**课 程 设 计 报 告**

**课程设计名称：软件工程与计算Ⅱ**

**课程设计题目**：**利用Java编写五子棋游戏**

院（系）：计算机科学技术学院

专 业：网络编程

班 级：2015网络编程

学 号：20151104711

姓 名：丁楷轩

指导教师：朝力萌

完成日期：2018.7

# 第1章 概要设计

## 1.1题目的内容与要求

**内容：**

**利用java 语言逻辑编写一款五子棋游戏。**

**要求：**

1. **能够进行人人对战**
2. **能够设置黑白棋先行**
3. **能够判断胜利的一方**

## 1.2总体结构

五子棋

先行设置

人人博弈

开始游戏

重新开始

退出游戏

# 第2章 详细设计

### 2.1 设置窗体

// 设置窗体的大小并居中

this.setSize(500, 600); // 设置窗体大小

this.setTitle("五子棋游戏"); // 设置窗体标题

int width = Toolkit.getDefaultToolkit().getScreenSize().width;// 获取屏幕的宽度

int height = Toolkit.getDefaultToolkit().getScreenSize().height;// 获取屏幕的高度

this.setLocation((width - 500) / 2, (height - 500) / 2); // 设置窗体的位置（居中）

this.setResizable(false); // 设置窗体不可以放大

// this.setLocationRelativeTo(null);//这句话也可以设置窗体居中

### 2.2 设置菜单栏

JMenuBar bar = new JMenuBar();

this.setJMenuBar(bar);

// 添加菜单栏目录

JMenu menu1 = new JMenu("游戏菜单"); // 实例化菜单栏目录

JMenu menu2 = new JMenu("设置");

JMenu menu3 = new JMenu("帮助");

bar.add(menu1); // 将目录添加到菜单栏

bar.add(menu2);

bar.add(menu3);

JMenu menu4 = new JMenu("博弈模式"); // 将“模式”菜单添加到“设置”里面

menu2.add(menu4);

// 设置“”目录下面的子目录

JRadioButtonMenuItem item1 = new JRadioButtonMenuItem("人人博弈");

**2.2.1设置按钮组并把黑棋先行和白起现行添加到一个按钮组里面**

ButtonGroup bg1 = new ButtonGroup();

bg1.add(item3);

bg1.add(item4);

// 将按钮组添加到菜单里面

menu5.add(item3);

menu5.add(item4);

item3.setSelected(true);

// 设置“帮助”下面的子目录

JMenuItem menu6 = new JMenuItem("帮助");

menu3.add(menu6);

/\*

\* 菜单栏的目录设置完毕

\*/

// 开始游戏菜单设置

JMenuItem menu7 = new JMenuItem("开始游戏");

menu1.add(menu7);

JMenuItem menu8 = new JMenuItem("重新开始");

menu1.add(menu8);

**2.2.2先行设置**

JMenu menu5 = new JMenu("先行设置"); // 将“先行设置”菜单添加到“设置”里面

menu2.add(menu5);

// 设置黑子先行还是白字先行的按钮

JRadioButtonMenuItem item3 = new JRadioButtonMenuItem("黑方先行");

JRadioButtonMenuItem item4 = new JRadioButtonMenuItem("白字先行");

### 2.3 设置窗体里容器操作

@SuppressWarnings("serial")

public class MyPanel extends JPanel implements MouseListener, Runnable {

private static final Toolkit ResourceUtil = null;

public Image boardImg; // 定义背景图片

static int[][] allChess = new int[15][15]; // 棋盘数组

static int[][] temporaryChess = new int[15][15];

int x;// 保存棋子的横坐标

int y;// 保存棋子的纵坐标

Boolean canPlay = false; // 游戏是否继续，默认为继续

Boolean isBlack = true;// 是否是黑子，默认为黑子

Boolean isManAgainst = false; // 判断是否是人人对战

String message = "";

Thread t = new Thread(this);

// 获取isBlack的值

public boolean getIsBlack() {

return this.isBlack;

}

### 2.4数据初始化

@Override

public void paint(Graphics g) {

super.paint(g);

int imgWidth = boardImg.getWidth(this); // 获取图片的宽度

int imgHeight = boardImg.getHeight(this); // 获取图片的高度

int FWidth = getWidth();

int FHeight = getHeight();

@SuppressWarnings("unused")

String message; // 标记谁下棋

int x = (FWidth - imgWidth) / 2;

int y = (FHeight - imgHeight) / 2;

g.drawImage(boardImg, x, y, null); // 添加背景图片到容器里面

g.setFont(new Font("宋体", 0, 14));

// 绘制棋盘

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

g.drawLine(30, 30 + 30 \* i, 450, 30 + 30 \* i);

g.drawLine(30 + 30 \* i, 30, 30 + 30 \* i, 450);

}

// 绘制五个中心点

g.fillRect(240 - 5, 240 - 5, 10, 10); // 绘制最中心的正方形

g.fillRect(360 - 5, 360 - 5, 10, 10); // 绘制右下的正方形

g.fillRect(360 - 5, 120 - 5, 10, 10); // 绘制右上的正方形

g.fillRect(120 - 5, 360 - 5, 10, 10);// 绘制左下的正方形

g.fillRect(120 - 5, 120 - 5, 10, 10);// 绘制左上的正方形

// 定义棋盘数组

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

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

draw(g, i, j); // 调用下棋子函数

}

}

}

### 2.5鼠标点击时发生的函数

public void mousePressed(MouseEvent e) {

if (canPlay == true) {// 判断是否可以开始游戏

x = e.getX(); // 获取鼠标的焦点

y = e.getY();

if (isManAgainst == true) {// 判断是否是人人对战

manToManChess();

} else { // 否则是人机对战，人机下棋

manToManChess();

}

}

}

### 2.6判断输赢的函数

private boolean checkWin(int x, int y) {

// TODO Auto-generated method stub

boolean flag = false;

// 保存共有多少相同颜色棋子相连

int count = 1;

// 判断横向 特点：allChess[x][y]中y值相同

int color = allChess[x][y];

// 判断横向

count = this.checkCount(x, y, 1, 0, color);

if (count >= 5) {

flag = true;

} else {

// 判断纵向

count = this.checkCount(x, y, 0, 1, color);

if (count >= 5) {

flag = true;

} else {

// 判断右上左下

count = this.checkCount(x, y, 1, -1, color);

if (count >= 5) {

flag = true;

} else {

// 判断左下右上

count = this.checkCount(x, y, 1, 1, color);

if (count >= 5) {

flag = true;

}

}

}

}

return flag;

}

### 2.7判断相同棋子连接的个数

private int checkCount(int x, int y, int xChange, int yChange, int color) {

// TODO Auto-generated method stub

int count = 1;

int tempX = xChange;

int tempY = yChange;

while (x + xChange >= 0 && x + xChange <= 14 && y + yChange >= 0 && y + yChange <= 14

&& 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 <= 14 && y - yChange >= 0 && y - yChange <= 14

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

count++;

if (xChange != 0) {

xChange++;

}

if (yChange != 0) {

if (yChange > 0) {

yChange++;

} else {

yChange--;

}

}

}

return count;

}

### 2.8 绘制黑白棋子

public void draw(Graphics g, int i, int j) {

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

g.setColor(Color.black);// 黑色棋子

g.fillOval(30 \* i + 30 - 7, 30 \* j + 30 - 7, 14, 14);

g.drawString(message, 230, 20);

}

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

g.setColor(Color.white);// 白色棋子

g.fillOval(30 \* i + 30 - 7, 30 \* j + 30 - 7, 14, 14);

g.drawString(message, 230, 20);

}

}

**2.9点击开始游戏设置属性，游戏开始**

@SuppressWarnings("deprecation")

public void Start() {

this.canPlay = true;

for (int i = 0; i < 14; i++) {

for (int j = 0; j < 14; j++) {

allChess[i][j] = 0;

}

}

if (canPlay == true) {

t.resume();

}

this.repaint();

JOptionPane.showMessageDialog(this, "游戏开始了，请开始下棋");

}

**2.10人人对战**

@SuppressWarnings("deprecation")

public void Start() {

this.canPlay = true;

for (int i = 0; i < 14; i++) {

for (int j = 0; j < 14; j++) {

allChess[i][j] = 0;

}

}

if (canPlay == true) {

t.resume();

}

this.repaint();

JOptionPane.showMessageDialog(this, "游戏开始了，请开始下棋");

}

**2.11退出游戏**

JMenuItem menu9 = new JMenuItem("退出游戏");

menu1.add(menu9);

menu9.addMouseListener(new MouseListener() {

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

Icon icon = new Icon() {

@Override

public void paintIcon(Component c, Graphics g, int x, int y) {

// TODO Auto-generated method stub

}

@Override

public int getIconWidth() {

// TODO Auto-generated method stub

return 0;

}

@Override

public int getIconHeight() {

// TODO Auto-generated method stub

return 0;

}

};

Object[] options = { "退出游戏", "不，谢谢" };

int n = JOptionPane.showOptionDialog(null, "是否退出游戏", "消息", 0, 1, icon, options, "保存并重新开始游戏");

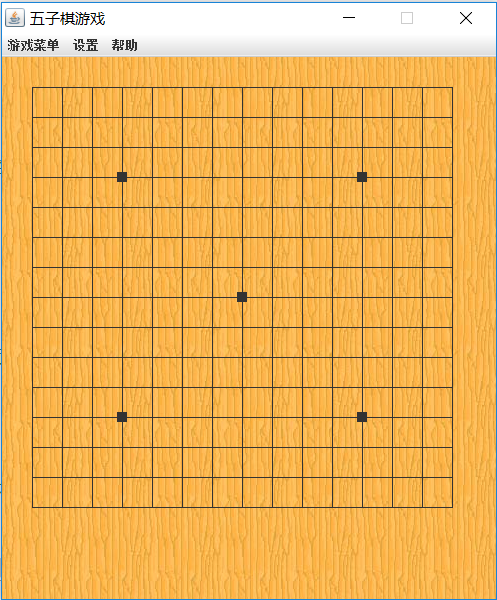
if (n == 0) {

System.exit(0);// 退出程序

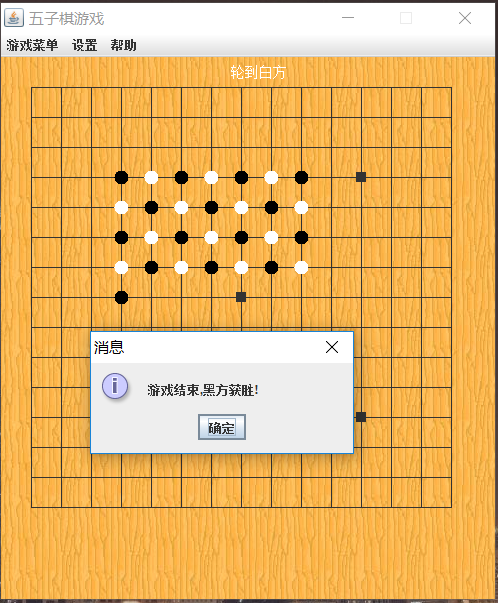
}

# 调试分析

**主界面**

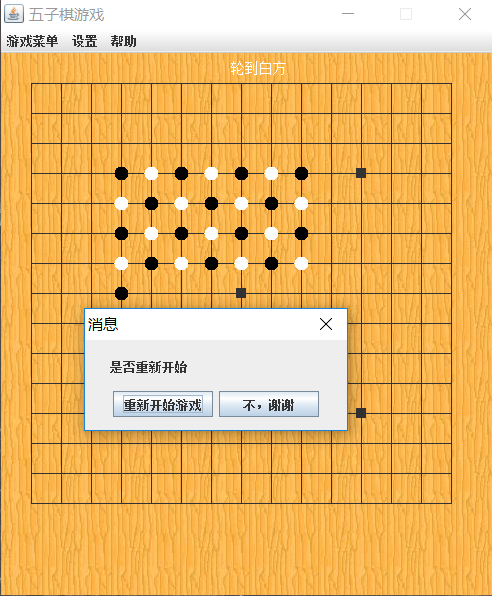


**一方获胜**

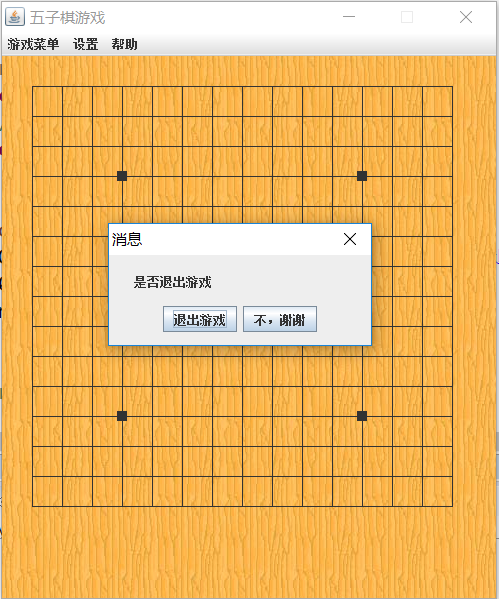


# 第4章 使用说明与执行结果

一、**运行操作及结果**



当游戏进行到一方获胜的时候，弹出是否重新开始的窗口，点击重新开始进行下一局游戏



点击游戏菜单，选择退出游戏，弹出对话框，询问是否退出游戏，点击退出游戏，游戏结束并且退出游戏。

1. 总结

**通过此次进行的软件工程2的课程学习，我学会了从搭建编程环境到完成一个简单项目，并且已经初具对项目开发的思维。而且此次的代码量不小，对个人能力是一个很大的挑战，不像之前的一二百行代码，让人只能粗浅的理解，而这次的实战，让我们学到了很多东西，而且个人积累的代码量也得到了很大的提升。而且额外学到了一些关于java web的知识，对Bean，Dao，Servlet也有了较为清楚的认识。这次的课程让我学到了很多，受益匪浅。**

**参考文献：**

**[1] 《Java基础入门》传智播客高教产品研发部 编著.清华大学出版社**

**[2] 《疯狂Java讲义》李刚 编著.电子工业出版社**

**[3] https://www.360kan.com/va/Y8ctbHNv7pg3Dj.html（Java入门视频）**