Lab assignment 7

Due date: Friday, June 17

40 points

Team of 4 students

Divide the team into two subgroups with two team members per group.

The group works on the lab assignment and can discuss the code with another group. Each group demonstrates the lab to each other. The team then upload the solution from one of the subgroups.

Write the contribution to the lab assignment, the contribution percentage (maximum of 100%) on the comment box for each team member.

All the team members must agree on the percentage. If the team members have 50% and the lab assignment grade is 40/40, they get only 20 points. The instructor will review the paper summarizing and validate the percentage of each team member. If you have problems with team members, please let me know.

Problem

In this lab assignment you build an app that uses the camera app (or at least one of them) of a device, takes a picture, processes that picture, and uses an email app to send it to a friend. The Android Studio environment does not include a camera so we need an actual device, phone or tablet, to test the apps

Design

Accessing the Camera App and Taking a Picture, Photo App, Version 0

If we want to take a picture within an app, we have two options:

- Access the camera app and process its results.
- Use the Camera API and access the camera itself.

In Version 0 of the app, we do three things:

- Enable the user to take a picture using a camera app.
- Capture the picture taken.
- Display the picture.

We can use the PackageManager class to gather information about application packages installed on the device: we can use a PackageManager reference to check if the device has a camera, a front-facing camera, a microphone, a GPS, a compass, a gyroscope, supports Bluetooth, supports Wi-Fi, etc.

We can test if the device has a given feature by calling the hasSystemFeature method of the PackageManager class, passing the constant of the PackageManager class that corresponds to that feature, and that methods returns true if the feature described by the constant is present on the device, false if it is not. The hasSystemFeature method has the following API:

public boolean hasSystemFeature(String feature)

Once we know that the device has a feature we want to use, we can launch an activity to use that feature. In this case, we are interested in capturing the photo picture and in displaying it on the screen.

activity_main.xml

```
<?xml version="1.0" encoding="UTF-8"?>

    <LinearLayout
    tools:context="com.android.example.photo.MainActivity"
    android:orientation="vertical"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <ImageView android:layout_height="0dp" android:layout_width="match_parent"
android:id="@+id/picture" android:scaleType="fitCenter" android:layout_weight="8"/>


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

    <SeekBar android:id="@+id/red_bar" style="@style/seekBarStyle"
android:thumb="@drawable/red_thumb"/>

    <SeekBar android:id="@+id/green_bar" style="@style/seekBarStyle"
android:thumb="@drawable/green_thumb"/>

    <SeekBar android:id="@+id/blue_bar" style="@style/seekBarStyle"
android:thumb="@drawable/blue_thumb"/>
    </LinearLayout>
    <LinearLayout android:orientation="horizontal" android:layout_height="0dp"
android:layout_width="match_parent" android:layout_weight="1">

    <TextView android:id="@+id/red_tv" style="@style/textStyle"/>

    <TextView android:id="@+id/green_tv" style="@style/textStyle"/>
```

```
    <TextView android:id="@+id/blue_tv" style="@style/textStyle"/>

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

    <Button
        android:id="@+id/take_Photo"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"

        android:text="TAKE PICTURE"/>
        <Button
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:onClick="send"
            android:text="EMAIL"/>


</LinearLayout>
</LinearLayout>
```

MainActivity class

```
ublic class MainActivity extends AppCompatActivity {

    Button takePhoto;
    ActivityResultLauncher<Intent> activityResultLauncher;
    private ImageView imageView;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        imageView = (__________) findViewById(___________);

        activityResultLauncher = registerForActivityResult(new
ActivityResultContracts.StartActivityForResult(), new
ActivityResultCallback<ActivityResult>() {
            @Override
            public void onActivityResult(ActivityResult result) {
                if (result._______________ == RESULT_OK && result._______ != null) {
                    Bundle bundle = result._____________;
                    bitmap = (__________) bundle.get(_______________);
                    imageView._____________(___________);
                }
            }
        });

PackageManager manager = this.getPackageManager();
if (manager.hasSystemFeature(_____________________)
   Toast.makeText(MainActivity.this, "There is A CAMERA",
         Toast.LENGTH_SHORT).show();
```

```
        takePhoto = findViewById(R.id.take_Photo);
        _____________.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MediaStore._____________);
                if (intent.resolveActivity(getPackageManager()) != null) {
                    activityResultLauncher.launch(intent);
                } else {                    Toast.makeText(MainActivity.this, "There
is no app that support this action",
                            Toast.LENGTH_SHORT).show();
                }
            }

        });

    }

 }
```

Run the app. Click the “Take Picture” button. The picture shows up on the screen.

The Model: Graying the Picture, Photo App, Version 1

In Version 1 of our app, we gray the picture with a hard coded formula. Our Model for this app is the BitmapGrayer class. It encapsulates a Bitmap and a graying scheme that we can apply to that Bitmap.

BitmapGrayer class

```
public class BitmapGrayer {
    private Bitmap originalBitmap;
    private float redCoeff;
    private float greenCoeff;
    private float blueCoeff;

    public BitmapGrayer( Bitmap bitmap, float newRedCoeff,
                         float newGreenCoeff, float newBlueCoeff ) {
        originalBitmap = _________________
        setRedCoeff( newRedCoeff );
        setGreenCoeff( newRedCoeff );
        setBlueCoeff( newRedCoeff );
    }

    public float getRedCoeff( ) {
        return redCoeff;
    }

    public float getGreenCoeff( ) {
        return greenCoeff;
    }

    public float getBlueCoeff( ) {
```

```
        return blueCoeff;
    }

    public void setRedCoeff( float newRedCoeff ) {
        if( newRedCoeff >= 0 && newRedCoeff <= 1 ) {
            if( greenCoeff + blueCoeff + newRedCoeff <= 1 )
                redCoeff = newRedCoeff;
            else
                redCoeff = 1 - greenCoeff - blueCoeff;
        }
    }

    public void setGreenCoeff( float newGreenCoeff ) {
        if( newGreenCoeff >= 0 && newGreenCoeff <= 1 ) {
            if( redCoeff + blueCoeff + newGreenCoeff <= 1 )
                greenCoeff = newGreenCoeff;
            else
                greenCoeff = 1 - redCoeff - blueCoeff;
        }
    }

    public void setBlueCoeff( float newBlueCoeff ) {
        if( newBlueCoeff >= 0 && newBlueCoeff <= 1 ) {
            if( redCoeff + greenCoeff + newBlueCoeff <= 1 )
                blueCoeff = newBlueCoeff;
            else
                blueCoeff = 1 - redCoeff - greenCoeff;
        }
    }

    public Bitmap grayScale( ) {
        int width = originalBitmap.getWidth( );
        int height = originalBitmap.getHeight( );
        Bitmap.Config config = originalBitmap.getConfig( );
        Bitmap bitmap = Bitmap.createBitmap( width, height, config);
        for( int i = 0; i < width; i++ ) {
            for( int j = 0; j < height; j++ ) {
                int color = originalBitmap.getPixel( i, j );
                int shade = ( int ) ( redCoeff * Color.red( color )
                        + greenCoeff * Color.green( color )
                        + blueCoeff * Color.blue( color ) ) ;
                color = Color.argb( Color.alpha( color ), shade, shade, shade );
                bitmap.setPixel( i, j, color );
            }
        }
        return bitmap;
    }
}
```

Updated MainActivity class

```
private _____________ grayer;

public void onActivityResult(ActivityResult result) {
    if (result……) {
        Bundle bundle = ……
        bitmap = ……
        grayer = new _____________(_____,0.34f, 0.33f, 0.33f);
        bitmap = grayer._____________;
        imageView.setImageBitmap(bitmap);
    }
}
```

Run the app. Click the "Take Picture" button. The picture is now black-and-white.

Defining Shades of Gray Using SeekBars, Photo App, Version 2

The progress bar and the thumb are light blue by default. We can customize a seek bar by replacing the progress bar or the thumb by a drawable resource, for example an image from a file or a shape. We can do that inside an XML layout file or programmatically. In this app, since each seek bar represents the coefficient of a particular color, we customize the thumb of each of the three seek bars to reflect the three colors that they represent: red, green, and blue. We define three drawable resources (folder), one per color, and use them in the activity_main.xml file.

Red_thumb.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<shape android:shape="oval"
xmlns:android="http://schemas.android.com/apk/res/android">

    <size android:height="30dp" android:width="30dp"/>

    <solid android:color="#FF00"/>

    <stroke android:width="20dp" android:color="#AF00"/>

</shape>
```

blue_thumb.xml

```
<?xml version="1.0" encoding="UTF-8"?>

    <shape android:shape="oval"
xmlns:android="http://schemas.android.com/apk/res/android">

    <size android:height="30dp" android:width="30dp"/>

    <solid android:color="#F00F"/>

    <stroke android:width="20dp" android:color="#A00F"/>
```

```
</shape>
```

green_thumb.xml

```
<?xml version="1.0" encoding="UTF-8"?>

    <shape android:shape="oval"
xmlns:android="http://schemas.android.com/apk/res/android">

    <size android:height="30dp" android:width="30dp"/>

    <solid android:color="#F0F0"/>

    <stroke android:width="20dp" android:color="#A0F0"/>

</shape>
```

styles.xml

```
<?xml version="1.0"?>

   <resources>

    <!-- Base application theme. -->


    <style parent="Theme.AppCompat.Light.DarkActionBar" name="AppTheme">

    <!-- 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="seekBarStyle">

    <item name="android:layout_weight">1</item>

    <item name="android:layout_width">wrap_content</item>

    <item name="android:layout_height">wrap_content</item>

    <item name="android:layout_gravity">center_vertical</item>

    <item name="android:progress">0</item>

    <item name="android:max">100</item>

</style>
    -<style parent="@android:style/TextAppearance" name="textStyle">
```

```
    <item name="android:layout_weight">1</item>

    <item name="android:layout_width">0dp</item>

    <item name="android:layout_height">match_parent</item>

    <item name="android:gravity">center</item>

    <item name="android:textStyle">bold</item>

    <item name="android:textSize">30sp</item>

    <item name="android:text">0.0</item>

</style>

</resources>
```

activity_main.xml

```
<?xml version="1.0" encoding="UTF-8"?>

    <LinearLayout
    tools:context="com.android.example.photo.MainActivity"
    android:orientation="vertical"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <ImageView android:layout_height="0dp" android:layout_width="match_parent"
android:id="@+id/picture" android:scaleType="fitCenter" android:layout_weight="8"/>


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

    <SeekBar android:id="@+id/red_bar" style="@style/seekBarStyle"
android:thumb="@drawable/red_thumb"/>

    <SeekBar android:id="@+id/green_bar" style="@style/seekBarStyle"
android:thumb="@drawable/green_thumb"/>

    <SeekBar android:id="@+id/blue_bar" style="@style/seekBarStyle"
android:thumb="@drawable/blue_thumb"/>
    </LinearLayout>
    <LinearLayout android:orientation="horizontal" android:layout_height="0dp"
android:layout_width="match_parent" android:layout_weight="1">
```

```
    <TextView android:id="@+id/red_tv" style="@style/textStyle"/>

    <TextView android:id="@+id/green_tv" style="@style/textStyle"/>

    <TextView android:id="@+id/blue_tv" style="@style/textStyle"/>

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

    <Button
        android:id="@+id/take_Photo"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"

        android:text="TAKE PICTURE"/>

        <Button
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:onClick="send"
            android:text="EMAIL"/>


</LinearLayout>
</LinearLayout>
```

MathRounding.java

```
public class MathRounding {
    public static float keepTwoDigits( float f ) {
        int n = ( int ) ( 100 * f );
        return ( ( float ) n ) / 100;
    }
}
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
      private static String MA = "1";
    Button takePhoto;
    private BitmapGrayer grayer;
    ActivityResultLauncher<Intent> activityResultLauncher;
    private Bitmap bitmap;
    private ImageView imageView;
    private SeekBar redBar, greenBar, blueBar;
    private TextView redTV, greenTV, blueTV;
    Uri uri;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
        imageView = _____________________________

        redBar = (SeekBar) findViewById(R.id.red_bar);
        greenBar = ______________________________
        blueBar = _______________________________
        redTV = (TextView) findViewById(R.id.red_tv);

        greenTV = _________________________________

        blueTV = ___________________________________________;

        GrayChangeHandler gch = new GrayChangeHandler();
        redBar.setOnSeekBarChangeListener(gch);
        greenBar._________________________________
        blueBar._________________________________
        activityResultLauncher = registerForActivityResult(new
ActivityResultContracts.StartActivityForResult(), new
ActivityResultCallback<ActivityResult>() {
            @Override
            public void onActivityResult(ActivityResult result) {
                _____________


            }
        });

     ____________________________________________________________

    }


    private class GrayChangeHandler
            implements SeekBar.OnSeekBarChangeListener {
        public void onProgressChanged(SeekBar seekBar,
                                      int progress, boolean fromUser) {
            if (fromUser) {
                if (seekBar == redBar) {
                    grayer.setRedCoeff(progress / 100.0f);
                    redBar.setProgress((int) (100 * grayer.getRedCoeff()));
                    redTV.setText( "" + MathRounding.keepTwoDigits(
grayer.getRedCoeff( ) ) );
                } else if (seekBar == greenBar) {
                    grayer.setGreenCoeff(progress / 100.0f);
                    greenBar.setProgress((int) (100 * grayer.getGreenCoeff()));
                    greenTV.setText( "" + MathRounding.keepTwoDigits(
grayer.getGreenCoeff( ) ) );
                } else if (seekBar == blueBar) {
                    grayer.setBlueCoeff(progress / 100.0f);
                    blueBar.setProgress((int) (100 * grayer.getBlueCoeff()));
                    blueTV.setText( ""+ MathRounding.keepTwoDigits(
grayer.getBlueCoeff( ) ) );
                }
                bitmap = grayer.grayScale();
                imageView.setImageBitmap(bitmap);
```

```
            }
        }

        public void onStartTrackingTouch(SeekBar seekBar) {
        }

        public void onStopTrackingTouch(SeekBar seekBar) {
        }
    }
}
```

Run the app. Interact with the seek bars.

Storing the Picture, and using the Email app: Sending the grayed picture to a friend Photo App, Version 3

StorageUtility.java

```
public class StorageUtility {
    /*
     * This method write its bitmap parameter to external storage
     *      in the Pictures directory
     * @param activity, an Activity
     * @param bitmap, a Bitmap reference
     * @return returns the File it wrote bitmap to
     */
   // public static String MA="Directory Name";
    public static File writeToExternalStorage
    ( Activity activity, Bitmap bitmap ) throws IOException {
        // get state of external storage
        String storageState = Environment.getExternalStorageState( );

        File file = null;
        if( storageState.equals( _____________________________ ) ) {
            // get external storage directory
            File dir
                    = activity.getExternalFilesDir( _____________________________ );
            // generate a unique file name
            Date dateToday = new Date( );
            long ms = SystemClock.elapsedRealtime( );
           // String filename = "/" + dateToday + "_" + ms + ".png";
            String filename = "/" + dateToday + "_" + ms + ".png";

            // create a file to write to
            file = new File( dir + filename );
            long freeSpace = dir.getFreeSpace( ); // in bytes
            int bytesNeeded = bitmap.getByteCount( ); // in bytes
            if( bytesNeeded * 1.5 < freeSpace ) {
                // there is space for the bitmap
                try {
                    FileOutputStream fos = new FileOutputStream( file );
                    // write to file
                    boolean result
                            =
bitmap._____________________________________________;
                    fos.close( );
                    if( result )
                        return file;
                    else
                        throw new IOException( "Problem compressing the Bitmap"
                                + " to the output stream" );
                } catch( Exception e ) {
                    throw new IOException( "Problem opening the file for writing" );
                }
            }
            else
                throw new IOException( "Not enough space in external storage"
```

```
                        + " to write Bitmap" );
        }
        else
            throw new IOException( "No external storage found" );
    }
}
```

AndroidManifest.xml

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

<application android:theme="@style/AppTheme" android:supportsRtl="true"
android:label="@string/app_name" android:icon="@mipmap/ic_launcher"
android:allowBackup="true" >
```

Sharing files with API 24 or higher

If you are targeting Android API 24 or higher, private File URI resources (file:///) cannot be shared. You must instead wrap the File object as a content provider (content://) using the FileProvider class.

First, you must declare this FileProvider in your AndroidManifest.xml file within the <application> tag:

```
<application android:theme="@style/AppTheme" android:supportsRtl="true"
android:label="@string/app_name" android:icon="@mipmap/ic_launcher"
android:allowBackup="true" >


    <activity android:name=".MainActivity" android:screenOrientation="portrait">


    <intent-filter>

    <action android:name="android.intent.action.MAIN"/>

    <category android:name="android.intent.category.LAUNCHER"/>

     </intent-filter>

    </activity>
        <provider
            android:name="androidx.core.content.FileProvider"
            android:authorities="com.android.example.photo.fileprovider"
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/fileprovider" />
        </provider>
</application>
```

Next, create a resource directory called xml under the res folder. and create a fileprovider.xml and store in the xml folder.. Assuming you wish to grant access to the application's specific external storage directory, which requires requesting no additional permissions, you can declare this line as follows:

fileprovider.xml

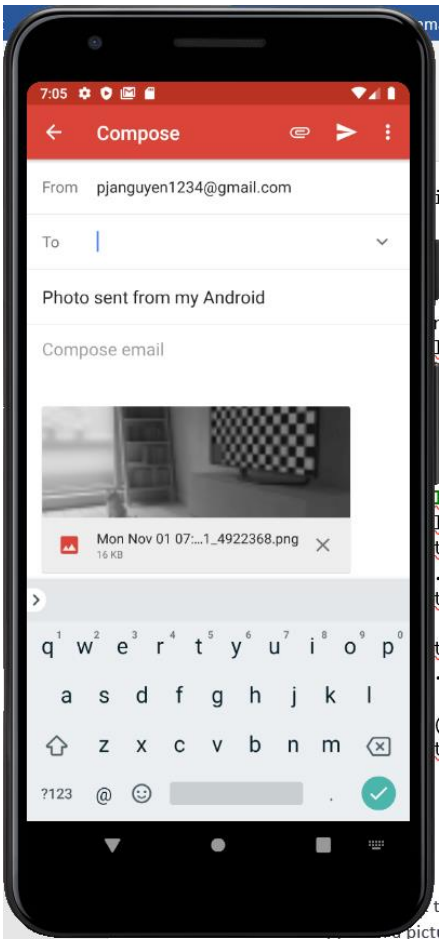
```
<?xml version="1.0" encoding="utf-8"?>
<paths>
        <external-files-path
        name="images"
        path="Pictures" />

</paths>
```

MainActivity.java

```
public void send(View view) {
          try {
              File file = StorageUtility._______________(____, _____);

              Intent emailIntent = new Intent(Intent._____________);
              emailIntent.setType("image/png");

              emailIntent.putExtra(Intent._________,
                      "Photo sent from my Android");

              Uri uri = FileProvider.getUriForFile(MainActivity.this,
"com.android.example.photo.fileprovider", file);
              emailIntent.putExtra(Intent.EXTRA_STREAM, _____);
              startActivity(Intent.createChooser(__________, "Share your
picture"));
              Toast.makeText(this, "EMAIL PICTURE",
Toast.LENGTH_LONG).show();

          } catch (IOException ioe) {
              Toast.makeText(this, ioe.getMessage()
                      + "; could not send it", Toast.LENGTH_LONG).show();
          }
}
```

Run the app.

Take the picture. Adjust the gray color. Click the “Email” button. Select the email app. We are now inside the email app with a picture attachment. Edit the email and send it.



Grading:

1. Submit the project in zip file (extension zip)
2. A pdf document that contains the content of the following files: MainActivity class, StorageUtility class, and AndroidManifest.xml. A zoom link or YouTube that demonstrate the user of the Hangman app (less than 5 minutes). Write the link on the comment section in the Dropbox.
   Your video should demonstrate the following features of the app:
   a. Taking the picture by pressing the button “TAKE PICTURE”
   b. Interacting with the seek bars to adjust the graying of the picture
   c. Using an email to send a grayed picture to yourself with an email title “Sending a picture” and email body “Attached is a picture for your review”.

d. Open an email and verify you got the picture

Note:
App is running correctly and met all the requirements is 40 points, but
Incomplete item 1(-30 points)
Incomplete item 2 (-10 points)
Incomplete item a (-5 points)
Incomplete item b (-5 points)
Incomplete item c (-5 points)
Incomplete item d (-5 points)