Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа № 3

По дисциплине ГИИС

Тема: «Разработка приложения «Алгоритмы растеризации»»

Выполнил:

Студент 4 курса

Группы ИИ-16

Журавлёв В.А.

Проверил:

Михно Е.В.

Брест, 2021

**Цель:** изучить алгоритмы растеризации.

**Задание:** написать программу, которая из отдельных пикселов рисует линию и окружность, используя алгоритм Брезенхейма.

**Код программы:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.util.ArrayList;

import java.util.HashSet;

public class DrawAlgorithm extends JFrame {

String mode;

HashSet<Point> points = new HashSet<>();

Point pointOffset = new Point(9,38);

public DrawAlgorithm() {

setTitle("Draw");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(1000, 1000);

setLocationRelativeTo(null);

setLayout(new BorderLayout());

JPanel submenu = createSubMenu();

add(submenu,BorderLayout.NORTH);

JPanel drawField = new JPanel();

drawField.setBackground(Color.lightGray);

add(drawField,BorderLayout.CENTER);

ArrayList<Point> list = new ArrayList<>();

drawField.addMouseListener(new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

}

@Override

public void mousePressed(MouseEvent e) {

}

@Override

public void mouseReleased(MouseEvent e) {

list.add(new Point(e.getX()+ pointOffset.x,e.getY() + pointOffset.y));

if(list.size() == 2){

if(mode.equals("line"))

drawBrasenhamLine(list.get(0),list.get(1));

else

drawCircle(list.get(0),list.get(1));

list.clear();

}

}

@Override

public void mouseEntered(MouseEvent e) {

}

@Override

public void mouseExited(MouseEvent e) {

}

});

setResizable(true);

setVisible(true);

pointOffset = new Point(pointOffset.x, pointOffset.y + submenu.getHeight());

}

@Override

public void paint(Graphics g) {

super.paint(g);

Graphics2D graphics2D = (Graphics2D)g;

graphics2D.setColor(Color.black);

for (Point p: points)

graphics2D.drawLine(p.x,p.y,p.x,p.y);

}

private JPanel createSubMenu(){

JPanel panel = new JPanel();

panel.setLayout(new FlowLayout(FlowLayout.LEFT));

JButton clear = new JButton("Clear field");

clear.addActionListener(e -> {

points.clear();

repaint();

});

JRadioButton line = new JRadioButton("Line");

mode = "line";

line.setEnabled(false);

line.setSelected(true);

JRadioButton circle = new JRadioButton("Circle");

line.addActionListener(e -> {

mode = "line";

circle.setSelected(false);

circle.setEnabled(true);

line.setSelected(true);

line.setEnabled(false);

});

circle.addActionListener(e -> {

mode = "circle";

line.setSelected(false);

line.setEnabled(true);

circle.setSelected(true);

circle.setEnabled(false);

});

panel.add(clear);

panel.add(line);

panel.add(circle);

return panel;

}

public void drawBrasenhamLine(Point start, Point end) {

int x, y, dx, dy, incx, incy, pdx, pdy, es, el, err;

dx = start.x - end.x;

dy = start.y - end.y;

incx = (dx > 0) ? 1 : (dx < 0) ? -1 : 0;

incy = (dy > 0) ? 1 : (dy < 0) ? -1 : 0;

if (dx < 0)

dx = Math.abs(dx);

if (dy < 0)

dy = Math.abs(dy);

if (dx > dy) {

pdx = incx;

pdy = 0;

es = dy;

el = dx;

}

else{

pdx = 0;

pdy = incy;

es = dx;

el = dy;

}

x = start.x;

y = start.y;

err = el/2;

points.add(new Point(x,y));

for (int t = 0; t < el; t++)//идём по всем точкам, начиная со второй и до последней

{

err -= es;

if (err < 0)

{

err += el;

x -= incx;//сдвинуть прямую (сместить вверх или вниз, если цикл проходит по иксам)

y -= incy;//или сместить влево-вправо, если цикл проходит по y

}

else

{

x -= pdx;//продолжить тянуть прямую дальше, т.е. сдвинуть влево или вправо, если

y -= pdy;//цикл идёт по иксу; сдвинуть вверх или вниз, если по y

}

points.add(new Point(x,y));

}

repaint();

}

private void drawCircle(Point center, Point pointRadius){

Integer radius = Math.round(new Float(Math.sqrt(Math.pow(center.x - pointRadius.x, 2) + Math.pow(center.y - pointRadius.y, 2))));

int x = 0;

int y = radius;

int delta = 1 - 2 \* radius;

int error;

while(y >= 0) {

if(center.x + x > pointOffset.x && center.y+y > pointOffset.y)

points.add(new Point(center.x + x,center.y + y));

if(center.x + x > pointOffset.x && center.y-y > pointOffset.y)

points.add(new Point(center.x + x,center.y - y));

if(center.x - x > pointOffset.x && center.y+y > pointOffset.y)

points.add(new Point(center.x - x,center.y + y));

if(center.x - x > pointOffset.x && center.y-y > pointOffset.y)

points.add(new Point(center.x - x,center.y - y));

error = 2 \* (delta + y) - 1;

if(delta < 0 && error <= 0) {

++x;

delta += 2 \* x + 1;

continue;

}

error = 2 \* (delta - x) - 1;

if(delta > 0 && error > 0) {

--y;

delta += 1 - 2 \* y;

continue;

}

++x;

delta += 2 \* (x - y);

--y;

}

repaint();

}

public static void main(String[] args) {

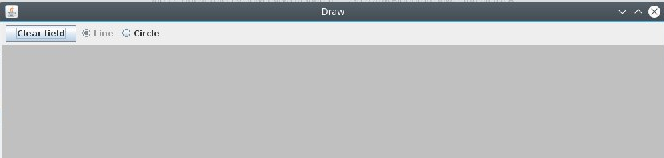
new DrawAlgorithm();

}

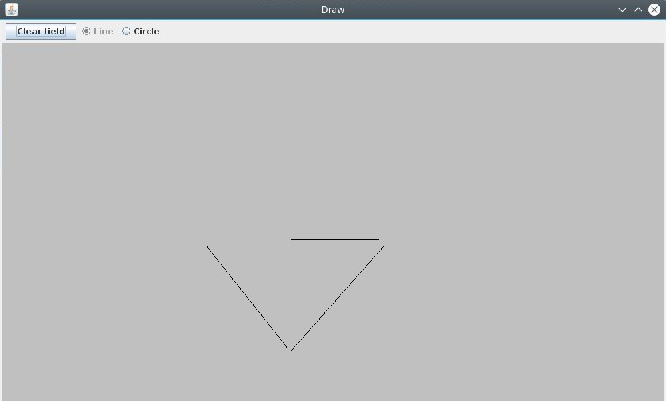
}

**Результат работы программы:**

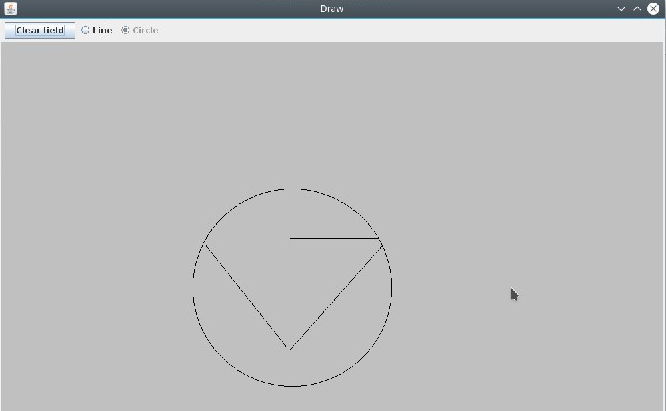
1). При запуске появляется окно:

****

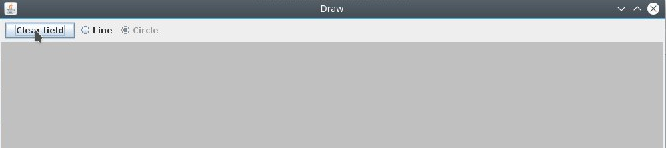
2). При выборе линии дважды кликаем на область рисования, чтобы нарисовать прямую линию.

****

3). При выборе опции круг. При первом клике выбирается центр окружности. При втором клике указывается радиус.

****

4). При щелчке на кнопку очистить поле, очищается область рисования.

****

**Вывод:** написал программу, которая из отдельных пикселов рисует линию и окружность, используя алгоритм Брезенхейма.