# Java实验报告

功能设计：代码可以完成五子棋棋盘、界面、棋子和弹窗的绘制，能判断获胜条件，也能完成存档与读档，终局回放的功能，没有完成联机功能和网上存盘功能。

核心类：Test类：用于调用MyJFrame类，仅有main函数。

public class Test {

public static void main(String[] args) {

MyJFrame mj=new MyJFrame();

mj.myJFrame();

}

}

MyJFrame类：实现了剩余所有功能。

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.awt.image.BufferedImage;

import java.io.\*;

public class MyJFrame extends JFrame implements MouseListener {

int qx = 20, qy = 40, qw = 490, qh = 490;

int bw = 150, bh = 50, bx = 570, by = 150;

int x = 0, y = 0;

int[][] SaveGame = new int[15][15];

int qc = 1;

int qn = 0;

boolean canplay = true;

String go = "黑子先行";

public void myJFrame() {

this.setTitle("五子棋");

this.setSize(800, 550);

this.setResizable(false);

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

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

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

this.setLocation((width - 800) / 2, (height - 600) / 2);

this.addMouseListener(this);

this.setVisible(true);

}

public void paint(Graphics g) {

BufferedImage bi = new BufferedImage(800, 550, BufferedImage.TYPE\_INT\_ARGB);

Graphics g2 = bi.createGraphics();

BufferedImage image = null;

g2.drawImage(image, 10, 10, this);

g2.setColor(Color.BLACK);

g2.setFont(new Font("黑体", 10, 50));

g2.drawString("五子棋", 525, 100);

g2.setColor(Color.getHSBColor(30, (float) 0.10, (float) 0.90));

g2.fillRect(qx, qy, qw, qh);

g2.setColor(Color.WHITE);

g2.fillRect(bx, by, bw, bh);

g2.setFont(new Font("黑体", 10, 30));

g2.setColor(Color.black);

g2.drawString("开始", 615, 185);

g2.setColor(Color.WHITE);

g2.fillRect(bx, by + 60, bw, bh);

g2.setFont(new Font("黑体", 10, 30));

g2.setColor(Color.black);

g2.drawString("存裆", 615, 245);

g2.setColor(Color.WHITE);

g2.fillRect(bx, by + 120, bw, bh);

g2.setFont(new Font("黑体", 10, 30));

g2.setColor(Color.black);

g2.drawString("读裆", 615, 305);

g2.setColor(Color.getHSBColor(30, (float) 0.10, (float) 0.90));

g2.fillRect(550, 350, 200, 150);

g2.setColor(Color.WHITE);

g2.fillRect(bx, by + 180, bw, bh);

g2.setFont(new Font("黑体", 10, 30));

g2.setColor(Color.black);

g2.drawString("终盘", 615, 365);

g2.setColor(Color.black);

g2.setFont(new Font("黑体", 10, 20));

g2.drawString(go, 610, 410);

g2.drawString("计算机001王昕海", 560, 440);

g2.drawString("2201412079", 560, 470);

for (int x = 0; x <= qw; x += 35) {

g2.drawLine(qx, x + qy, qw + qx, x + qy);

g2.drawLine(x + qx, qy, x + qx, qh + qy);

}

for (int i = 3; i <= 11; i += 4) {

for (int y = 3; y <= 11; y += 4) {

g2.fillOval(35 \* i + qx - 3, 35 \* y + qy - 3, 6, 6);

}

}

for (int i = 0; i < 15; i++) {

for (int j = 0; j < 15; j++) {

if (SaveGame[i][j] == 1)

{

int sx = i \* 35 + qx;

int sy = j \* 35 + qy;

g2.setColor(Color.BLACK);

g2.fillOval(sx - 13, sy - 13, 26, 26);

}

if (SaveGame[i][j] == 2)

{

int sx = i \* 35 + qx;

int sy = j \* 35 + qy;

g2.setColor(Color.WHITE);

g2.fillOval(sx - 13, sy - 13, 26, 26);

g2.setColor(Color.BLACK);

g2.drawOval(sx - 13, sy - 13, 26, 26);

}

}

}

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

}

private boolean WinLose() {

boolean flag = false;

int count = 1;

int color = SaveGame[x][y];

int i = 1;

while (color == SaveGame[x + i][y]) {

count++;

i++;

}

i = 1;

while (color == SaveGame[x - i][y]) {

count++;

i++;

}

if (count >= 5) {

flag = true;

}

count = 1;

i = 1;

while (color == SaveGame[x][y + i]) {

count++;

i++;

}

i = 1;

while (color == SaveGame[x][y - i]) {

count++;

i++;

}

if (count >= 5) {

flag = true;

}

count = 1;

i = 1;

while (color == SaveGame[x - i][y - i]) {

count++;

i++;

}

i = 1;

while (color == SaveGame[x + i][y + i]) {

count++;

i++;

}

if (count >= 5) {

flag = true;

}

count = 1;

i = 1;

while (color == SaveGame[x + i][y - i]) {

count++;

i++;

}

i = 1;

while (color == SaveGame[x - i][y + i]) {

count++;

i++;

}

if (count >= 5) {

flag = true;

}

return flag;

}

public void Initialize() {

for (int i = 0; i < 15; i++) {

for (int j = 0; j < 15; j++) {

SaveGame[i][j] = 0;

}

}

qc = 1;

go = "轮到黑子";

}

@Override

public void mouseClicked(MouseEvent e) {

}

@Override

public void mousePressed(MouseEvent e) {

x = e.getX();

y = e.getY();

if (canplay == true) {

if (x > qx && x < qx + qw && y > qy && y < qy + qh) {

if ((x - qx) % 35 > 17) {

x = (x - qx) / 35 + 1;

} else {

x = (x - qx) / 35;

}

if ((y - qy) % 35 > 17) {

y = (y - qy) / 35 + 1;

} else {

y = (y - qy) / 35;

}

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

SaveGame[x][y] = qc;

qn = 0;

} else {

qn = 1;

}

if (qn == 0) {

if (qc == 1) {

qc = 2;

go = "轮到白子";

} else {

qc = 1;

go = "轮到黑子";

}

}

this.repaint();

boolean wl = this.WinLose();

if (wl) {

JOptionPane.showMessageDialog(this, "游戏结束，" + (SaveGame[x][y] == 1 ? "黑方赢" : "白方赢"));

canplay = false;

try {

File writeName = new File("D:\\endsave.txt");

if(!writeName.exists()) {

writeName.createNewFile();

}

FileWriter writer = new FileWriter(writeName);

BufferedWriter out = new BufferedWriter(writer);

for(int i=0;i<15;i++)

for(int j=0;j<15;j++)

out.write(SaveGame[i][j]);

out.write(qc);

out.flush();

out.close();

} catch (IOException ee) {

ee.printStackTrace();

}

}

} else {

}

}

if (e.getX() > bx && e.getX() < bx + bw && e.getY() > by && e.getY() < by + bh) {

if (canplay == false) {

canplay = true;

JOptionPane.showMessageDialog(this, "游戏开始");

Initialize();

this.repaint();

} else {

JOptionPane.showMessageDialog(this, "重新开始");

Initialize();

this.repaint();

}

}

if (e.getX() > bx && e.getX() < bx + bw && e.getY() > by + 60 && e.getY() < by + 60 + bh) {

if (canplay == true) {

int z = 0;

for (int i = 0; i < 15; i++) {

for (int j = 0; j < 15; j++) {

if (SaveGame[i][j] != 0) {

z++;

}

}

}

if (z != 0) {

try {

File writeName = new File("D:\\data.txt");

if(!writeName.exists()) {

writeName.createNewFile();

}

FileWriter writer = new FileWriter(writeName);

BufferedWriter out = new BufferedWriter(writer);

for(int i=0;i<15;i++)

for(int j=0;j<15;j++)

out.write(SaveGame[i][j]);

out.write(qc);

out.flush();

out.close();

JOptionPane.showMessageDialog(this, "已存档！");

} catch (IOException ee) {

ee.printStackTrace();

}

} else {

JOptionPane.showMessageDialog(this, "无棋子");

}

} else {

JOptionPane.showMessageDialog(this, "还没开始");

}

}

if (e.getX() > bx && e.getX() < bx + bw && e.getY() > by + 120 && e.getY() < by + 120 + bh)

{

if (canplay == true)

{

try{

File file = new File("D:\\data.txt");

if(!file.exists()) {

JOptionPane.showMessageDialog(this, "没有存档数据！");

}else

{

BufferedReader br = new BufferedReader(new FileReader(file));

int t= 0;

for(int i=0;i<15;i++)

for(int j=0;j<15;j++)

{

t = br.read();

SaveGame[i][j]=t;

}

qc= br.read();

br.close();

this.repaint();

}

}catch(Exception ee){

ee.printStackTrace();

}

}

else {

JOptionPane.showMessageDialog(this, "还没开始");

}

}

if (e.getX() > bx && e.getX() < bx + bw && e.getY() > by+180 && e.getY() < by + bh+180) {

try{

File file = new File("D:\\endsave.txt");

if(!file.exists()) {

JOptionPane.showMessageDialog(this, "还没有下过一盘完整的棋");

}else

{

BufferedReader br = new BufferedReader(new FileReader(file));

int t= 0;

for(int i=0;i<15;i++)

for(int j=0;j<15;j++)

{

t = br.read();

SaveGame[i][j]=t;

}

qc=br.read();

br.close();

this.repaint();

}

}catch(Exception ee){

ee.printStackTrace();

}

}

}

@Override

public void mouseReleased(MouseEvent e) {

}

@Override

public void mouseEntered(MouseEvent e) {

}

@Override

public void mouseExited(MouseEvent e) {

}

}

测试结果：



出现的问题：

1..图形和文字覆盖混乱，已解决

2..判断胜负的算法部分有错判，已解决

3. 重开时画板绘制难以处理，已解决