
 yesBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startVoice(R.raw.*yes*);  
 newDialog.dismiss();  
 saveDrawing();  
 paintView.startNew();  
 saveName = setSaveName();  
 createThumbNail();  
 }  
 });  
 noBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startVoice(R.raw.*no*);  
 newDialog.dismiss();  
 }  
 });  
 newDialog.getWindow().getAttributes().windowAnimations = R.style.*CustomDialogAnimation*;  
 newDialog.show();  
 }  
  
 */\*\*  
 \* Eraser clicked.  
 \*  
 \* @param view the view  
 \*/* void eraserClicked(View view) {  
 startVoice(R.raw.*erase*);  
 ImageButton imgView = (ImageButton) view;  
 currPaint.setImageDrawable(ResourcesCompat.*getDrawable*(getResources(), R.drawable.*color\_pallete*, null));  
 imgView.setImageDrawable(ResourcesCompat.*getDrawable*(getResources(), R.drawable.*color\_selected*, null));  
 currPaint = imgView;  
 paintView.drawingChanging = true;  
 view.startAnimation(fadeIn);  
 setMusic(R.raw.*erasor*);  
 paintView.setErase(true);  
 }  
  
 private void saveDrawing() {  
 int permissionCheck = ContextCompat.*checkSelfPermission*(this,  
 Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*);  
 *// Here, thisActivity is the current activity* if (permissionCheck != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this,  
 new String[]{Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*},  
 REQUEST\_EXTERNAL\_STORAGE);  
 } else {  
 paintView.setDrawingCacheEnabled(true);  
 paintView.setDrawingCacheQuality(View.*DRAWING\_CACHE\_QUALITY\_HIGH*);  
 Bitmap bitmap = paintView.getDrawingCache();  
 File pictureFile = new File(savePath + File.*separator* + saveName);  
 try {  
 FileOutputStream fos = new FileOutputStream(pictureFile);  
 bitmap.compress(Bitmap.CompressFormat.*PNG*, 100, fos);  
 fos.close();  
 Toast savedToast = Toast.*makeText*(getApplicationContext(),  
 "Drawing Saved! " + pictureFile.getAbsolutePath(), Toast.*LENGTH\_SHORT*);  
 savedToast.show();  
 } catch (Exception e) {  
 Toast unsavedToast = Toast.*makeText*(getApplicationContext(),  
 "Oops! Image could not be saved. " + pictureFile.getAbsolutePath(), Toast.*LENGTH\_SHORT*);  
 unsavedToast.show();  
 }  
 }  
 paintView.setDrawingCacheEnabled(false);  
 }  
  
 */\*\*  
 \* Create thumb nail.  
 \*/* public void createThumbNail() {  
 if (savePath != null) {  
 File files[] = new File(savePath).listFiles();  
 prevFilePath = new String[3];  
 if (files.length > 0) {  
 File file1 = files[0];  
 File file2 = null;  
 File file3 = null;  
  
 for (int i = 1; i < files.length; i++) {  
 if (file1.lastModified() < files[i].lastModified()) {  
 file3 = file2;  
 file2 = file1;  
 file1 = files[i];  
 } else if ((file2 == null) || (file2.lastModified() < files[i].lastModified())) {  
 file3 = file2;  
 file2 = files[i];  
 } else if ((file3 == null) || (file3.lastModified() < files[i].lastModified())) {  
 file3 = files[i];  
 }  
 }  
 Log.*i*("info", "createThumbNail: " + files.length);  
*// try {  
// Log.i("file", "createThumbNail: " + file1.getName() + " " + file2.getName() + " " + file3.getName());  
// } catch (NullPointerException e) {  
// Log.e("file", "createThumbNail: ", e);  
// }* if (file1 != null) {  
 prevFilePath[0] = file1.getAbsolutePath();  
 Bitmap ThumbImage = ThumbnailUtils.*extractThumbnail*(BitmapFactory.*decodeFile*(prevFilePath[0]), 64, 64);  
 thumbNail1.setImageBitmap(ThumbImage);  
 }  
 if (file2 != null) {  
 prevFilePath[1] = file2.getAbsolutePath();  
 Bitmap ThumbImage2 = ThumbnailUtils.*extractThumbnail*(BitmapFactory.*decodeFile*(prevFilePath[1]), 64, 64);  
 thumbNail2.setImageBitmap(ThumbImage2);  
 }  
 if (file3 != null) {  
 prevFilePath[2] = file3.getAbsolutePath();  
 Bitmap ThumbImage3 = ThumbnailUtils.*extractThumbnail*(BitmapFactory.*decodeFile*(prevFilePath[2]), 64, 64);  
 thumbNail3.setImageBitmap(ThumbImage3);  
 }  
 }  
 }  
 }  
  
*// @Nullable  
// private File getOutputMediaFile() {  
// // Create a media file name  
// savePath = setSavePath();  
// if (savePath != null) {  
// File mediaFile;  
// mediaFile = new File(savePath + File.separator + saveName);  
// return mediaFile;  
// }  
// return null;  
// }* private String setSaveName() {  
 String timeStamp = new SimpleDateFormat("ddMMyy\_kkmmss", Locale.*getDefault*()).format(new Date());  
 return "BD\_" + timeStamp + ".png";  
 }  
  
 @Nullable  
 private String setSavePath() {  
 int permissionCheck = ContextCompat.*checkSelfPermission*(this,  
 Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*);  
 Log.*e*(*TAG*, "setSavePath: " + permissionCheck);  
 *// Here, thisActivity is the current activity* if (permissionCheck != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this,  
 new String[]{Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*},  
 REQUEST\_EXTERNAL\_STORAGE);  
 } else {  
 Log.*e*(*TAG*, "setSavePath: " + isExternalStorageWritable());  
 if (!isExternalStorageWritable()) {  
 return null;  
 }  
 File mediaStorageDir = new File(Environment.*getExternalStoragePublicDirectory*(  
 Environment.*DIRECTORY\_PICTURES*).getAbsolutePath(), albumName);  
 Log.*e*(*TAG*, "setSavePath: " + mediaStorageDir);  
 Log.*e*(*TAG*, "setSavePath: mediaStorageDir path - " + mediaStorageDir.getAbsolutePath());  
 if (!mediaStorageDir.exists()) {  
 Log.*e*(*TAG*, "setSavePath: not exist " + mediaStorageDir);  
 if (!mediaStorageDir.mkdirs()) {  
 Log.*e*(*TAG*, "setSavePath: cannot make " + mediaStorageDir);  
 return null;  
 } else {  
 Log.*e*(*TAG*, "setSavePath: created " + mediaStorageDir);  
 }  
 }  
 return mediaStorageDir.getAbsolutePath();  
 }  
 return null;  
 }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 switch (requestCode) {  
 case REQUEST\_EXTERNAL\_STORAGE: {  
 *// If request is cancelled, the result arrays are empty.* if (grantResults.length > 0  
 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {  
 *// permission was granted* saveDrawing();  
  
 } else {  
 *// permission denied* Toast unsavedToast = Toast.*makeText*(getApplicationContext(),  
 "Permission Denied! Image could not be saved.", Toast.*LENGTH\_SHORT*);  
 unsavedToast.show();  
 }  
 }  
 }  
 }  
  
 */\*\*  
 \* Is external storage writable boolean.  
 \*  
 \* @return the boolean  
 \*/  
 /\* Checks if external storage is available for read and write \*/* public boolean isExternalStorageWritable() {  
 String state = Environment.*getExternalStorageState*();  
 return Environment.*MEDIA\_MOUNTED*.equals(state);  
 }  
  
 */\* Checks if external storage is available to at least read  
 public boolean isExternalStorageReadable() {  
 String state = Environment.getExternalStorageState();  
 return Environment.MEDIA\_MOUNTED.equals(state) ||  
 Environment.MEDIA\_MOUNTED\_READ\_ONLY.equals(state);  
 }\*/  
  
 /\*\*  
 \* Start voice.  
 \*  
 \* @param voiceId the voice id  
 \*/* void startVoice(int voiceId) {  
 if (fg\_voice != null) {  
 fg\_voice.release();  
 }  
 fg\_voice = MediaPlayer.*create*(context, voiceId);  
 fg\_voice.start();  
 }  
  
 */\*\*  
 \* Sets music.  
 \*  
 \* @param musicId the music id  
 \*/* void setMusic(int musicId) {  
 if (paintView.bg\_music != null) {  
 paintView.bg\_music.release();  
 }  
 paintView.bg\_music = MediaPlayer.*create*(context, musicId);  
  
 }  
  
 private void openContactPage(View view) {  
 view.startAnimation(slideUD);  
 paintView.drawingChanging = true;  
 final Dialog newDialog = new Dialog(context);  
 newDialog.setContentView(R.layout.*activity\_contact\_us*);  
 newDialog.getWindow().getAttributes().windowAnimations = R.style.*CustomDialogAnimation*;  
 newDialog.show();  
 }  
  
}

* Paint View

package com.ksa.apkcrew.childdrawing;  
  
import android.content.Context;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.graphics.Canvas;  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.graphics.Path;  
import android.graphics.PorterDuff;  
import android.media.MediaPlayer;  
import android.util.AttributeSet;  
import android.view.MotionEvent;  
import android.view.View;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
  
*/\*\*  
 \* The type Paint view.  
 \*/*public class PaintView extends View {  
 *//drawing path* private Path drawPath;  
 *//drawing and canvas paint* private Paint drawPaint, canvasPaint;  
 *//initial color* private int paintColor = Color.*RED*;  
 *//canvas* private Canvas drawCanvas;  
 *//canvas bitmap* private Bitmap canvasBitmap;  
 private boolean bitmapSet = false;  
 */\*\*  
 \* The Drawing changed.  
 \*/* public boolean drawingChanged = false, */\*\*  
 \* The Drawing changing.  
 \*/* drawingChanging = false;  
 private ArrayList<PathColorPair> paths = new ArrayList<>();  
 private ArrayList<PathColorPair> undonePaths = new ArrayList<>();  
 */\*\*  
 \* The Bg music.  
 \*/* public MediaPlayer bg\_music;  
  
 private class PathColorPair {  
 */\*\*  
 \* The Path.  
 \*/* Path path;  
 */\*\*  
 \* The Color.  
 \*/* int color;  
  
 */\*\*  
 \* Instantiates a new Path color pair.  
 \*  
 \* @param path the path  
 \* @param color the color  
 \*/* PathColorPair(Path path, int color) {  
 this.path = path;  
 this.color = color;  
 }  
  
 */\*\*  
 \* Instantiates a new Path color pair.  
 \*  
 \* @param other the other  
 \*/* PathColorPair(PathColorPair other) {  
 this.path = other.path;  
 this.color = other.color;  
 }  
 }  
  
 */\*\*  
 \* Instantiates a new Paint view.  
 \*  
 \* @param context the context  
 \* @param attrs the attrs  
 \*/* public PaintView(Context context, AttributeSet attrs) {  
 super(context, attrs);  
 drawPath = new Path();  
 drawPaint = new Paint();  
 drawCanvas = new Canvas();  
 canvasPaint = new Paint(Paint.*DITHER\_FLAG*);  
 int brushSize = getResources().getInteger(R.integer.*brush\_size*);  
 drawPaint.setAntiAlias(true);  
 drawPaint.setDither(true);  
 drawPaint.setStrokeWidth(brushSize);  
 drawPaint.setStyle(Paint.Style.*STROKE*);  
 drawPaint.setStrokeJoin(Paint.Join.*ROUND*);  
 drawPaint.setStrokeCap(Paint.Cap.*ROUND*);  
 drawPaint.setColor(paintColor);  
 }  
  
 @Override  
 protected void onSizeChanged(int w, int h, int oldw, int oldh) {  
 *//view given size* super.onSizeChanged(w, h, oldw, oldh);  
 canvasBitmap = Bitmap.*createBitmap*(w, h, Bitmap.Config.*ARGB\_8888*);  
 drawCanvas = new Canvas(canvasBitmap);  
 }  
  
 */\*\*  
 \* Sets color.  
 \*  
 \* @param newColor the new color  
 \*/* public void setColor(String newColor) {  
 *//set color* invalidate();  
 paintColor = Color.*parseColor*(newColor);  
 drawPaint.setColor(paintColor);  
 }  
  
 */\*\*  
 \* Sets erase.  
 \*  
 \* @param isErase the is erase  
 \*/* public void setErase(boolean isErase) {  
 if (isErase) {  
 paintColor = Color.*WHITE*;  
 drawPaint.setColor(Color.*WHITE*);  
 }  
 }  
  
 */\*\*  
 \* Start new.  
 \*/* public void startNew() {  
 bitmapSet = false;  
 paths.clear();  
 drawCanvas.drawColor(0, PorterDuff.Mode.*CLEAR*);  
 invalidate();  
 }  
  
 */\*\*  
 \* Sets bit map.  
 \*  
 \* @param fileName the file name  
 \*/* public void setBitMap(String fileName) {  
*// startNew();* paths.clear();  
 drawCanvas.drawColor(0, PorterDuff.Mode.*CLEAR*);  
 canvasBitmap = BitmapFactory.*decodeFile*(fileName).copy(Bitmap.Config.*ARGB\_8888*, true);  
*// drawCanvas = new\_icon Canvas(bmp);* bitmapSet = true;  
 drawCanvas.drawBitmap(canvasBitmap, 0, 0, canvasPaint);  
 invalidate();  
 }  
  
 private static final float *TOUCH\_TOLERANCE* = 4;  
 private float mX, mY;  
  
 private void touch\_start(float x, float y) {  
 undonePaths.clear();  
 drawPath.reset();  
 drawPath.moveTo(x, y);  
 mX = x;  
 mY = y;  
 drawingChanging = true;  
 }  
  
 private void touch\_move(float x, float y) {  
 float dx = Math.*abs*(x - mX);  
 float dy = Math.*abs*(y - mY);  
 if (dx >= *TOUCH\_TOLERANCE* || dy >= *TOUCH\_TOLERANCE*) {  
 drawPath.quadTo(mX, mY, (x + mX) / 2, (y + mY) / 2);  
 mX = x;  
 mY = y;  
 }  
 }  
  
 private void touch\_up() {  
 drawPath.lineTo(mX, mY);  
 drawCanvas.drawPath(drawPath, drawPaint);  
 paths.add(new PathColorPair(drawPath, paintColor));  
 drawPath = new Path();  
 drawingChanging = false;  
 drawingChanged = true;  
 }  
  
 */\*\*  
 \* Can undo boolean.  
 \*  
 \* @return the boolean  
 \*/* boolean canUndo() {  
 return paths.size() > 0;  
 }  
  
 */\*\*  
 \* Can redo boolean.  
 \*  
 \* @return the boolean  
 \*/* boolean canRedo() {  
 return undonePaths.size() > 0;  
 }  
  
 */\*\*  
 \* On click undo.  
 \*/* public void onClickUndo() {  
 if (canUndo()) {  
 undonePaths.add(new PathColorPair(paths.remove(paths.size() - 1)));  
 drawingChanged = true;  
 invalidate();  
 } else {  
 Toast.*makeText*(getContext(), "Nothing to undo", Toast.*LENGTH\_SHORT*)  
 .show();  
 }  
 }  
  
 */\*\*  
 \* On click redo.  
 \*/* public void onClickRedo() {  
 if (canRedo()) {  
 paths.add(new PathColorPair(undonePaths.remove(undonePaths.size() - 1)));  
 drawingChanged = true;  
 invalidate();  
 } else {  
 Toast.*makeText*(getContext(), "Nothing to redo", Toast.*LENGTH\_SHORT*)  
 .show();  
 }  
 }  
  
  
 @Override  
 protected void onDraw(Canvas canvas) {  
 if (bitmapSet) {  
 canvas.drawBitmap(canvasBitmap, 0, 0, canvasPaint);  
 }  
 for (PathColorPair pc : paths) {  
 drawPaint.setColor(pc.color);  
 canvas.drawPath(pc.path, drawPaint);  
 }  
 drawPaint.setColor(paintColor);  
 canvas.drawPath(drawPath, drawPaint);  
 }  
  
 @Override  
 public boolean onTouchEvent(MotionEvent event) {  
 float x = event.getX();  
 float y = event.getY();  
  
 switch (event.getAction()) {  
 case MotionEvent.*ACTION\_DOWN*:  
 bg\_music.start();  
 bg\_music.setLooping(true);  
 touch\_start(x, y);  
 break;  
 case MotionEvent.*ACTION\_MOVE*:  
 touch\_move(x, y);  
 break;  
 case MotionEvent.*ACTION\_UP*:  
 bg\_music.pause();  
 touch\_up();  
 break;  
 default:  
 return false;  
 }  
 invalidate();  
 return true;  
 }  
}

* Activity Contact Us

package com.ksa.apkcrew.childdrawing;  
  
  
import android.os.Bundle;  
import android.support.v7.app.AppCompatActivity;  
  
public class ContactActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_contact\_us*);  
 }  
}