**CENTRO UNIVERSITÁRIO BELAS ARTES**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS EAD**

**PROGRAMAÇÃO ORIENTADA A OBJETOS**

**FELIPE SCHAITEL RA: 24112424**

**ATIVIDADE 02 - MÓDULO 2**

**SÃO PAULO**

**2024**

**1. ENUNCIADO**

Neste trabalho, você deverá construir um software utilizando Orientação a Objetos e interface gráfica na linguagem Java que permita um usuário desenhar com formas geométricas bidimensionais conhecidas, como Retângulo, Círculo, Triângulo, ou de maneira livre utilizando o mouse. A interface da aplicação deve contemplar os elementos da figura.

Retângulos de 1 a 5 são botões e região 6 é onde o usuário pode desenhar. Funcionalidade dos botões são:

1-) Abre uma caixa de diálogo para salvar a imagem que foi desenhada até o momento.

2-) Abre uma caixa de diálogo que permite carregar uma imagem na região 6.

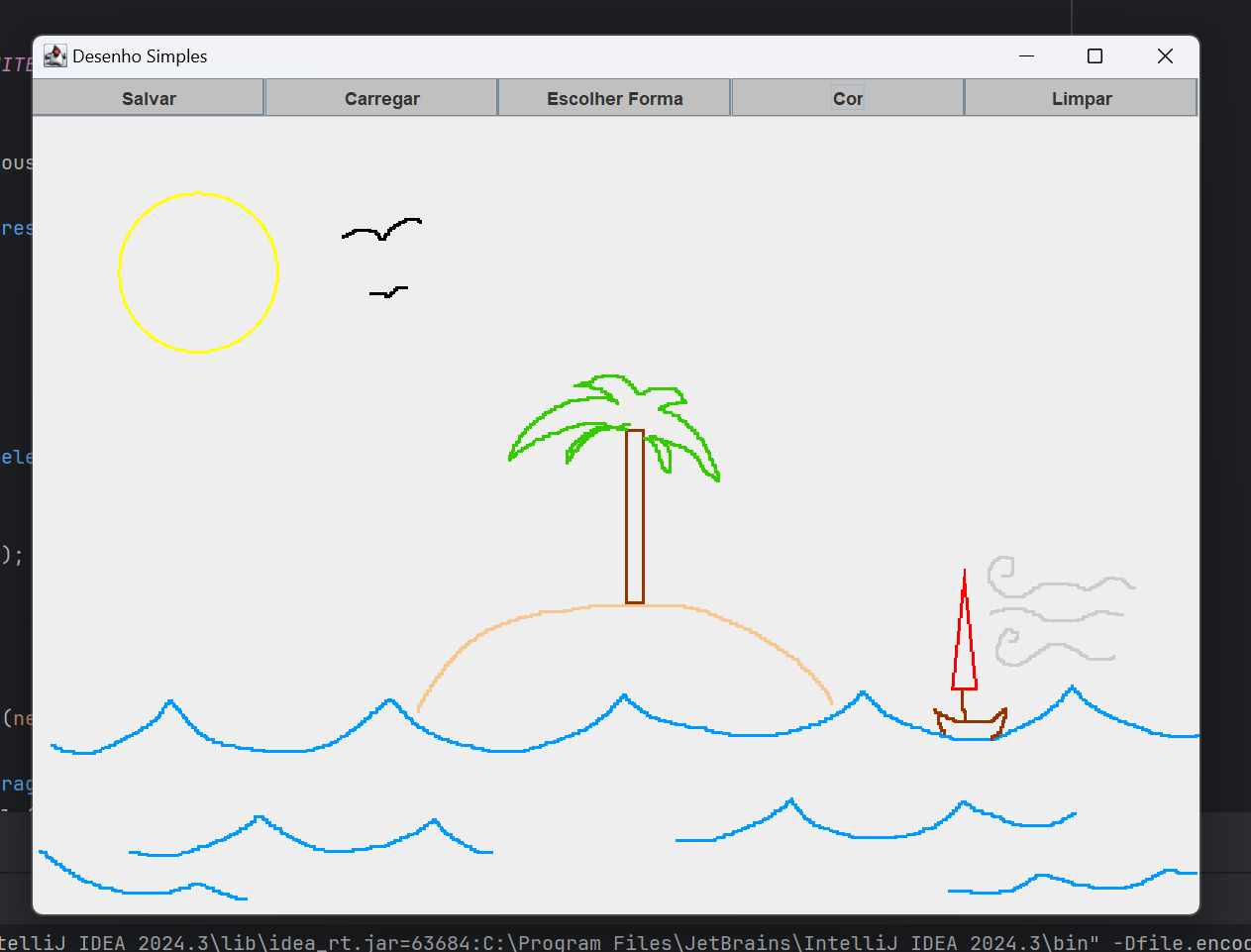
3-) Abre uma caixa de diálogo para selecionar uma forma básica entre: quadrado, retângulo, círculo, elipse e triângulo.

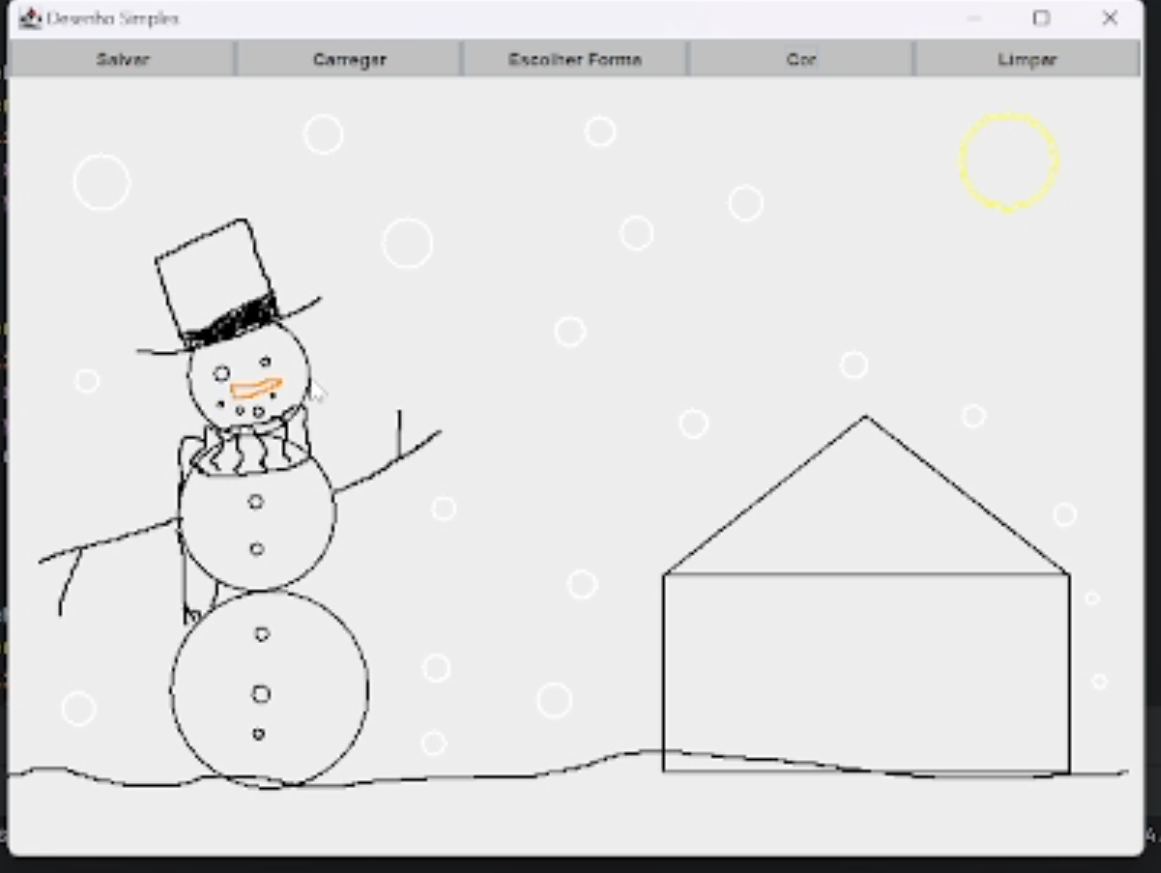
4-) Abre um diálogo que permite escolher uma cor para o pincel.

5-) Apagar tudo o que estiver desenhado na região 6.

Sobre o botão 2, Java possui classes para ler e escrever nos formatos mais comuns de imagens, como JPG, GIF, PNG, TIF, mas você pode utilizar um único formato. Formato mais simples de trabalhar é o BMP.

**2. RESULTADOS**





**3. CÓDIGO**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import java.awt.image.RenderedImage;

import javax.imageio.ImageIO;

class Desenhando extends JFrame {

private Painel painel;

public Desenhando() {

setTitle("Desenho Simples");

setSize(800, 600);

setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

setLayout(new BorderLayout());

JPanel painelBotoes = new JPanel();

painelBotoes.setLayout(new GridLayout(1, 5));

JButton salvar = new JButton("Salvar");

JButton carregar = new JButton("Carregar");

JButton forma = new JButton("Escolher Forma");

JButton cor = new JButton("Cor");

JButton limpar = new JButton("Limpar");

salvar.setBackground(Color.*LIGHT\_GRAY*);

carregar.setBackground(Color.*LIGHT\_GRAY*);

forma.setBackground(Color.*LIGHT\_GRAY*);

cor.setBackground(Color.*LIGHT\_GRAY*);

limpar.setBackground(Color.*LIGHT\_GRAY*);

painelBotoes.add(salvar);

painelBotoes.add(carregar);

painelBotoes.add(forma);

painelBotoes.add(cor);

painelBotoes.add(limpar);

painel = new Painel();

add(painelBotoes, BorderLayout.*NORTH*);

add(painel, BorderLayout.*CENTER*);

salvar.addActionListener(e -> painel.salvar());

carregar.addActionListener(e -> painel.carregar());

forma.addActionListener(e -> painel.mudarForma());

cor.addActionListener(e -> painel.mudarCor());

limpar.addActionListener(e -> painel.limpar());

setVisible(true);

}

public static void main(String[] args) {

new Desenhando();

}

}

class Painel extends JPanel {

private Image img;

private Graphics2D g2d;

private Color cor = Color.*BLACK*;

private String forma = "Livre";

private int x1, y1, x2, y2;

public Painel() {

setBackground(Color.*WHITE*);

addMouseListener(new MouseAdapter() {

@Override

public void mousePressed(MouseEvent e) {

x1 = e.getX();

y1 = e.getY();

}

@Override

public void mouseReleased(MouseEvent e) {

x2 = e.getX();

y2 = e.getY();

desenharForma();

}

});

addMouseMotionListener(new MouseAdapter() {

@Override

public void mouseDragged(MouseEvent e) {

if (forma.equals("Livre")) {

x2 = e.getX();

y2 = e.getY();

desenharLinhaLivre();

}

}

});

}

private void desenharForma() {

if (g2d == null) {

preparar();

}

switch (forma) {

case "Retângulo":

g2d.drawRect(Math.*min*(x1, x2), Math.*min*(y1, y2),

Math.*abs*(x2 - x1), Math.*abs*(y2 - y1));

break;

case "Círculo":

int d = Math.*max*(Math.*abs*(x2 - x1), Math.*abs*(y2 - y1));

g2d.drawOval(Math.*min*(x1, x2), Math.*min*(y1, y2), d, d);

break;

case "Triângulo":

int[] xPoints = {x1, (x1 + x2) / 2, x2};

int[] yPoints = {y2, y1, y2};

g2d.drawPolygon(xPoints, yPoints, 3);

break;

}

repaint();

}

private void desenharLinhaLivre() {

if (g2d == null) {

preparar();

}

g2d.drawLine(x1, y1, x2, y2);

x1 = x2;

y1 = y2;

repaint();

}

private void preparar() {

img = createImage(getWidth(), getHeight());

g2d = (Graphics2D) img.getGraphics();

g2d.setColor(cor);

g2d.setStroke(new BasicStroke(2));

}

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

if (img != null) {

g.drawImage(img, 0, 0, null);

}

}

public void salvar() {

File file = new File("desenho.bmp");

try {

ImageIO.*write*((RenderedImage) img, "bmp", file);

} catch (Exception e) {

e.printStackTrace();

}

}

public void carregar() {

File file = new File("desenho.bmp");

try {

img = ImageIO.*read*(file);

g2d = (Graphics2D) img.getGraphics();

repaint();

} catch (Exception e) {

e.printStackTrace();

}

}

public void mudarForma() {

String[] opcoes = {"Livre", "Retângulo", "Círculo", "Triângulo"};

forma = (String) JOptionPane.*showInputDialog*(this, "Escolha uma forma:",

"Forma", JOptionPane.*QUESTION\_MESSAGE*, null, opcoes, forma);

}

public void mudarCor() {

cor = JColorChooser.*showDialog*(this, "Escolha uma cor", cor);

if (g2d != null) {

g2d.setColor(cor);

}

}

public void limpar() {

img = null;

g2d = null;

repaint();

}

}

**Referências bibliográficas:**

**ORACLE CORPORATION.** *Java Platform, Standard Edition 8 Documentation*. Disponível em:<https://docs.oracle.com/javase/8/docs/api/>. Acesso em: 26 nov. 2024.

**ORACLE CORPORATION.** *The Java Tutorials*. Disponível em:<https://docs.oracle.com/javase/tutorial/>. Acesso em: 26 nov. 2024.

DEITEL, Paul; DEITEL, Harvey. *Java: How to Program*. 11. ed. Boston: Pearson, 2017.

ALURA. *Como criar interface gráfica com Swing no Java*. Disponível em:<https://www.alura.com.br/artigos/como-criar-interface-grafica-swing-java>. Acesso em: 26 nov. 2024.

TUTORIALSPOINT. *Swing Tutorial*. Disponível em:<https://www.tutorialspoint.com/swing/index.htm>. Acesso em: 26 nov. 2024.

TUTORIALSPOINT. *Swing Examples*. Disponível em:<https://www.tutorialspoint.com/swingexamples/index.htm>. Acesso em: 26 nov. 2024.

GUJ. Criando um mini paint em Java. Disponível em:<https://www.guj.com.br/t/criando-um-mini-paint-em-java/393297>. Acesso em: 27 nov. 2024.

PUCRS. *Interface gráfica em Java (GUI - Parte 2)*. Disponível em:<https://www.inf.pucrs.br/flash/lapro2/aula_gui2/aula_gui2.html>. Acesso em: 27 nov. 2024.

JAVADIVERTIDO. 15 – Programação Gráfica em Java: Desenhos com a classe Graphics. Disponível em:<https://javadivertido.blogspot.com/2015/04/15-programacao-grafica-em-java-desenhos_17.html>. Acesso em: 27 nov. 2024.

YOUTUBE. *[Java] How to make a Swing Paint and Drawing application ?*. Disponível em:<https://www.youtube.com/watch?v=OOb1eil4PCo>. Acesso em: 29 nov. 2024.

YOUTUBE. *Aula de Java 082 - Programa de desenhar, mini Paint*. Disponível em:<https://www.youtube.com/watch?v=CxU1TnD-xH8>. Acesso em: 29 nov. 2024.