```
package com.example.nedtaylor.scramblegame;

import android.content.ClipData;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.util.DisplayMetrics;
import android.view.DragEvent;
import android.view.View;
import android.widget.Button;
import android.widget.GridLayout;
import android.widget.ImageView;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.Collections;

/**
 * Created by nedtaylor on 12/8/16.
 */

public class GameScreen extends AppCompatActivity implements View.OnDragListene
r, View.OnLongClickListener, View.OnClickListener {


    /**
     * the grid layout the image views will populate
     */
    private GridLayout layout;
    /**
     * the arrayList of bitmaps in their unscrambled order
     */
    private ArrayList<Tile> original;
    /**
     * the arrayList of bitmaps after they have been re-arranged
     */
    private ArrayList<Tile> scramble;
    /**
     * the 2D array of Tile objects
     */
    private Tile[][] tilearray;
    /**
     * the 2D array of imageViews that actually show the user the image
     */
    private ImageView[][] iv;
    /**
     * the dimensions of the grid
     */
    private int numCells;
    /**
     * the help button that displays the instructions
     */
    private Button help;
    /**
     * the quit button that solves the puzzle for a user
     */
    private Button quit;
    /**
     * the home button that sends the user back to the WelcomeActivity
     */
    private Button home;

    /**
```

```
     * This method displays the original state of the Screen, i.e. the
     * first screen the user sees
     *
     * @param savedInstanceState
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_game);//stretch if bitmaps don't take w
hole screen


        //get dimensions from size screen
        numCells = getIntent().getIntExtra("size", 0 );

        iv = new ImageView[numCells][numCells];
        tilearray = new Tile[numCells][numCells];

        int count = 0;


        //set parameters of grid layouts
        layout = (GridLayout) findViewById(R.id.grid);

        layout.setColumnCount(numCells);
        layout.setRowCount(numCells);
        layout.setId(count);
        layout.setAlignmentMode(GridLayout.ALIGN_BOUNDS);
        layout.setRowOrderPreserved(false);
        layout.setColumnOrderPreserved(false);
        layout.setOnDragListener(this);

        original = new ArrayList<Tile>();
        scramble = new ArrayList<Tile>();

        ImageView img;
        Bitmap b = null;

        //
        original = ImageItem.bm;

        //get the image scrambled
        for(int i = 0; i < original.size(); i++){
            scramble.add(i, original.get(i));
        }

        Collections.shuffle(scramble);

        DisplayMetrics displaymetrics = new DisplayMetrics();
        getWindowManager().getDefaultDisplay().getMetrics(displaymetrics);

        int width = displaymetrics.widthPixels;

        int bmWidth = width / numCells;
        int usefulHeight = bmWidth * (scramble.get(0).getBitmap().getHeight()/sc
ramble.get(0).getBitmap().getWidth());

        int c = 0;

        //set grid layout of image views
        for(int i = 0; i < layout.getColumnCount(); i++){
            for(int j = 0; j < layout.getRowCount(); j++){

                tilearray[i][j] = scramble.get(c);
```

```
                iv[i][j] = new ImageView(this);

                iv[i][j].setImageBitmap(tilearray[i][j].getBitmap());

                iv[i][j].setScaleType(ImageView.ScaleType.CENTER_CROP);
                layout.addView(iv[i][j], bmWidth, usefulHeight);

                iv[i][j].setOnLongClickListener(this);

                c++;

            }
        }
        //set buttons and click listeners
        home = (Button) findViewById(R.id.home);
        home.setOnClickListener(this);
        help = (Button) findViewById(R.id.Help);
        help.setOnClickListener(this);
        quit = (Button) findViewById(R.id.Quit);
        quit.setOnClickListener(this);

    }

    /**
     * TAKEN FROM: http://patrick-iv.github.io/2015/05/04/drag-n-drop/
     *
     * Utilized to facilitate the user dragging a tile of the GridLayout
     * and dropping it
     *
     * @param v the View
     * @param event an Event that recognizes that a drag is happening
     * @return boolean true
     */
    @Override
    public boolean onDrag(View v, DragEvent event) {

        final View view = (View) event.getLocalState();
        switch (event.getAction()) {
            //while dragging to a given index
            case DragEvent.ACTION_DRAG_LOCATION:

                if (view == v)
                    return true;

                final int index = calculateNewIndex(event.getX(), event.getY());

                layout.removeView(view);

                layout.addView(view, index);
                break;

            //when the tile is dropped
            case DragEvent.ACTION_DROP:

                view.setVisibility(View.VISIBLE);

                break;

            //when the drag has stopped
            case DragEvent.ACTION_DRAG_ENDED:
                if (!event.getResult()) {
                    view.setVisibility(View.VISIBLE);
                }
                //check for a win condition
                if(isWon()){
                    Toast.makeText(this, "YOU WIN!", Toast.LENGTH_SHORT).show();
```

```
                }

                break;
            }

        return true;
    }

    /**
     * Taken from: http://patrick-iv.github.io/2015/05/04/drag-n-drop/
     *
     * Calculating the new index of the tile that is being dragged
     *
     * @param x the x value of the tile being dragged
     * @param y the y value of the tile being dragged
     * @return int - the new index of the tile after being dragged
     */
    public int calculateNewIndex(float x, float y) {
        // calculate which column to move to
        final float cellWidth = layout.getWidth() / numCells;
        final int column = (int)(x / cellWidth);

        // calculate which row to move to
        final float cellHeight = layout.getHeight() / numCells;
        final int row = (int)Math.floor(y / cellHeight);

        // the items in the GridLayout is organized as a wrapping list
        // and not as an actual grid, so this is how to get the new index
        int index = row * numCells + column;
        if (index >= layout.getChildCount()) {
            index = layout.getChildCount() - 1;
        }

        return index;
    }

    /**
     *
     * Taken from: http://patrick-iv.github.io/2015/05/04/drag-n-drop/
     *
     * Recognizes a long click by the user (or press with a finger)
     * and raises the tile above the GridLayout, allowing for it to be
     * dragged
     *
     * @param v the View v
     * @return boolean true
     */
    @Override
    public boolean onLongClick(View v) {

        final ClipData data = ClipData.newPlainText("", "");

        //elevates the tile off the grid layout and allows for it to be dragged
while being held down

        View.DragShadowBuilder shadowBuilder = new View.DragShadowBuilder(v);

        v.startDrag(data, shadowBuilder, v, 0);
        v.setVisibility(View.INVISIBLE);

        return true;

    }

    /**
     *
```

```
     * Checks if the array has been won or not
     *
     *
     * @return true if a = original and false otherwise
     */
    public boolean isWon(){

        int win = 0;

        int count = 0;
        int bs = 0;


        //if every index of the scrambled arrayList equals the same index of the
 original arraylist
        //return true
        for(int i = 0; i < numCells*numCells; i++) {


            if(layout.getChildAt(i) instanceof ImageView) {

                ImageView bitmap = (ImageView) layout.getChildAt(i);
                Bitmap b = ((BitmapDrawable) bitmap.getDrawable()).getBitmap();

                if (!b.sameAs(original.get(count).getBitmap())) {
                    return false;
                }
            }
                count++;
            }

        return true;
    }

    /**
     *
     * Based on a button click, do different tasks
     *
     * @param v the View v
     */
    @Override
    public void onClick(View v) {

        //if button click = help, set alert dialog
        if(v.getId() == help.getId()){

            AlertDialog alertDialog = new AlertDialog.Builder(this).create();
            alertDialog.setTitle("Help");
            alertDialog.setMessage("You've made it to the game screen. Hold your finger down on an
y tile you " +
                    "want to move and drag it to the cell desired. Good luck solving! Press QUIT to have the
 puzzle " +
                    "solved for you or press HOME to return to the home screen.");
            alertDialog.setButton(AlertDialog.BUTTON_NEUTRAL, "OK",
                    new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int which) {
                            dialog.dismiss();
                        }
                    });
            alertDialog.show();

        }
        //if button click = home, go back to welcome screen
        else if(v.getId() == home.getId()){

            Intent intent = new Intent(this, WelcomeActivity.class);
```

```
            startActivity(intent);

        }
        //if button click = quit, solve the puzzle for them
        else if(v.getId() == quit.getId()){

            int count = 0;

            for(int i = 0; i < numCells; i++){
                for(int j = 0; j < numCells; j++){
                    iv[i][j].setImageBitmap(original.get(count).getBitmap());
                    count++;
                }
            }

        }

    }

}
```

```
package com.example.nedtaylor.scramblegame;

import android.graphics.Bitmap;
import android.provider.ContactsContract;

import java.util.ArrayList;

/**
 * Created by nedtaylor on 12/13/16.
 */

public class ImageItem {

    /**
     * A static reference to an ArrayList of Tiles
     * that is being sent from the Size Screen to the Game Screen
     *  which contains the original, unscrambled sequence of
     *  Tile objects
     */
    public static ArrayList<Tile> bm = null;


}
```

```
package com.example.nedtaylor.scramblegame;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.GridLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.view.View.OnClickListener;
/**
 * Created by Shayla Moore on 12/8/16.
 */

public class SendScreen extends Activity implements View.OnClickListener {

    /**
     * The method that populates the screen that the user sees upon its
     * creation.
     *
     * @param savedInstanceState
     */
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_email);

        Button sendButton = (Button) findViewById(R.id.send);
        sendButton.setOnClickListener(this);

        Button back = (Button) findViewById(R.id.backemail);
        back.setOnClickListener(this);

    }

    /**
     * Method to determine which button was clicked and act accordingly
     *
     * @param v View v
     */
    @Override
    public void onClick(View v) {

        if(v.getId() == R.id.send){
            Intent emailIntent = new Intent(Intent.ACTION_SEND);
            emailIntent.putExtra(Intent.EXTRA_EMAIL, new String[]{" "});
            emailIntent.putExtra(Intent.EXTRA_CC, new String[]{" "});
            emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Here's an Image for Your scramble
puzzle!");
            emailIntent.putExtra(Intent.EXTRA_TEXT, "Download the Attachment for your scra
mblepuzzle!");

            emailIntent.setType("text/plain");
            startActivity(Intent.createChooser(emailIntent, "Choose your email client.."
));
        }
        else if(v.getId() == R.id.backemail){
            Intent intent = new Intent(this, WelcomeActivity.class);
            startActivity(intent);
        }

    }
}
```

```
package com.example.nedtaylor.scramblegame;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;

import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.util.DisplayMetrics;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.GridLayout;
import android.widget.GridLayout.LayoutParams;
import android.widget.ImageView;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.Collections;


/**
 * Created by nedtaylor on 12/8/16.
 */

public class SizeScreen extends AppCompatActivity implements View.OnClickListener
 {
    /**
     * button to choose a 3x3 board
     */
    private Button b;
    /**
     * button to choose a 4x4 board
     */
    private Button b2;
    /**
     * button to choose a 5x5 board
     */
    private Button b3;
    /**
     * help button that sets an alert dialog
     */
    private Button help;
    /**
     * button to go back to the welcome activity
     */
    private Button back;
    /**
     * value to achieve entry into the gallery of photos
     */
    public static final int IMAGE_GALLERY_REQUEST = 20;
    /**
     * Bitmap to be parsed into chunks according to the difficulty
     * chosen
     */
    private Bitmap image;
    /**
```

```
     * the difficulty results in either 3, 4, or 5
     */
    private int numCols;

    /**
     *
     * the method that populates the screen as the user is navigated on to it,
     * centered on putting the buttons on the screen and setting onClickListener
s
     *
     * @param savedInstanceState
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        GridLayout layout = new GridLayout(this);

        setContentView(layout);

        int count = 0;

        //sets the buttons' text, id, and onClickListener

        b = new Button(this);
        b.setOnClickListener(this);
        b.setText("3x3");
        b.setId(count+1);
        layout.addView(b);

        b2 = new Button(this);
        b2.setOnClickListener(this);
        b2.setText("4x4");
        b2.setId(count+2);
        layout.addView(b2);

        b3 = new Button(this);
        b3.setOnClickListener(this);
        b3.setText("5x5");
        b3.setId(count+3);
        layout.addView(b3);

        help = new Button(this);
        help.setOnClickListener(this);
        help.setText("Help");
        help.setId(count+4);
        layout.addView(help);

        back = new Button(this);
        back.setOnClickListener(this);
        back.setText("Back");
        back.setId(count + 5);
        layout.addView(back);
    }

    /**
     *
     * Method to find out which button was clicked and act accordingly
     *
     * @param v the View v
     */
    @Override
    public void onClick(View v) {

            if(v.getId() == b.getId()) {
                numCols = 3;
```

```
                Intent photoIntent = new Intent(Intent.ACTION_PICK);
                File picDir = Environment.getExternalStoragePublicDirectory(Envi
ronment.DIRECTORY_PICTURES);
                String picturePath = picDir.getPath();
                Uri data = Uri.parse(picturePath);
                photoIntent.setDataAndType(data, "image/*");
                startActivityForResult(photoIntent, IMAGE_GALLERY_REQUEST);

            }
            else if(v.getId() == b2.getId()) {
                numCols = 4;
                Intent photoIntent = new Intent(Intent.ACTION_PICK);
                File picDir = Environment.getExternalStoragePublicDirectory(Envi
ronment.DIRECTORY_PICTURES);
                String picturePath = picDir.getPath();
                Uri data = Uri.parse(picturePath);
                photoIntent.setDataAndType(data, "image/*");
                startActivityForResult(photoIntent, IMAGE_GALLERY_REQUEST);
            }
            else if(v.getId() == b3.getId()) {
                numCols = 5;
                Intent photoIntent = new Intent(Intent.ACTION_PICK);
                File picDir = Environment.getExternalStoragePublicDirectory(Envi
ronment.DIRECTORY_PICTURES);
                String picturePath = picDir.getPath();
                Uri data = Uri.parse(picturePath);
                photoIntent.setDataAndType(data, "image/*");
                startActivityForResult(photoIntent, IMAGE_GALLERY_REQUEST);
            }
            else if(v.getId() == help.getId()) {

                AlertDialog alertDialog = new AlertDialog.Builder(SizeScreen.thi
s).create();
                alertDialog.setTitle("Help");
                alertDialog.setMessage("You have chosen to solve a puzzle! This screen will determ
ine the difficulty you would" +
                        " like to compete at\nEasy: 3x3\nMedium: 4x4\nHard 5x5\nThe next screen will be
 choosing your photo!");
                alertDialog.setButton(AlertDialog.BUTTON_NEUTRAL, "OK",
                        new DialogInterface.OnClickListener() {
                            public void onClick(DialogInterface dialog, int whic
h) {
                                dialog.dismiss();
                            }
                        });
                alertDialog.show();

            }
            else if(v.getId() == back.getId()){
                Intent intent = new Intent(this, WelcomeActivity.class);
                startActivity(intent);
            }


    }

    /**
     *
     * This method accesses the camera gallery and upon the user choosing a phot
o,
     * it parses it into an appropriate number of bitmap pieces, and then sends
that
     * arrayList to a static variable in ImageItem which is accessed in Game Scr
een
     *
     * @param requestCode requests access to gallery
```

```
     * @param resultCode contains whether or not the picture choosing was succes
sful
     * @param data the data of the picture
     */
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data
) {

        DisplayMetrics d = new DisplayMetrics();
        getWindowManager().getDefaultDisplay().getMetrics(d);


        //if
        if(resultCode == RESULT_OK){
            //
            if(requestCode == IMAGE_GALLERY_REQUEST){
                Uri imgData = data.getData();
                InputStream input;

                try{

                    input = getContentResolver().openInputStream(imgData);
                    image = BitmapFactory.decodeStream(input);

                    ArrayList<Bitmap> test = new ArrayList<>();
                    test.add(image);

                    ArrayList<Bitmap> array = chunkImage(image, numCols);
                    ArrayList<Tile> tileArray = new ArrayList<>();

                    for(int i = 0; i < numCols*numCols; i++){
                        Tile t = new Tile(array.get(i), i);
                        tileArray.add(t);
                    }

                    ImageItem.bm = tileArray;

                    Intent i = new Intent(this, GameScreen.class);

                    i.putExtra("size", numCols);
                    startActivity(i);

                }catch(FileNotFoundException fo){
                    System.out.println("can't find image");
                }
            }
        }
    }
    /**
     *
     * Parses the bitmap into pieces based on the difficulty chosen by the
     * user and puts those pieces into an ArrayList<Bitmap>
     *
     * @param b the bitmap of the image
     * @param i a number associated with the difficulty the user chooses
     * @return the ArrayList<Bitmap> containing the parsed image
     */
    public ArrayList<Bitmap> chunkImage(Bitmap b, int i){

        ArrayList<Bitmap> alb = new ArrayList<>( (int) Math.pow(i,2));

        Bitmap scale = Bitmap.createScaledBitmap(b, b.getWidth(), b.getHeight(),
 true);

        int chunkHeight = b.getHeight()/i;
        int chunkWidth = b.getWidth()/i;
```

```
        int yc = 0;

        for(int x=0; x<i; x++){
            int xc= 0;
            for(int y = 0; y < i; y++){

                alb.add(Bitmap.createBitmap(scale, xc, yc, chunkWidth, chunkHeig
ht));
                xc += chunkWidth;
            }
            yc += chunkHeight;
        }
        return alb;
    }

}
```

```
package com.example.nedtaylor.scramblegame;

import android.graphics.Bitmap;
import android.widget.ImageView;

import java.util.ArrayList;

/**
 * Created by nedtaylor on 12/16/16.
 */

public class Tile {

    /**
     * instance for the bitmap of the Tile object
     */
    private Bitmap b;
    /**
     * index of the Tile object
     */
    private int i;

    /**
     *
     * An overloaded constructor that populates a Tile object with a bitmap and
an int
     *
     * @param bm Bitmap sent in
     * @param id int sent in
     */
    public Tile(Bitmap bm, int id){
        this.b = bm;
        this.i = id;
    }

    /**
     * returns the bitmap of the Tile object
     * @return a Bitmap object associated with the Tile object
     */
    public Bitmap getBitmap(){
        return this.b;
    }

    /**
     * returns the id of the Tile object
     * @return the int associated with the Tile object
     */
    public int getId(){
        return this.i;
    }

    /**
     * sets bitmap to a passed in parameter
     * @param bit the parameter to set the bitmap to
     */
    public void setBitmap(Bitmap bit){
        this.b = bit;
    }

    /**
     * sets the id instance field on the tile object
     * @param id the parameter to set the id to
     */
    public void setId(int id){
        this.i = id;
    }
```

```
}
```

```
package com.example.nedtaylor.scramblegame;

import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Environment;
import android.provider.MediaStore;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Gallery;
import android.widget.ImageView;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.InputStream;


/**
 * This class represents the first screen of our application: the Welcome Screen
 */

public class WelcomeActivity extends AppCompatActivity implements View.OnClickLis
tener {


    /**
     * used to access the camera if "take a picture" is selected
     */
    static final int REQUEST_IMAGE_CAPTURE = 1;

    /**
     * Populates the WelcomeActivity screen according to the activity_welcome xm
l file
     *
     * @param savedInstanceState
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_welcome);

        Button b = (Button) findViewById(R.id.button);
        Button b2 = (Button) findViewById(R.id.button2);
        Button b3 = (Button) findViewById(R.id.button3);
        Button b4 = (Button) findViewById(R.id.button4);
        b4.setOnClickListener(this);

    }

    /**
     *
     * Based on which button was clicked, either access the camera
     * or navigate to a different screen
     *
     * @param v the View v
     */
    @Override
    public void onClick(View v){

        switch (v.getId()){
            case R.id.button:
```

```
                Intent intent = new Intent(this, SizeScreen.class);
                startActivity(intent);
                break;

            case R.id.button2:

                Intent intent2 = new Intent(this, SendScreen.class);
                startActivity(intent2);
                break;

            case R.id.button3:

                AlertDialog alertDialog = new AlertDialog.Builder(WelcomeActivit
y.this).create();
                alertDialog.setTitle("Help");
                alertDialog.setMessage("This is the Welcome Screen of the Scrambler! You have tw
o options at this point: " +
                "do you want to solve a puzzle or send a picture to your friend? Solving a puzzle will bring you
 to the " +
                        "next screen where you pick the difficulty of your puzzle, while sending a picture
will bring you" +
                        "to your gallery of pictures where you can select which photo to send!");
                alertDialog.setButton(AlertDialog.BUTTON_NEUTRAL, "OK",
                        new DialogInterface.OnClickListener() {
                            public void onClick(DialogInterface dialog, int whic
h) {
                                dialog.dismiss();
                            }
                        });
                alertDialog.show();
                break;

            case R.id.button4:

                Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CA
PTURE);
                if (takePictureIntent.resolveActivity(getPackageManager()) != nu
ll) {
                    startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPT
URE);
                }
        }
    }

}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:text="Directions for Sending the Email:"
        android:ems="10"
        android:id="@+id/editText"
        android:layout_marginTop="25dp"
        android:layout_width="match_parent"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:textAppearance="@android:style/TextAppearance.Material.Large.Inv
erse" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textMultiLine"
        android:ems="10"
        android:id="@+id/editText2"
        android:layout_marginTop="20dp"
        android:text="Click the &amp;quot;Send Email&amp;quot; button and choose the Ema
il Client of your choice. Then click &amp;quot;Choose Attachment&amp;quot; to send an im
age to another user through their email that you'd like them to solve a puzzle o
f!"
        android:layout_below="@+id/editText"
        android:layout_centerHorizontal="true" />

    <Button
        android:text="Send Email"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/send"
        android:layout_weight="1"
        android:layout_below="@+id/editText2"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="28dp" />

    <Button
        android:text="Back"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/backemail"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginLeft="22dp"
        android:layout_marginStart="22dp" />

</RelativeLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/relativelayout"
    android:orientation="vertical">

    <Button
        android:text="Home"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="29dp"
        android:id="@+id/home" />

    <Button
        android:text="Quit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/Help"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginRight="22dp"
        android:layout_marginEnd="22dp"
        android:id="@+id/Quit" />

    <GridLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@+id/home"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="19dp"
        android:id="@+id/grid">

    </GridLayout>

    <Button
        android:text="Help"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/Help"
        android:layout_below="@+id/grid"
        android:layout_centerHorizontal="true"
        android:layout_alignParentBottom="true"/>
</RelativeLayout>
```

```
<?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_welcome"
    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.nedtaylor.scramblegame.WelcomeActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to Scrambler!"
        android:id="@+id/textView"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginLeft="112dp"
        android:layout_marginStart="112dp" />

    <Button
        android:text="Play a Game"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button"
        android:onClick="onClick"
        android:layout_alignBaseline="@+id/button2"
        android:layout_alignBottom="@+id/button2"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_marginRight="30dp"
        android:layout_marginEnd="30dp" />

    <Button
        android:text="Send a picture"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button2"
        android:onClick="onClick"
        android:layout_below="@+id/textView"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="99dp" />

    <Button
        android:text="Help"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button3"
        android:onClick="onClick"
        android:layout_marginBottom="51dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginLeft="21dp"
        android:layout_marginStart="21dp" />

    <Button
        android:text="Take a Picture"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button4"
        android:onClick="onClick"
```

```
        android:layout_alignBaseline="@+id/button3"
        android:layout_alignBottom="@+id/button3"
        android:layout_alignLeft="@+id/button"
        android:layout_alignStart="@+id/button" />
</RelativeLayout>
```