package wuziqi;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.Toolkit;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import javax.imageio.ImageIO;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

@SuppressWarnings("serial")

public class wuziqi extends JFrame implements MouseListener,Runnable {

int width = Toolkit.getDefaultToolkit().getScreenSize().width;

int height = Toolkit.getDefaultToolkit().getScreenSize().height;

BufferedImage bgimage = null;

int x = 0;

int y = 0;

int[][] allChess = new int[19][19];

boolean isBlack = true;

boolean canPlay = true;

String message="黑方先行";

public wuziqi() {

this.setTitle("五子棋");

this.setSize(390, 450);

this.setLocation((width - 500) / 2, (height - 500) / 2);

this.setResizable(false);

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

this.setVisible(true);

this.addMouseListener(this);

this.repaint();

try {

bgimage = ImageIO.read(new File("/home/os/workspace/1.jpg"));

} catch (IOException e) {

e.printStackTrace();

}

}

public void paint(Graphics g) {

BufferedImage bi=new BufferedImage(500, 500,BufferedImage.TYPE\_INT\_ARGB );

Graphics g2=bi.createGraphics();

g2.drawImage(bgimage, 3, 22, this);

g2.setFont(new Font("黑体", Font.BOLD, 20));

g2.setColor(Color.black);

g2.drawString(" "+message, 120, 60);

g2.setFont(new Font("黑体", 0, 18));

for (int i = 0; i < 19; i++) {

g2.drawLine(13, 72 + 20 \* i, 373, 72 + 20 \* i);

g2.drawLine(13 + 20 \* i, 72, 13 + 20 \* i, 432);

}

g2.fillOval(71, 130, 4, 4);

g2.fillOval(71, 370, 4, 4);

g2.fillOval(311, 130, 4, 4);

g2.fillOval(311, 370, 4, 4);

g2.fillOval(191, 250, 5, 5);

g2.fillOval(71, 250, 4, 4);

g2.fillOval(311, 250, 4, 4);

g2.fillOval(191, 130, 4, 4);

g2.fillOval(191, 370, 4, 4);

for (int i = 0; i < 19; i++) {

for (int j = 0; j < 19; j++) {

if (allChess[i][j] == 1) {

int tempX = i \* 20 + 13;

int tempY = j \* 20 + 73;

g2.fillOval(tempX - 7, tempY - 7, 14, 14);

}

if (allChess[i][j] == 2) {

int tempX = i \* 20 + 13;

int tempY = j \* 20 + 73;

g2.setColor(Color.white);

g2.fillOval(tempX - 7, tempY - 7, 14, 14);

g2.setColor(Color.black);

g2.drawOval(tempX - 7, tempY - 7, 14, 14);

}

}

}

g.drawImage(bi,0,0,this );

}

public void mouseClicked(MouseEvent e) {

}

public void mousePressed(MouseEvent e) {

if (canPlay == true) {

x = e.getX();

y = e.getY();

if (x >= 13 && x <= 373 && y >= 73 && y <= 433) {

x = (x - 13) / 20;

y = (y - 72) / 20;

if (allChess[x][y] == 0) {

if (isBlack == true) {

allChess[x][y] = 1;

isBlack = false;

} else {

allChess[x][y] = 2;

isBlack = true;

}

boolean Winflag = this.checkWin();

if (Winflag == true) {

JOptionPane.showMessageDialog(this, "游戏结束"

+ (allChess[x][y] == 1 ? "黑方" : "白方") + "胜利");

canPlay = false;

}

} else {

JOptionPane.showMessageDialog(this, "这个地方已经有棋子了，请选择其他地方！");

}

this.repaint();

}}

}

public void mouseReleased(MouseEvent e) {

}

private boolean checkWin() {

boolean flag = false;

int count = 1;

int color = allChess[x][y];

count = this.checkChess(1, 0, color);

if (count >= 5) {

flag = true;

} else {

count = this.checkChess(0, 1, color);

if (count >= 5) {

flag = true;

} else {

count = this.checkChess(1, -1, color);

if (count >= 5) {

flag = true;

} else {

count = this.checkChess(1, 1, color);

if (count >= 5) {

flag = true;

}

}

}

}

return flag;

}

private int checkChess(int xChange, int yChange, int color) {

int count = 1;

int tempX = xChange;

int tempY = yChange;

while (x+xChange>=0&&x+xChange<=18&&y+yChange>=0&&y+yChange<=18&&

color == allChess[x + xChange][y + yChange]) {

count++;

if (xChange != 0)

xChange++;

if (yChange != 0){

if(yChange>0)

yChange++;

else{

yChange--;

}

}

}

xChange = tempX;

yChange = tempY;

while (x-xChange>=0&&x- xChange <= 18&&y - yChange>= 0&&y- yChange<=18&&

color == allChess[x - xChange][y - yChange]) {

count++;

if (xChange != 0)

xChange++;

if (yChange != 0){

if(yChange>0)

yChange++;

else{

yChange--;

}

}

}

return count;

}

public void mouseEntered(MouseEvent e) {

}

public void mouseExited(MouseEvent e) {

}

public void run() {

}

public static void main(String[] args) {

new wuziqi();

}

}