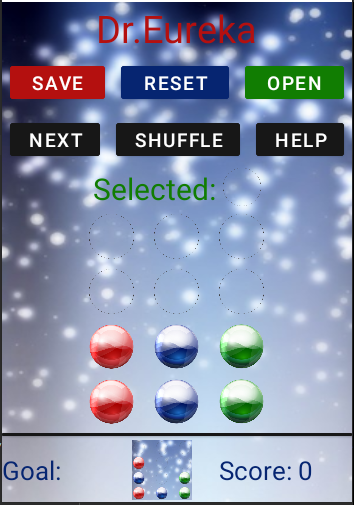
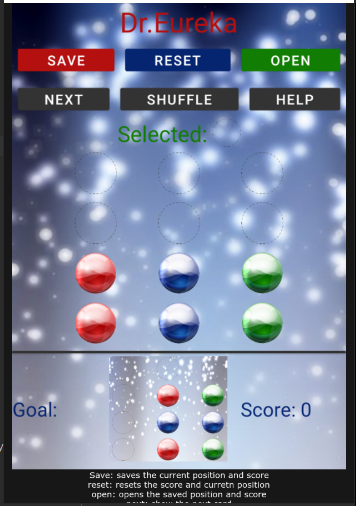
Gurjap Singh Hajra

App: Dr.Eureka



Layout

Acitity\_Main.xml

<?xml version="1.0" encoding="utf-8"?><!--  
Name: Gurjap Singh Hajra  
Date: Jan 29, 2021  
Purpose: Make Dr.eureka using objects in android studio  
-->  
<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\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/grey"  
 android:gravity="center"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
 <!-- shows the logo -->  
 <ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/red\_ball" />  
 <!-- Shows the text " By: gurjap Singh Hajra"-->  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="By: Gurjap Singh Hajra"  
 android:textColor="@color/white"  
 android:textSize="20dp" />  
  
</LinearLayout>

Activity\_home\_screen.xml

<?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\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".Home\_Screen"  
 android:orientation="vertical"  
 android:background="@color/grey">  
 <!--first Linear layout to hold the image and the textview -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:gravity="center"  
 android:orientation="vertical">  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:src="@drawable/red\_ball"  
 android:layout\_weight="1"/>  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:text="MineSweeper"  
 android:textStyle="bold"  
 android:textColor="@color/white"  
 android:textSize="65dp"  
 android:layout\_weight="2"  
 android:gravity="center"/>  
 </LinearLayout>  
 <!-- second linear loyout to hold the buttons-->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:layout\_weight="1.2">  
 <Button  
 android:layout\_weight="1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Play"  
 android:onClick="play"  
 android:backgroundTint="@color/green"  
 android:textSize="40dp"/>  
 <Button  
 android:layout\_weight="1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Instruction"  
 android:onClick="instruction"  
 android:backgroundTint="@color/blue"  
 android:textSize="40dp"/>  
 <Button  
 android:layout\_weight="1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Quit"  
 android:onClick="quit"  
 android:backgroundTint="@color/red"  
 android:textSize="40dp"/>  
 </LinearLayout>  
</LinearLayout>

Activity\_intructions.xml

<?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\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/grey"  
 android:gravity="center\_horizontal"  
 tools:context=".intructions">  
 <!-- initials ScrollView -->  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <!-- makes a Linear Layout in the Scroll View-->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center\_horizontal"  
 android:orientation="vertical">  
 <!-- adds all the instruction pictures to the scroll view-->  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="710dp"  
 android:src="@drawable/ins1" />  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="710dp"  
 android:src="@drawable/ins2" />  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="710dp"  
 android:src="@drawable/ins3" />  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="710dp"  
 android:src="@drawable/ins4" />  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="710dp"  
 android:src="@drawable/ins5" />  
 <!-- add back button -->  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:backgroundTint="@color/white"  
 android:onClick="back"  
 android:text="Back"  
 android:textColor="@color/grey"  
 android:textSize="40dp" />  
 <!-- adds play button -->  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:backgroundTint="@color/white"  
 android:onClick="play"  
 android:text="PLay"  
 android:textColor="@color/grey"  
 android:textSize="40dp" />  
 </LinearLayout>  
 </ScrollView>  
</LinearLayout>

Activity\_game\_screen.xml

<?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\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".Game\_Screen"  
 android:background="@drawable/game\_background"  
 android:gravity="center\_horizontal"  
 android:orientation="vertical">  
 <!-- Title -->  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Dr.Eureka"  
 android:textSize="50dp"  
 android:textColor="@color/red"  
 android:onClick="ins"/>  
 <!-- Linear layout for the first three buttons-->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="center\_horizontal">  
 <Button  
 android:layout\_weight="1"  
 android:onClick="save"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Save"  
 android:textSize="25dp"  
 android:layout\_margin="10dp"  
 android:backgroundTint="@color/red"/>  
 <Button  
 android:layout\_weight="1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="reset"  
 android:text="reset"  
 android:textSize="25dp"  
 android:layout\_margin="10dp"  
 android:backgroundTint="@color/blue"/>  
 <Button  
 android:layout\_weight="1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="open"  
 android:text="open"  
 android:textSize="25dp"  
 android:layout\_margin="10dp"  
 android:backgroundTint="@color/green"/>  
  
 </LinearLayout>  
 <!-- Linear layout for the second half of the buttons-->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="center\_horizontal">  
 <Button  
 android:layout\_weight="1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="next"  
 android:textSize="25dp"  
 android:layout\_margin="10dp"  
 android:backgroundTint="@color/grey"  
 android:onClick="next"/>  
 <Button  
 android:layout\_weight="1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Shuffle"  
 android:textSize="25dp"  
 android:layout\_margin="10dp"  
 android:backgroundTint="@color/grey"  
 android:onClick="shuffleClick"/>  
 <Button  
 android:layout\_weight="1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Help"  
 android:textSize="25dp"  
 android:layout\_margin="10dp"  
 android:backgroundTint="@color/grey"  
 android:onClick="help"/>  
 </LinearLayout>  
 <!-- Linear layout to show the selected ball -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="center\_horizontal">  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="40dp"  
 android:text="Selected: "  
 android:textColor="@color/green"/>  
 <ImageView  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:id="@+id/heldItem"  
 android:src="@drawable/empty"/>  
 </LinearLayout>  
 <!-- Linear layout for the grid -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="center\_horizontal">  
 <!-- 3 grid layout for each column -->  
 <GridLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:columnCount="1"  
 android:rowCount="4"  
 android:id="@+id/grid1"  
 android:onClick="click1"/>  
 <GridLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:columnCount="1"  
 android:rowCount="4"  
 android:id="@+id/grid2"  
 android:onClick="click2"/>  
 <GridLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:columnCount="1"  
 android:rowCount="4"  
 android:id="@+id/grid3"  
 android:onClick="click3"/>  
 </LinearLayout>  
 <!-- Button shown as a bar to divide the screen -->  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="15dp"  
 android:backgroundTint="@color/grey"/>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="center">  
 <!-- text : "Goal" -->  
 <TextView  
 android:layout\_weight="1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/blue"  
 android:text="Goal: "  
 android:textSize="35dp"/>  
 <!-- shows a picture of pattern is required -->  
 <ImageView  
 android:id="@+id/cardImage"  
 android:layout\_width="100dp"  
 android:layout\_height="80dp"  
 android:layout\_weight="1"  
 android:src="@drawable/p1" />  
 <!-- Text view for score -->  
 <TextView  
 android:layout\_weight="1"  
 android:id="@+id/scoreB"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/blue"  
 android:text="Score: 0"  
 android:textSize="35dp"/>  
 </LinearLayout>  
</LinearLayout>

Ball.java

package com.example.dreureka;  
  
  
import android.widget.ImageView;  
  
public class Ball {  
  
 // instance variable  
 private char colour;  
  
 //constructors  
 public Ball(){  
 colour = 'e';  
 }  
 public Ball(char color)  
 {  
 this.setColour(color);  
 }  
  
 //accessors  
 public char getColour(){  
 return colour;  
 }  
 public int getPic(){  
 if (colour == 'r')  
 return R.drawable.*red\_ball*;  
 else if(colour == 'b')  
 return R.drawable.*blue\_ball*;  
 else if(colour == 'g')  
 return R.drawable.*green\_ball*;  
 else if(colour == 'e')  
 return R.drawable.*empty*;  
 return -1;  
 }  
 public String toString(){  
 return String.*format*("the colour of the ball is %s.",colour);  
 }  
  
 //Mutator  
 public void setColour(char color){  
 colour = color;  
 }  
 // sets the picture of the ball to the image view  
 public void setPic(ImageView view){  
 if (colour == 'r')  
 view.setImageResource(R.drawable.*red\_ball*);  
 else if(colour == 'b')  
 view.setImageResource(R.drawable.*blue\_ball*);  
 else if(colour == 'g')  
 view.setImageResource(R.drawable.*green\_ball*);  
 else if(colour == 'e')  
 view.setImageResource(R.drawable.*empty*);  
 }  
  
 //facilitators  
 public boolean equals(Ball ball){  
 if(colour==ball.getColour()){  
 return true;  
 }  
 return false;  
 }  
 //compares the colour of the balls  
 public int compareTo(Ball ball){  
 if(colour>ball.getColour()){  
 return 1;  
 }else if(colour == ball.getColour()){  
 return 0;  
 }  
 return -1;  
 }  
}

Card.java

package com.example.dreureka;  
  
import android.widget.ImageView;  
  
public class Card {  
 //instance variables  
 int cardNum;  
  
 //holds all the picture for the cards  
 int picNum[] = {R.drawable.*p0*, R.drawable.*p1*, R.drawable.*p2*, R.drawable.*p3*, R.drawable.*p4*, R.drawable.*p5*, R.drawable.*p6*, R.drawable.*p7*,  
 R.drawable.*p8*, R.drawable.*p9*, R.drawable.*p10*, R.drawable.*p11*, R.drawable.*p12*, R.drawable.*p13*, R.drawable.*p14*,  
 R.drawable.*p15*, R.drawable.*p16*, R.drawable.*p17*, R.drawable.*p18*, R.drawable.*p19*, R.drawable.*p20*, R.drawable.*p21*,  
 R.drawable.*p22*, R.drawable.*p23*, R.drawable.*p24*, R.drawable.*p25*, R.drawable.*p26*, R.drawable.*p27*, R.drawable.*p28*,  
 R.drawable.*p29*, R.drawable.*p30*, R.drawable.*p31*, R.drawable.*p32*, R.drawable.*p33*, R.drawable.*p34*, R.drawable.*p35*,  
 R.drawable.*p36*, R.drawable.*p37*, R.drawable.*p39*, R.drawable.*p40*, R.drawable.*p41*, R.drawable.*p42*, R.drawable.*p43*,  
 R.drawable.*p44*, R.drawable.*p45*, R.drawable.*p46*, R.drawable.*p47*, R.drawable.*p48*, R.drawable.*p49*, R.drawable.*p50*,  
 R.drawable.*p51*, R.drawable.*p52*, R.drawable.*p53*, R.drawable.*p54*, R.drawable.*p55*, R.drawable.*p56*, R.drawable.*p57*,  
 R.drawable.*p58*, R.drawable.*p59*, R.drawable.*p60*, R.drawable.*p61*};  
  
 //constructors  
 public Card() {  
 cardNum = (int) Math.*random*() \* picNum.length - 1;  
 }  
  
 public Card(int CardNum) {  
 cardNum = CardNum;  
 }  
  
 //accessors  
 public int getCardNum() {  
 return cardNum;  
 }  
  
 //return the picture  
 public int getPic() {  
 return picNum[cardNum];  
 }  
 public String toString() {  
 return String.*format*("the card num is %d.", cardNum);  
 }  
  
 //Mutators  
 public void setCardNum(int CardNum) {  
 cardNum = CardNum;  
 }  
  
  
 //facilitators  
 public boolean equals(Card card) {  
 if (cardNum == card.getCardNum()) {  
 return true;  
 }  
 return false;  
 }  
 // compare the card numbers  
 public int compareTo(Card card) {  
 if (cardNum > card.getCardNum()) {  
 return 1;  
 } else if (cardNum == card.getCardNum()) {  
 return 0;  
 }  
 return -1;  
 }  
 // sets the pic  
 public void setPic(ImageView picture) {  
 picture.setImageResource(picNum[cardNum]);  
  
 }  
  
}

Deck.java

package com.example.dreureka;  
  
  
public class Deck {  
  
 //instance variables  
 private int count;  
 private Card data[] = new Card[70];  
  
 public Deck(){  
 shuffle();  
 }  
 public Deck(Card card){  
 data[count] = card;  
 count++;  
 }  
 // pushes the new card into the deck  
 public void push(Card addMe) {  
 data[count] = addMe;  
 count++;  
 }  
 //return the size  
 public int size() {  
 return count;  
 }  
 //returns the if the deck is full  
 public boolean isFull() {  
 return (count == data.length);  
 }  
 public Card pop() {  
 count--;  
 return data[count];  
 }  
 // return the top card  
 public Card peek() {  
 return data[count-1];  
 }  
 // returns if the deck is empty  
 public boolean isEmpty() {  
 return count == 0;  
 }  
 // clears the deck  
 public void clear() {  
 count = 0;  
 }  
 public void shuffle(){  
 //TO DO: Make an array for each instance variable. A card's pieces are all in the same index  
 int cardNum[] = new int [62];  
 for(int i=0;i<61;i++){  
 cardNum[i] = i;  
 }  
 //TO DO: Randomize the order of the arrays  
 for (int i = 0; i < 100; i++) {  
 int r1 = (int) (Math.*random*() \* cardNum.length);  
 int r2 = (int) (Math.*random*() \* cardNum.length);  
 int temp = cardNum[r1];  
 cardNum[r1] = cardNum[r2];  
 cardNum[r2] = temp;  
  
 }  
 count = 0;  
 //TO DO: push all (now in random order) into the Deck  
 for (int i = 0; i < cardNum.length; i++) {  
 Card c = new Card(cardNum[i]);  
 push(new Card(cardNum[i]));  
 }  
 }  
  
}

Gamge\_Screen.java

package com.example.dreureka;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.Activity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
import android.util.Log;  
import android.view.View;  
import android.view.WindowManager;  
import android.view.animation.Animation;  
import android.view.animation.AnimationUtils;  
import android.widget.FrameLayout;  
import android.widget.GridLayout;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
public class Game\_Screen extends AppCompatActivity {  
 int row = 4;  
 // 3 tubes for the 3 column  
 Tube tube1 = new Tube();  
 Tube tube2 = new Tube();  
 Tube tube3 = new Tube();  
 int score = 0;  
  
 // stores the pictures for each column  
 ImageView pics1[] = new ImageView[row];  
 ImageView pics2[] = new ImageView[row];  
 ImageView pics3[] = new ImageView[row];  
  
 // held stores the ball that is currently on hold  
 Ball held = null;  
  
 // initializes the deck  
 Deck deck = new Deck();  
 //runs when the Activity starts  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_game\_\_screen*);  
  
  
 //hides the notification bar  
 this.getWindow().setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*, WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);  
 //hides the navigation bar  
 getWindow().getDecorView().setSystemUiVisibility(View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION*);  
  
  
 //pushes the default values into the tubes  
 tube1.push(new Ball('r'));  
 tube1.push(new Ball('r'));  
 tube2.push(new Ball('b'));  
 tube2.push(new Ball('b'));  
 tube3.push(new Ball('g'));  
 tube3.push(new Ball('g'));  
  
 // Grid layout for each of the columns  
 GridLayout g1 = (GridLayout) findViewById(R.id.*grid1*);  
 GridLayout g2 = (GridLayout) findViewById(R.id.*grid2*);  
 GridLayout g3 = (GridLayout) findViewById(R.id.*grid3*);  
  
 // dimensions of the balls shown in the columns  
 int height = 45;  
 int width = 45;  
  
 // margins for the of the balls shown in the columns  
 int topMargin = 5;  
 int bottomMargin = 5;  
 int rightMargin = 10;  
 int leftMargin = 10;  
  
 //for each of the girview this set the images  
  
 //for g1  
 int m = 0;  
 for (int i = 0; i < row; i++) {  
 pics1[m] = new ImageView(this);  
 setpic(pics1[m], m, 1);  
 pics1[m].setId(m);  
 //LinearLayout.LayoutParams parms = new LinearLayout.LayoutParams(100, 100);  
 FrameLayout.LayoutParams layoutParams = new FrameLayout.LayoutParams(FrameLayout.LayoutParams.*WRAP\_CONTENT*, FrameLayout.LayoutParams.*WRAP\_CONTENT*);  
 layoutParams.leftMargin = leftMargin;  
 layoutParams.rightMargin = rightMargin;  
 layoutParams.topMargin = topMargin;  
 layoutParams.bottomMargin = bottomMargin;  
 layoutParams.width = width;  
 layoutParams.height = height;  
 //pics[m].setLayoutParams(parms);  
 pics1[m].setLayoutParams(layoutParams);  
 pics1[m].setScaleType(ImageView.ScaleType.*FIT\_XY*);  
 g1.addView(pics1[m]);  
 m++;  
 }  
 // g2  
 m = 0;  
 for (int i = 0; i < row; i++) {  
 pics2[m] = new ImageView(this);  
 setpic(pics2[m], m, 2);  
 pics2[m].setId(m);  
 //LinearLayout.LayoutParams parms = new LinearLayout.LayoutParams(100, 100);  
 FrameLayout.LayoutParams layoutParams = new FrameLayout.LayoutParams(FrameLayout.LayoutParams.*WRAP\_CONTENT*, FrameLayout.LayoutParams.*WRAP\_CONTENT*);  
 layoutParams.leftMargin = leftMargin;  
 layoutParams.rightMargin = rightMargin;  
 layoutParams.topMargin = topMargin;  
 layoutParams.bottomMargin = bottomMargin;  
 layoutParams.width = width;  
 layoutParams.height = height;  
 //pics[m].setLayoutParams(parms);  
 pics2[m].setLayoutParams(layoutParams);  
 pics2[m].setScaleType(ImageView.ScaleType.*FIT\_XY*);  
 g2.addView(pics2[m]);  
 m++;  
 }  
 //g3  
 m = 0;  
 for (int i = 0; i < row; i++) {  
 pics3[m] = new ImageView(this);  
 setpic(pics3[m], m, 3);  
 pics3[m].setId(m);  
 //LinearLayout.LayoutParams parms = new LinearLayout.LayoutParams(100, 100);  
 FrameLayout.LayoutParams layoutParams = new FrameLayout.LayoutParams(FrameLayout.LayoutParams.*WRAP\_CONTENT*, FrameLayout.LayoutParams.*WRAP\_CONTENT*);  
 layoutParams.leftMargin = leftMargin;  
 layoutParams.rightMargin = rightMargin;  
 layoutParams.topMargin = topMargin;  
 layoutParams.bottomMargin = bottomMargin;  
 layoutParams.width = width;  
 layoutParams.height = height;  
 //pics[m].setLayoutParams(parms);  
 pics3[m].setLayoutParams(layoutParams);  
 pics3[m].setScaleType(ImageView.ScaleType.*FIT\_XY*);  
 g3.addView(pics3[m]);  
 m++;  
 }  
 display();  
 }  
  
 //hides the navigation bar  
 @Override  
 public void onWindowFocusChanged(boolean hasFocus) {  
 super.onWindowFocusChanged(hasFocus);  
 if (hasFocus) {  
 getWindow().getDecorView().setSystemUiVisibility(  
 View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* | View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* | View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* | View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* | View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* | View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY*);  
 }  
 }  
 // set the pictures on the grid views  
 public void setpic(ImageView i, int pos, int gridNum) {  
 // calculate the 2d positions of the button clicked  
 if (gridNum == 1) {  
 i.setImageResource(tube1.getBallAt(Math.*abs*(pos - 3)).getPic());  
 } else if (gridNum == 2) {  
 i.setImageResource(tube2.getBallAt(Math.*abs*(pos - 3)).getPic());  
 } else if (gridNum == 3) {  
 i.setImageResource(tube3.getBallAt(Math.*abs*(pos - 3)).getPic());  
 }  
  
 }  
 // runs when grid 1 is clicked  
 //code inside the click1 is the same in click2 and click3  
 // with just different grids  
 public void click1(View view) {  
 Log.*d*("tag","click1 ran");  
 // Makes an animation  
 Animation slideDown;  
 Animation slideUp;  
 slideDown = AnimationUtils.*loadAnimation*(getApplicationContext(), R.anim.*slide\_down*);  
 slideUp = AnimationUtils.*loadAnimation*(getApplicationContext(), R.anim.*slide\_up*);  
 //gets the id of the heldItem  
 ImageView heldItem = findViewById(R.id.*heldItem*);  
 // runns if there is no ball selected  
 if (held == null && !(tube1.isEmpty())) {  
 Log.*d*("tag","first if ran");  
 // get the top item in the stack  
 held = tube1.pop();  
 // set the image to the held item  
 heldItem.setImageResource(held.getPic());  
 heldItem.startAnimation(slideDown);  
 //runs if there is a ball selected  
 } else if (held != null) {  
 Log.*d*("tag","second if ran");  
 tube1.push(held);  
 heldItem.startAnimation(slideUp);  
 heldItem.setImageResource(R.drawable.*empty*);  
 held = null;  
 //error message  
 } else {  
 Log.*d*("tag","else ran");  
 Toast.*makeText*(getApplicationContext(), "Not possible", Toast.*LENGTH\_SHORT*).show();  
 }  
 //redaws after the changes have been made  
 redraw();  
 }  
 // runs when grid 2 is clicked  
 public void click2(View view) {  
 // Makes an animation  
 Animation slideDown;  
 Animation slideUp;  
 slideDown = AnimationUtils.*loadAnimation*(getApplicationContext(), R.anim.*slide\_down*);  
 slideUp = AnimationUtils.*loadAnimation*(getApplicationContext(), R.anim.*slide\_up*);  
 ImageView heldItem = findViewById(R.id.*heldItem*);  
 if (held == null && !(tube2.isEmpty())) {  
 // get the top item in the stack  
 held = tube2.pop();  
 // set the image to the held item  
 heldItem.setImageResource(held.getPic());  
 heldItem.startAnimation(slideDown);  
 } else if (held != null) {  
 tube2.push(held);  
 heldItem.startAnimation(slideUp);  
 heldItem.setImageResource(R.drawable.*empty*);  
 held = null;  
 } else {  
 Toast.*makeText*(getApplicationContext(), "Not possible", Toast.*LENGTH\_SHORT*).show();  
 }  
 redraw();  
 }  
 // runs when grid 3 is clicked  
 public void click3(View view) {  
 // Makes an animation  
 Animation slideDown;  
 Animation slideUp;  
 slideDown = AnimationUtils.*loadAnimation*(getApplicationContext(), R.anim.*slide\_down*);  
 slideUp = AnimationUtils.*loadAnimation*(getApplicationContext(), R.anim.*slide\_up*);  
 ImageView heldItem = findViewById(R.id.*heldItem*);  
 if (held == null && !(tube3.isEmpty())) {  
 // get the top item in the stack  
 held = tube3.pop();  
 // set the image to the held item  
 heldItem.setImageResource(held.getPic());  
 heldItem.startAnimation(slideDown);  
 } else if (held != null && !(tube3.isFull())) {  
 tube3.push(held);  
 heldItem.startAnimation(slideUp);  
 heldItem.setImageResource(R.drawable.*empty*);  
 held = null;  
 } else {  
 Toast.*makeText*(getApplicationContext(), "Not possible", Toast.*LENGTH\_SHORT*).show();  
 }  
 redraw();  
 }  
 // redraw the grid  
 public void redraw() {  
 // draw the first column  
 pics1[0].setImageResource(tube1.getBallAt(3).getPic());  
 pics1[1].setImageResource(tube1.getBallAt(2).getPic());  
 pics1[2].setImageResource(tube1.getBallAt(1).getPic());  
 pics1[3].setImageResource(tube1.getBallAt(0).getPic());  
 // draws the second column  
 pics2[0].setImageResource(tube2.getBallAt(3).getPic());  
 pics2[1].setImageResource(tube2.getBallAt(2).getPic());  
 pics2[2].setImageResource(tube2.getBallAt(1).getPic());  
 pics2[3].setImageResource(tube2.getBallAt(0).getPic());  
 // draws the third comumn  
 pics3[0].setImageResource(tube3.getBallAt(3).getPic());  
 pics3[1].setImageResource(tube3.getBallAt(2).getPic());  
 pics3[2].setImageResource(tube3.getBallAt(1).getPic());  
 pics3[3].setImageResource(tube3.getBallAt(0).getPic());  
 // shows the score  
 TextView s = findViewById(R.id.*scoreB*);  
 s.setText("Score: " + score);  
 }  
 // resets the grid  
 public void reset(View view) {  
 // clears all the columns  
 tube1.clear();  
 tube2.clear();  
 tube3.clear();  
 // pushes the default values of the into the columns  
 tube1.push(new Ball('r'));  
 tube1.push(new Ball('r'));  
 tube2.push(new Ball('b'));  
 tube2.push(new Ball('b'));  
 tube3.push(new Ball('g'));  
 tube3.push(new Ball('g'));  
 held = null;  
 ImageView heldItem = findViewById(R.id.*heldItem*);  
 heldItem.setImageResource(R.drawable.*empty*);  
 //resets the score  
 score = 0;  
 //redraws  
 redraw();  
 }  
 // displays any changes  
 public void display() {  
 // if the deck is empty we display the next card  
 if (!deck.isEmpty()) {  
 Card c = deck.pop();  
 ImageView i = (ImageView) findViewById(R.id.*cardImage*);  
 i.setImageResource(c.getPic());  
 }//if the deck is empty it shuffles and displays the next card  
 else {  
 deck.shuffle();  
 display();  
 }  
 }  
 //shows the next card  
 public void next(View view) {  
 display();  
 score++;  
 TextView s = findViewById(R.id.*scoreB*);  
 s.setText("Score: " + score);  
 // check if won  
 if (win(score)) {  
 Toast.*makeText*(getApplicationContext(), "Congrats, you win!", Toast.*LENGTH\_SHORT*).show();  
 final Handler handler = new Handler();  
 handler.postDelayed(new Runnable() {  
 @Override  
 public void run() {  
 // Do something after 2s = 2000ms  
 Toast.*makeText*(getApplicationContext(), "CLick reset to start again or keep on going", Toast.*LENGTH\_SHORT*).show();  
 }  
 }, 5000);  
 }  
 }  
 //shuffles the card  
 public void shuffleClick(View view) {  
 deck.shuffle();  
 display();  
 }  
 // goes to the instructions screen  
 public void ins(View view){  
 Intent i = new Intent(this, intructions.class);  
 startActivity(i);  
 }  
 // save the current position of the grid and the score  
 public void save(View view) {  
 //tries to save the  
 try {  
 // first three files for each of the columns and last file for extra information  
 FileOutputStream out1 = openFileOutput("data1.txt", Activity.*MODE\_PRIVATE*);  
 FileOutputStream out2 = openFileOutput("data2.txt", Activity.*MODE\_PRIVATE*);  
 FileOutputStream out3 = openFileOutput("data3.txt", Activity.*MODE\_PRIVATE*);  
 FileOutputStream out4 = openFileOutput("data4.txt", Activity.*MODE\_PRIVATE*);  
  
 for(int i=0;i<tube1.size();i++) {  
 out1.write(tube1.getBallAt(i).getColour());  
 }  
 for(int i=0;i<tube2.size();i++) {  
 out2.write(tube2.getBallAt(i).getColour());  
 }  
 for(int i=0;i<tube3.size();i++) {  
 out3.write(tube3.getBallAt(i).getColour());  
 }  
 out4.write(score);  
  
 out1.flush();  
 out1.close();  
 out2.flush();  
 out2.close();  
 out3.flush();  
 out3.close();  
  
 }  
 //catches if it does work  
 catch (FileNotFoundException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 }  
 // open the sved grid and the score  
 public void open(View view) {  
 // tries to open the files  
 try{  
 FileInputStream in1 = openFileInput("data1.txt");  
 FileInputStream in2 = openFileInput("data2.txt");  
 FileInputStream in3 = openFileInput("data3.txt");  
 FileInputStream in4 = openFileInput("data4.txt");  
  
 tube1.clear();  
 tube2.clear();  
 tube3.clear();  
  
 add(tube1, in1.read());  
 add(tube1, in1.read());  
 add(tube1, in1.read());  
 add(tube1, in1.read());  
 Log.*d*("tag","size 1"+tube1.size());  
  
 add(tube2, in2.read());  
 add(tube2, in2.read());  
 add(tube2, in2.read());  
 add(tube2, in2.read());  
  
 add(tube3, in3.read());  
 add(tube3, in3.read());  
 add(tube3, in3.read());  
 add(tube3, in3.read());  
  
 score = in4.read();  
  
 redraw();  
 held = null;  
 }  
 // catches if gets an error  
 catch (FileNotFoundException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 }  
 // add adds ball after reading from a file  
 public void add(Tube t, int a){  
 // e is for empty  
 if(a==-1)  
 t.push(new Ball('e'));  
 else  
 t.push(new Ball((char)a));  
 }  
 // checks if the user has won  
 public Boolean win(int s) {  
 // score is 10 then the user win  
 if (s == 10) {  
 return true;  
 } else {  
 return false;  
 }  
 }  
 // Ai does a move also recusive  
 public void help(View view) {  
 // calles the helper method  
 helper();  
 }  
 // to make the method recusive I have to remove the View view by making a new method  
 public void helper(){  
 // generates radom moves unit it finds one that is valid  
 int r1, r2;  
 do {  
 r1 = (int) (Math.*random*() \* 3) + 1;  
 r2 = (int) (Math.*random*() \* 3) + 1;  
 } while (r1 == r2);  
 Log.*d*("mytag", r1 + ", " + r2);  
  
 if (r1 == 1 && !(tube1.isEmpty())) {  
 switch (r2) {  
 case 2:  
 if (!(tube2.isFull())&&!(tube1.isEmpty()))  
 tube2.push(tube1.pop());  
 case 3:  
 if (!(tube3.isFull())&&!(tube1.isEmpty()))  
 tube3.push(tube1.pop());  
 }  
 } else if (r1 == 2 && !(tube2.isEmpty())) {  
 switch (r2) {  
 case 1:  
 if (!(tube1.isFull())&&!(tube2.isEmpty()))  
 tube1.push(tube2.pop());  
 case 3:  
 if (!(tube3.isFull())&&!(tube2.isEmpty()))  
 tube3.push(tube2.pop());  
 }  
 } else if (r1 == 3 && !(tube3.isEmpty())) {  
 switch (r2) {  
 case 2:  
 if (!(tube2.isFull())&&!(tube1.isEmpty()))  
 tube2.push(tube3.pop());  
 case 1:  
 if (!(tube1.isFull())&&!(tube1.isEmpty()))  
 tube1.push(tube3.pop());  
 }  
 }else{  
 helper();  
 }  
 redraw();  
 }  
}

Home\_Screen.java

package com.example.dreureka;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.view.WindowManager;  
  
public class Home\_Screen extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_home\_\_screen*);  
 //hides the notification bar  
 this.getWindow().setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*, WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);  
 //hides the navigation bar  
 getWindow().getDecorView().setSystemUiVisibility(View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION*);  
 }  
 // runs method when the play button is clicked  
 public void play(View view){  
 Intent i = new Intent(this, Game\_Screen.class);  
 startActivity(i);  
 }  
 // runs method when the instructions button is clicked  
 public void instruction(View view){  
 Intent i = new Intent(this, intructions.class);  
 startActivity(i);  
 }// method when the quit button is clicked  
 public void quit(View view){  
 finish();  
 moveTaskToBack(true);  
 }  
 //hides the navigation bar  
 @Override  
 public void onWindowFocusChanged(boolean hasFocus) {  
 super.onWindowFocusChanged(hasFocus);  
 if (hasFocus) {  
 getWindow().getDecorView().setSystemUiVisibility(  
 View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* | View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* | View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* | View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* | View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* | View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY*);  
 }  
 }  
}

instructions.java

package com.example.dreureka;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.view.WindowManager;  
  
public class intructions extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_intructions*);  
 //hides the notification bar  
 this.getWindow().setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*, WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);  
 //hides the navigation bar  
 getWindow().getDecorView().setSystemUiVisibility(View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION*);  
 }  
 //hides the navigation bar  
 @Override  
 public void onWindowFocusChanged(boolean hasFocus) {  
 super.onWindowFocusChanged(hasFocus);  
 if (hasFocus) {  
 getWindow().getDecorView().setSystemUiVisibility(  
 View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* | View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* | View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* | View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* | View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* | View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY*);  
 }  
 }  
 // runs method when the back button is clicked  
 public void back(View view){  
 Intent i = new Intent(this, Home\_Screen.class);  
 startActivity(i);  
 }  
 // runs method when the play button is clicked  
 public void play(View view){  
 Intent i = new Intent(this, Game\_Screen.class);  
 startActivity(i);  
 }  
}

MainActivity.java

package com.example.dreureka;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
import android.view.View;  
import android.view.WindowManager;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 //hides the notification bar  
 this.getWindow().setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*, WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);  
 //hides the navigation bar  
 getWindow().getDecorView().setSystemUiVisibility(View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION*);  
  
 Intent i = new Intent(this, Home\_Screen.class);  
  
 //this code add a delay for the splash screen  
 final Handler handler = new Handler();  
 handler.postDelayed(new Runnable() {  
 @Override  
 public void run() {  
 // Do something after 2s = 2000ms  
 startActivity(i);  
 }  
 }, 2000);  
 }  
 //hides the navigation bar  
 @Override  
 public void onWindowFocusChanged(boolean hasFocus) {  
 super.onWindowFocusChanged(hasFocus);  
 if (hasFocus) {  
 getWindow().getDecorView().setSystemUiVisibility(  
 View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* | View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* | View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* | View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* | View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* | View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY*);  
 }  
 }  
}

package com.example.dreureka;  
  
public class Tube {  
  
 //instance variables  
 private int count;  
 private Ball data[] = new Ball[4];  
  
 //constructors  
 public Tube(){  
  
 for(int i=0;i<4;i++){  
 data[i]= new Ball();  
 }  
 count = 0;  
 }  
 public Tube(Ball ball){  
 data[count] = ball;  
 count++;  
 for(int i = count;i<4;i++){  
 data[i] = new Ball();  
 }  
 }  
  
 //accessors  
 public int size() {  
 return count;  
 }  
 //return if the tube is full  
 public boolean isFull(){  
 return(count == data.length);  
 }  
 //returns the ball at ith possition  
 public Ball getBallAt(int i){  
 return data[i];  
 }  
 // return the top ball  
 public Ball peek(){  
 return data[count-1];  
 }  
 // return if the tube is empty  
 public Boolean isEmpty(){  
 if(count==0){  
 return true;  
 }  
 return false;  
 }  
 //mutator  
 public Ball pop(){  
 count--;  
 Ball returnthis = data[count];  
 data[count] = new Ball();  
 return returnthis;  
 }  
 // pushes a new ball in the tube  
 public void push(Ball ball){  
 data[count] = ball;  
 if(ball.getColour()!='e')  
 count++;  
 }  
 // clears the tube  
 public void clear(){  
 count = 0;  
 for(int i=0;i<4;i++){  
 data[i]= new Ball();  
 }  
 }  
}