

南 开 大 学

JAVA 语 言 程 序 设 计 课 程 作 业

中文题目： 基于 JAVA 的五子棋小游戏

外文题目： JAVA-based Gobang game

学 号： 2311983

姓 名： 余辰民

年 级： 2023 级

学 院： 网络空间安全学院

专 业： 信息安全、法学双学位班

完成日期： 2024 年 12 月

摘 要

在 JAVA 中采用 MVC 思想设计了一个具有联机对战功能的五子棋网络对战游戏，具备五子棋对战的基本功能。如落子、判断胜负；通过客户端连接服务器，实现用户的远程登录、游戏大厅、选座落座、房间聊天等多种功能。

关键词：五子棋；JAVA；网络对战

Abstract

In JAVA using the MVC ideology to design a Gobang network game with online battle function, with the basic functions of Gobang battle. Such as drop, judge winners and losers; through the client to connect to the server, to realize the user's remote login, the game lobby, selecting the seat drop, room chat and other functions.

Key Words: Gobang;JAVA;network battle

目 录

摘要	I
Abstract	II
目录	III
第一章 概述	1
第一节 前言	1
第二节 背景介绍	1
1.2.1 MVC 模式	1
1.2.2 五子棋规则	3
第二章 技术选型	4
第一节 选用 JAVA 的理由	4
2.1.1 JAVA 语言特性	4
2.1.2 C++ 语言特性	5
2.1.3 选用 Java 语言开发五子棋小游戏的原因	5
第二节 选用 Eclipse 的理由	6
2.2.1 Eclipse 的特性	6
2.2.2 IntelliJ IDEA 的特性	6
2.2.3 选择 Eclipse 的开发五子棋小游戏的原因	6
第三章 系统设计与实现	7
第一节 用户登录界面	7
3.1.1 设计思路	7
3.1.2 布局实现方式	8
3.1.3 具体逻辑和跳转处理	8
第二节 房间界面	10
3.2.1 设计思路	10
3.2.2 布局实现方式	10
3.2.3 具体逻辑和传输信息	12
第三节 游戏界面	13
3.3.1 设计思路	13

3.3.2 聊天交互实现	14
3.3.3 下棋功能的实现与胜负判断	15
第四章 总结与展望	17
参考文献	18
致 谢	XIX

第一章 概述

第一节 前言

随着互联网技术的飞速发展，网络游戏已成为人们休闲娱乐的重要方式之一。五子棋作为一种经典的棋类游戏，以其简单易懂的规则和策略性玩法，深受广大玩家的喜爱。传统的五子棋游戏主要局限于面对面的对局模式，然而，在当今数字化时代，人们对于能够跨越地域限制，与远方的朋友或其他玩家进行实时对战的需求日益增长。

本研究旨在利用 JAVA 编程语言开发一款具有联机对战功能的五子棋网络对战游戏，旨在打破传统游戏的空间限制，为玩家提供更加便捷、多样化的游戏体验。游戏具备五子棋对战的基本功能，包括棋子的落子操作以及准确的胜负判断机制，以确保游戏的公平性和竞技性；通过客户端与服务器的有效连接，实现了一系列丰富的附加功能。用户能够进行远程登录，轻松进入游戏世界；在游戏大厅中，玩家可以直观地查看各个游戏房间的状态，自由选择心仪的座位，实现快速入座，准备展开激烈的对战；此外，为了增强玩家之间的互动性和社交性，游戏还特别设计了房间聊天功能，让玩家在紧张的对战之余，能够与对手或其他房间内的玩家进行实时交流，分享游戏心得、交流生活趣事，营造出更加活跃、有趣的游戏氛围。

第二节 背景介绍

1.2.1 MVC 模式

经典 MVC 模式中，M 是指模型，V 是视图，C 则是控制器，使用 MVC 的目的是将 M 和 V 的实现代码分离，从而使同一个程序可以使用不同的表现形式。其中，View 的定义比较清晰，就是用户界面。MVC 开始是存在于桌面程序

中的，M 是指业务模型，V 是指用户界面，C 则是控制器，使用 MVC 的目的是将 M 和 V 的实现代码分离，从而使同一个程序可以使用不同的表现形式。

比如一批统计数据可以分别用柱状图、饼图来表示。C 存在的目的则是确保 M 和 V 的同步，一旦 M 改变，V 应该同步更新。模型—视图—控制器（MVC）是 Xerox PARC 在二十世纪八十年代为编程语言 Smalltalk — 80 发明的一种软件设计模式，已被广泛使用。后来被推荐为 Oracle 旗下 Sun 公司 Java EE 平台的设计模式，并且受到越来越多的使用 ColdFusion 和 PHP 的开发者的欢迎。模型—视图—控制器模式是一个有用的工具箱，它有很多好处，但也有一些缺点。

M 即 model 模型是指模型表示业务规则。在 MVC 的三个部件中，模型拥有最多的处理任务。被模型返回的数据是中立的，模型与数据格式无关，这样一个模型能为多个视图提供数据，由于应用于模型的代码只需写一次就可以被多个视图重用，所以减少了代码的重复性。^[1]

V 即 View 视图是指用户看到并与之交互的界面。比如由 html 元素组成的网页界面，或者软件的客户端界面。MVC 的好处之一在于它能为应用程序处理很多不同的视图。在视图中其实没有真正的处理发生，它只是作为一种输出数据并允许用户操作的方式。

C 即 controller 控制器是指控制器接受用户的输入并调用模型和视图去完成用户的需求，控制器本身不输出任何东西和做任何处理。它只是接收请求并决定调用哪个模型构件去处理请求，然后再确定用哪个视图来显示返回的数据。

视图是用户看到并与之交互的界面。对老式的 Web 应用程序来说，视图就是由 HTML 元素组成的界面，在新式的 Web 应用程序中，HTML 依旧在视图中扮演着重要的角色，但一些新的技术已层出不穷，它们包括 Adobe Flash 和像 XHTML, XML/XSL, WML 等一些标识语言和 Web services.

MVC 好处是它能为应用程序处理很多不同的视图。在视图中其实没有真正的处理发生，不管这些数据是联机存储的还是一个雇员列表，作为视图来讲，它只是作为一种输出数据并允许用户操纵的方式。

模型表示企业数据和业务规则。在 MVC 的三个部件中，模型拥有最多的处理任务。例如它可能用像 EJBs 和 ColdFusion Components 这样的构件对象来处理数据库，被模型返回的数据是中立的，就是说模型与数据格式无关，这样一

个模型能为多个视图提供数据，由于应用于模型的代码只需写一次就可以被多个视图重用，所以减少了代码的重复性。

控制器接受用户的输入并调用模型和视图去完成用户的需求，所以当单击 Web 页面中的超链接和发送 HTML 表单时，控制器本身不输出任何东西和做任何处理。它只是接收请求并决定调用哪个模型构件去处理请求，然后再确定用哪个视图来显示返回的数据。

视图层和业务层分离，这样就允许更改视图层代码而不用重新编译模型和控制器代码，同样，一个应用的业务流程或者业务规则的改变只需要改动 MVC 的模型层即可。因为模型与控制器和视图相分离，所以很容易改变应用程序的数据层和业务规则。

MVC 模式允许使用各种不同样式的视图来访问同一个服务器端的代码，因为多个视图能共享一个模型，它包括任何 WEB (HTTP) 浏览器或者无线浏览器 (wap)，比如，用户可以通过电脑也可通过手机来订购某样产品，虽然订购的方式不一样，但处理订购产品的方式是一样的。由于模型返回的数据没有进行格式化，所以同样的构件能被不同的界面使用。

MVC 使开发和维护用户接口的技术含量降低。使用 MVC 模式使开发时间得到相当大的缩减，它使程序员 (Java 开发人员) 集中精力于业务逻辑，界面程序员 (HTML 和 JSP 开发人员) 集中精力于表现形式上。分离视图层和业务逻辑层也使得 WEB 应用更易于维护和修改。

1.2.2 五子棋规则

对局双方各执一色棋子。空棋盘开局。黑先、白后, 交替下子, 每次只能下一子。棋子下在棋盘的空白点上, 棋子下定后, 不得向其它点移动, 不得从棋盘上拿掉或拿起另落别处。黑方的第一枚棋子必须下在天元点上, 即中心交叉点。五子棋对局, 执行黑方指定开局、三手可交换、五手两打的规定。整个对局过程中黑方有禁手, 白方无禁手。黑方禁手有三三禁手、四四禁手和长连禁手三种。

第二章 技术选型

第一节 选用 JAVA 的理由

2.1.1 JAVA 语言特性

跨平台性：Java 凭借 Java 虚拟机（JVM）实现了卓越的跨平台能力。只要目标平台安装了对应的 JVM，Java 程序就能顺利运行，无需重新编译。这一特性极大地降低了开发成本和维护难度，拓宽了应用的受众范围。对于五子棋小游戏，无论玩家使用何种主流操作系统，都能轻松畅玩，无需开发者针对不同系统进行复杂的代码适配。

自动内存管理：Java 的自动垃圾回收机制（GC）是其一大亮点。它自动处理不再使用的内存空间，开发者无需手动操作内存的分配与释放，有效避免了内存泄漏、悬空指针等问题。这使得代码更加简洁、安全，在五子棋游戏开发中，开发者能专注于游戏逻辑的实现，减少因内存管理不善带来的调试困扰。

丰富的类库和框架支持：Java 拥有庞大且实用的标准类库，涵盖网络编程、图形界面、多线程等多个方面，同时还有众多开源框架可供选择。在开发五子棋游戏时，利用其网络编程类库（如 `java.net` 包）能轻松构建客户端与服务器的通信连接，图形界面类库（如 `Swing`）可快速搭建出美观的游戏界面，大大提高了开发效率。

多线程支持：Java 对多线程编程提供了良好的支持，通过 `Thread` 类和 `Runnable` 接口，开发者可以方便地创建和管理多个线程，实现并发执行。在五子棋游戏中，可将网络通信和游戏逻辑分别放在不同线程处理，避免网络延迟导致游戏卡顿，提升玩家的游戏体验。

2.1.2 C++ 语言特性

高效的性能：C++ 接近底层硬件，代码执行效率高，在对性能要求极高的场景，如大型游戏引擎开发中表现出色。然而，对于五子棋小游戏这类对性能要求并非十分苛刻的应用，Java 的性能足以满足需求，且其开发效率更具优势。

直接的内存操作能力：C++ 允许开发者直接操控内存，能根据需求精细优化内存使用，但这也增加了内存管理的难度和出错风险。对于五子棋游戏开发，Java 的自动内存管理机制更为合适，可降低开发复杂性。

丰富的语言特性和模板编程：C++ 具有多重继承、运算符重载、模板编程等丰富特性，模板编程可提高代码复用性，但也使代码复杂度增加。相比之下，Java 的语言特性在五子棋游戏开发中更能平衡开发效率和代码可读性。

与底层系统紧密结合：C++ 便于与底层操作系统 API 交互，在系统级开发中优势明显。但五子棋小游戏更侧重于应用层开发，Java 的高层抽象和易用性更符合需求。

2.1.3 选用 Java 语言开发五子棋小游戏的原因

开发效率与易用性：Java 的丰富类库和自动内存管理机制，在开发五子棋小游戏时能够快速搭建框架，高效实现各种功能，如棋盘绘制、棋子落子和网络对战等，将更多精力投入到游戏逻辑的完善上，缩短开发周期。

跨平台性需求：考虑到游戏的潜在用户可能使用不同的操作系统，Java 的跨平台特性确保游戏无需针对每个系统单独开发，降低了开发和维护成本，提高了游戏的可用性，让更多玩家能够轻松体验游戏。

代码的可读性和可维护性：Java 的面向对象特性使代码结构清晰、易于理解和维护。

网络编程和多线程支持的便利性：Java 的网络编程类库和多线程支持为实现五子棋的网络对战功能提供了便捷的方式。通过简单的 Socket 和 ServerSocket 类就能建立稳定的网络连接，多线程机制可确保游戏的流畅性，而 C++ 在这方面的实现相对复杂，开发难度较大。

第二节 选用 Eclipse 的理由

2.2.1 Eclipse 的特性

丰富的插件生态系统：拥有庞大的插件库，涵盖了代码分析、版本控制、数据库管理、Web 开发等众多领域。通过安装不同的插件，开发者可以根据项目需求灵活扩展 Eclipse 的功能，例如安装 EGit 插件进行高效的版本控制，或者安装 Spring Tools 插件方便进行 Spring 框架项目的开发。

良好的 Java 开发基础支持：对 Java 语言的基本特性提供了稳定且成熟的支持，包括语法高亮、代码自动完成、代码模板、错误检查与修复等功能，能够帮助开发者快速编写高质量的 Java 代码，提高开发效率。

2.2.2 IntelliJ IDEA 的特性

智能代码辅助：具有强大的代码智能提示和自动完成功能，能够根据代码上下文准确预测开发者的意图，提供相关的代码片段和方法建议，极大地提高了编码速度和准确性，减少了代码编写过程中的错误。

先进的代码导航与重构功能：方便开发者在大型项目中快速定位代码，轻松进行代码的重构操作，如提取方法、重命名变量、移动类等，有助于优化代码结构，提高代码的可维护性和可读性。

内置的工具集成：集成了众多实用的开发工具，如版本控制系统（Git、SVN 等）、构建工具（Maven、Gradle）、应用服务器等，无需额外安装和配置，即可在 IDE 中方便地进行项目的构建、部署和调试，简化了开发流程。

2.2.3 选择 Eclipse 的开发五子棋小游戏的原因

资源占用与性能：Eclipse 在系统资源占用方面相对较为轻量化，对于一些配置较低的计算机，能够更加流畅地运行，避免因 IDE 性能问题导致开发过程卡顿，确保开发工作的高效进行。

学习曲线与定制化：对于初学者而言，Eclipse 的界面和操作相对较为简洁直观，学习成本较低。其开源的特性使得开发者可以深入了解 IDE 的内部机制，根据自身的习惯和项目需求进行高度定制，打造个性化的开发环境。

第三章 系统设计与实现

第一节 用户登录界面

3.1.1 设计思路

创建一个五子棋游戏的登录界面，用户可以在该界面输入用户名，选择头像，并通过按钮进行连接、重置和退出操作。对用户输入的用户名进行合法性验证，当用户点击连接按钮且输入合法时，进入游戏大厅界面。

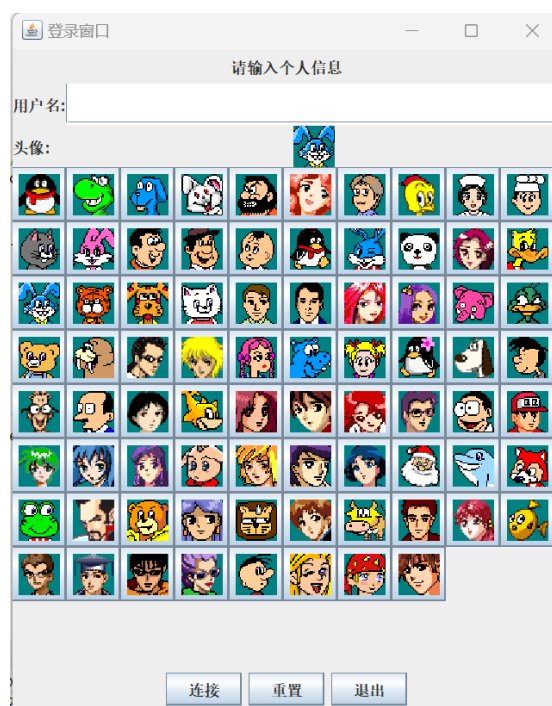


图 3.1 用户登录窗口

整体采用边框布局（BorderLayout），将界面分为上（北）、中、下（南）三个区域。北部区域用于显示提示信息和输入框相关的组件，采用内部的边框布局和网格布局来组织用户名和头像相关的标签和文本框。中部区域用于显示可

选头像，采用绝对布局（`null`）来精确放置头像按钮。南部区域用于放置操作按钮（连接、重置、退出），采用流式布局（`FlowLayout`）来排列按钮。

用户可以通过文本框输入用户名；通过点击头像按钮选择头像；点击连接、重置、退出按钮来执行相应操作。当用户点击连接按钮时，会验证输入内容的合法性，合法则进入游戏大厅，不合法则弹出相应错误提示。当用户点击重置按钮时，会清空输入内容并恢复默认头像。当用户点击退出按钮时，会退出整个程序。

3.1.2 布局实现方式

定义 `LoginWindow` 类，继承自 `Component` 并实现 `ActionListener` 接口，用于处理按钮点击等事件。声明各种成员变量，包括 `JFrame` (`jf`) 用于承载整个界面，`JLabel` 用于显示提示信息和标签，`JTextField` 用于输入用户名，`JBButton` 用于操作按钮，`Random` 用于生成随机头像编号等。

在构造函数 `LoginWindow()` 中，使用 `jf.getContentPane().setLayout(new BorderLayout());` 设置整体边框布局。创建北部（`north`）、中部（`center`）、南部（`south`）三个面板。北部面板内部再次采用边框布局，创建 `north_north`、`north_left`、`north_center` 三个子面板。在 `north_north` 中添加提示信息标签（`info`）。在 `north_left` 中采用网格布局（`new GridLayout(3,1,1,1)`）添加用户名、头像标签。在 `north_center` 中采用网格布局添加对应的文本框和头像显示标签（`use_face`）。中部面板采用绝对布局（`center.setLayout(null);`），通过循环创建 85 个头像按钮，每个按钮对应一个头像图标，并设置按钮位置、添加动作监听器。南部面板采用流式布局（`south.setLayout(new FlowLayout());`），添加连接（`Connect`）、重置（`Reset`）、退出（`Exit`）按钮，并为每个按钮添加动作监听器（`button.addActionListener(this);`）。设置窗口大小（`jf.setSize(440,550);`）、可见性（`jf.setVisible(true);`）、初始位置（`jf.setLocation(50,50);`）和关闭操作（`jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);`）。

3.1.3 具体逻辑和跳转处理

可选头像的显示：

```

int x=0,y=0;
for(int i=1;i<=85;i++){//创建头像图标按钮
    if(x<420){//设置按钮位置，满420换行
        JButton jb=new JButton(new
            ImageIcon(LoginWindow.class.getResource("/resources/face/"+i+"-1.gif")));
        jb.setBounds(x,y,42,42);
        jb.addActionListener(this);
        center.add(jb);
        x+=42;
    }else{
        y+=42;
        x=0;
        JButton jb=new JButton(new
            ImageIcon(LoginWindow.class.getResource("/resources/face/"+i+"-1.gif")));
        jb.setBounds(x,y,42,42);
        jb.addActionListener(this);
        center.add(jb);
    }
}
}

```

按钮点击事件处理：

```

@Override
public void actionPerformed(ActionEvent e) {
    //退出键
    if(e.getSource()==Exit){
        System.exit(0);
    }
    //重置键，将头像恢复为默认，同时清空用户名输入框中的内容
    else if(e.getSource()==Reset){
        use_face.setIcon(new ImageIcon("/resources/face/1-1.gif"));
        userName.setText(null);
        Server.setText(null);
    }
    //连接键
    else if(e.getSource()==Connect){
        isTextField(e);//调用合法性验证
    }
    //点击选择头像
    else{//设置新头像
        JButton btn = (JButton)e.getSource();
        use_face.setIcon(btn.getIcon());
    }
}
}

```

输入验证处理:

```
public void isTextField(ActionEvent e) {
    //1、判断姓名是否为空;
    String name=userName.getText();
    int name_long=name.length();
    if(name_long>0){
        JOptionPane.showMessageDialog(this,"欢迎进入游戏大厅!", "结果",JOptionPane.PLAIN_MESSAGE);
        Icon face=use_face.getIcon();
        new UserHouse(name,face);
        jf.setVisible(false);//将当前登录界面设置为不可见
    }else if(name_long==0){
        JOptionPane.showMessageDialog(this,"输入不能为空!", "结果",JOptionPane.ERROR_MESSAGE);
    }
}
```

第二节 房间界面

3.2.1 设计思路

构建五子棋游戏登录后的房间界面，该界面要展示用户信息（头像、用户名等）、提供游戏操作按钮（开始游戏、自动入座、退出游戏、退出登录等），同时呈现游戏座位布局情况，方便玩家入座参与游戏。实现与服务器的交互功能，包括连接服务器、向服务器发送用户操作相关信息（如入座、开始游戏等指令）以及接收服务器传来的数据并进行相应处理（如更新座位状态、显示欢迎信息等）。

3.2.2 布局实现方式

定义 UserHouse 类，继承自 JFrame 并实现 ActionListener 接口，用于承载整个房间界面以及处理界面上的各种动作事件。声明成员变量，包括网络相关的 Socket (client) 和用于向服务器发送数据的 DataOutputStream (dos)；用于存储用户信息（用户名、头像）的数组 str 和 ico；布局组件如 JFrame (jf)、JTabbedPane (jtp_1、jtp_2、jtp_3)、JSplitPane (jsp_main、jsp_left)，还有用于展示界面不同部分信息的 JLabel、JButton 以及用于座位相关布局 and 操作的数组 (jb_noone、jl_xqnoone、jl_name、jl_number)。

在构造函数 UserHouse(String my_name, Icon my_face) 实现界面初始化和基

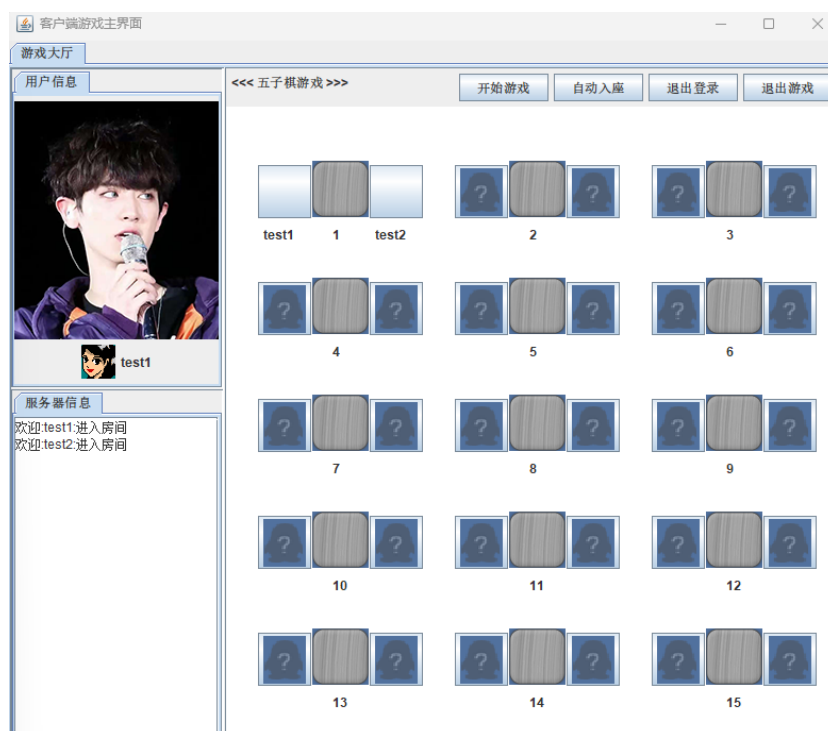


图 3.2 房间选座界面

本设置，设置分割面板 `jsp_left` 的分割位置 (`jsp_left.setDividerLocation(300);`) 和分割线大小 (`jsp_left.setDividerSize(3);`)，并通过内部布局添加用户信息相关的面板（如 `left_north`）到顶部组件，服务器信息相关的面板（通过 `JTextArea` 和 `JScrollPane` 构建的 `left_south`）到底部组件。在用户信息区域布局 `left_north` 面板中采用 `BorderLayout`，添加包含用户头像的 `left_north_center` 到中心区域，添加包含用户名和头像展示的 `left_north_south` 到南部区域，然后将 `left_north` 通过 `jtp_2` 选项卡添加到 `jsp_left` 的顶部。在游戏座位及操作按钮所在区域，首先设置 `right` 面板采用 `BorderLayout`，在其北部的 `right_north` 面板中，通过 `BorderLayout` 将标题 (`title`) 添加到西部，将游戏操作按钮 (`start_game`、`auto_seat`、`exit_login`、`exit_game`) 添加到东部。接着设置 `right_center` 面板（用于放置游戏座位相关组件）采用绝对布局 (`right_center.setLayout(null);`) 并设置背景颜色 (`right_center.setBackground(Color.WHITE);`)，调用 `seat(my_name, my_face)` 方法来进行座位的布局 and 监听器添加，最后将 `right_north` 和 `right_center` 分别添加到 `right` 面板的北部和中心区域。

3.2.3 具体逻辑和传输信息

界面选座逻辑:

1. 数组 `jb_noone[]` 为座位, 共 30 个; 数组 `jl_xqnoone[]` 为桌子; 数组 `jl_name[]` 为用户名; 数组 `jl_number[]` 为桌号。使用 `for` 循环排座位, 如果 `i` 是偶数, 则当前为左侧座位按钮, 添加完组件后 `x` 坐标右移一个桌子的宽度; 否则当前为右侧座位按钮, 添加完组件后 `x` 坐标右移一个桌距。
2. 用户点击座位后, 系统判断该座位是否已经有人坐下, 有的话在本地删除并且通知服务器删除用户; 若无则使用包含 `s`, `i`, `my_face`, 和 `my_name` 的字符串 `String` 将座位坐标位置和用户头像写入服务器。之后调起五子棋页面, 并且发送信息告诉服务器, 服务器分发给其他登录用户。
3. 如果用户点击“自动落座”按钮, 在 0-29 里面获取随机数, 判断是否有空位, 如果没空位就重复获取, 直到获取到空的位置, 不再有空位就提示用户。将座位取出后模拟一次点击。

```
else if (e.getSource() == auto_seat) {
    while (true) {
        int i = r.nextInt(30);
        if (String.valueOf(jb_noone[i].getIcon()).equals(String.valueOf(noone))) {
            //将座位坐标和头像写入服务器
            Rectangle s = jb_noone[i].getBounds();
            try {
                dos = new DataOutputStream(client.getOutputStream());
                dos.writeUTF(s + " " + i + " " + ico[0] + " " + str[0]);
                System.out.println(s + " " + i + " " + ico[0] + " " + str[0]);
            } catch (IOException ioException) {
                ioException.printStackTrace();
            }
            break;
        }
    }
}
```

服务器交互:

定义了 `clientReader` 类, 继承自 `Thread`, 可以在后台独立运行而不阻塞主线程。声明 `DataInputStream dis` 用于从服务器接收数据; `JTextArea jta` 用于在客户端界面展示相关信息 (如服务器发送的欢迎消息等), 通过构造函数进行初始化, 接收外部传入的相应对象。在 `run` 方法用 `while (true)` 实现持续监听与数据处理, `try` 块内首先通过 `String info = dis.readUTF();` 从服务器传来的数据流中读取数据。

接着使用 `String[] str01 = info.split(":"); String[] str02 = info.split(" ");` 分隔符方便提取信息片段。然后根据不同数据内容进行相应处理。

1. 处理欢迎消息：读取 `str01` 数组的第一个元素（即按照: 分割后得到的第一部分内容），将整个接收到的 `info` 字符串添加到 `JTextArea (jta)` 中，在客户端界面的文本显示区域展示。

```
if (str01[0].equals("欢迎")) {jta.append(info + "\n");}
```

2. 处理游戏开始相关数据：用户点击了客户端界面上的“开始游戏”按钮后服务器接收到请求，此时开始判断座位是否满座以决定是否正式开始游戏。首先初始化一个变量 `time = 0;`，用于记录满座的组数（这里以两个座位为一组来判断满座情况）。接着循环遍历座位，在循环内判断每两个相邻座位的图标是否都不为空座图标（即是否都有玩家入座），如果满足这个条件，说明找到了一组满座的情况，此时弹出一个提示框告知用户准备开始游戏。创建新的游戏窗口 `new GameWindow(str[0], ico[0]);` 并将当前房间界面设置为不可见，实现界面切换，引导用户进入游戏进行界面，并向服务器发送当前用户的头像和名字信息。

```
if(!String.valueOf(jb_noone[i].getIcon()).equals(String.valueOf(noone))&&
!String.valueOf(jb_noone[i+1].getIcon()).equals(String.valueOf(noone))){
    JOptionPane.showMessageDialog(jf, "准备开始游戏!", "结果",
        JOptionPane.PLAIN_MESSAGE);
    new GameWindow(str[0], ico[0]); //创建新窗口，传入名字头像
    jf.setVisible(false);
    time++;
    //点开始游戏后，将头像和名字传入服务器，用于给下一个页面数据
    String touxiang=ico[0]+" "+str[0];
    try {dos = new DataOutputStream(client.getOutputStream());
        dos.writeUTF(touxiang);
    } catch (IOException e) {e.printStackTrace();}}
```

第三节 游戏界面

3.3.1 设计思路

`GameWindow` 类主要用于构建五子棋游戏进行时的界面，集成游戏相关的功能组件及布局，将界面划分为展示玩家自身与对手信息的区域、聊天区域、

游戏棋盘区域以及操作按钮区域等，同时实现与服务器的连接及数据交互功能，以保证游戏过程中的信息更新与同步，例如接收服务器消息以及向服务器发送玩家操作相关信息等。

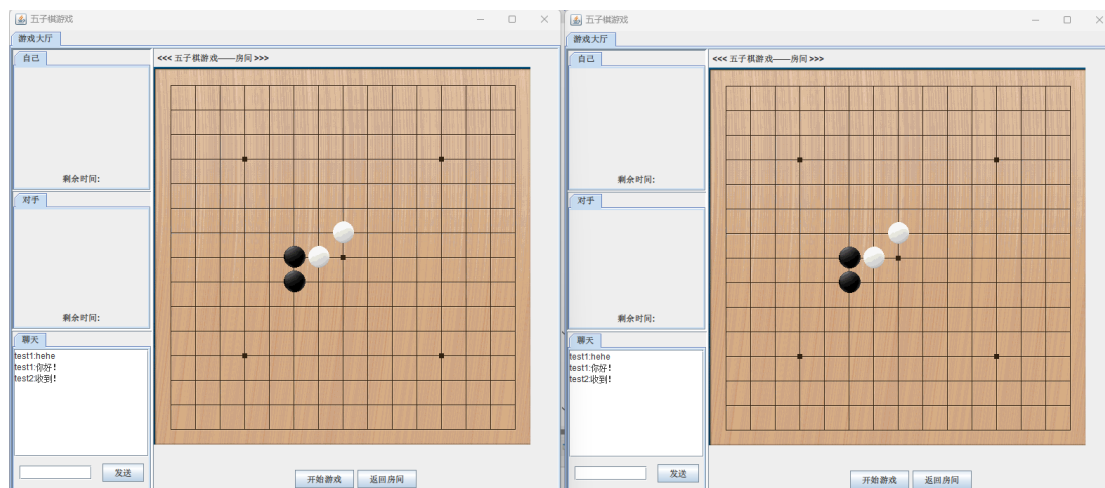


图 3.3 游戏界面

构造函数 `public GameWindow(String my_name, Icon my_face)` 实现界面初始化与基本设置，接收并存储用户信息。`actionPerformed(ActionEvent e)` 实现点击按钮处理，`connectServer()`、`createReaderThread(JTextArea jta)` 连接服务器和多线程处理接收信息。

3.3.2 聊天交互实现

用户点击“发送”按钮时，执行如下：

```
dos = new DataOutputStream(client.getOutputStream());
dos.writeUTF("info" + "::" + str[0] + "::" + sent_info.getText());
```

在 `clientReader` 方法中，执行如下：

```
String[] str02 = info.split("::");//聊天框信息
if (str02[0].equals("info")) {
    System.out.println("客户端运行!");
    gw.jta.append(str02[1] + "::" + str02[2] + "\n");
    gw.sent_info.setText("");
}
```

将聊天消息的发送者(str02[1])和消息内容(str02[2])拼接后添加到 GameWindow 类中的 JTextArea (jta, 通过 gw 引用访问) 中, 并添加换行符, 实现聊天消息在客户端聊天框区域的显示功能, 最后通过 gw.sent_info.setText(""); 将用于输入聊天消息的文本框(sent_info) 内容清空, 方便用户继续输入下一条聊天消息, 保持良好的聊天交互体验。

3.3.3 下棋功能的实现与胜负判断

定义了 Image 类型的变量 qipan (棋盘图片)、blackqizi (黑棋子图片)、whiteqizi (白棋子图片), 用于在棋盘界面上绘制相应的图形元素, 提供可视化的棋盘和棋子展示效果。棋子坐标及状态记录变量如下:

```
int row=-1;int line=-1;//记录的是鼠标点击时在面板上的原始坐标
int arrayX=-1;int arrayY=-1;//经过换算后对应到棋盘数组中的坐标位置
int[][] all_chess=new int[15][15];//二维数组用于记录棋盘上每个位置的棋子状态
//0表示该位置为空, 1表示有黑棋, 2表示有白棋
```

构造函数 public ChessBoard() 实现初始化操作, 通过 try-catch 块来加载棋盘、黑棋子和白棋子的图片资源。使用通过类加载器获取资源路径的方式, 先获取资源的 URL, 再将其转换为 File 对象, 最后使用 ImageIO.read 方法读取图片资源并赋值给相应的变量 (qipan、blackqizi、whiteqizi)。

paint 方法实现棋盘和棋子的绘制:

```
@Override
public void paint(Graphics g) {
    super.paint(g); //调用父类的paint方法
    g.drawImage(qipan,0,0,this);
    for(int i=0;i<15;i++){
        for(int j=0;j<15;j++){
            if(all_chess[i][j]==1){
                g.drawImage(blackqizi,i*35+10,j*35+10,this);
            }else if(all_chess[i][j]==2){
                g.drawImage(whiteqizi,i*35+10,j*35+10,this);
            }
        }
    }
}
```

使用传入的 Graphics 对象 g, 将之前加载的棋盘图片绘制到面板上, 从面板的左上角开始绘制, 绘制的目标组件就是当前的 ChessBoard 面板, 以此实现棋盘背景图片在界面上的展示。通过嵌套的 for 循环遍历整个棋盘的二维数组 all_chess, 根据每个位置存储的棋子状态值 (0、1、2) 来决定绘制哪种棋子图片, 绘制位

置通过计算坐标以此实现在棋盘上正确展示各个已放置棋子的可视化效果，根据棋盘状态动态更新棋子的显示情况。

鼠标点击时，触发：

```
dos = new DataOutputStream(client.getOutputStream());
dos.writeUTF(color+" "+arrayX+" "+arrayY);
```

继而在多线程交互中，储存棋子信息：

```
if(str_01[0].equals("black")) {
    int x=Integer.parseInt(str_01[1]);
    int y=Integer.parseInt(str_01[2]);
    all_chess[x][y]=1;
}else if(str_01[0].equals("white")){
    int x=Integer.parseInt(str_01[1]);
    int y=Integer.parseInt(str_01[2]);
    all_chess[x][y]=2;
}
```

胜负判断方法通过检查以该坐标为起点，向四个不同方向（垂直、水平、两条对角线方向）进行遍历，统计连续相同棋子的数量，若在某个方向上达到连续五个相同棋子，则判定该棋子所在方获胜。

```
// 定义四个方向的偏移量数组，分别对应垂直、水平、左上右下对角线、左下右上对角线方向
int[][] directions = {
    {0, -1}, {0, 1}, {-1, 0}, {1, 0}, {-1, -1}, {1, 1}, {-1, 1}, {1, -1} };
for (int[] direction : directions) {
    int count = 1;
    int curRow = row + direction[0];
    int curCol = col + direction[1];
    // 循环检查当前方向上连续相同棋子的数量
    while (curRow >= 0 && curRow < all_chess.length && curCol >= 0 && curCol <
        all_chess[0].length
        && all_chess[curRow][curCol] == all_chess[row][col]) {
        count++;
        curRow += direction[0];
        curCol += direction[1];
        if (count == 5) { return all_chess[row][col];}}}
```

第四章 总结与展望

在本次使用 Java 开发五子棋小游戏的过程中，我经历了从功能设计到代码实现，再到反复调试的完整流程，收获颇丰。

在设计阶段，我参考之前玩过的五子棋游戏加以简化思路，包括棋盘的构建、玩家操作的响应、与服务器交互实现联机对战以及胜负判断等核心功能。通过如 `GameWindow` 负责游戏界面展示与布局，`ChessBoard` 处理棋盘相关逻辑，`clientReader` 线程类实现与服务器的数据通信。代码实现环节，利用 Java 的图形界面库（如 `Swing`）绘制简单的游戏界面，实现了鼠标点击放置棋子等交互操作，同时借助网络编程知识，建立客户端与服务器的连接，确保联机功能的可行性。在胜负判断方面，`winCol` 方法准确检测五子连珠的情况，保证游戏规则的正确执行。

同时也遇到了许多 bug。通信传输流读取错误导致数据传输错误。后来采用字符串 `String` 进行数据传输，在客户端和服务端之间建立可靠的通信协议，将字符串 `s`, `i`, `my_face`, 和 `my_name` 写入客户端的输出流，服务器读取信息，采用 `split(" ")` 根据空格将信息分割成一个字符串数组 `str_01[]`，从而进行后续的操作。界面组件操作引发的显示异常。通过微信截图功能获取分辨率，不断调整资源图大小以及组件位置。以及因参数传递或边界判断失误造成的逻辑错误等问题。通过查看异常堆栈信息、上网查询以及逐步调试代码逻辑，让我深刻体会到严谨的逻辑和细致的测试对于开发的重要性。

虽然目前五子棋小游戏已基本实现简单功能，但仍有许多预想的没有实现。例如优化游戏的界面设计。以及悔棋、求和、利用权值法、博弈树算法实现人机对战。同时优化网络通信部分，增强程序的稳定性，处理好各种异常情况输出，确保联机对战的流畅性。

参考文献

- [1] 刘潇. 基于 JAVA 的五子棋游戏设计与实现. 电脑知识与技术, 2011, 7(06): 1353–1354.

致 谢

在完成这篇论文的过程中，我得到了许多人的帮助和支持。在此，我要向他们表示衷心的感谢。

首先，我要感谢我的老师。他在我论文写作过程中给予了悉心的指导和帮助。无论是选题、构思、撰写还是修改，导师都耐心地指导我，并提出了宝贵的意见和建议。导师的严谨的学术态度、深厚的学术造诣和无私的奉献精神，让我深受感染和启发。我不仅在论文写作上学到了很多知识，更重要的是学到了做学问的方法和态度。在此，我向导师表示最诚挚的感谢和敬意。

其次，我要感谢学院的各位老师。他们在我论文写作过程中给予了我很多宝贵的意见和建议。他们的专业知识和丰富的教学经验让我受益匪浅，也让我更加深入地理解了论文的主题和内容。同时，他们也在我的学习和生活中给予了很多关心和帮助，让我更加坚定了自己的学术追求。

此外，我还要感谢我的同学们。他们在我论文写作过程中给予了我很多帮助和支持。我们一起探讨学术问题、交流心得体会，互相鼓励、互相支持。他们的陪伴让我的学习生活更加丰富多彩，也让我更加珍惜这段美好的时光。

最后，我要感谢我的家人。他们一直支持我的学习和成长，为我提供了良好的学习和生活环境。在我论文写作过程中，他们也给予了我很多支持和鼓励，让我能够顺利地完成论文的撰写。家人的支持和鼓励是我前进的动力和源泉，也是我坚持不懈追求学术事业的坚实后盾。

在未来的学习和工作中，我将继续努力，不断进步，为学术事业的发展贡献自己的力量。再次向所有给予我帮助和支持的人表示最诚挚的感谢和敬意。

同时，我也要感谢参与审阅、答辩的专家教授们。他们对我的论文进行了认真的评议和指导，提出了宝贵的意见和建议。他们的专业素养和严谨的学术作风让我深受启发和感悟，也让我更加明白了做学问要注重细节、严谨认真。他们的指点和帮助让我更加深入地思考问题、提高自己的学术水平。在此，我向审阅、答辩的专家教授们表示最诚挚的感谢和敬意。

此外，我还要感谢那些在学术研究中做出杰出贡献的前辈们。他们的研究成果和学术思想是我学习和研究的基石和导向，也是推动学术事业不断发展的

重要力量。他们的精神和成就激励着我不断向前迈进、攀登学术高峰。在此，我向前辈们表示最崇高的敬意和感激之情。

最后，我要感谢那些一直关注我成长的人们。他们的鼓励和支持是我不断进步的动力源泉，也是我勇敢面对困难和挑战的力量支撑。他们的关心和期望让我更加坚定了自己的信念和追求，也让我更加有信心去迎接未来的挑战 and 机遇。在此，我向他们表示最真挚的谢意和感激之情。

总之，在完成这篇论文的过程中，我得到了许多人的帮助和支持。他们的关心、支持和鼓励是我前进的动力和源泉。再次向所有给予我帮助和支持的人表示最诚挚的感谢和敬意。在未来的学习和工作中，我将继续努力、不断进步，为学术事业的发展贡献自己的力量。