package battleship;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.\*;

import BreezySwing.\*;

import java.io.IOException;

import javax.sound.sampled.\*;

import java.io.\*;

public class OpenScreen extends GBFrame implements ActionListener

{

// ImageIcon introLogo = new ImageIcon(getClass().getClassLoader().getResource("battleship.png"));

// ImageIcon play = new ImageIcon(getClass().getClassLoader().getResource("play.jpeg"));

JButton logo = new JButton();

IntegerField X;

IntegerField Y;

JLabel middle1 = new JLabel();

JLabel XSide1 = new JLabel();

JLabel YSide1 = new JLabel();

IntegerField Size;

JButton addShip;

JButton quit;

JButton easy;

JButton medium;

JButton hard;

JLabel size;

JButton Finish;

// int boats = 210;

boolean ready=false;

int diff = -1;

int[][] spots = new int[8][8];

public Ship[] user = new Ship[4];

public ArrayList<String> locations = new ArrayList<String>();

String direction = "";

JButton Reset;

int count = 0;

JComboBox comboDirections;

MyBoard myBoard = new MyBoard(this);

OpponentBoard oppBoard = new OpponentBoard(this);

public OpenScreen()

{

logo = addButton("",1,2,1,2);

myBoard.setVisible(true);

oppBoard.setVisible(false);

// logo.setIcon(introLogo);

XSide1 = addLabel("Enter row:",3,1,1,1);

YSide1 = addLabel("Enter column:",3,3,1,1);

X = addIntegerField(0,3,2,1,1);

Y = addIntegerField(0,3,4,1,1);

X.setText("1");

Y.setText("1");

size = addLabel("Size",4,1,1,1);

Size = addIntegerField(0,4,2,1,1);

Size.setText("2");

addShip = addButton("Add Ship",5,1,1,1);

quit = addButton("Quit",5,2,1,1);

Reset = addButton("Reset",5,3,1,1);

comboDirections = addComboBox(4,3,1,1);

comboDirections.addItem("Right");

comboDirections.addItem("Up");

comboDirections.addItem("Down");

comboDirections.addItem("Left");

direction = "Up";

Finish = addButton("",6,2,1,1);

// Finish.setIcon(play);

Finish.setEnabled(false);

easy = addButton("Easy",4,4,1,1);

medium = addButton("Medium",5,4,1,1);

hard = addButton("Hard ",6,4,1,1);

Size.setEditable(false);

}

public void buttonClicked(JButton obj)

{

if(obj == easy){

diff = 0;

easy.setVisible(false);

medium.setVisible(false);

hard.setVisible(false);

ready=true;

setFinish();

}

else if(obj == medium){

diff = 1;

easy.setVisible(false);

medium.setVisible(false);

hard.setVisible(false);

ready=true;

setFinish();

}

else if(obj == hard){

diff = 2;

easy.setVisible(false);

medium.setVisible(false);

hard.setVisible(false);

ready=true;

setFinish();

}

else if(obj == Finish)

{

Master.passShipArray(user,diff);

addShip.setVisible(false);

oppBoard.setVisible(true);

dispose();

}

else if(obj == quit)

{

System.exit(1);

}

else if(obj == Reset)

{

// boats=210;

count = 0;

locations = new ArrayList<String>();

user = new Ship[4];

X.setText("1");

Y.setText("1");

Size.setText("2");

addShip.setVisible(true);

Finish.setEnabled(false);

spots=new int[8][8];

ready=false;

MyBoard.reset();

easy.setVisible(true);

medium.setVisible(true);

hard.setVisible(true);

}

else if(obj == addShip)

{

int way = -1;

direction = String.valueOf(comboDirections.getSelectedItem());

// System.out.println(direction);

if(direction.equals("Up"))

way = 0;

if(direction.equals("Down"))

way = 1;

if(direction.equals("Left"))

way = 2;

if(direction.equals("Right"))

way = 3;

if(validate(X.getNumber() - 1,Y.getNumber() - 1,Size.getNumber(),way))

{

user[count] = new Ship(X.getNumber() - 1,Y.getNumber() - 1,direction,Size.getNumber());

count++;

myBoard.setShip(X.getNumber() - 1,Y.getNumber() - 1, Size.getNumber(),String.valueOf(comboDirections.getSelectedItem()));

if (Size.getNumber()==5) {

addShip.setVisible(false);

} else {

Size.setText(Size.getNumber()+1+"");

}

}

else

this.messageBox("ERROR: Invalid Ship");

setFinish();

}

}

@Override

public void actionPerformed(ActionEvent e)

{

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

String direction1 = (String)cb.getSelectedItem();

direction = direction1;

}

public boolean validate(int y1, int x1, int size1, int way)

{

if (size1<2 || size1>5 || x1<0 || x1>7 || y1<0 || y1>7) {

return false;

} if (way==0 && y1-size1+1<0) { //up

return false;

} if (way==1 && y1+size1-1>7) { //down

return false;

} if (way==2 && x1-size1+1<0) { //left

return false;

} if (way==3 && x1+size1-1>7) { //right

return false;

}

for (int i=0; i<size1; i++) {

if (way==0) { //up

if (spots[y1-i][x1]!=0) {

return false;

}

} else if (way==1) { //down

if (spots[y1+i][x1]!=0) {

return false;

}

} else if (way==2) { //left

if (spots[y1][x1-i]!=0) {

return false;

}

} else { //right

if (spots[y1][x1+i]!=0) {

return false;

}

}

}

// if (size1==2) {

// if (boats%2!=0) {

// return false;

// }

// boats/=2;

// } else if (size1==3) {

// if (boats%3!=0) {

// return false;

// }

// boats/=3;

// } else if (size1==4) {

// if (boats%5!=0) {

// return false;

// }

// boats/=5;

// } else {

// if (boats%7!=0) {

// return false;

// }

// boats/=7;

// }

for (int i=0; i<size1; i++) {

if (way==0) { //up

spots[y1-i][x1]=1;

} else if (way==1) { //down

spots[y1+i][x1]=1;

} else if (way==2) { //left

spots[y1][x1-i]=1;

} else { //right

spots[y1][x1+i]=1;

}

}

return true;

} private void setFinish() {

if(count >= 4 && ready)

Finish.setEnabled(true);

}

public static void main(String [] args)

{

OpenScreen theGUI = new OpenScreen();

theGUI.setSize(500,300);

theGUI.setVisible(true);

music();

}

public static void music(){

try {

// Open an audio input stream.

File soundFile = new File("/Users/marcos\_a/Desktop/EclipseProjects/Battleship/ImagesLowercase/FILENAME.wav"); //you could also get the sound file with an URL

AudioInputStream audioIn = AudioSystem.getAudioInputStream(soundFile);

// Get a sound clip resource.

Clip clip = AudioSystem.getClip();

// Open audio clip and load samples from the audio input stream.

clip.open(audioIn);

clip.start();

} catch (UnsupportedAudioFileException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} catch (LineUnavailableException e) {

e.printStackTrace();

}

} public static void test() {

}

}