**Федеральное агентство связи**

**Ордена трудового Красного Знамени федеральное государственное бюджетное**

**образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математическая кибернетика и информационные технологии

Отчет по лабораторной работе №6

по дисциплине «Объектно-ориентированное программирование»

Выполнил: студент группы БВТ2002

ФИО: Мосева Алеся Сергеевна

Проверил:Полянцева Ксения Андреевна

Москва, 2020

**Цель работы**: необходимо реализовать возможность рисования фрактала с несколькими фоновыми потоками

**Код программы:**

**Измененный код класса FractelExplorer**

public class FractalExplorer {

private int size;

private JImageDisplay jDisplay;

private FractalGenerator fractal;

private Rectangle2D.Double range;

JComboBox comboBox;

JButton resetButton;

JButton saveButton;

private int rowsLost;

public FractalExplorer (int display\_size) {

size = display\_size;

range = new Rectangle2D.Double();

fractal = new Mandelbrot();

fractal.getInitialRange(range);

jDisplay = new JImageDisplay(display\_size, display\_size);

}

public void createAndShowGUI () {

jDisplay.setLayout(new BorderLayout());

JFrame frame = new JFrame();

frame.add(jDisplay, BorderLayout.CENTER);

resetButton = new JButton("Reset");

ResetButtonHandler clearAction = new ResetButtonHandler();

resetButton.addActionListener(clearAction);

MyMouseListener mouse = new MyMouseListener();

jDisplay.addMouseListener(mouse);

frame.setDefaultCloseOperation(frame.EXIT\_ON\_CLOSE);

String[] items = {"Mandelbrot", "Tricorn", "BurningShip"};

comboBox = new JComboBox(items);

JLabel label = new JLabel("Fractal:");

JPanel panelBox = new JPanel();

panelBox.add(label);

panelBox.add(comboBox);

frame.add(panelBox, BorderLayout.NORTH);

ChooseButtonHandler chooseAction = new ChooseButtonHandler();

comboBox.addActionListener(chooseAction);

saveButton = new JButton("Save Image");

SaveImageButton saveAction = new SaveImageButton();

saveButton.addActionListener(saveAction);

JPanel panelButtons = new JPanel();

panelButtons.add(resetButton);

panelButtons.add(saveButton);

frame.add(panelButtons, BorderLayout.SOUTH);

frame.pack();

frame.setVisible(true);

frame.setResizable(false);

}

private void drawFractal () {

enableIO(false);

rowsLost = size;

for (int y = 0; y < size; y++){

FractalWorker frac = new FractalWorker(y);

frac.execute();

}

}

private void enableIO(boolean val){

comboBox.setEnabled(val);

resetButton.setEnabled(val);

saveButton.setEnabled(val);

}

public class ResetButtonHandler implements ActionListener {

@Override

public void actionPerformed (ActionEvent e) {

fractal.getInitialRange(range);

drawFractal();

}

}

public class ChooseButtonHandler implements ActionListener{

@Override

public void actionPerformed(ActionEvent e) {

JComboBox combo = (JComboBox)e.getSource();

String name = (String) combo.getSelectedItem();

if (name == "Mandelbrot"){

fractal = new Mandelbrot();

}

if (name == "Tricorn") {

fractal = new Tricorn();

}

if (name == "BurningShip") {

fractal = new BurningShip();

}

fractal.getInitialRange(range);

drawFractal();

}

}

public class SaveImageButton implements ActionListener{

@Override

public void actionPerformed(ActionEvent e) {

JFileChooser chooser = new JFileChooser();

FileFilter filter = new FileNameExtensionFilter("PNG Images", "png");

chooser.setFileFilter(filter);

chooser.setAcceptAllFileFilterUsed(false);

int result = chooser.showSaveDialog(jDisplay);

if (result == JFileChooser.APPROVE\_OPTION) {

File dir = chooser.getSelectedFile();

String dir\_string = dir.toString();

try{

BufferedImage image = jDisplay.getImage();

ImageIO.write(image, "png", dir);

}

catch(Exception exception){

JOptionPane.showMessageDialog(chooser, exception.getMessage(),"Can not save image", JOptionPane.ERROR\_MESSAGE);

}

}

}

}

private class MyMouseListener extends MouseAdapter {

@Override

public void mouseClicked(MouseEvent e) {

if (rowsLost!=0) {

return;

}

int x = e.getX();

double xCoord = fractal.getCoord(range.x, range.x+range.width, size,x);

int y = e.getY();

double yCoord = fractal.getCoord(range.y, range.y+range.height, size,y);

fractal.recenterAndZoomRange(range, xCoord, yCoord, 0.5);

drawFractal();

}

}

private class FractalWorker extends SwingWorker<Object,Object>{

private int y;

private int[] valuesRGB;

public FractalWorker (int y){

this.y = y;

}

@Override

protected Object doInBackground() throws Exception {

valuesRGB = new int[size];

for (int x = 0; x < valuesRGB.length; x ++) {

double xCoord = fractal.getCoord(range.x,range.x + range.width, size, x);

double yCoord = fractal.getCoord(range.y, range.y + range.height, size, y);

int iterations = fractal.numIterations(xCoord,yCoord);

int rgbColor;

if (iterations == -1)

rgbColor = 0;

else {

float hue = 0.7f + (float) iterations / 200f;

rgbColor = Color.HSBtoRGB(hue, 1f, 1f);

}

valuesRGB[x] = rgbColor;

}

return null;

}

@Override

protected void done() {

for (int i = 0; i < valuesRGB.length; i ++) {

jDisplay.drawPixel(i,y,valuesRGB[i]);

}

jDisplay.repaint(0,0,y,size,1);

rowsLost--;

if (rowsLost == 0)

enableIO(true);

}

}

public static void main (String[] args) {

FractalExplorer display = new FractalExplorer(800);

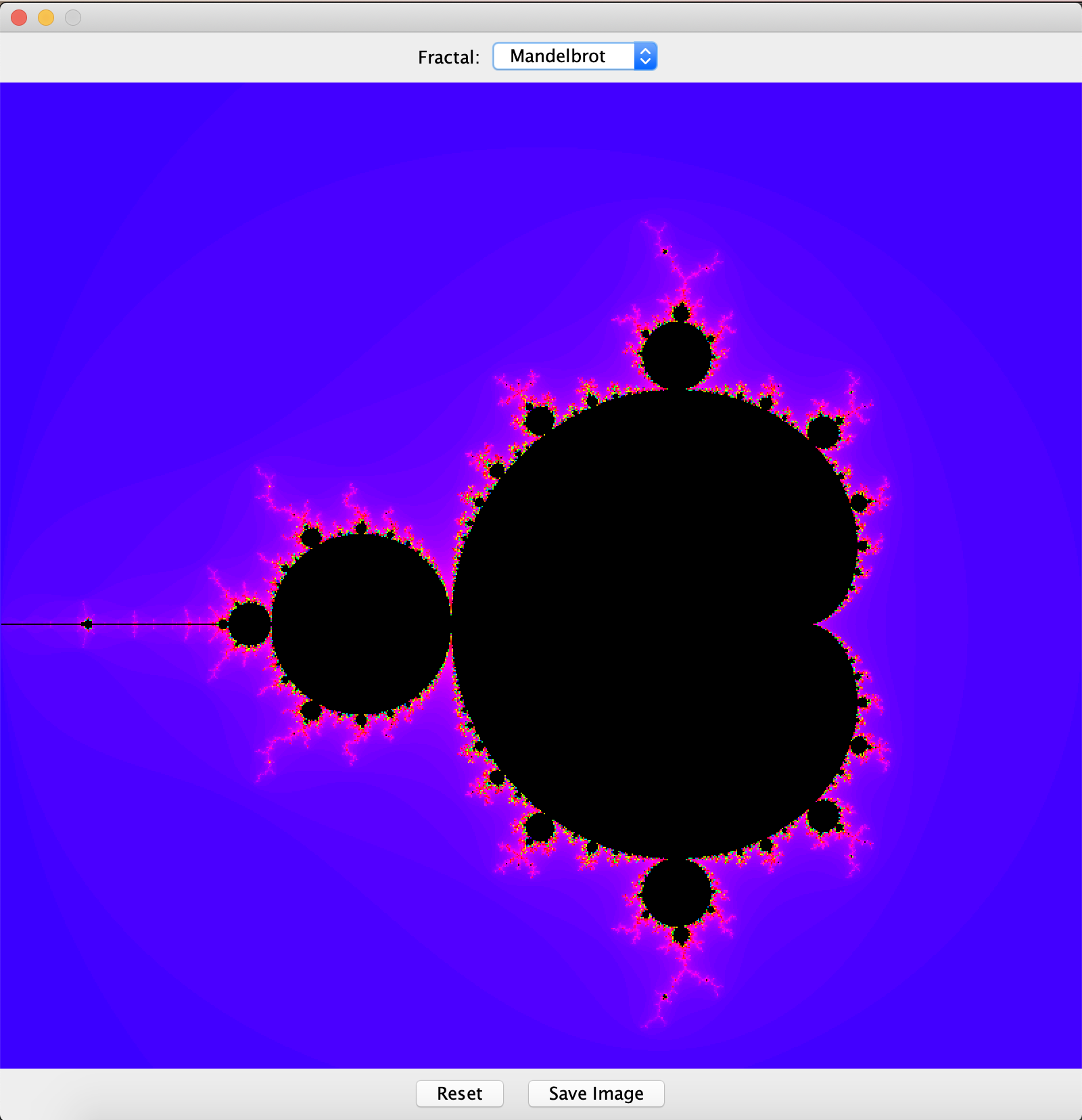
display.createAndShowGUI();

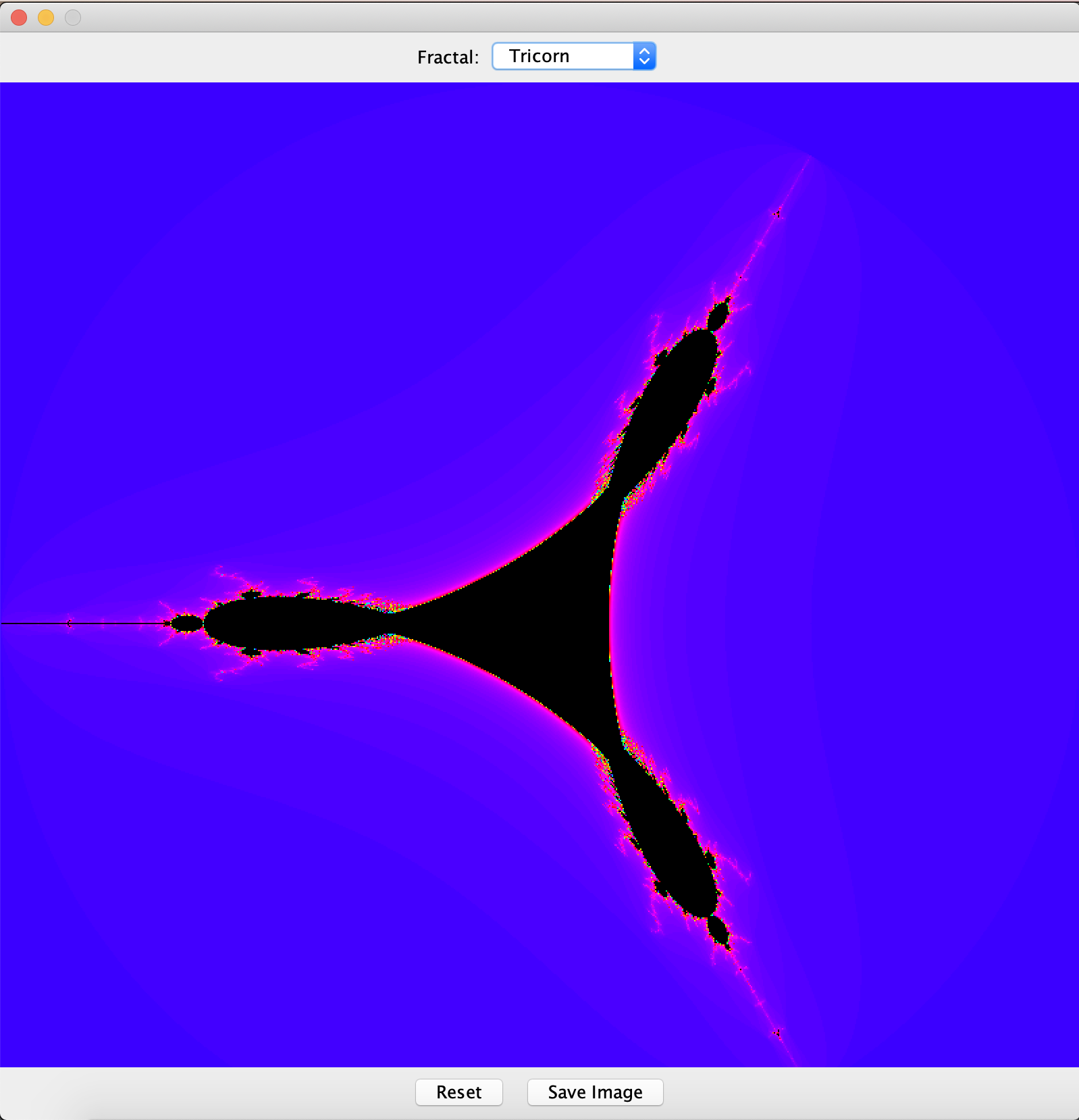
display.drawFractal();

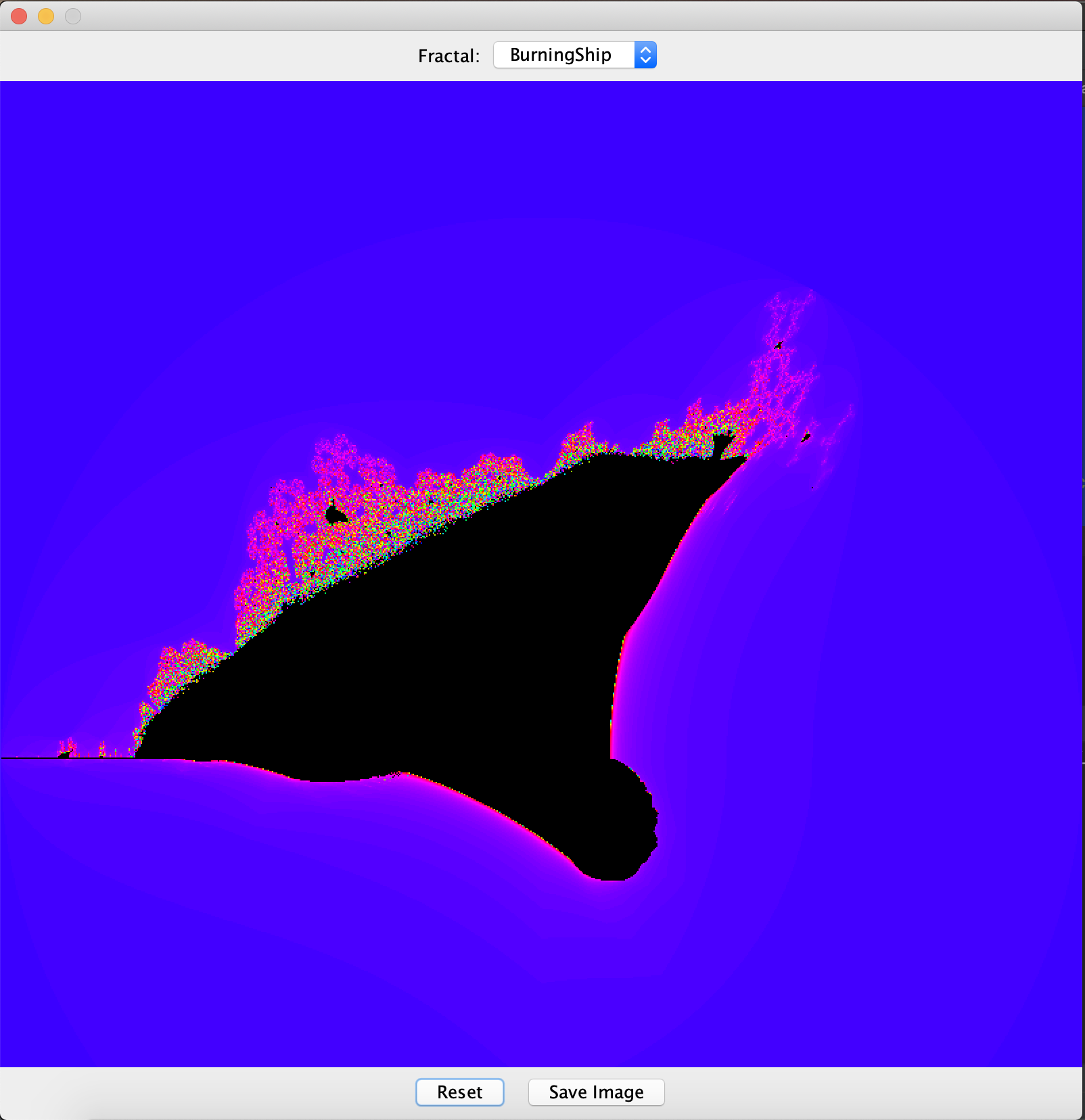
}

}

**Скриншоты выполнения:**

****

****

****

**Заключение:**

Таким образом, мы реализовали рисование фракталов с несколькими фоновыми потоками.