

南开大学

本科生毕业论文（设计）

中文题目： 用Java开发一个基于网络联机的五子棋小游戏

外文题目： Using Java to develop a net-based gobang game

学 号： 1910278

姓 名： 李向阳

年 级： 2019级

专 业： 软件工程

系 别： 计算机科学与技术

学 院： 计算机学院

指导教师： 刘嘉欣

完成日期： 2021年12月18日

摘要

随着人们生活越来越富裕，人们追求的中心从物质生活逐渐转移到精神娱乐，而游戏正是精神娱乐里面的重要的版块之一。2021年，中国游戏市场实际销售收入2965.13亿元，较去年增收178.26亿元，同比增长6.4%，所以我打算用学到的Java中的UDP网络连接，多线程,Swing GUI，事件监听事件的知识做一款基于网络联机的五子棋小游戏，从而熟练应用Java知识，提高Java编程能力，并且希望能为劳逸结合提供一份乐趣。

关键字：精神娱乐；游戏；网络联机；五子棋

Abstract

With people's life getting richer and richer, people's pursuit of the center from the material life gradually shifted to the spiritual entertainment, and the game is one of the important parts of the spiritual entertainment. In 2021, the actual sales revenue of China's game market is 296.513 billion yuan, 17.826 billion yuan more than last year, with a year-on-year growth of 6.4%. Therefore, I plan to use the knowledge I have learned about UDP network connection in Java, multi-threading, Swing GUI and event monitoring events to make a small online backgamgo game. So as to skillfully apply Java knowledge, improve Java programming ability, and hope to combine work and rest to provide a fun.

Key words: spiritual entertainment; The game; Network online; gobang

目录

摘要.....	2
Abstract.....	2
第一章 引言.....	3
第二章 五子棋的基本常识和原理.....	4
(一) 五子棋的起源.....	4
(二) 五子棋的基本常识.....	4
(三) 五子棋比赛的相关规定.....	4
1. 职业连珠规则.....	4
2. 比赛中的规定和特例.....	4
3. 特殊规定.....	5
第三章 五子棋的系统设计.....	5
(一) 图形化界面的设计.....	5
(二) 实现游戏规则的设计(概述).....	5
1. 游戏界面的设计方案.....	5
2. 网络连接和接收信息.....	5
3. 不同信息的接收的设计方案.....	5
4. 判断胜负的算法的设计方案.....	6
第四章 五子棋的实现.....	6
(一) 网络联机的实现.....	6
(二) 开局.....	6
(三) 下棋.....	7
(四) 聊天功能的实现.....	7
(五) 悔棋.....	8
(六) 判断胜负（包含认输）.....	8
1. 认输.....	8
2. 五子连珠.....	8

(七) 复盘的实现.....	8
第五章 五子棋的核心代码实现.....	9
(一) 网络连接.....	9
(二) 下棋.....	9
(三) 接收信息后的辨别过程.....	10
(四) 悔棋的算法.....	11
(五) 复盘的算法.....	12
(六) 认输的算法.....	12
(七) 判断五子连珠的算法.....	13
(八) 按钮的监听器.....	14
第六章 五子棋的测试.....	15
(一) 网路联机的测试.....	15
(二) 开局的测试.....	15
(三) 下棋的测试.....	16
(四) 聊天功能的测试.....	17
(五) 悔棋的测试.....	19
(六) 判断胜负(包含认输).....	20
1. 认输.....	21
2. 五子连珠.....	22
(七) 复盘的实现.....	24
第七章 总结与展望.....	24
(一) 主要工作.....	24
(二) 展望.....	24
参考文献.....	25

第一章 引言

随着时代不断发展,游戏已经被称为“第九大艺术”。人们追求的中心从物质生活逐渐转移到精神娱乐,而游戏正是精神娱乐里面的重要的版块之一。2021年,中国游戏市场实际销售收入2965.13亿元,较去年增收178.26亿元,同比增长6.4%,国内游戏用户规模达到6.66亿。由此可见,游戏已经成为了人们生活中不可缺少的一部分。科学技术升级加速游戏业态变革,产业发展又助益着科技创新,使游戏称为虚拟现实、人工智能等新技术应用的实验场。与此同时,“游戏+”形成的多元融合,延伸了游戏的社会功能,为其多元赋能提供了更为宽广的实践平台。近年来,国内游戏企业开始逐步探索“游戏+”模式,发挥游戏多元价值,使产品和服务广泛赋能与科技、文化、教育、医疗、公益等领域。游戏不单单是一种娱乐消遣,更是建设数字文化生态、推动多领域社会实践的重要助力。在健康有序规范的前提下,进一步发挥游戏的多元赋能作用,强化社会价值和社会功能将是未来行业发展的重要方式。游戏发展的潮流趋势不可阻挡,所以有必要深入去研究和开发游戏。

Java的出现是源于对独立于平台语言的需要,希望这种编程语言能编写出嵌入各种家用电器等设备的芯片上、且易于维护的程序。人们发现当时的编程语言都有一个共同的缺点,就是针对CPU芯片进行编译。这样,一旦电器设备更换了芯片就不能保证程序正确运行。90年Sun公司成立了开发小组,开始致力于开发一种可移植、跨平台的编程语言,该语言能生成正确运行于各种操作系统各种芯片上的代码。Java的出现标志着真正的分布式系统的到来。与C++相比,Java更简单,操作起来更方便,类库函数也更多,是一种跨平台、可移植的多线程的动态语言,具有平台无关性,安全性,健壮性。虽然C++的效率比Java更强,但是

Java发展到现在,其性能效率也高了起来。本文中开发的游戏为经典的五子棋游戏,我选择用Java开发。开发环境使用eclipse集成开发环境,与IDEA相比,eclipse更加轻便,小巧。

第二章 五子棋的基本常识和原理

(一) 五子棋的起源

五子棋,是一种两人对弈的纯策略型棋类游戏,也称为“串珠”、“连五子”;是中国民间非常熟知的一个古老棋种。相传,它起源于四千多年前的尧帝时期,比围棋的历史还要悠久。也有传说,五子棋最初流行于少数民族地区,以后渐渐演变为围棋并在炎黄子孙后代中遍及开来。

五子棋发展于日本,流行于欧美。容易上手,老少皆宜,而且趣味横生,引人入胜;不仅能增强思维能力,提高智力,而且富含哲理,有助于修身养性。

传统五子棋的棋具与围棋相同,棋子分为黑白两色,棋盘为 19×19 ,棋子放置于棋盘线交叉点上。两人对局,各执一色,轮流下一子,先将横、竖或斜线的5个或5个以上同色棋子连成不间断的一排者为胜。因为传统五子棋在落子后不能移动或拿掉,所以也可以用纸和笔来进行游戏。

(二) 五子棋的基本常识

与任何一种竞技棋一样,五子棋的每一局棋也分为三个阶段:开局,中局和残局。

五子棋的开始阶段称为开局,或称布局。其开局阶段是十分短暂的,大约在七着与十几着之间。在这一阶段的争夺中,双方的布局,应对将对以后的胜负起着极为关键的作用。开局阶段取得的形式好坏,主动与被动,先手与后手的优劣程度,往往直接影响中局的战斗。因此积极处理好开局和开局向中局的过渡十分重要。

五子棋是一到五,逐渐布子,发展连系,同时运用限制和反限制的智慧,在连子的过程中为自己的棋子争得相对的主动权和优势,逐步扩展优势,或者从劣势转换为优势,击溃对方的防线,最后连五取胜或抓禁手取胜或迫使对方投子认负。

(三) 五子棋比赛的相关规定

1. 职业连珠规则

①黑方先下子,白后下,从天元开始相互顺序落子

②最先在棋盘横向、竖向、斜向形成连续的相同色五个棋子的一方为胜。

黑棋禁手判负,白棋无禁手。黑棋禁手包括“三三”(包括“四三三”)、“四四”(包括“四四三”)、“长连”。即黑棋只能以“四三”取胜。

③如分不出胜负,则定为平局。对局中拔子、中途退场均判为负。

④如分不出胜负,则定为平局。对局中拔子、中途退场均判为负。

五连与禁手同时形成,判胜。

2. 比赛中的规定和特例

①对局开始前,双方猜子,大数减小数,单数交换,偶数不换。

②白棋第一手(即盘面第二着棋)应在天元为界自己一侧布子,主要以示尊重对方,之后双方可任意行子。

③对局中双方应遵守“职业连珠五子棋规则”,如出现争议,应由裁判判定。

④对局中应礼貌行棋,防止不正常声音、动作影响对方思考。比赛结束后,双方应将各自棋子放回原处。

⑤对局中如黑方出现禁手，白方应立即指出禁手点，黑方即负。如白方在黑方出现禁手后，又下一步白子，黑棋禁手则不成立了。

⑥比赛对局时间，双方分别为15分钟、1小时、2小时、5小时四个等级标准，超时判负。

3. 特殊规定

职业五子棋虽然对黑棋采取了种种限制，但黑子先行的优势依然很大。因此，在高段位的职业比赛中，又出现了三种特殊的规定。

指定打法：是指比赛双方按照约定好的开局进行对弈，由白棋先行。

三手可交换：是指黑棋下盘面第三手后，白方在下第四手之前，如感觉黑方棋形不利于己方，可提出交换，即执白棋一方变为执黑棋一方，而黑方不可以不交换。

五手两打法：是指黑棋在下盘面第五手棋时，必须下两步棋，让白棋在这两部棋中拿掉一粒棋子，然后再继续对弈。一般来说，白棋肯定拿掉对白方不利的一点，而保留对黑方较为不利的那点让黑方行棋。

第三章 五子棋的系统设计

（一）图形化界面的设计

为了实现所需要的功能，以及便于玩家使用，所以有必要使用GUI界面来开发游戏。

1. 继承了JPanel的FiveChessPane

FiveChessPane重写了paint()方法，paint()方法中用划线的方法绘制了19×19的棋盘，用g.setColor()设置棋子的颜色，用g.fillOval()方法绘制棋子。在(3,3), (3,9), (3,15), (9,3), (9,9), (9,15), (15,3), (15,9), (15,15)设置了标注点位，以提醒玩家棋盘的中心位置，便于把握有利位置落子。

同时，FiveChessPane也继承了MouseListener接口，重写了MousePressed方法。

2. 继承了JFrame的FiveChessFrame

FiveChessFrame包含一个FiveChessPane对象，以及一系列选项，比如，一个记录对方Ip地址的JTextField对象，三个JButton对象，分别对应开局，悔棋，认输，一个记录要发送的信息的JTextField对象，一个对应发送信息的JButton对象，以及一个显示聊天信息的JTextArea对象的聊天框。

（二）实现游戏规则的设计(概述)

1. 游戏界面的设计方案

设计一个继承JFrame的类作为窗体，将这个类的布局在开始时设为null，设计一个继承JPanel的类，重写paint()函数，用来显示棋盘，棋盘上的线调用画笔的drawLine()方法，棋盘上的棋子调用画笔的drawOval()方法，颜色的不同则调用画笔的setColor()方法，用这个方法改变颜色。将继承JPanel的类通过setBounds()放到继承JFrame的类的左上方，将一系列按钮放在继承JFrame类的右方，右方包括连接Ip按钮，开局按钮，悔棋按钮，认输按钮，以及发送文字框和消息框

2. 网络连接和接收信息

对于网络连接可以考虑使用TCP协议，也可以考虑UDP协议。为了得到ip，要有获取ip的文本框。接收信息应该另外开一个线程。

3. 不同信息的接收的设计方案

对于不同类型信息，则使用了不同的标识符来区别它们，比如，开局的信息，会发送“ready，”，落下棋子的信息会发送“POS，”+X+“，”+Y，悔棋会发送“

Back, ” 等等, 接收方先通过String的split()方法分辨出是哪一种信息, 然后再做进一步的操作。

4. 判断胜负的算法的设计方案

本程序中判断胜负主要有两种方式, 某一方认输或者某一方达成五子连珠。

第四章 五子棋的实现

(一) 网络联机的实现

网络联机使用了网络和多线程的知识。UDP 是User Datagram Protocol的简称, 中文名是用户数据包协议, 是OSI (Open System Interconnection, 开放式系统互联) 参考模型中一种无连接的传输层协议, 提供面向事务的简单不可靠信息传送服务, IETF RFC 768 是UDP的正式规范。UDP协议与TCP协议一样用于处理数据包, 在OSI模型中, 两者都位于传输层, 处于IP协议的上一层。UDP有不提供数据包分组、组装和不能对数据包进行排序的缺点, 也就是说, 当报文发送之后, 是无法得知其是否安全完整到达的。UDP用来支持那些需要在计算机之间传输数据的网络应用。包括网络视频会议系统在内的众多的客户/服务器模式的网络应用都需要使用UDP协议。UDP协议从问世至今已经被使用了很多年, 虽然其最初的光彩已经被一些类似协议所掩盖, 但即使在今天UDP仍然不失为一项非常实用和可行的网络传输层协议。UDP是OSI参考模型中一种无连接的传输层协议, 它主要用于不要求分组顺序到达的传输中, 分组传输顺序的检查与排序由应用层完成 [4], 提供面向事务的简单不可靠信息传送服务。UDP 协议基本上是IP协议与上层协议的接口。UDP协议适用端口分别运行在同一台设备上的多个应用程序。UDP提供了无连接通信, 且不对传送数据包进行可靠性保证, 适合于一次传输少量数据, UDP传输的可靠性由应用层负责。常用的UDP端口号有: 53 (DNS)、69 (TFTP)、161 (SNMP), 使用UDP协议包括: TFTP、SNMP、NFS、DNS、BOOTP。UDP报文没有可靠性保证、顺序保证和流量控制字段等, 可靠性较差。但是正因为UDP协议的控制选项较少, 在数据传输过程中延迟小、数据传输效率高, 适合对可靠性要求不高的应用程序, 或者可以保障可靠性的应用程序, 如DNS、TFTP、SNMP等。

基于上述信息, 本程序的网络连接使用UDP传输。而发送和接收信息则使用了多线程的知识, 主线程负责发送信息, 副线程负责接收信息。

对于不同类型信息, 则使用了不同的标识符来区别它们, 比如, 开局的信息, 会发送 “ready, ”, 落下棋子的信息会发送 “POS, ” +X+ “,” +Y, 悔棋会发送 “Back, ” 等等, 接收方先通过String的split()方法分辨出是哪一种信息, 然后再做进一步的操作。

(二) 开局

FiveChessFrame中包含了一个boolean型的enemyState对象, true表示对方已经准备好开局。

FiveChessFrame中包含了一个boolean型的State对象, true表示我方已经准备好开局。

FiveChessPane中包含了一个boolean型的canPlay对象, true表示我方可以下棋, false表示我方不可下棋

FiveChessFrame中添加了一个Start按钮,按下按钮,就会触发相应的事件,当对方按下Start按钮时,会向我们发送“ready,”信息,接收到信息后,我们将enemyState设置为true,同时对方的State对象设置为true

只有当enemyState和State同时为true时,canPlay设置为true,才可以下棋。

(三) 下棋

FiveChessPane中声明了一个 19×19 的二维数组allChess,用来存储棋盘上棋子的信息,0为没有棋子,1为黑子,2为白子。

FiveChessPane中声明了一个 $391(19 \times 19) \times 2$ 的二维数组memory,用来依次存储下棋的落点坐标。

FiveChessFrame中包含了一个int型的变量role,role为1时为黑子,role为2时为白子,双方默认均为黑子。

FiveChessPane中实现了MouseListener接口,重新写了MousePressed()方法,利用e.getX()和e.getY()获得鼠标按压的坐标。要对获得的坐标还要进行进一步的处理。

进一步的处理:将获得的坐标映射到距离这一点最近的棋盘点上,将这一棋盘点的坐标除以棋盘上每个格子的长度,获得相对于原点的坐标。

FiveChessPane中重写了paint()方法,这个方法会根据allChess中棋子颜色信息,将allChess中所有点遍历一遍,如果某一点的信息为1,则调用g.setColor(Color.Black),将画笔的颜色设置为黑色,在这一点调用g.drawOval()函数,画出黑色实心圆,如果在某一点的信息为2,则调用g.setColor(Color.White),将画笔的颜色设置为白色,在这一点调用g.drawOval(),画出白色实心圆,如果在某一点的信息为0,则什么都不调用,直接跳过。

下棋的一方将棋子的落点坐标存储到memory中,将棋子的颜色信息存储到allChess中,然后将棋子的落点坐标和棋子的颜色信息发送到另一方,调用paint()方法。然后另一方执行同样的操作,将对手发来的棋子的落点坐标存储到memory中,将棋子的颜色信息存储到allChess中,调用paint()方法。

开局后,先下棋的一方(先发送棋子落点信息的一方)发送落下的棋子相对坐标和棋子颜色的同时会将1发送过去,接收到1的另一方将己方的role对象设置为2。然后下完棋的一方的canPlay对象设置为false,接收到另一方信息的canPlay对象设置为true,将接收的棋子颜色信息存储到memory中,开始下棋,MouseListener监听鼠标按压事件,获得棋子的相对坐标,将棋子相对坐标存在allChess中,并且发送棋子坐标信息和棋子颜色信息,canPlay被设置为false,调用paint()方法。

(四) 聊天功能的实现

FiveChessFrame中包含了一个JTextArea的一个对象message,用来在图形界面充当发送的信息框,当用户在信息框内打出信息时,可以通过message.getText()方法查看要发送的信息。

FiveChessFrame中包含了一个JScrollPane对象Js,用来做另外一个JTextArea对象showArea的容器。

FiveChessFrame中包含了一个JButton的对象send,这个对象注册了ActionListener,当一方点击send时,注册的ActionListener会调用actionPerformed(),将message中信息通过UDP传输协议发送到另一方,同时会将

message中的信息加入到showArea中，接收到信息的一方，通过标识符辨别后，将接收的message加入到自己的showArea对象上。

（五）悔棋

FiveChessPane中包含的memory数组中存储着依次下棋的信息，将倒数后两个存进来的棋子的颜色在allChess中设置为0，然后重绘，调用repaint()函数，对画板进行重新绘制

（六）判断胜负（包含认输）

1. 认输

FiveChessFrame中包含了一个JButton对象surrender, 这个对象注册了ActionListener, 当一方点击surrender时，注册的ActionListener会调用actionPerformed(), 这时会根据调用的JOptionPane.showConfirmMessage()弹出一个确认框，需要一方进行确认，如果点击是，则将认输的信息通过UDP传输协议发送到另一方，同时会将allChess中的所有元素都设置为0，通过调用repaint()方法，将棋盘上的棋子全部清空，

2. 五子连珠

FiveChessFrame类中包含了定义了一个可以判断五子连珠的方法，算法为，当落下棋子时，方法会以落下的棋子为中心，判断4个方向，包含X轴方向，Y轴方向，左偏45度方向，右偏45度方向。

当判断X轴方向时，会以落下的棋子为中心，左右各伸展4个单位，从最左边到最右边，依次遍历allChess中对应的元素的数值，并将这个数值添加到一个String类的空字符串S中，声明一个含有5个当前落下的棋子颜色的字符串作为对比的字符串，如果在S中查找到了对比的字符串，那么则说明在X轴方向上满足五子连珠。

当判断Y轴方向时，会以落下的棋子为中心，上下各伸展4个单位，从最左边到最右边，依次遍历allChess中对应的元素的数值，并将这个数值添加到一个String类的空字符串S中，声明一个含有5个当前落下的棋子颜色的字符串作为对比的字符串，如果在S中查找到了对比的字符串，那么则说明在X轴方向上满足五子连珠。

当判断左偏45度轴方向时，会以落下的棋子为中心，沿着左偏45度的方向各伸展4个单位，从最左边到最右边，依次遍历allChess中对应的元素的数值，并将这个数值添加到一个String类的空字符串S中，声明一个含有5个当前落下的棋子颜色的字符串作为对比的字符串，如果在S中查找到了对比的字符串，那么则说明在X轴方向上满足五子连珠。

当判断右偏45度方向时，会以落下的棋子为中心，右偏45度方向各伸展4个单位，从最左边到最右边，依次遍历allChess中对应的元素的数值，并将这个数值添加到一个String类的空字符串S中，声明一个含有5个当前落下的棋子颜色的字符串作为对比的字符串，如果在S中查找到了对比的字符串，那么则说明在X轴方向上满足五子连珠。

如果上述有一个方向满足五子连珠，则属于当前棋子的颜色的一方获胜。

（七）复盘的实现

FiveChessPane类中包含一个memory的数组，声明一个19×19的二维数组reShow对象，将allChess中的元素依次复制到reShow中，然后遍历allChess, 并将allChess中的每一个元素都设置为0，调用repaint()方法，进行重绘来达到清

空的目的，然后从前到后，每隔一秒，将memory中坐标对应的棋子的颜色复制到allChess中，然后调用repaint()方法，进行重绘。

第五章 五子棋的核心代码实现

（一）网络连接

```
socket=new DatagramSocket(12346);
DatagramPacket dp =new DatagramPacket(buf,buf.length);
while(true)
{
    socket.receive(dp);
    InetAddress ia=dp.getAddress();
}
```

（二）下棋

```
public void mousePressed(MouseEvent e) {
    // TODO Auto-generated method stub
    if(canPlay==true)
    {
        x=e.getX();
        y=e.getY();
        if( x>=startX && x<=startX+18*interval && y>=startY &&
            y<=startY+18*interval)
        {
            if((x-startX)>((x-startX)/interval)*interval+interval/2)
                selectX=((x-startX)/interval)+1;
            else
                selectX=((x-startX)/interval);

            if((y-startY)>((y-startY)/interval)*interval+interval/2)
                selectY=((y-startY)/interval)+1;
            else
                selectY=((y-startY)/interval);
            if(allChess[selectX][selectY]==0)
            {
                memory[count][0]=selectX;
                memory[count][1]=selectY;
                count++;
                allChess[selectX][selectY]=role;
                reshov[selectX][selectY]=role;
                System.out.println("POIs"+selectX+", "+selectY);
                NetTool.sendUDPBroadCast(fcf.Ip,
                    "POS,"+selectX+", "+selectY+", "+role);
                if(role==1)
                    message="轮到白方";
                else
                    message="轮到黑方";

                this.setClickable(false);

                boolean winFlag = this.checkWin();
                System.out.println(winFlag);
                if(winFlag==true)
                {
                    clear();
                    int result;
```

```

        if(allChess[memory[count-1][0]][memory[count-1][1]]==role)
        {
            result = JOptionPane.showConfirmDialog(this, "
游戏结束，你输了，要复盘吗?");
        }
        else
        {
            result = JOptionPane.showConfirmDialog(this, "
游戏结束，你赢了，要复盘吗?");
        }
        if(result==0)
        {
            reShow();
        }
        clear();
        JButton s=fcf.getStartButton();
        s.setText("Restart");
        s.setEnabled(true);
        canPlay=false;
    }
}
else
    JOptionPane.showMessageDialog(this,"当前位置已经有棋子
，请重新落子");

    this.repaint();
}
}
}

```

(三) 接收信息后的辨别过程

```

while(true)
{
    socket.receive(dp);
    InetAddress ia=dp.getAddress();
    String data=new String(dp.getData(),0,dp.getLength());
    if(data.isEmpty())
    {
        fp.setClickable(false);
    }
    else
    {
        System.out.println(data);
        String[] msg=data.split(",");
        System.out.println(msg[0]+" "+msg[1]);

        if(msg[0].equals("ready"))
        {
            enemyState="ready";
            start.setText("对方已经准备");
            fp.setClickable(true);

            System.out.println("-----");
            System.out.println("你的剩余时间:"+timeCount);
            System.out.println("-----");

```

```

    }

    else if(msg[0].equals("POS"))
    {
        System.out.println("接收坐标");

        fp.setCoord(Integer.parseInt(msg[1]),Integer.parseInt(msg[2]),Integer.p
arsetInt(msg[3]));

    }
    else if(msg[0].equals("back"))
    {
        int result=JOptionPane.showConfirmDialog(fp, "你同
意对方悔棋吗?");

        if(result==0)
        {
            fp.back2();
            NetTool.sendUDPBroadCast(Ip, "agree, ");
        }
        else if(result==1)
        {
            NetTool.sendUDPBroadCast(Ip, "disagree, ");
        }
    }
    else if(msg[0].equals("agree"))
    {
        JOptionPane.showMessageDialog(fp, "对方同意你悔棋
");

        fp.back2();
    }
    else if(msg[0].equals("disagree"))
    {
        JOptionPane.showMessageDialog(fp, "对方不同意你悔
棋");

    }
    else if(msg[0].equals("message"))
    {
        showArea.append("enemy: "+msg[1]+"\\n");
    }
    }
}
} catch (Exception e)
{
    e.printStackTrace();
}
}

```

(四) 悔棋的算法

```

void back()
{
    int result=JOptionPane.showConfirmDialog(this, "你确定要悔棋吗?");
    if(result==0)
    {
        NetTool.sendUDPBroadCast(fcf.getIp(),"back, ");
    }
}
void back2()
{

```

```

        allChess[memory[count-1][0]][memory[count-1][1]]=0;
        allChess[memory[count-2][0]][memory[count-2][1]]=0;
        repaint();
    }
}

```

(五) 复盘的算法

```

void reShow()
{
    clear();
    repaint();
    System.out.println(count);
    for(int i=0;i<count;i++)
    {
        allChess[memory[i][0]][memory[i][1]]=
        reshow[memory[i][0]][memory[i][1]];

        System.out.println("--"+memory[i][0]+" "+memory[i][1]+"
        "+allChess[memory[i][0]][memory[i][1]]);

        repaint();

        //间隔1秒后显示下一个棋子
        long startTime=System.currentTimeMillis();
        while(true)
        {
            long currentTime=System.currentTimeMillis();
            long time=currentTime-startTime;
            if(time>=1000)
            {
                break;
            }
        }
    }
    for(int i=1;i<19;i++)
    {
        for(int j=1;j<19;j++)
            System.out.println(allChess[i][j]);
    }
}

```

(六) 认输的算法

```

void surrender()
{
    int result=JOptionPane.showConfirmDialog(this, "你确定要认输吗?");
    surrender1(result);
}
//投降
void surrender1(int result)
{
    if(result==0)
    {
        NetTool.sendUDPBroadCast(fcf.getIp(), "surrender, ");
        if(role==1)
            JOptionPane.showMessageDialog(this, "黑方认输, 白方获胜");
        else
            JOptionPane.showMessageDialog(this, "白方认输, 黑方获胜");
        for(int i=0;i<19;i++)

```

```

        {
            for(int j=0;j<19;j++)
            {
                allChess[i][j]=0;
            }
        }
        setClickable(false);
        repaint();
    }
}

```

(七) 判断五子连珠的算法

```

private boolean checkWin()
{
    boolean[] flag = new boolean[4];
    for(int i=0;i<4;i++)
        flag[i]=false;
    int sign=0;
    sign=allChess[memory[count-1][0]][memory[count-1][1]];
    String nine="";
    String compare="";
    for(int i =0;i<5;i++)
        compare=compare+sign;

    //X轴方向
    for(int i=-4;i<=4;i++)
    {
        if((selectX+i)>=0 && (selectX+i)<=18)
            nine=nine+allChess[selectX+i][selectY];
    }
    if(nine.indexOf(compare)!=-1)
    {
        flag[0]=true;
    }
    System.out.println("X "+nine+" "+compare);
    nine="";

    //Y轴方向
    for(int i=-4;i<=4;i++)
    {
        if((selectY+i)>=0 && (selectY+i)<=18)
            nine=nine+allChess[selectX][selectY+i];
    }
    if(nine.indexOf(compare)!=-1)
    {
        flag[1]=true;
    }
    System.out.println("Y "+nine+" "+compare);
    nine="";

    //左偏45度
    for(int i=-4;i<=4;i++)
    {
        if(selectX+i>=0 && selectX+i<=18 && (selectY+i)>=0 &&
(selectY+i)<=18)
            nine=nine+allChess[selectX+i][selectY+i];
    }
}

```

```

    }
    if(nine.indexOf(compare)!=-1)
    {
        flag[2]=true;
    }
    System.out.println("45 "+nine+" "+compare);
    nine="";

    //右偏45度
    for(int i=-4;i<=4;i++)
    {
        if(selectX+i>=0 && selectX+i<=18 && (selectY-i)>=0 &&
(selectY-i)<=18)
            nine=nine+allChess[selectX+i][selectY-i];
    }
    if(nine.indexOf(compare)!=-1)
    {
        flag[3]=true;
    }
    System.out.println("135 "+nine+" "+compare);
    System.out.println(flag[3]);
    nine="";

    return flag[0]||flag[1]||flag[2]||flag[3];
}

```

(八) 按钮的监听器

```

public void actionPerformed(ActionEvent e)
{
    System.out.println(123);
    System.out.println(e.getSource()==start);
    if(e.getSource()==start)
    {
        if(start.getText().equals("Start"))
        {
            if(!ip.getText().isEmpty()&& ip.getText()!="不能为空")
            {
                Ip=ip.getText();
                start.setEnabled(false);
                start.setText("等待对方准备");
                NetTool.sendUDPBroadCast(Ip, "ready, ");
                if(enemyState=="ready")
                {
                    fp.setClickable(true);
                }
            }
        }
        else if(start.getText()=="Restart")
        {
            NetTool.sendUDPBroadCast(Ip, "ready, ");
            if(enemyState=="ready")
            {
                fp.setRole(1);
                fp.setClickable(true);
                start.setEnabled(false);
            }
        }
    }
}

```

```

    }
    else if(e.getSource()==back)
    {

        fp.back();

    }

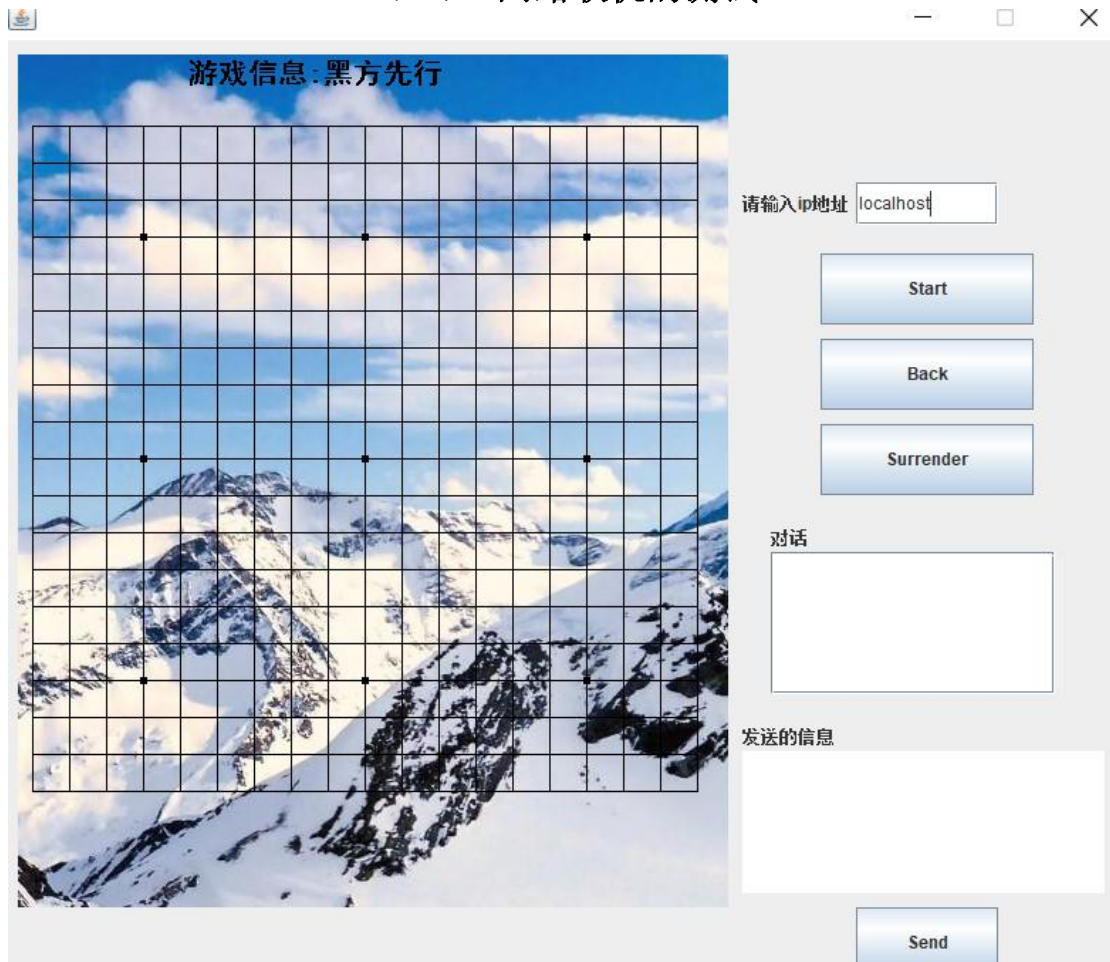
    else if(e.getSource()==surrender)
    {
        fp.surrender();
        start.setText("Restart");
        start.setEnabled(true);
    }

    else if(e.getSource()==send)
    {
        showArea.append("me: "+message.getText()+"\n");
        NetTool.sendUDPBroadCast(Ip,"message,"+message.getText());
        message.setText("");
        message.requestFocus();
    }
}

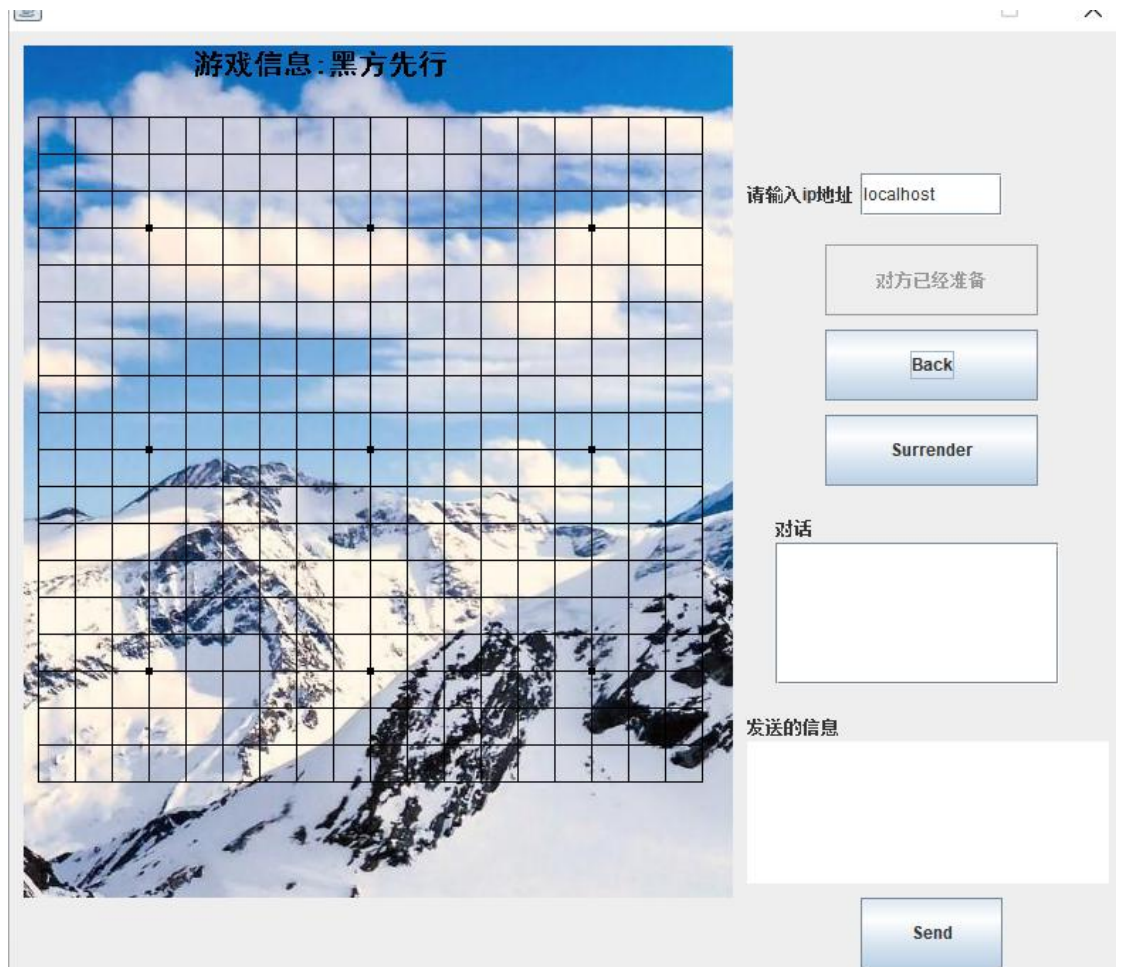
```

第六章 五子棋的测试

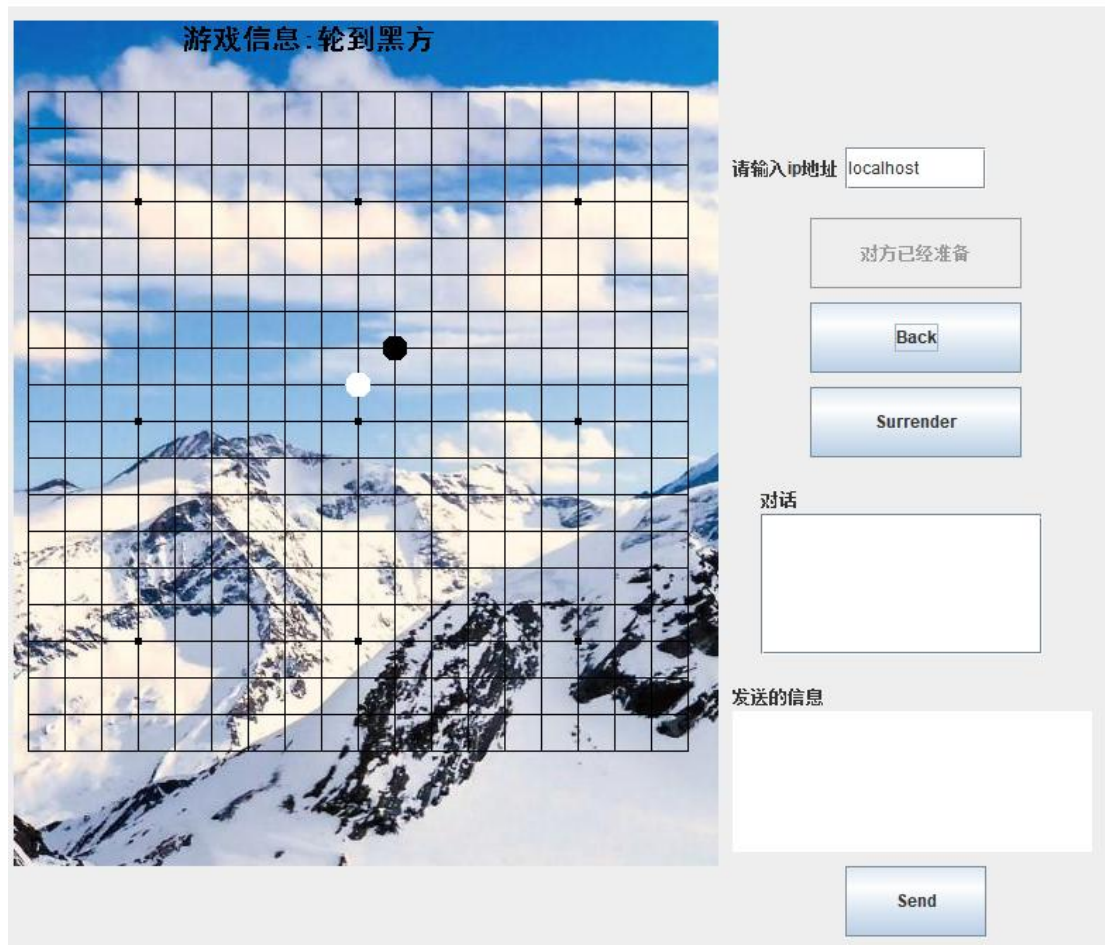
(一) 网路联机的测试



(二) 开局的测试

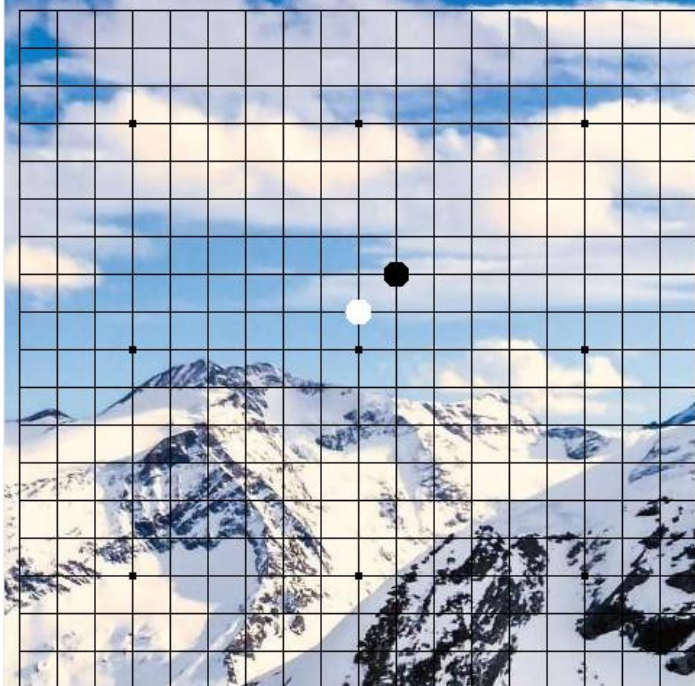


(三) 下棋的测试



(四) 聊天功能的测试

游戏信息: 轮到黑方



请输入ip地址

localhost

对方已经准备

Back

Surrender

对话

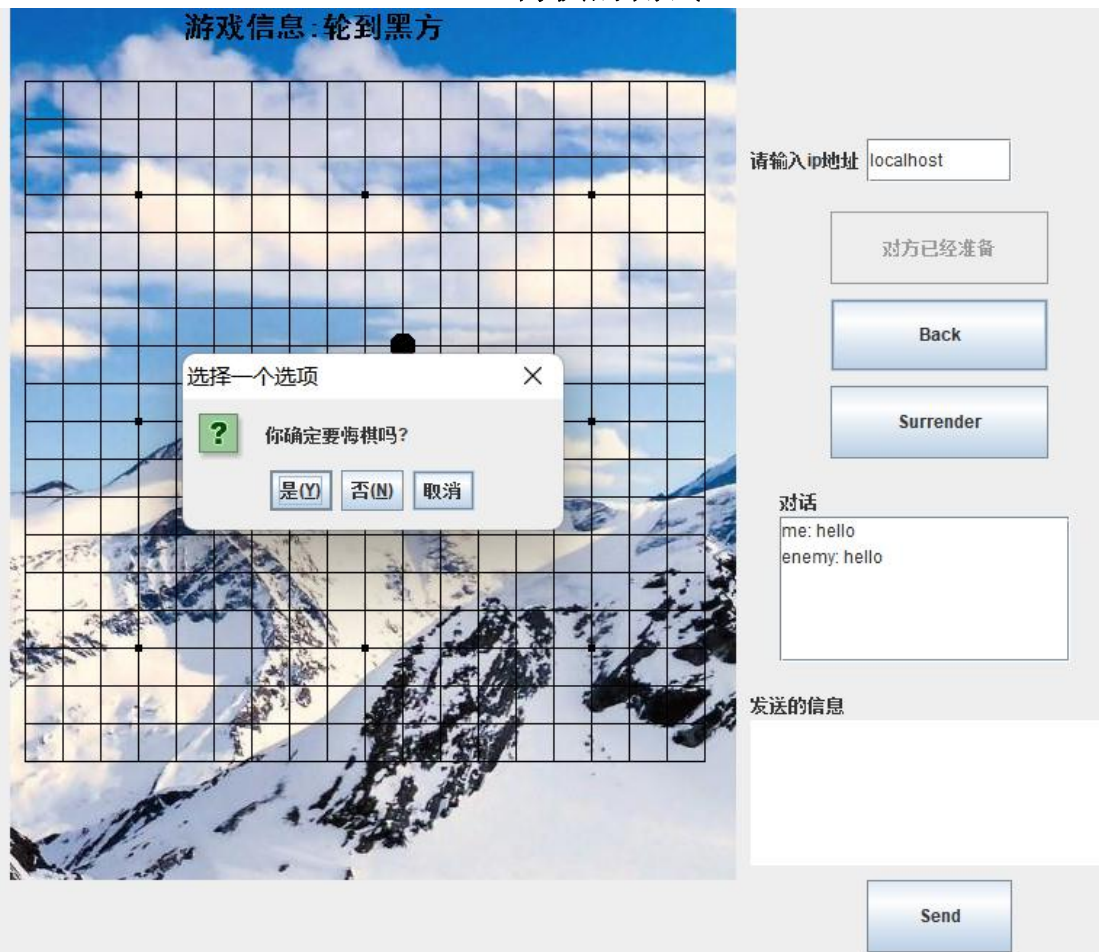
me: hello

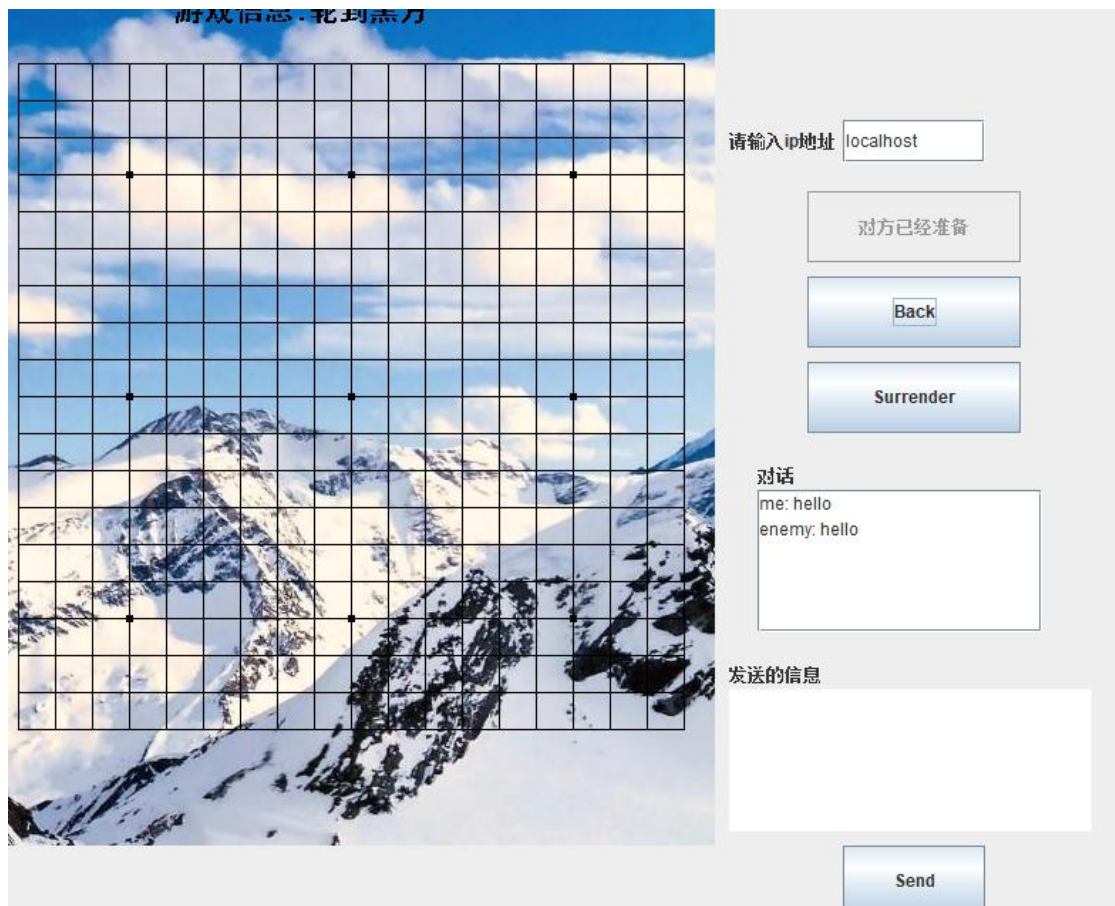
enemy: hello

发送的信息

Send

(五) 悔棋的测试






(六) 判断胜负(包含认输)

1. 认输



游戏信息: 轮到黑方

消息

 黑方认输, 白方获胜

确定

请输入ip地址

localhost

对方已经准备

Back

Surrender

对话

me: hello
enemy: hello

发送的信息

Send

2.五子连珠

请输入ip地址

对方已经准备

Back

Surrender

对话

me: hello
enemy: hello

发送的信息

Send

游戏信息: 轮到白方



选择一个选项



游戏结束，你赢了，要复盘吗？

是(Y)

否(N)

取消

请输入ip地址

localhost

对方已经准备

Back

Surrender

对话

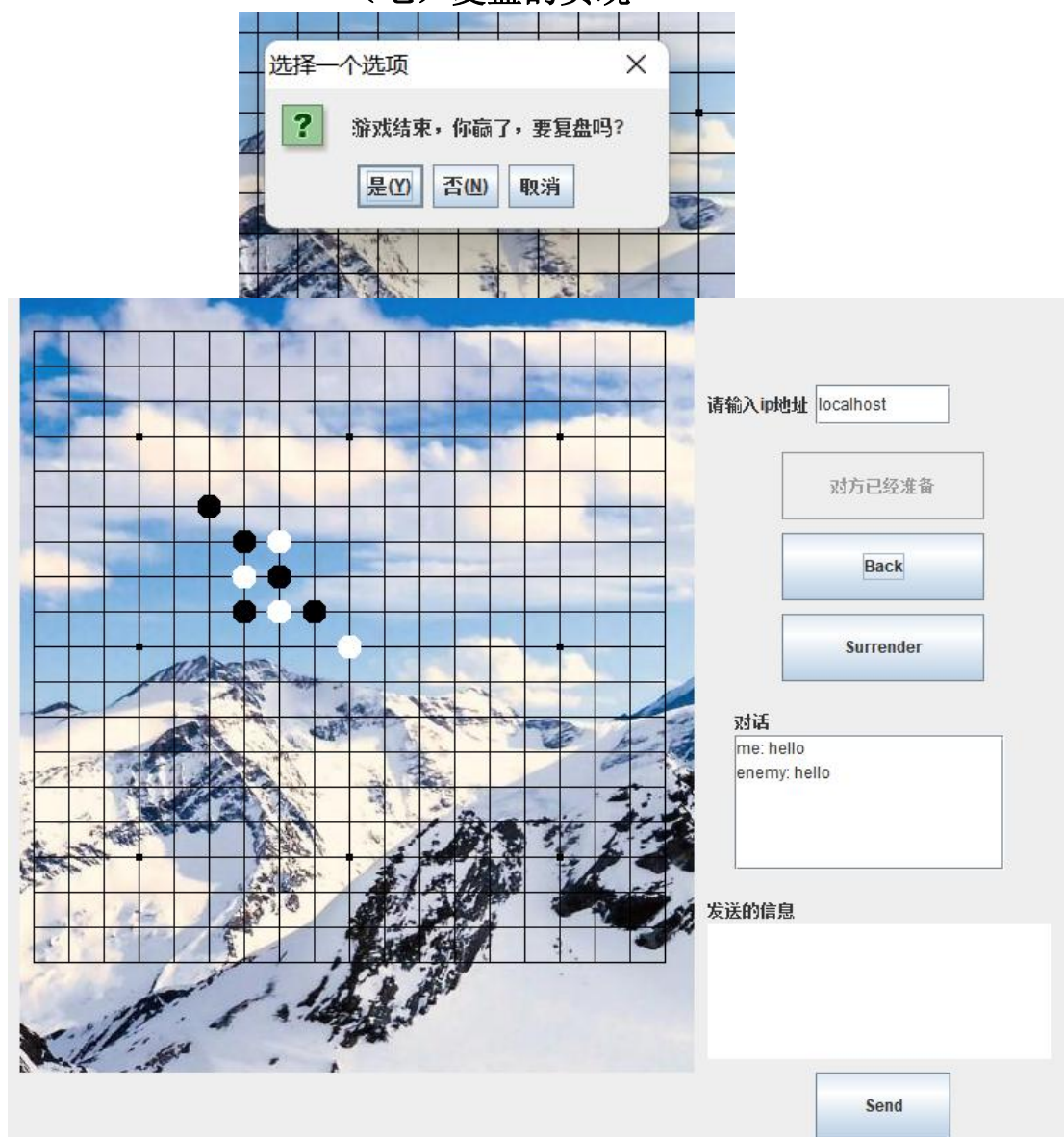
me: hello

enemy: hello

发送的信息

Send

（七）复盘的实现



第七章 总结与展望

（一）主要工作

这个用Java写的五子棋程序利用了网络、多线程、GUI和事件监听的机制基本上实现了网络连接，开局，下棋，悔棋，认输，判断输赢等基本功能。整个设计上仍有欠缺，功能较少，外观不是那么优美。通过写这个程序，我明白了提前设计的重要性，代码组织的重要性，因为这关系到日后的维护以及扩展。

在设计过程中，一开始也是觉得这个程序比较困难，然后去网络上搜索资源观看，发现他们都是先设计后开发，虽然我自己设计的算法可能比较慢，通过与其他人的算法进行比较，我发现了自己的问题。

（二）展望

尽管上述程序实现基本的一些功能，但是这个游戏仍然存在许多问题。

但在严肃的比赛中，整局游戏的时间不是无限制的，而是有限制的，所以后续应该再这个程序里面加上一个计时器，并规定好时间，当某一方思考的时间超过规定好的时间时，程序会自动提示“时间超时，自动认输”。

这个游戏在下棋时并没有音效，这与现实不符合，不能很好的拟合现实，所以后续应该在程序中加入下棋时的棋子与棋盘的碰撞音效。

黑子优先的优势太大，在这个程序中并没有对黑子进行限制，所以后续应该在程序中加入对黑子的限制。

程序中只有人人对战，而没有人机对战，而随着人工智能的发展，人机对战已经成为了必然，所以后续应该在程序中添加人机对战，并且设置难度分级，比如小白，入门，高手，大师，人工智能类似的分级，为这个程序提供多样化。

程序中应该设置游戏大厅，相当于指路的作用，以便玩家能够快速找到自己想要玩的内容，同样也应可以在游戏大厅里面加入聊天的功能。

参考文献

[1] 《2021年中国游戏产业报告》

[2] 北京青年报 2021 . 12 . 17