import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import java.util.\*;

import javax.swing.\*;

import javax.swing.event.\*;

import javax.swing.filechooser.\*;

public class TextEditor extends JFrame implements ActionListener{

JTextArea textArea;

JScrollPane scrollPane;

JLabel fontLabel;

JSpinner fontSizeSpinner;

JButton fontColorButton;

JComboBox fontBox;

JMenuBar menuBar;

JMenu fileMenu;

JMenuItem openItem;

JMenuItem saveItem;

JMenuItem exitItem;

TextEditor(){

getContentPane().setBackground(Color.ORANGE);

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

this.setTitle("Bro text Editor");

this.setSize(500, 500);

getContentPane().setLayout(new FlowLayout());

this.setLocationRelativeTo(null);

textArea = new JTextArea();

textArea.setLineWrap(true);

textArea.setWrapStyleWord(true);

textArea.setFont(new Font("Arial",Font.PLAIN,20));

scrollPane = new JScrollPane(textArea);

scrollPane.setPreferredSize(new Dimension(450,450));

scrollPane.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL\_SCROLLBAR\_ALWAYS);

fontLabel = new JLabel("Font: ");

fontSizeSpinner = new JSpinner();

fontSizeSpinner.setPreferredSize(new Dimension(50,25));

fontSizeSpinner.setValue(20);

fontSizeSpinner.addChangeListener(new ChangeListener() {

@Override

public void stateChanged(ChangeEvent e) {

textArea.setFont(new Font(textArea.getFont().getFamily(),Font.PLAIN,(int) fontSizeSpinner.getValue()));

}

});

fontColorButton = new JButton("Color");

fontColorButton.setForeground(Color.WHITE);

fontColorButton.setBackground(Color.RED);

fontColorButton.addActionListener(this);

String[] fonts = GraphicsEnvironment.getLocalGraphicsEnvironment().getAvailableFontFamilyNames();

fontBox = new JComboBox(fonts);

fontBox.addActionListener(this);

fontBox.setSelectedItem("Arial");

// ------ menubar ------

menuBar = new JMenuBar();

fileMenu = new JMenu("File");

openItem = new JMenuItem("Open");

saveItem = new JMenuItem("Save");

exitItem = new JMenuItem("Exit");

openItem.addActionListener(this);

saveItem.addActionListener(this);

exitItem.addActionListener(this);

fileMenu.add(openItem);

fileMenu.add(saveItem);

fileMenu.add(exitItem);

menuBar.add(fileMenu);

// ------ /menubar ------

this.setJMenuBar(menuBar);

getContentPane().add(fontLabel);

getContentPane().add(fontSizeSpinner);

getContentPane().add(fontColorButton);

getContentPane().add(fontBox);

getContentPane().add(scrollPane);

this.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource()==fontColorButton) {

JColorChooser colorChooser = new JColorChooser();

Color color = colorChooser.showDialog(null, "Choose a color", Color.black);

textArea.setForeground(color);

}

if(e.getSource()==fontBox) {

textArea.setFont(new Font((String)fontBox.getSelectedItem(),Font.PLAIN,textArea.getFont().getSize()));

}

if(e.getSource()==openItem) {

JFileChooser fileChooser = new JFileChooser();

fileChooser.setCurrentDirectory(new File("."));

FileNameExtensionFilter filter = new FileNameExtensionFilter("Text files", "txt");

fileChooser.setFileFilter(filter);

int response = fileChooser.showOpenDialog(null);

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

File file = new File(fileChooser.getSelectedFile().getAbsolutePath());

Scanner fileIn = null;

try {

fileIn = new Scanner(file);

if(file.isFile()) {

while(fileIn.hasNextLine()) {

String line = fileIn.nextLine()+"\n";

textArea.append(line);

}

}

} catch (FileNotFoundException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

finally {

fileIn.close();

}

}

}

if(e.getSource()==saveItem) {

JFileChooser fileChooser = new JFileChooser();

fileChooser.setCurrentDirectory(new File("."));

int response = fileChooser.showSaveDialog(null);

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

File file;

PrintWriter fileOut = null;

file = new File(fileChooser.getSelectedFile().getAbsolutePath());

try {

fileOut = new PrintWriter(file);

fileOut.println(textArea.getText());

}

catch (FileNotFoundException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

finally {

fileOut.close();

}

}

}

if(e.getSource()==exitItem) {

System.exit(0);

}

}

}