# 南开大学

2022学年度Java语言与应用大作业论文

中文题目：基于Java的五子棋的单机版和网络版设计

外文题目：Java-based Gomoku stand-alone and web version design

学号：2113285

姓名：吴静

专业：信息安全

学院：网络空间安全学院

指导教师：刘嘉欣

完成日期：2022年12月12日

摘 要

五子棋是学习java过程中十分典型的一个案例，几乎每个初学者都会尝试将他打出来，其中涉及到的诸多与java相关知识比如各类监听器的使用，各Java类（JFrame，JTextField，JTextArea，JLabel）的使用，以及对于MVC模式，线程的处理等都是java中比较经典且常用的方法，能够对我们学习Java起一个很好的促进作用。

本论文旨在阐述这一次打单机版五子棋和网络版五子棋所遇到的困难和一些心得体会，重点在于聊天消息的输送，下棋功能的实现，悔棋功能的实现，TCP的连接于监听，监听器的使用，图形化界面的使用。

关键字：Java；五子棋；网络版；聊天界面

**Abstract**

Gobang is a very typical case in the process of learning Java, almost every beginner will try to play him, which involves a lot of Java-related knowledge such as the use of various listeners, the use of Java classes (JFrame, JTextField, JTextArea, JLabel), and for MVC mode, thread processing, etc. are more classic and commonly used methods in Java. It can be a good promotion for us to learn Java.

This paper aims to explain the difficulties and some experiences encountered in playing the single-player version of Gobang and the online version of Gobang, focusing on the transmission of chat messages, the implementation of chess playing functions, the realization of repentant chess functions, TCP connection to monitoring, the use of listeners, and the use of graphical interfaces.

**Keywords：**Java；gobang；web version；chat interface

目 录

摘要………………………………………………………………Ⅰ

Abstract…………………………………………………………Ⅱ

目录………………………………………………………………Ⅲ

第一章 概述………………………………………………………1

1.1 目的……………………………………………………………1

1.2 介绍……………………………………………………………1

第二章 工具选型…………………………………………………2

2.1 Java语言………………………………………………………2

2.2 Java Swing包…………………………………………………2

2.2.1布局管理器………………………………………………2

2.2.1.1 BorderLayout……………………………………2

2.2.1.2 FlowLayout………………………………………3

2.2.1.3 GridLayout………………………………………3

2.2.1.4 BoxLayout…………………………………………4

2.2.2 基本组件……………………………………………4

2.2.2.1 JLabel……………………………………………5

2.2.2.2 JButton……………………………………………5

2.2.2.3 JRadioButton……………………………………6

2.2.2.4 JTextField………………………………………6

2.2.2.5 JTextArea………………………………………6

2.2.3其他……………………………………………………6

2.2.3.1 JFrame……………………………………………7

2.2.3.2 JPanel……………………………………………7

2.2.3.3 JOptionPanel……………………………………7

2.3 Java net包……………………………………………………8

2.3.1 ServerSocket……………………………………………8

2.3.2 Socket………………………………………………9

2.4 Java io包…………………………………………………10

2.4.1 BufferedReader………………………………………10

2.4.2 PrintWriter……………………………………………10

2.5 一些特别的函数………………………………………………11

2.5.1 substring()……………………………………………11

2.5.2 split()………………………………………………11

第三章 系统设计………………………………………………12

3.1 MVC模式的使用………………………………………………12

3.2 开始界面………………………………………………………12

3.3 单机版准备模式………………………………………………13

3.3.1 行列数的选择…………………………………………14

3.3.2 先手棋子颜色的选择…………………………………15

3.4 网络版准备模式……………………………………………16

3.4.1 选择“开房”还是“进入房间”………………………17

3.4.2 相应的选择功能……………………………………18

3.5 下棋界面………………………………………………19

3.5.1 棋盘实现………………………………………………20

3.5.2 功能键的实现…………………………………………21

3.5.3 聊天区的实现…………………………………………22

3.5.4 音乐播放界面…………………………………………23

3.6 游戏说明…………………………………………………24

第四章 系统实现与关键问题描述………………………………24

4.1 单机版五子棋系统实现………………………………………24

4.2 网络版五子棋系统实现………………………………………25

4.3 关键问题…………………………………………………25

4.3.1 实现网络版的聊天…………………………………25

4.3.2 悔棋…………………………………………………25

第五章 总结与展望………………………………………………25

5.1 总结………………………………………………………25

5.2 展望……………………………………………………25

致谢………………………………………………………………26

**第一章 概述**

**1.1 目的**

五子棋是一个老少皆宜的游戏，同时他作为初学java者不可多得的好的训练案例，这一次我将使用Java语言，在eclipse平台上做出一个简易版的五子棋游戏。

**1.2 介绍**

我做的这个简易版的五子棋首先会有开始界面，并且打算用开始界面来选择“单机版”（即双人模式），“网络版”（即联机模式）和“游戏介绍”。考虑到行列数不同五子棋难度不同，我这一次特地设置了“选择行列数”的环节，将行列数框定在“9~25”中间供用户选择；同时破除以往黑子先落的规则，我将棋子颜色也设置为可选择。而网络版模式则是要选择“开房间作为房主”还是“进入别人的房间”，如果选择“开房间”，就有选择行列数和棋子颜色的权力。在下棋阶段，五子棋要具备最基本的“悔棋”“和盘”功能，在网络模式阶段还应具有“聊天沟通”的功能，这则又需要用到线程与网络连接的知识，也是我在这一次常识中重点突破的内容。作为点睛之笔的音乐，也可以加在面板上面。

**第二章 工具选型**

**2.1 Java语言**

Java是20世纪90年代发展起来的一门编程语言，具有简单性，面向对象，分布性，编译和解释性，稳健性，安全性，可移植性，高性能，多线程性，动态性等特点；并且由编程语言，类文件格式，虚拟机，应用程序接口四个方面组成。

**2.2 Java Swing包**

Swing是一个为了Java涉及的工具包，包括了众多图形化界面。

**2.2.1 布局管理器**

能控制动态布局模式的是setLayout(LayoutManager )，常用的有五种布局模式，BorderLayout,BoxLayout,FlowLayout,GridLayout,GridBagLayout,由于我这一次只用上了这其中的四种，故在这里不阐述GridBagLayout。

**2.2.1.1 BorderLayout**

BorderLayout，俗称麻将布局管理器，他的布局模式大体分为NORTH,SOUTH,WEST,EAST,CENTER这五个方面，细分下来还有NORTHWEST,NORTHEAST,SOUTHWEST,SOUTHEAST这几个方面，可以将各个控件分别安在这些对应的位置上面；在我们将JFrame的大小变大变小后，SOUTH,NORTH的高度不变但是宽度会变，WEST,EAST的高度会变但是宽度不变，这也是我最常用的一种布局，一般来说，CENTER位置的都是放的最大的组件，但是它又可以根据各组件可能需要的空间来分配。

**2.2.1.2 FlowLayout**

FlowLayout，俗称流式布局管理器，它会将各组件按规定的排列方式排列，位置不够了自动转到下一行，构造函数有FlowLayout(),FlowLayout(int align,int hgap,int vgap)，其中align是对齐方式，有LEFT,RIGHT,CENTER三种方式，默认是居中对齐CENTER，hgap是组件水平间距，vgap是组件垂直间距，均默认5个像素。

**2.2.1.3 GridLayout**

GridLayout，俗称表格布局管理器，因为它会将各组件满满地平均分配至各个行列，构造函数有GridLayout(int rows,int cols)， GridLayout(int rows,int cols,int hgap,int vgap),先将一行放满再放下一行，缺点就是看起来太方正了，有点呆板。

**2.2.1.4 BoxLayout**

BoxLayout，俗称抽屉布局管理器，它可以横着摆放，也可以竖着摆放，组件的大小是由他每一行的字数决定的。

**2.2.2 基本组件**

**2.2.2.1 JLabel**

JLabel 对象可以显示文本、图像或同时显示二者。可以通过设置垂直和水平对齐方式，指定标签显示区中标签内容在何处对齐。默认情况下，标签在其显示区内垂直居中对齐。只显示文本的标签是开始边对齐；而只显示图像的标签则水平居中对齐。

常用的构造方法有JLabel()为无参构造函数,JLabel(Icon image)为创建具有指定图像的JLabel实例,JLabel(Icon imag, int horizontalAlignment)创建具有指定图像和水平对齐方式的JLabel实例,JLabel(String text)设置默认文本为text，JLabel(String text, Icon icon, int horizontalAlignment)设置了默认文本，默认图片，水平对齐方式,JLabel(String text,int horizontalAlignment)则设置了默认文本和水平对齐方式。

在常用方法上面，主要是设置文本函数setText(Sting text)和设置图片函数setIcon(Icon icon),设置内容在标签内的水平对齐方式setHotizontalTextPosition(int textPosition)和是垂直对齐方式setVerticalTextPosition(int textPosition)

**2.2.2.2 JButton**

JButton是一个按钮。常用的构造函数有JButton()创建一个按钮，JButton(String text)为创建一个按钮，其上名字为text，Jbutton(Icon icon)创建一个带图标icon的按钮。

常用方法有setEnable(Boolean enable)为设置是否可用，当按钮只能点一次时就可以调用这个函数，结果时当我们点了一次后这个按钮会变灰显示为不可用。

由于是按钮，所以我们经常与监听器连用，最常加的监听器是ActionListener，表示我们以任何方式（比如空格，回车，点击）触发这个按钮时这个按钮会有的反应。

**2.2.2.3 JRadioButton**

JRadioButton是一个单选按钮，由一个圆圈和文本组成，常见的构造函数有三种，可以分别选择是否有文本，是否被选中。

而常用的方法也很简单，首先是setText(String text)设置文本，setSelected(Boolean b)设置是否为选中状态，isSelected()判断是否被选中，setIcon(Icon icon）添加默认图片，setPressedIcon(Icon icon)添加选中状态时显示的图片，setDisabledICcon(Icon icon)设置不可用时选中的图片，setEnabled(Boolean enable)设置是否可以被选择。

同JButton一样，JRadioButton也需要添加监听器，但他常用的监听器则为ChangeListener(),可以在状态改变时监听到并作出相应反应。

**2.2.2.4 JTextField**

JTextField是一个专门用来编辑单行文本的文本框。其常见的构造函数有：JTextField()为无参构造函数，JTextField(String text),构造一个文本框，默认文本为text，JTextField(int columns)则为构造一个文本框，文本框内文字大小默认为columns，JTextField(String text,int columns)则为构造一个默认文本为text，文本大小默认为columns的单行文本框。

其常用函数有：

1.设置文本setText(String text)即将文本框内的内容赋值为text

2.获得文本框内的内容getText()

3.设置是否可以编辑setEditable(boolean b)当b为false时不可编辑，当b为true时可以编辑

4.设置文本内容的水平对齐方式setHorizontalAlignment()

5.设置文本内容的字体样式setFont(new Font())Font也是Java中的一个类，其常用构造函数是Font(String familyName,int style,int size), familyName是字体类型，style是字体风格，Java默认给出三种字体风格：“Font.BOLD”“Font.ITALIC”“Font.PLAIN”

**2.2.2.5 JTextArea**

JTextArea是一个专门用来编辑多行文本的文本框。其用法与JTextField机会一模一样，先来看构造函数，JTextArea(String text)表示默认文本为text；JTextArea(int rows, int columns)表示默认可见的行数和列数分别为rows和cols，JTextArea(String text, int rows, int columns)则同时设置了默认文本和默认可见行列数。

在常用方法上面，其与JTextField最大的区别就在于它可以设置是否自动换行，函数为setLineWrap(boolean wrap)，在不显性调用的情况下wrap默认为false；此外还有函数可以设置自动换行的方式setWrapStyleWord(boolean word) ，如果word为 true，则将在单词边界（空白）处换行反之则将在字符边界处换行，不显性调用的情况下word默认为false。

**2.2.3 其他**

**2.2.3.1 JFrame**

JFrame是屏幕上Windows的对象，能够实现“最大化”“最小化”“关闭”的功能；同时是一个容器，可以容纳许多组件，帮助程序员最终把界面呈现给用户。

常见的构造函数有两种：JFrame()为构造一个不可见的，没有标题的新窗体；JFrame(String title)则是构造一个标题为title的不可见的窗体。特别需要注意的是令JFrame组件可见还需要特别设置setVisible(true)，因为这个组件默认情况下是不可见的。

JFrame类中有几个常见的函数。

其中能控制其大小的有setBounds(int x,int y,int width,int height),setSize(int width,int height),setSize(new Dimension①(int width,int height))，其中x和y代表起始位置的横纵坐标，width和height代表JFrame这个矩形容器的长与宽。

在向其中添加组件时，由于JFrame是一个容器，他只有其中的内容面板可以容纳东西，所以我们调用其add()函数前应该先调用getContentPane()函数获得其内容面板后再添加组件。

**2.2.3.2 JPanel**

JPanel 是一个透明面板，他和JFrame十分类似，他们的布局模式都是用setLayout()，但不同的是在向其中添加组件时，JPanel不用调用getContentPanel()函数，可以直接添加。

**2.2.3.3 JOptionPanel**

JOptionPanel是Java中的对话框，主要有四种消息提示框方法，showConfirmDialog()确认对话框,showInputDialog()输入对话框,showMessageDialog()消息对话框,showOptionDialoge()选择对话框。也提供了五种不同的图标：ERROR\_MESSAGE，INFORMATION\_MESSAGE，WARNING\_MESSAGE， QUESTION\_MESSAGE，PLAIN\_MESSAGE

**2.3 Java net包**

Java net包中提供了两种常见的网络协议的支持，一个是TCP，一个是UDP。

TCP是一种面向连接的，可靠的，基于字节流的传输层通信协议，TCP层是位于 IP 层之上，应用层之下的中间层。TCP 保障了两个应用程序之间的可靠通信。通常用于互联网协议，被称 TCP / IP。

而UDP位于 OSI 模型的传输层。一个无连接的协议。提供了应用程序之间要发送数据的数据报。由于UDP缺乏可靠性且属于无连接协议，所以应用程序通常必须容许一些丢失、错误或重复的数据包。

在这一次的实践中我只用上了TCP，即Socket编程。套接字使用TCP提供了两台计算机之间的通信机制。 客户端程序创建一个套接字，并尝试连接服务器的套接字。当连接建立时，服务器会创建一个 Socket 对象。客户端和服务器现在可以通过对 Socket 对象的写入和读取来进行通信。Socket 类代表一个套接字，并且 ServerSocket 类为服务器程序提供了一种来监听客户端，并与他们建立连接的机制。具体方法为：

1. 服务器实例化一个 ServerSocket 对象，表示通过服务器上的端口通信。

2. 服务器调用 ServerSocket 类的 accept() 方法，该方法将一直等待，直到客户端连接到服务器上给定的端口。

3. 服务器正在等待时，一个客户端实例化一个 Socket 对象，指定服务器名称和端口号来请求连接。

4. Socket 类的构造函数试图将客户端连接到指定的服务器和端口号。如果通信被建立，则在客户端创建一个 Socket 对象能够与服务器进行通信。

5. 在服务器端，accept() 方法返回服务器上一个新的 socket 引用，该 socket 连接到客户端的 socket。

**2.3.1 ServerSocket**

服务器可以使用ServerSocket类创建一个端口并且侦听客户端的请求，常见的构造函数是ServerSocket(int port)可以创建绑定到特定端口的服务器套接字，最常用的方法则是accept()方法，可以监听Socket对象

**2.3.2 Socket**

Socket 类代表客户端和服务器都用来互相沟通的套接字。客户端要获取一个 Socket 对象通过实例化 ，而服务器获得一个 Socket 对象则通过 accept() 方法的返回值。常见的构造函数有Socket(String host,int port)，或者说Socket(InetAddress host,int port)都是连接到指定主机或者指定ip地址的指定端口号当 Socket 构造方法返回，并没有简单的实例化了一个 Socket 对象，它实际上会尝试连接到指定的服务器和端口。

关于常用的一些方法，首先是getInetAddress()返回套接字的地址；getPort()返回连接到的远程端口,getLocalPort()返回绑定的本地端口,getRemoteSocketAddress()返回连接的端点的地址，未连接则返回null,getInputStream()属于输入输出流中的输入流，getOutputStream()则为输入输出流中的输出流,close()则可以关闭此套接字。

值得注意的是其中的getInetAddress()可以访问ip地址，进而得到其他对象信息，比如getHostName()返回ip地址主机名,getHostAddress()返回ip地址字符串,getLocalHost()返回本地主机。

**2.4 Java io包**

**2.4.1 BufferedReader**

BufferedReader是从缓冲区之中读取内容，所有的输入的字节数据都将放在缓冲区之中。构造函数是BufferedReader(Reader in)，常用的方法是readLine()，一次性从缓冲区中将内容全部读取进来。

在这一次实践中用的主要是new BufferedReader(new InputStreamReader(socket.getInputStream()))，InputStreamReader可以将字节流变成字符流。调用这个函数之后就可以给键盘输入数据的标准格式。

**2.4.2 PrintWriter**

PrintWriter也属于输出流，他只能写数据，不能读数据；可以针对文件直接进行操作；同时它还具备自动刷新功能，可以在构造函数中得到体现printWriter(OutputStream out/Writer out,boolean autoflush)这第二个参数就是是否自动刷新，赋值为true即可。

其他常用方法有打印函数println(String text)，如果没有设定自动刷新时还可能会用上flush()函数手动刷新。

**2.5 一些特别的函数**

在这一次实践中，我还用上了以前从没有用过的一些函数，但是由于十分好用，并且我觉得在以后打代码过程中也有可能用到于是在这里也记录下来。

**2.5.1 substring()**

Substring()函数可以很方便地用来切割函数，提取我们需要的信息。它有两种构造函数，一种构造函数的参数只有beginIndex，即起始位置的索引，可以返回索引为beginIndex后的字符串；另一种是同时具有beginIndex和endIndex的构造函数，他可以提取索引值位于beginIndex和endIndex中间的字符串。

**2.5.2 split()**

split()方法可以根据匹配给定的正则表达式来拆分字符串，构造函数为split(String regex，int limit)，regex表示正则表达式分隔符，limit则是分割的份数，这一项可以省略不写，默认为1。

**第三章 系统设计**

**3.1 MVC模式的使用**

MVC是指“Model”，“View”，“Controller”三个板块的使用，即模型，界面，和控制台，可以将应用程序的输入，处理和输出分开，大大提高程序的效率。View作为视图，是一种输出数据并允许用户操纵的方式；Model拥有最多的处理任务，模型返回的数据都是中立的，与数据格式无关，能为多个视图提供数据；Controller则接受用户的输入并调用模型和视图去完成用户的请求。简单来说，View和用户连接，Model和计算机算法相关，而Controller则将两者有机地联合在一起。

在这一次实践中，我主要将下棋部分应用到MVC模式中，所以Model中肯定会有一些棋子变量，比如说棋子的颜色，棋盘的存储，判断输赢等；Controller中则应该有各种下棋的函数（和Model连接），同时控制棋盘的重绘函数（和用户端连接）等一系列功能；View则要能显示下棋界面。

**3.2 开始界面**

开始界面最基本的组件应该有三个按钮JButton，也要有一个JLabel显示“五子棋”三个字，基于此，我们可以设计我们得到的页面布局。但是考虑到JButton上面要添加监听器，写起来可能会有点长，于是把这三个按钮组件放在另一个类中，用setFont(),setLayout()等函数完善界面，这就是界面的大概思路。

在按下“双人模式”后，应该单机版的准备模式；而在按下“网络模式”后应该进入网络版五子棋的准备模式；在按下“游戏说明”后则应该跳转至游戏说明的界面，这其中会有开始界面的关闭，使用close()即可。

**3.3 单机版准备模式**

有两个选择，一个是选择行列值，一个是确定先手棋子的颜色，这里我用了两个JOptionPanel，一个是showInputDialog(),一个是showOptionDialog(),在两个都完成选择后进入下棋界面。

**3.3.1 行列数的选择**

利用showInputDialog()提示用户输入行列值，由于五子棋棋盘不能太大也不能太小，所以要先经过一个对输入数据大小的判断，大于25或者小于9都会提示重新输入，如果满足要求，那么所输入的数据则会传入Model中保存，便于之后绘图以及算法使用。

**3.3.2 先手棋子颜色的选择**

在选择完行列值之后，就会进入先手棋子颜色的选择。有两个选项：黑色和白色，点击确认后所选择的颜色会传入Controller中的localColor中作为先手棋子颜色

**3.4 网络版准备模式**

同样是有两个选择，一个是“开房还是进入别人房间”，一个是行列数的选择和棋子颜色的选择（和单机模式一样）。

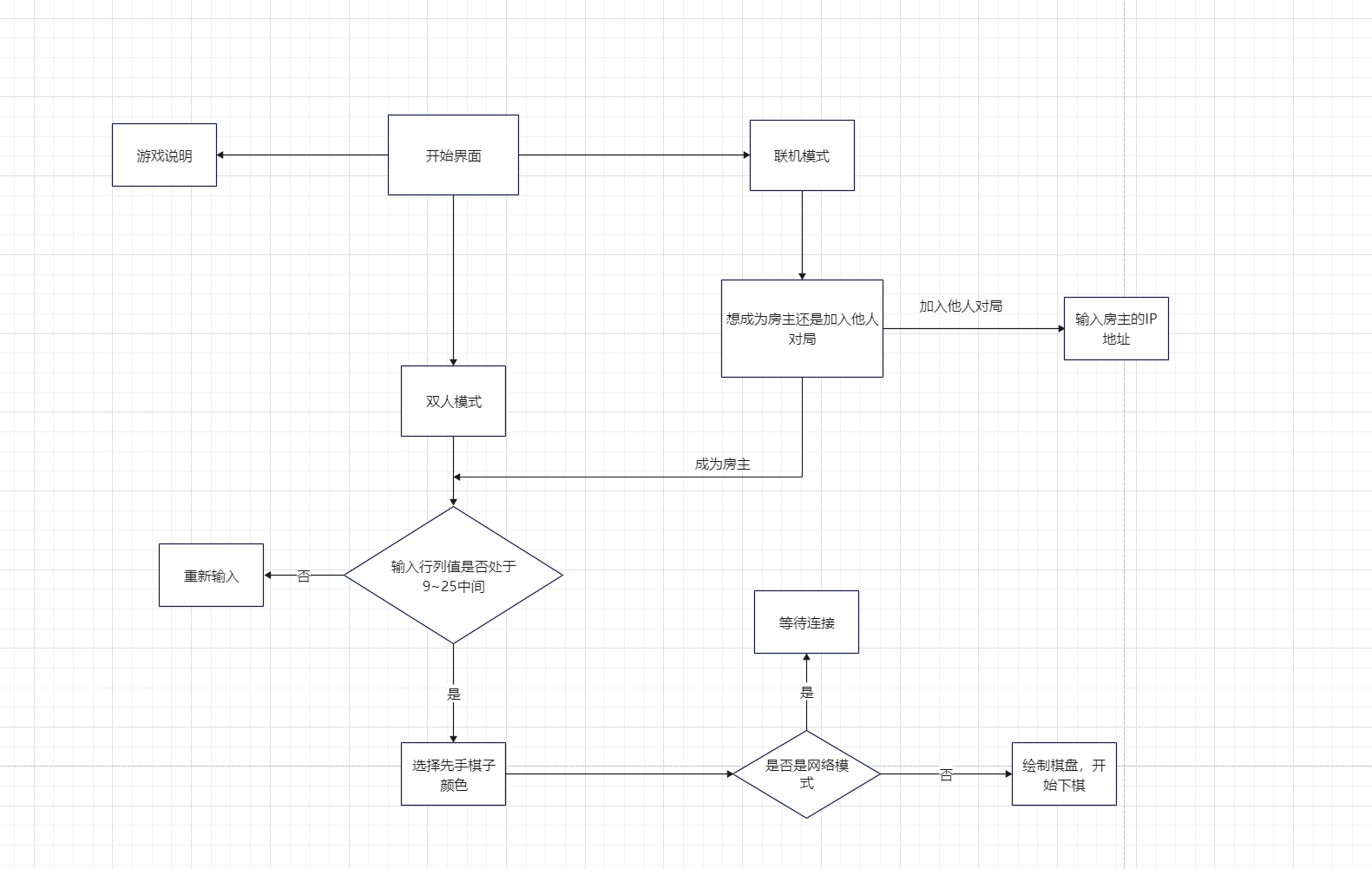
**3.4.1 选择“开房”还是“进入房间”**

也是利用JOptionPanel中的showOptionDialog供用户选择，此时的“开房”和“不开房”对应着“服务器”还是“客户端”，当用户选择了“开房”就应该创建一个ServerSocket对象开始监听，我给定的端口是8090，而如果用户选择了“进入别人房间”，那么我们应该实例化一个Socket来帮助连接，同样，考虑到ServerSocket的ip地址不一定是Socket端的本机地址，于是特地加了一个选择ip地址的对话框，不输入则默认本机地址，输入则会创建一个Socket尝试连接该ip地址。

**3.4.2 相应的选择功能**

在做好前面的选择后，此时房主也会对应几个选择，都是和单机版五子棋模式一样的选择，于是将单机版五子棋的选择抽象成函数，直接调用即可。

这里还存在一个问题，房客进入后是不存在选择行列数和先手棋子的环节的，那如何能让房主选择的行列数和先手棋子颜色这些信息传给房客让他也绘图呢？我想到的是在连接成功后利用输入输出流将房主的localColor和WIDTH传给房客，房客再利用这些信息进行绘图。



**3.5 下棋界面**

**3.5.1 棋盘实现**

棋盘是整个五子棋最核心的一部分，他要包括棋子，要在每一次下棋成功后都更新一次棋盘，所作的工作很多，于是我们抽象一个chessPanel类继承至JPanel，专门用来记录棋盘，重绘棋盘以及监听用户的点击事件。

需要注意的是，由于我们用的是动态布局，同时窗口大小可以变大变小，所以棋盘的位置不是固定的，我们需要让他根据窗口的大小实时变化棋盘的大小。正巧有一个监听器ComponentAdapter可以检测窗口大小，在监听后得到一系列参数然后调用repaint()函数即可实现棋盘随着窗口大小变化而变化，也能画出大小适当的棋盘。

另一个方面，棋盘还需要监听我们鼠标点击的位置，算出棋子的行列值，并传给Model的data数组保存，在每一次重绘时就可以循环一遍data数组进行“下棋”功能，总结来说就是View监听用户鼠标点击的位置，通过Controller的数据处理（能否下棋？棋子颜色？）等再传入Model，View根据Model的数据进行重绘操作，完成“下棋”步骤。

**3.5.2 功能键的实现**

五子棋最基本的功能有悔棋，和棋，网络版的五子棋还有聊天功能。这里先说悔棋和和棋，要实现悔棋功能，首先得记录每一次下棋时棋子的位置，可以用一个链表chess存储于Model中，当下棋成功后记录这一次下棋成功的位置（行和列）。当需要悔棋并且满足悔棋条件的时候，就向chess中查找上一次的落点是哪里，取出上一次的落点的行列值，在data中将行列值为该落点的行列值的棋子设置为SPACE，再调用repaint()函数实现悔棋功能。

和棋，在单机版中只需要将Model中的data数组循环遍历，将所有值都设为SPACE，再调用repaint()即可，而在网络版五子棋时还需要向对方传一个信息，征求对方的同意，如果对方同意，则清空两方的棋盘，不同意则不执行。

**3.5.3 聊天区的实现**

聊天区是网络版五子棋很重要的一个部分，需要有发送信息界面，信息窗口，和一个JButton按钮，这时候需要用到TCP网络协议，如前文所说，需要两者交换信息的地方有很多，所以我特地新建了一个NetHelper用来管理服务端和客户端的信息交流与传递。

首先是beginListen()函数，其中要有监听，创建输入输出流以及读取客户端发来的消息；其次是connect()函数，其中要有连接，创建输入输出流和接受客户端发来的消息两个功能。因为“接受消息”这一个功能都是共有的，所以我们能再次将他抽象出来成一个函数startReadLine();

现在开始思考接收函数startReadLine()需要完成哪些功能，根据前文论述，首先要有接受对方发来的聊天消息，要能接受服务器发来的行列值和先手棋子颜色（对于客户端来说）要能接受对方落子的行列值，还要能接受对方和棋请求的消息……由于这些都是通过一个输入输出流传递的，所以我们要能很好地区分他们分别对应哪一类消息，这时候很自然地想到String的标记，可以在消息头加上每一类消息对应的字符表示比如“PutChess”，那么如何做到消息的识别呢，String中有一个函数叫做startsWith(),可以用来判断消息头的开头是否以某一个字符串开头，那么这个问题就迎刃而解了。在发消息的同时给消息做一定的标记，在收消息的同时对消息做出选择，输送到对应的消息处理机制中，此时再将给予String的标志用substring函数清除即可得到原始信息，然后在做相应的处理即可。

拿输送聊天消息举例子，在获取了每一条要发送的消息（String）后，传入输出流时再字符串头加上“Chat”，对方接受消息的时候，发现这属于聊天信息，于是输送到聊天信息的处理函数，执行相应的操作：在聊天框中打出相应的字符，信息输送完成。

还有一个很重要的点，由于下棋和聊天是两个独立的功能，不能因为一个功能没有完成就影响了另一个功能，所以在这里最好添加一个新线程，使两个线程独立运行，就可以避免界面无法操作的情况。

**3.5.4 音乐播放界面**

音乐的播放需要使用AudioInputStream类，AudioSystem类以及AudioFormat类。

1.首先导入相应的音乐文件（wav格式）

2.定义一个AudioInputStream用于接收输入的音频数据

3.使用AudioSystem来获取音频的音频输入流

4.使用AudioFormat来获取AudioInputStream的格式

5.获取受数据行支持的音频格式 DataLine.info

6.获取与上面类型相匹配的行并且打开具有指定格式的行

7.写数据，从音频流读取指定的最大数量的数据字节，并放入给定的字节数组中

8.如果要实现循环播放，则可以在一个线程里面加上while循环即可

**3.6 游戏说明**

游戏说明就只有一个简单的页面，由于没有什么可操作性，为了让界面变得更美观，我在手机上先做了一个图片，打算直接把图片插入进去，于是这里就涉及JFrame的背景插入。

我们直到JFrame只是一个容器，其上有很多内容面板，但是有不同的层次，最常用的方法是借用JLayeredPanel类，这是一个可以用于分层的类。

1.创建JLable，ImageIcon，JPanel将图片导入ImageIcon的一个实例，添加至JLable的实例中，再添加至JPanel的实例中。

2. 由于JLayeredPanel没有layoutManager所以必须完成设置setBounds()函数对每个添加到JLayeredPanel的对象，否则会导致无法显示该组件。，所以我们设置JPanel实例的大小（或者位置），以及各个组件的大小。

3.调用setLayeredPane()函数添加JLayeredPanel实例，再调用JLayeredPanel的add函数添加包含图片的JPanel和各个组件，同时把图片放在最底层（JLayeredPane.DEFAULT\_LAYER），将其他各组件放在高一层的地方（JLayeredPane.MODAL\_LAYER），将JFrame设置为可见即可。

**第四章 系统实现与关键问题描述**

**4.1 单机版五子棋系统实现**

对于开始界面，中间的选择过程主要就是对于布局方法和各类组件的运用，在开始界面和下棋界面我都使用了BorderLayout的布局方法，在一些小界面的组合上面，常用FlowLayout和GridLayout。

如下图（1）是我的开始界面。

*五子棋*

游戏说明

联网挑战

双人模式

图（1）

而对于下棋界面来说，界面主体也是用的BorderLayout来完成，只是各个小组件内部的布局方式会比开始界面复杂很多。

如下图（2）是我的下棋界面，聊天界面是只有网络模式才有的，单机版五子棋没有这样的界面。

音乐播放部分

棋盘

聊天信息显示的界面

和棋

悔棋

信息发送窗口

图（2）

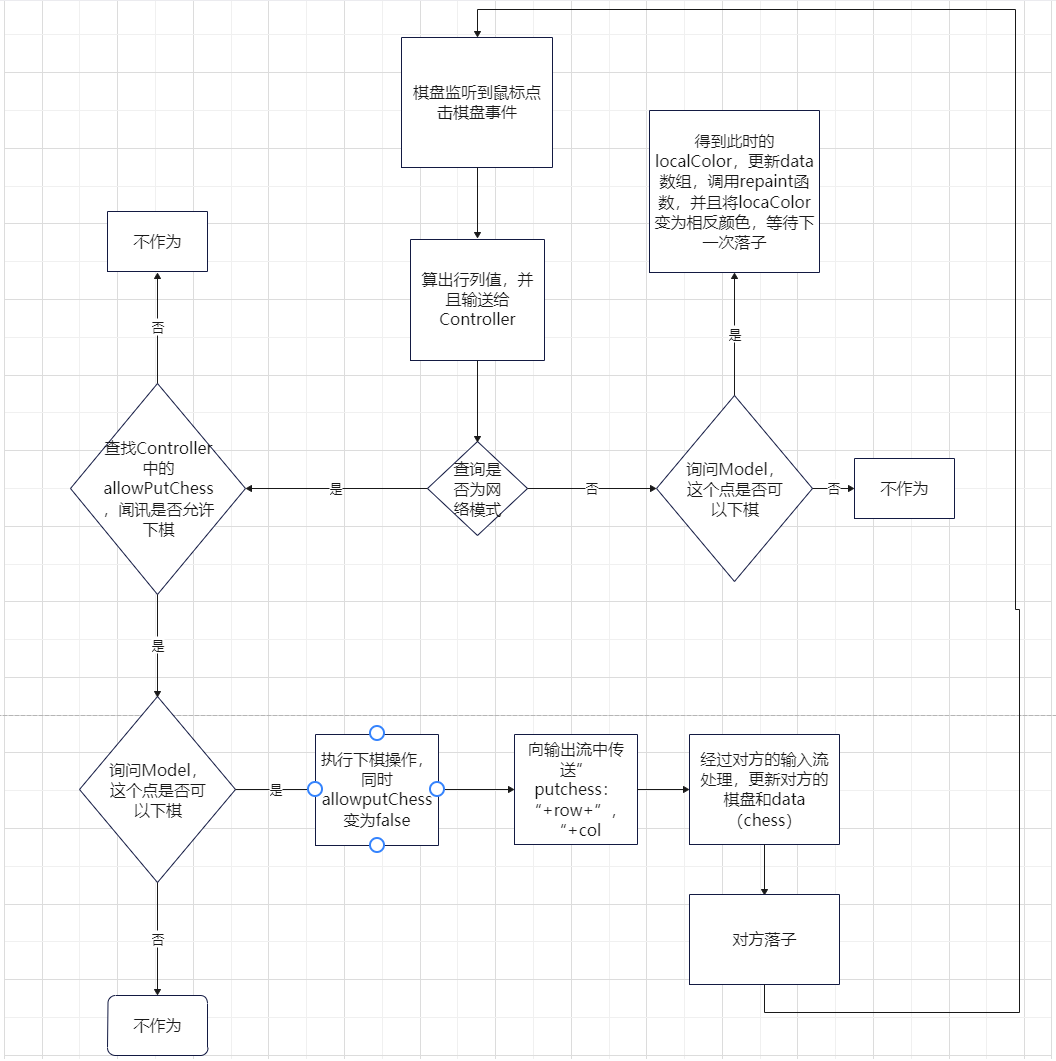
提示信息JLabel

发送按钮

信息输入窗口

使用了一个空的组件来提高高度

另外就是向Model，Controller中设置相应的变量。将棋子颜色设置于Model中，使用int型将SPACE,BLACK,WHITE分别赋值为0，+1和-1，这样在下完棋子后就方便给转换颜色。将WIDTH也设置在Model中，因为很多函数都要用上，放在Model里面具有良好的系统性。而Controller里面则主要负责调用Model中的各个变量对View发回来的消息做出响应。



图（3）

**4.2 网络版五子棋系统实现**

与单机版相比，网络版多了传达信息的过程，这一点我在系统设计里面有详细说明，可以再补充的一点是，关于线程的问题，比如说在accept()函数调用的时候没有收到及时的回应，这个主线程就会变成阻塞状态，所以我们在accept的时候加上一个新线程是很必要的，同时在聊天接收消息，在接受对方落子的行列值的时候都应该加上相应的线程，防止主线程阻塞。

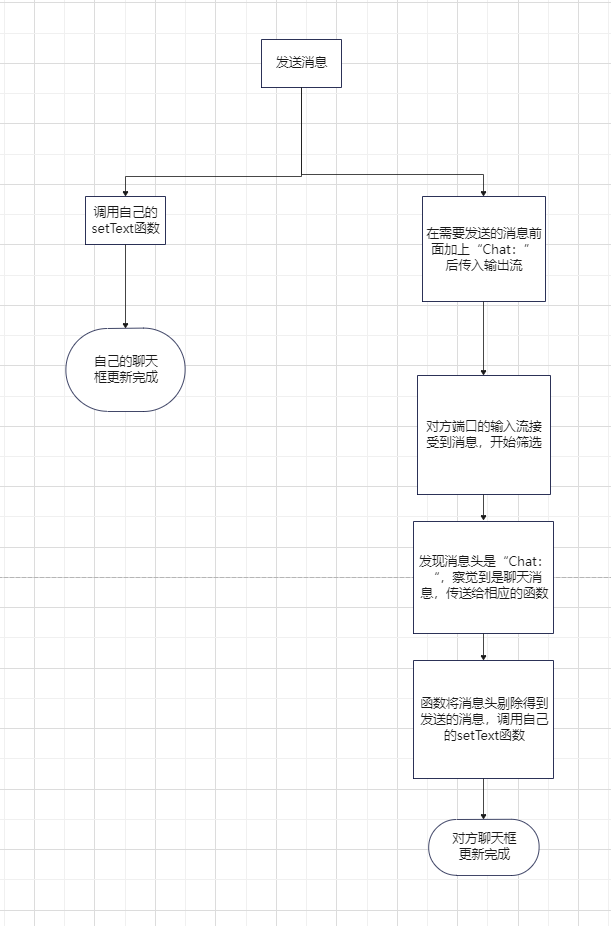
另一方面，如果需要一个服务器连接多个客户端，可以在accept()外面加一个循环，不断连接是都有新的客户端加入，同时定义一个Socket数组，保存每一个时候接收到的客户端的Socket变量，这样就可以在其他时候也给对应的客户端发送消息以及接受相应客户端的消息了，由于这一次我做的是针对双方的五子棋联机版，所以在这一次的事实践中没有体现出来。

**4.3 关键问题**

对于这一次的实践，最难的就是联机部分，因为要处理相应的客户端与服务器的连接与监听，消息的传送与接受功能。

**4.3.1 实现网络版的聊天**

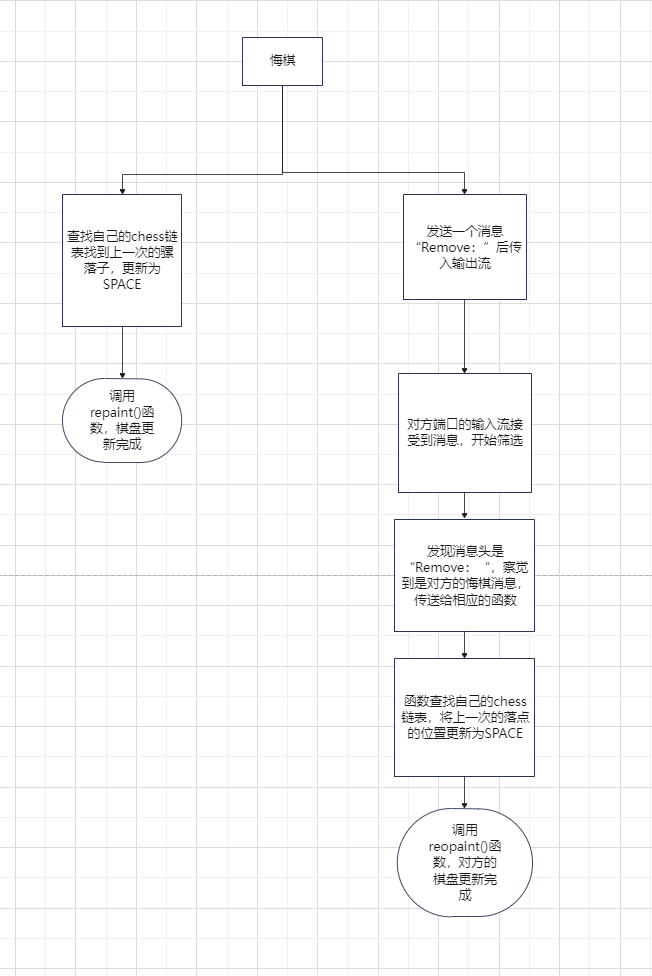
这一块的难点主要在于如何传送消息至双方的聊天版，让对方也看得见。对于自己的聊天版来说，直接调用自己的JTextArea中的setText函数，将其更新为原先的text加上我最新输入的text（中间可能有名字和换行）；而对于对方的聊天版来说，首先，对方要接受到我们的消息就得通过输入输出流和TCP协议传送，先将我们要发送的信息前面加上“Chat：”封装一下，然后传入输出流，待对方的输入流函数接受到之后，识别到字符串开头是“Chat：”，意识到这是个聊天信息，于是传入处理聊天信息的parseChat()中，在parseChat()中先用substring函数剔除掉我们的封装，得到我们输出的消息，再调用自己的setText函数将JTextArea中的内容更新，完成数据的输送。具体流程见图（4）



图（4）

**4.3.2 悔棋**

同聊天一样，悔棋部分在单机版中可以直接查找上一次的落点，将上一次落点的值设为SPACE即可；但网络版呢，网络版的悔棋还要将自己悔的棋子变成SPACE，后输送给对方，再次同时保持“可落子”状态，再下一个棋子。在自己的面板上面悔棋并不难，那么如何告诉对方却是个问题。我们是需要传送棋子的行列值吗，其实我们可以直接输送一个消息，叫做“我要悔棋”的消息，因为双方都存在一个链表chess记录着每一次的落点，只要对方得到了这个消息，可以利用自己的chess链表找出上一次的落点，变为SPACE即可，具体过程与输送消息十分类似，这里也不过多阐述。



图（5）

## 第五章 总结与展望

**5.1 总结**

这一次的五子棋，我学到了很多，不仅是各种Java组件，Java类的运用，还有各种思考方法，比如MVC模式的思考方式，比如每一个类之间的连接，最最重要的是我还学习了输入输出流，以及在自己的程序中加入图片，音乐，还有利用网络协议达到两台或多台电脑的连接，同时在查阅资料的过程中，也学到了很多函数，同时，也对于命令行指令有了一定的认识，比如查网络ip地址可以输入ipconfig/all，查询连接状态可以输入netstat -na等等。

**5.2 展望**

虽然这一次也有很用心地在做这一个可以联机和单机的五子棋，但是还是有很多不足，比如所没有做到输送语音，没有做到网络大厅，多人观战，还有一只很想做的在JFrame中插入图片北京也没有很好地完成，以及虽然每一个函数写前都有认真地思考过，还是会觉得有更简单的方法……这一系列的问题我都将在之后的学习生活中逐渐完善学习。

致 谢

首先当然要感谢我的刘嘉欣老师，在各种方面他都让我受益匪浅。曾经有一次绘图作业，也是很荣幸地得到了点拨，让我知道我的问题出在了哪里，平时上课也是言简意赅，一语中的，能让我们学到很多东西。

同时当然也要感谢我的室友，谢谢他在我每一次打不出来的时候都能不厌其烦地帮我看问题出在哪里，帮我debug，为我解答我很多奇怪的问题。