package student;

import java.awt.Color;

import java.util.ArrayList;

import java.util.Random;

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author p0073862

\*/

public class ObjectCell {

private int width;

private int height;

private int noOfSqares = 300;

private int noOftriangles = 300;

private int noOfhexagons = 300;

private Color[] colors = {Color.RED, Color.GREEN, Color.BLUE, Color.CYAN,

Color.MAGENTA, Color.ORANGE, Color.PINK, Color.YELLOW};

// private ArrayList<KSquare> squares = new ArrayList<KSquare>();

//private ArrayList<KTriangle> triangles = new ArrayList<KTriangle>();

private ArrayList<KShape> shapes = new ArrayList<KShape>();

private int nudge = 5;

public ObjectCell(int width, int height) {

this.width = width;

this.height = height;

reset();

}

public ArrayList<int[]> getShapeData() {

ArrayList<int[]> data = new ArrayList<int[]>();

// System.out.println(""+squareList.size());

for(KShape s: shapes){

data.add(s.getData());

}

return data;

//return getRedBlackSquares();

}

public void reset() {

Random numberGen = new Random();

for (int i = 0; i < noOfSqares; i++) {

int x = numberGen.nextInt(width);

int y = numberGen.nextInt(height);

Color c = colors[numberGen.nextInt(colors.length)];

int w = numberGen.nextInt(30);

shapes.add(new KSquare(x, y, w, c));

}

for (int i = 0; i < noOftriangles; i++) {

int x = numberGen.nextInt(width);

int y = numberGen.nextInt(height);

Color c = colors[numberGen.nextInt(colors.length)];

int w = numberGen.nextInt(30);

shapes.add(new KTriangle(x, y, w, c));

}

for (int i = 0; i < noOfhexagons; i++) {

int x = numberGen.nextInt(width);

int y = numberGen.nextInt(height);

Color c = colors[numberGen.nextInt(colors.length)];

int w = numberGen.nextInt(30);

shapes.add(new KHexagon(x, y, w, c));

}

}

public void but1() {

}

public void but2() {

}

public void nudge() {

for(KShape s : shapes){

s.nudge(width, height, nudge);

}

}

private ArrayList<int[]> getRedBlackSquares() {

ArrayList<int[]> result = new ArrayList<int[]>();

int squareWidth = Math.min(width, height) / 50;

int sep = squareWidth \* 3 / 2;

boolean black = true;

for (int i = 0;

i < width;

i += sep) {

for (int j = 0; j < height; j += sep) {

int[] data = new int[11];

data[0] = black ? 0 : 255;

data[1] = 0;

data[2] = 0;

data[3] = i;

data[4] = j;

data[5] = i + squareWidth;

data[6] = j;

data[7] = i + squareWidth;

data[8] = j + squareWidth;

data[9] = i;

data[10] = j + squareWidth;

result.add(data);

black = !black;

}

}

return result;

}

}

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package student;

/\*\*

\*

\* @author 09092543

\*/

import java.awt.Color;

public class KSquare extends KShape {

private int width;

public KSquare(int x, int y, int w, Color color) {

super(x,y,color);

this.width = w;

}

public int getWidth() {

return width;

}

@Override

public int[] getData() {

int[] data = new int[11];

data[0]=getColor().getRed();

data[1]=getColor().getGreen();

data[2]=getColor().getBlue();

data[3]=getX()-getWidth()/2;

data[4]=getY()-getWidth()/2;

data[5]=getX()+getWidth()/2;

data[6]=getY()-getWidth()/2;

data[7]=getX()+getWidth()/2;

data[8]=getY()+getWidth()/2;

data[9]=getX()-getWidth()/2;

data[10]=getY()+getWidth()/2;

return data;

}

}

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package student;

import java.awt.Color;

import java.util.Random;

/\*\*

\*

\* @author 09092543

\*/

public class KTriangle extends KShape {

private int d;

public KTriangle(int x, int y, int d, Color color) {

super(x,y,color);

this.d = d;

}

public int getD() {

return d;

}

@Override

public int[] getData() {

int[] data = new int[11];

int xStore = getX();

int yStore = getY();

int dStore = getD();

data[0]=getColor().getRed();

data[1]=getColor().getGreen();

data[2]=getColor().getBlue();

//top vertex

data[3]=xStore;

data[4]=yStore-dStore;

//bottom right vertex

data[5] = (int) (xStore + ((Math.cos(Math.toRadians(30))\*dStore)));

data[6] = (int) (yStore + ((Math.sin(Math.toRadians(30))\*dStore)));

//bottom left vertex

data[7] = (int) (xStore - ((Math.cos(Math.toRadians(30))\*dStore)));

data[8] = (int) (yStore + ((Math.sin(Math.toRadians(30))\*dStore)));

return data;

}

}

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package student;

import java.awt.Color;

/\*\*

\*

\* @author 09092543

\*/

public class KHexagon extends KShape {

private int s;

public KHexagon(int x, int y, int s, Color color) {

super(x,y,color);

this.s = s;

}

public int getS() {

return s;

}

@Override

public int[] getData() {

int[] data = new int[16];

double angle = 2\*Math.PI/6;

for (int i = 0; i < 6; i++) {

data[3+2\*i] = (int) (getX() + s \* Math.cos(angle \* i));

data[4+2\*i] = (int) (getY() + s \* Math.sin(angle \* i));

}

return data;

}

}

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package student;

/\*\*

\*

\* @author 09092543

\*/

import java.awt.Color;

import java.util.Random;

public class KShape {

private int x;

private int y;

private Color color;

private Random randNo;

public KShape(int x, int y,Color color) {

this.x=x;

this.y=y;

this.color = color;

}

public int getX() {

return x;

}

public int getY() {

return y;

}

public Color getColor() {

return color;

}

void nudge(int xMax,int yMax,int nudgeVal){

randNo = new Random();

if((x < xMax) && (x > 0))

x = x - (nudgeVal + randNo.nextInt((2\*nudgeVal)+1));

if((y < yMax) && (y > 0))

y = y - (nudgeVal + randNo.nextInt((2\*nudgeVal)+1));

}

public int[] getData(){

return null;

}

}