

# 南 开 大 学

## JAVA 语 言 程 序 设 计 课 程 作 业

中文题目: 基于 Java 的五子棋游戏

外文题目: Gobang game based on Java

学 号: 2313721

姓 名: 许洋

年 级: 2023 级

学 院: 计算机学院

系 别: 计算机科学与技术

专 业: 计算机科学与技术

完成日期: 2024 年 12 月

指导教师: 刘嘉欣 教授

## 关于南开大学本科生毕业论文（设计）的声明

本人郑重声明：所呈交的学位论文，是本人在指导教师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或没有公开发表的作品内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年 月 日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

## 摘 要

五子棋游戏作为一项历史悠久的传统游戏，因为其简单的游戏规则，灵活的游戏策略，时至今日仍然坐拥众多的玩家。由于五子棋是一款双人游戏，对战时对游戏双方有一定的空间和时间要求。所以我们可以考虑利用计算机技术开发一款五子棋游戏，实现网络上的联机对战，这样就可以让对战双方很大程度上解除空间限制。

在执行输入的计算机指令时，利用 Java 编程语言通过解释执行,它是面向对象的,关键是在许多开发语言中,Java 编程语言具有简单、可靠、安全、高效等非常明显的优势，而且还可以移植,节省了大量开发工作，并且具备中立性和多线程的结构特点，因此 Java 编程语言关注度非常高。[1]

本文使用的是 GUI 中的 swing，并配合 MVC 模式进行设计，同时在 MVC 的大体框架之上添加了 net 网络部分，实现了人机对战、网络对战两种模式，在游戏中提供悔棋、求和、认输、重开等快捷功能，除了整个游戏的逻辑层次的构架，还补充了基本的背景图形音乐等设计，丰富游戏体验，最终设计成型了一个多功能五子棋小游戏。

**关键词：**五子棋游戏；网络对战；GUI；JAVA 语言

## **Abstract**

Gobang game, as a traditional game with a long history, still has many players today because of its simple game rules and flexible game strategies. Because gobang is a two person game, there are certain space and time requirements for both sides of the game in wartime. Therefore, we can consider using computer technology to develop a Gobang game to realize the online battle on the network, so that both sides of the battle can largely remove the space restrictions.

When executing the input computer instructions, Java programming language is used to execute through interpretation. It is object-oriented. The key is that among many development languages, Java programming language has obvious advantages such as simplicity, reliability, safety and efficiency. It can also be transplanted, which saves a lot of development work, and has the structural characteristics of neutrality and multi-threading. Therefore, the Java programming language is highly concerned.

In this paper, swing in GUI is used and designed in cooperation with MVC mode. At the same time, net network part is added on the general framework of MVC to realize two modes of man-machine war and network war. It provides quick functions such as repentance, sum, admit defeat and reopen in the game. In addition to the logical framework of the whole game, it also complements the design of basic background graphics and music, Enrich the game experience and finally design a multifunctional Gobang game.

**Key Words:** Gobang game; Network warfare; GUI; Java language

# 目 录

摘要	.....	I
Abstract	.....	II
目录	.....	III
第一章 绪论	.....	1
第一节 背景介绍	.....	1
第二节 开发价值	.....	2
第二章 开发工具	.....	3
第一节 Java 和 C++	.....	3
第二节 GUI 工具集	.....	4
2.2.1 AWT	.....	5
2.2.2 Swing	.....	6
2.2.3 对比	.....	6
第三节 IDE 的选择	.....	7
2.3.1 IDEA	.....	7
2.3.2 Eclipse	.....	8
第三章 模型与协议设计	.....	9
第一节 模型层	.....	10
3.1.1 棋盘模型与棋子表示	.....	11
3.1.2 下棋动作的实现	.....	11
3.1.3 输赢判断逻辑	.....	11
3.1.4 悔棋与清空棋盘的操作	.....	11
3.1.5 棋盘状态的维护	.....	11
第二节 视图层	.....	12
3.2.1 ChessPanel: 绘制棋盘与棋子	.....	12
3.2.2 NorthPanel: 监听面板	.....	12
3.2.3 WestPanel: 按钮面板	.....	12
3.2.4 ChatPanel: 聊天面板	.....	13
第三节 控制层	.....	13

3.3.1	游戏逻辑控制 .....	13
3.3.2	游戏结束与对话框管理 .....	13
3.3.3	悔棋与游戏重置 .....	14
3.3.4	用户界面交互 .....	14
第四节	网络层 .....	14
3.4.1	网络通信管理 .....	14
3.4.2	消息解析与处理 .....	14
3.4.3	单例设计模式 .....	14
3.4.4	发送消息功能 .....	15
3.4.5	用户界面交互 .....	15
第四章	程序设计 .....	16
第一节	图形化界面 .....	16
4.1.1	登录窗口 .....	16
4.1.2	人人对战 .....	16
4.1.3	人机对战 .....	17
第二节	按钮功能 .....	18
4.2.1	悔棋 .....	18
4.2.2	认输 .....	18
4.2.3	求和 .....	19
第三节	菜单栏 .....	19
第五章	总结和展望 .....	20
第一节	全文总结 .....	20
第二节	未来展望 .....	20
参考文献	.....	21
致 谢	.....	XXII

## 第一章 绪论

### 第一节 背景介绍

随着科学技术的飞速发展，我们正处在一个信息爆炸的时代。计算机技术的迅猛进步不仅拓宽了其应用范围，也加速了其发展速度。计算机的普及极大地满足了人们日益增长的需求，并为日常生活带来了便利。在娱乐领域，随着人们对娱乐需求的增加，网络游戏和单机游戏不断涌现，极大地丰富了人们的娱乐生活。

五子棋作为一种历史悠久且广受欢迎的棋类游戏，以其简单的规则和策略性思考而著称。尽管历经多年，五子棋依然受到众多玩家的喜爱，成为他们闲暇时的娱乐选择。利用计算机和网络技术，开发一款基于网络的五子棋游戏，可以让人们足不出户地享受游戏的乐趣，满足不同场景下玩家的娱乐需求。

在开发一款功能全面、符合用户需求的应用程序时，需要考虑多方面的因素。除了程序设计者和使用者，选择正确的编程语言也是一个关键决策。在众多程序设计语言中，Java 以其强大的功能脱颖而出。游戏开发往往涉及复杂的情况，有时甚至包括美学问题，因此开发过程常常需要不断修改和调整。能够方便地修改游戏内容对于游戏开发至关重要。

许多游戏引擎都配备了自定义的游戏脚本解释器，用于解释特定的游戏脚本语言，这些脚本语言通常是命令式的，例如 `movePlayer(int x, int y)` 或 `playerSmile()`。此外，一些游戏引擎还支持如 Lua 和 Ruby 等脚本语言，以便访问游戏引擎的功能。Java 语言由于其虚拟机特性，本身就适合编写游戏脚本。目前，还有如 Beanshell 和 Groovy 等脚本语言可以与 Java 无缝交互，这为 Java 游戏编程提供了极大的便利。[2]

## 第二节 开发价值

在时代的快速发展中，我们不仅要学习知识文化，更要注重将所学知识应用于实践。国家的繁荣复兴依赖于技术精湛的工匠，技术的价值只有在实际应用中才能得到体现。五子棋游戏的开发，使用 Java 语言，不仅满足了玩家的需求，也是对实践能力和知识运用的一次检验，其深远的意义不言而喻。

Java 作为一种广泛应用于软件开发的编程语言，以其面向对象的特性和独特的优势——如简单性、安全性、可移植性和可扩展性——在提升软件开发效率方面发挥着核心作用。结合人们对游戏的需求和网络技术的现状，选择 Java 作为开发语言具有明显的优势和便利。[3]

本次开发不仅明显提升了开发者的技术能力，还锻炼了思考能力。在开发过程中，开发者需要站在玩家的角度思考，考虑玩家的需求和操作，预见可能出现的各种情况并制定对策。同时，还需考虑如何让用户在使用过程中不依赖对内部程序的了解，以及如何提供良好的游戏体验，避免因程序问题给玩家带来困扰。这种锻炼对于提升个人的工作能力和社交能力都具有重要意义。

娱乐行业的发展是社会进步的体现。在快节奏和竞争激烈的现代社会中，人们面临着各种压力和烦躁，通过业余娱乐和放松来缓解压力是一种有效的途径。通过游戏开发推动娱乐行业的进步，对社会发展和人们精神状态的调整具有积极影响。同时，通过本次游戏开发，也能提升个人的编程能力和 Java 应用能力。



## 第二章 开发工具

### 第一节 Java 和 C++

Java 和 C++ 都是流行的面向对象编程语言，它们都采用了封装、继承和多态等面向对象的核心概念，这使得它们在代码重用性方面表现出色。尽管如此，这两种语言在编译和执行过程中有着显著的不同，主要有以下 5 方面：[4, 5] • 解释与编译：

- Java 是一种解释型语言，其代码在运行时会被 Java 虚拟机（JVM）解释执行。现代 Java 执行环境结合了解释执行和即时编译（JIT）技术，以提高运行效率。这意味着 Java 代码在执行时会被“翻译”成二进制形式，但随着 JIT 技术的应用，热点代码会被编译成机器码以提高性能。

- C++ 是一种编译型语言，其代码在编译时直接转换成特定操作系统的机器语言，这意味着 C++ 程序一旦编译完成，就可以在目标操作系统上直接运行，无需额外的解释或转换步骤。[6]

- 内存安全：

- Java 是一种内存安全的语言，它提供了自动垃圾回收机制和数组边界检查，以防止内存泄漏和越界访问。在 Java 中，即使数组访问超出了预定义的范围，也只会抛出异常，而不会导致程序崩溃。

- C++ 提供了更高的灵活性，允许开发者直接管理内存，包括使用指针和动态内存分配。这种灵活性的代价是，如果开发者不当使用，可能会导致内存泄漏、越界访问甚至程序崩溃。

- 性能：

- Java 的性能通常受到 JVM 解释执行的影响，尽管 JIT 技术已经显著提高了性能，但在某些情况下，特别是在大型程序和经过优化的 C++ 代码面前，Java 的性能可能会稍逊一筹。

- C++ 由于在编译时直接生成机器码，因此通常能够提供更快的运行速度。对于大型和复杂的程序，C++ 的性能优势可能会更加明显。

- 指针：

- C++ 支持指针，允许开发者直接操作内存地址，这为内存管理和优化提供了强大的工具，但也增加了编程的复杂性和出错的风险。

- Java 不支持指针操作，而是通过引用和对象管理来处理内存，这减少了直接内存操作的风险，但也限制了某些底层操作的可能性。

- 重载：

- Java 支持方法重载，即同一个类中可以有多个同名方法，只要它们的参数列表不同即可。这增加了代码的灵活性和可读性。

- C++ 不仅支持方法重载，还支持运算符重载，允许开发者定义或修改大多数内置运算符的行为，以适应特定的数据类型。这些差异使得 Java 和 C++ 在不同的应用场景下各有优势。Java 的跨平台特性和内存安全使其在企业级应用和网络编程中非常受欢迎，而 C++ 的高性能和底层控制能力使其在系统编程、游戏开发和高性能计算领域中占据重要地位。

## 第二节 GUI 工具集

图形用户界面（Graphical User Interface，简称 GUI）是一种通过图形化元素展示的计算机操作界面，用户可以通过点击菜单、按钮、标签等界面组件以及使用鼠标来执行指令和启动应用程序。与传统的命令行界面相比，GUI 在视觉上更加直观和易于接受。现代程序开发者越来越重视 GUI 设计，认为它的重要性不亚于程序逻辑本身，甚至在某些情况下更为关键。

在 Java 出现之前，不同的操作系统如 Windows、Linux、UNIX 等都有自己独特的图形用户界面，它们的主要问题是缺乏跨平台的兼容性，即在一个平台上开发的程序往往不能直接在其他平台上运行。Java 的出现改变了这一局面。Java 提供了一套跨平台的 GUI 组件，使得开发者能够编写一次代码，然后在多个平台上运行，无需为每个平台单独编写界面。Java 中常用的 GUI 组件包括：

- AWT（Abstract Window Toolkit）：Java 最初的 GUI 工具包，提供了基础的

窗口管理功能，但由于其组件不是完全由 Java 实现，因此在跨平台性和外观一致性上有所限制。

- **Swing**: 建立在 AWT 之上的纯 Java GUI 工具包，提供了更丰富的组件和更好的跨平台支持。Swing 组件完全由 Java 编写，因此可以在任何安装了 Java 运行环境的平台上保持一致的外观和行为。[7]
- **SWT/JFace**: 不是 Java 标准版的一部分，而是 Eclipse 项目的一部分。SWT (Standard Widget Toolkit) 提供了与本地操作系统组件紧密集成的 GUI 组件，因此可以提供更接近本地应用程序的外观和感觉，但牺牲了一定的跨平台性。

随着技术的发展，Java GUI 编程也在不断进步，新的框架和库不断涌现，以满足现代应用程序的需求。开发者可以根据项目的具体需求选择合适的 GUI 工具包，以创建既美观又功能强大的用户界面。

### 2.2.1 AWT

AWT (Abstract Windowing Toolkit) 确实是 Java 的一个抽象窗口工具集，它提供了一套与本地窗口系统进行交互的接口。

AWT 的“对等”机制是指 Java 中的 GUI 组件与本地窗口系统中的组件之间存在一对一的映射关系。这种设计允许 Java 程序在不同的操作系统上运行时，能够利用该系统的 GUI 功能。

AWT 的“最大公约数”方法意味着它只提供了在所有支持的操作系统中都存在的 GUI 功能，这样做是为了保持 Java 的跨平台特性，即“一次编写，到处运行” (write once, run anywhere)。由于 AWT 提供的是最基本的 GUI 功能，对于那些需要更丰富、更高级特性的应用程序，开发者可能需要使用 Swing 或者 JavaFX 等更高级的 GUI 工具集。Swing 是构建于 AWT 之上的，它提供了更多的组件和更好的跨平台一致性，而 JavaFX 是 Java 的一个较新的图形和媒体包，它提供了更丰富的用户界面控件和更强大的图形渲染能力。

总的来说，AWT 是 Java GUI 编程的基础，但它的局限性促使开发者探索其他更高级的 GUI 框架来满足现代应用程序的需求。

### 2.2.2 Swing

Swing 是 Java 的一个 GUI（图形用户界面）组件库，它在 1997 年的 JavaOne 大会上首次亮相，并在 1998 年 5 月正式推出，作为对早期 AWT（Abstract Window Toolkit）的改进。Swing 的设计初衷是为了提供一个更加强大和灵活的界面开发工具，它通过“模拟”机制，不依赖于本地操作系统的窗口组件，而是完全由 Java 代码绘制界面元素，从而实现了跨平台的一致性和可定制性。

尽管 Sun 公司曾预测 Java 应用和 applets 将成为主流，并且人们会转向 JavaOS 操作系统，这一愿景并未完全实现。然而，Swing 作为 Java 应用程序和小程序的 GUI 库，仍然在一段时间内取得了相当的成功。Swing 的主要优势在于其独立于本地代码和窗口组件，能够灵活地在不同平台上绘制窗口组件，提供了一个强大且完整的开发工具包。这种设计使得 Swing 能够在保持界面一致性的同时，为开发者提供更多的自定义选项。

### 2.2.3 对比

AWT 是 Java 早期的 GUI 组件库，虽然 Sun 公司现在不推荐使用，但在一些低资源环境，如移动或嵌入式设备中，它依然有其独特的优势，例如低环境要求、少内存消耗、快速启动和响应，以及成熟稳定的性能。然而，AWT 也存在一些明显的不足：

- 组件种类有限：缺少桌面应用中常见的表格和树形视图等重要组件。
- 组件特性不足：例如，按钮不支持图片显示。
- 扩展性差：AWT 的组件是本地组件，JVM 中的 AWT 类实例仅包含对本地组件的引用，不支持继承和重用，限制了编程的灵活性。

相比之下，Swing 是一个功能强大的 GUI 工具集，也是 Sun 推荐使用的 GUI 工具集。经过多个版本的改进和扩展，Swing 在某些领域展现出了明显的优势：

- 组件类型丰富：Swing 提供了广泛的标准组件，与 SWT 一样丰富，并且得益于其良好的可扩展性，除了标准组件外，还有许多第三方组件库可供选择。
- 组件特性丰富：Swing 不仅包含了跨平台的特性，还支持根据运行平台添加额外特性，遵循易于扩展的原则，提供了比 SWT 和 AWT 更多的功能。
- 优秀的组件 API 模型：Swing 遵循 MVC 设计模式，这种模式非常成功，其

API 成熟且设计精良，被认为是最成功的 GUI 设计之一。

基于这些原因，我们的教材和教师选择让我们学习 Swing 而不是 AWT，以更好地适应现代 GUI 开发的需求。

### 第三节 IDE 的选择

在 TIOBE 编程语言排行榜中，Java 一直稳居前三甲，目前全球有大约 700 万到 1000 万的 Java 开发者。由于许多应用程序完全用 Java 编写，因此集成开发环境（IDE）对于开发人员来说至关重要，它们是编写、测试和运行 Java 程序的核心工具。

在众多 Java 开发工具中，IDEA（IntelliJ IDEA）和 Eclipse 以其卓越的性能和广泛的用户基础脱颖而出，成为市场上最受欢迎的选择。这两款 IDE 提供了强大的功能，包括代码自动完成、错误检测、版本控制集成等，极大地提高了开发效率和代码质量。

#### 2.3.1 IDEA

IntelliJ IDEA，即 IntelliJ Integrated Development Environment for Java，是一款专为 Java 编程语言开发的集成开发环境。它在软件开发领域享有盛誉，被许多开发者认为是 Java 开发的最佳工具。IntelliJ IDEA 的卓越性能尤其在智能代码辅助、代码自动提示、代码重构、JavaEE 支持、版本控制工具集成、JUnit 测试、CVS 整合、代码分析以及创新的图形用户界面设计等方面表现得尤为突出。

在众多优点中，IDEA 的调试功能尤为突出，它能够对 Java 代码以及 JavaScript、jQuery、Ajax 等技术进行深入的调试。例如，在查看 Map 类型的对象时，如果其实现是基于哈希映射，IDEA 会自动过滤掉空的 Entry 实例，从而简化调试过程。此外，当开发者需要动态评估一个表达式的值，尤其是在面对不熟悉 API 的类实例时，可以通过代码补全功能快速找到支持的方法。在处理多线程调试时，IDEA 提供的在控制台记录日志的功能可以帮助开发者监控多线程的执行情况，进一步优化代码和程序性能。这些功能使得 IntelliJ IDEA 成为了一个强大的开发工具，极大地提高了开发效率和代码质量。

### 2.3.2 Eclipse

Eclipse 是一个开源的、基于 Java 的可扩展开发平台，它本身提供了一个框架和一系列服务，允许通过插件组件来构建定制化的开发环境。Eclipse 附带了一套标准的插件，其中包括 Java 开发工具（Java Development Kit, JDK），这使得它在 Java 开发领域尤其受欢迎。以下是 Eclipse 的一些显著优点：[7]

Eclipse 提供了强大的代码辅助功能，包括代码完成、重构和语法检查，这些功能对于提高编程效率和代码质量至关重要。Eclipse 的 Java 开发工具（JDT）及其插件工具集，为开发各种 Java 应用程序提供了全面的支持。Eclipse 不仅支持 Java，还支持多种编程语言，如 C/C++ 和 PHP，使其成为一个多语言开发的一站式资源。Eclipse 的免费特性也是其广受欢迎的一个重要原因，许多公司因此选择使用这款 IDE，并且它良好的性能在市场上占有一席之地。Eclipse 还提供了便捷的操作方式，例如使用 Ctrl+M 快捷键或双击标签来最大化当前编辑器窗口，以及鼠标悬停在代码上即可查看部分 Javadoc 文档的功能。

经过与 IntelliJ IDEA 的对比，我们最终选择了 Eclipse 作为开发工具。尽管 IntelliJ IDEA 在某些方面可能更为出色，但 Eclipse 的便捷性和易用性，尤其是对于新手来说，使得它成为一个更容易上手的选择。这些小优势，加上 Eclipse 的灵活性和社区支持，使其成为我们团队的优选开发环境。

## 第三章 模型与协议设计

MVC，即模型-视图-控制器（Model-View-Controller）模式，是一种软件设计架构，旨在将应用程序分为三个核心组件：模型（Model）、视图（View）和控制器（Controller），以实现业务逻辑、数据和界面显示的分离。这种分层结构有助于降低各部分之间的耦合度，提高代码的可维护性和可复用性。[8]MVC 模式将应用程序划分为如下三个部分：

1. 模型（Model）：作为被观察者，模型负责封装与业务逻辑相关的数据和处理方法。它包含了数据的增加、删除以及更新操作，并且能够在数据发生变化时通知观察者（通常是视图）进行更新。

2. 视图（View）：作为观察者，视图负责提供用户界面的可视化表示。它是用户直接交互的界面，例如通过浏览器访问的网页或在移动设备上显示的界面。视图从模型获取数据，并根据这些数据渲染用户界面。

3. 控制器（Controller）：控制器作为视图和模型之间的中介，负责处理用户的输入，将输入转化为模型状态的变更，并选择合适的视图进行展示。它从视图接收数据，向模型发送数据，并在需要时更新视图以反映模型的最新状态。

通过采用 MVC 模式，当需要对用户界面或交互进行定制或改进时，开发者可以不必重写业务逻辑，从而使得 Web 项目的开发和维护变得更加容易。这种模式提高了应用程序的灵活性和可扩展性，使得开发团队能够更高效地协作和迭代产品。

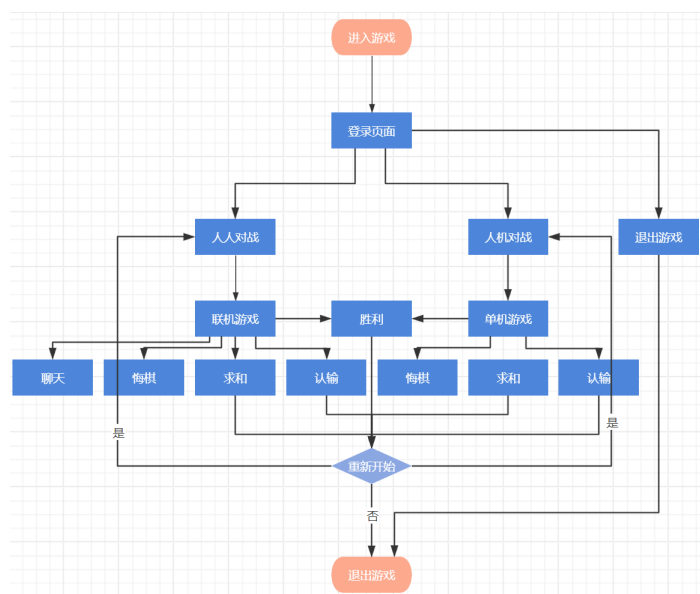


图 3.1 总览

## 第一节 模型层

在 MVC（模型-视图-控制器）架构模式中，模型层是程序的核心，主要负责实现主要功能和数据信息的表达与访问。它按照特定的业务逻辑进行操作，确保程序保持在合理的状态。模型层通过面向对象的方式，将抽象的概念转化为具体的程序对象，这些对象包含了属性和逻辑等信息。模型层的作用是响应视图的查询请求，并根据系统设定的要求做出反应。当模型内的数据发生变化时，它可以通知视图进行相应的状态调整和更新，从而确保用户界面能够反映最新的数据状态。

控制器在模型和视图之间起到中介的作用，它接收用户的输入，将这些输入转化为对模型的操作，并从模型层获取数据以更新视图。这样，控制器确保了用户的任务要求得到满足。模型层、视图层和控制器的紧密协作，为用户提供了一个流畅且响应迅速的交互体验，使得应用程序易于使用和维护。通过这种分层的设计，MVC 模式提高了代码的可维护性和可扩展性，使得开发和更新变得更加高效。



### 3.1.1 棋盘模型与棋子表示

Model 层定义了五子棋的棋盘模型，包括棋盘的尺寸和棋子的表示方法。棋盘被表示为一个二维数组，其中空位用 0 表示，黑棋用 1 表示，白棋用 -1 表示。这种表示方法简单直观，便于程序处理。

### 3.1.2 下棋动作的实现

Model 层中的 ‘putChess’ 函数负责处理玩家的下棋动作。该函数首先检查玩家选择的位置是否在棋盘范围内，并且该位置是否为空。如果条件满足，函数将在棋盘数组的对应位置放置棋子，并更新最后落子的坐标。

### 3.1.3 输赢判断逻辑

输赢判断是五子棋游戏的核心逻辑之一。Model 层通过 ‘whoWin’ 函数来检查是否有玩家赢得了游戏。该函数会检查最后一次落子是否形成了五子连珠，包括水平、垂直和两个对角线方向。这是通过 ‘checkDirection’ 和 ‘countConsecutive’ 函数实现的，它们负责计算特定方向上连续相同颜色棋子的数量。

### 3.1.4 悔棋与清空棋盘的操作

Model 层还处理游戏中的悔棋和重开操作。‘removeChess’ 函数用于悔棋，它从棋盘上移除最后一个落子的棋子。而 ‘resetBoard’ 函数用于重开游戏，它将棋盘上所有的位置重置为 0，清空棋盘。这些操作使得游戏更加灵活，允许玩家在游戏过程中更改决策。

### 3.1.5 棋盘状态的维护

Model 层维护着棋盘的当前状态，包括棋子的位置和游戏的输赢状态。这种状态管理对于游戏的进行至关重要，它确保了游戏逻辑的正确性和数据的一致性。

总结来说，Model 层为五子棋游戏提供了一个坚实的基础，它不仅管理着棋盘的状态，还处理着游戏的核心逻辑，包括下棋动作、输赢判断以及悔棋和重开操作。通过将这些功能封装在 Model 层，游戏的其他部分，如视图层和控制

器层，可以更加专注于它们自己的职责，从而实现清晰和高效的代码结构。

## 第二节 视图层

视图层在五子棋游戏中扮演着至关重要的角色，它负责提供程序与用户之间的交互基础，确保用户界面能够准确地展示模型层的数据，并在数据变化时更新视图。在设计五子棋游戏过程中，建立多个 `panel` 来实现面板上不同位置的功能：

### 3.2.1 ChessPanel: 绘制棋盘与棋子

‘`ChessPanel`’ 是一个自定义的 ‘`JPanel`’，负责绘制五子棋的棋盘和棋子。它通过覆盖 ‘`paintComponent`’ 方法来绘制棋盘的线条和棋子。棋盘的大小和棋子的位置基于模型层提供的数据。棋子的颜色（黑或白）也由模型层决定，并通过不同的颜色填充圆形来表示在 ‘`JPanel`’ 上。此外，‘`ChessPanel`’ 还监听鼠标事件，当用户在棋盘上点击时，计算出相应的坐标，并通知控制器层处理落子逻辑。

### 3.2.2 NorthPanel: 监听面板

‘`NorthPanel`’ 提供了一个顶部面板，用于显示当前轮到谁下棋以及网络连接的状态。它包含 IP 和端口标签，以及连接和监听按钮。用户可以通过这些控件输入服务器的 IP 地址和端口号，并发起连接或开始监听。按钮的动作监听器会响应用户的点击事件，调用网络层的方法来处理连接请求，并更新界面上的轮到谁下棋的标签。

### 3.2.3 WestPanel: 按钮面板

‘`WestPanel`’ 包含三个按钮：悔棋、认输和求和。这些按钮允许玩家在游戏中进行相应的操作。每个按钮都有一个事件监听器，当被点击时，会通知控制器层执行相应的游戏逻辑。悔棋按钮会调用控制器层的方法来撤销最后一步棋，而认输和求和按钮则分别通知游戏结束。

### 3.2.4 ChatPanel: 聊天面板

‘ChatPanel’ 提供了一个聊天界面，允许玩家在游戏中进行交流。它包含一个文本区域用于显示聊天消息，一个文本框用于输入消息，以及一个发送按钮。当用户输入消息并点击发送按钮时，消息会被发送到网络上，同时在聊天区域显示发送的消息。面板还包含一个方法 ‘receiveChatMessage’，用于接收并显示来自对方的消息。

综上所述，视图层通过这些面板组件提供了一个直观的用户界面，使得玩家可以进行下棋、聊天和游戏控制等操作。这些组件与模型层和控制器层紧密协作，确保用户的操作能够及时反映到游戏状态中，同时也将游戏的状态变化展示给用户。通过这种方式，视图层增强了游戏的交互性和用户体验。

## 第三节 控制层

控制层（Controller）在五子棋游戏中扮演着核心的调控角色，它负责协调模型层（Model）和视图层（View）之间的交互，并管理游戏的整体流程。

### 3.3.1 游戏逻辑控制

包括游戏的基本逻辑控制，如棋步的合法性检查、回合的切换、以及游戏状态的更新。Control 类通过 isAllow 变量控制当前轮到谁下棋，并通过 moveHistory 栈记录每一步棋的历史。localPutChess 和 otherPutChess 方法处理玩家下棋的逻辑，包括在棋盘上放置棋子、更新回合状态、发送网络消息（如果是在线游戏）以及检查游戏是否结束。

### 3.3.2 游戏结束与对话框管理

负责游戏结束时的逻辑处理，包括判断游戏的获胜者（checkWinner 方法）和显示游戏结束对话框（showGameOverDialog 方法）。对话框询问玩家是否想要重新开始游戏或退出，根据玩家的选择执行相应的操作。

### 3.3.3 悔棋与游戏重置

允许玩家悔棋（undoMove 方法）和认输（surrender 方法），以及在游戏结束后或玩家请求时重置游戏（resetGame 方法）。悔棋功能会撤销最后一步棋，并更新游戏状态，包括棋盘显示和回合状态。游戏重置则会清空棋盘和棋步历史，准备开始新的游戏。

### 3.3.4 用户界面交互

Control 类还负责与用户界面的交互，如更新棋盘（ChessPanel.getInstance().repaint）、更新回合标签（NorthPanel.getInstance().updateTurnLabel）和更新悔棋按钮的状态（WestPanel.getInstance().setCanUndo）。这些方法确保用户界面能够及时反映游戏状态的变化。

## 第四节 网络层

### 3.4.1 网络通信管理

‘Net’类通过 ‘ServerSocket’ 监听指定端口（‘PORT’）来接受客户端连接，或者通过 ‘Socket’ 主动连接到服务器。一旦建立了网络连接，它会创建一个线程来持续读取从对方发送过来的消息，并根据消息类型进行相应的处理。

### 3.4.2 消息解析与处理

‘Net’类包含了 ‘parseMessage’ 方法，用于解析接收到的消息。根据不同的消息前缀（如 “undo:”、“chess:”、“chat:” 和 “resetgame”），它会调用不同的处理函数。例如，“undo:” 消息会触发悔棋操作，“chess:” 消息会放置对方的棋子，“chat:” 消息会显示聊天内容，而 “resetgame” 消息则会重置游戏。

### 3.4.3 单例设计模式

‘Net’类采用单例设计模式，确保整个应用程序中只有一个 ‘Net’ 实例，通过 ‘getInstance’ 方法提供全局访问点，这有助于统一管理网络连接和通信。

#### 3.4.4 发送消息功能

‘Net’类提供了发送棋步、聊天消息、撤回操作和游戏重置请求的方法。这些方法允许游戏逻辑在需要时发送相应的网络消息。

#### 3.4.5 用户界面交互

‘Net’类还负责将接收到的聊天消息传递给用户界面组件 ‘ChatPanel’，以显示聊天内容，增强了游戏的交互性。

## 第四章 程序设计

### 第一节 图形化界面

#### 4.1.1 登录窗口

首先游戏有一个的主窗口，包含三个按钮，分别是“人人对战”、“人机对战”、“退出游戏”。



图 4.1 登录窗口

#### 4.1.2 人人对战

联网模式中一个用户可以和另一个用户进行对战。

监听和连接按钮：默认连接本地主机 2345 端口。

悔棋按钮：每回合只允许悔棋一次，符合正常的下棋逻辑。

认输按钮：会弹失败对话框，同时会询问是否重新开始，不重开则退出游戏。

求和按钮：会向对方发出求和询问，一旦对方确认求和，则判定平局，并询问是否重新开始。

聊天窗口：最右侧的文本框为玩家提供聊天功能，双方能够进行实时连续

对话。

菜单栏：游戏菜单可以进行更换游戏背景，帮助界面可以查询游戏规则，以及制作者信息。

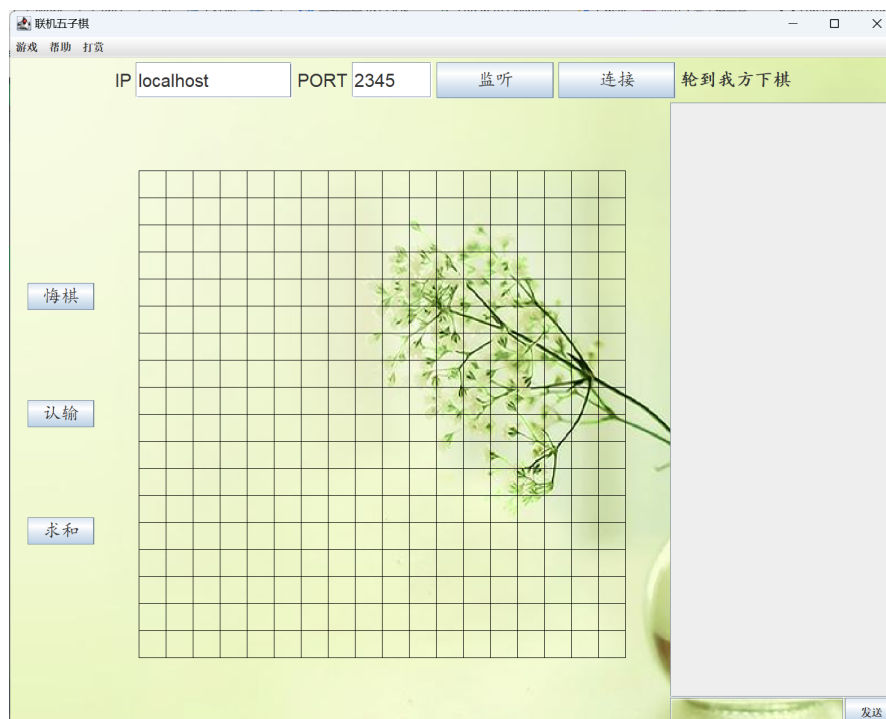


图 4.2 人人对战

### 4.1.3 人机对战

单机模式是用户和电脑对战的界面，用户先手下黑棋。

悔棋按钮：每回合只允许悔棋一次，符合正常的下棋逻辑。

认输按钮：会弹失败对话框，同时会询问是否重新开始，不重开则退出游戏。

求和按钮：会向发出求和询问，一旦确认求和，则判定平局，并询问是否重新开始。

菜单栏：游戏菜单可以进行更换游戏背景，帮助界面可以查询游戏规则，以及制作者信息。

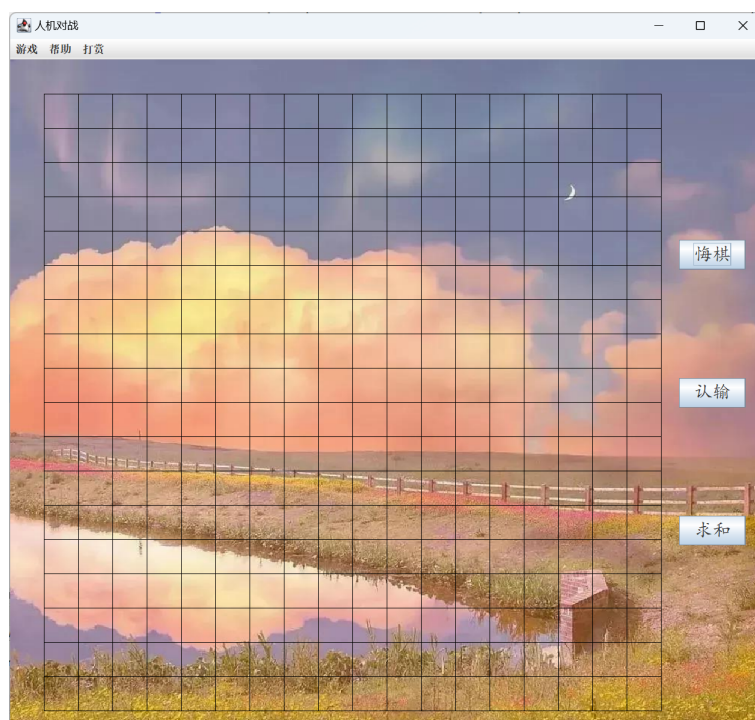


图 4.3 人机对战

## 第二节 按钮功能

功能概述：提供了悔棋、认输、求和三个按钮，单机模式下实现较为简单，网络模式下需要通过 Net 给对手发送消息，根据对手是否同意来进行下一步处理。

### 4.2.1 悔棋

人人对战：先进行判断是否能进行悔棋，可以进行一次悔棋，则 Control 调用 undoMove 发送消息通知 Net，Control 通知 Model 调用函数清除最后下的一个棋子。

人机对战：简化版直接调用 undoMove 函数，撤回我方和 AI 方的棋子。

### 4.2.2 认输

联网模式：Control 调用 surrender 发送消息通知 Net，通知 Model 调用 surrender 弹出对话框告知玩家对方赢棋，并询问是否重开棋局。



单机模式：简化版直接调用 `surrender` 函数，告知 AI 胜利，询问是否重开。

#### 4.2.3 求和

联网模式：Control 调用 `offerDraw` 发送消息通知 Net 对手想要求和，后续 Net 调用消息处理函数，弹出确认框询问是否同意求和，如果同意则使用 Net 发送同意消息，弹出对话框表示平局，并询问是否重开棋局。

单机模式：简化版直接调用 `surrender` 函数，告知平局，询问是否重开。

### 第三节 菜单栏

游戏：包含更换背景选项，点击后可以读取电脑图片进行更换背景。

帮助：包含两个下拉菜单，游戏规则点击可以查看游戏规则，关于我们介绍作者。

## 第五章 总结和展望

### 第一节 全文总结

在本次游戏开发之旅中，我深刻体会到了游戏背后的复杂性和开发者的辛勤付出。每一款游戏，无论是在内容还是界面设计上，都是一项庞大的工程。这次经历彻底改变了我对游戏的看法，让我对这一领域有了全新的认识。

在开发过程中，我意识到自己知识的浅薄，许多在课堂上没有掌握好的知识点和内容在实践中暴露无遗。同时，这次经历也加深了我对 Java 语言精髓的理解。回顾自己的程序，我发现还有许多不足之处，如界面美化、功能完善以及代码的高内聚低耦合原则等，都需要进一步改进。

尽管如此，面对最初的挑战时的恐惧和犹豫已经消失，看到自己的程序成功运行，我感到非常高兴。这次经历不仅积累了我论文写作的经验和方法，还提高了我使用  $\text{\LaTeX}$  的技巧。我意识到，未来的挑战仍然很多，需要学习的知识也很多。因此，我将更加注重日常的积累，不畏惧困难，勇于探索，不断前进。

### 第二节 未来展望

虽然 Java 课程已经结束，但我对这门语言的热爱不会停止。目前我所学的可能只是 Java 的冰山一角，但已经为我打开了新世界的大门。我计划在接下来的假期和学习中继续练习、巩固和运用 Java。根据自身的能力，我将选择更深入地学习 Java。这不是结束，而是一个美好的开始。

## 参考文献

- [1] 郑莉. Java 语言程序设计. 清华大学出版社有限公司, 2006.
- [2] 吴锦涛, 薛益鸽. Java 语言的桌面程序开发——以推箱子为例. 智能计算机与应用, 2018, 8(2): 179–183.
- [3] 田丰维, 冯永安. 计算机软件 JAVA 编程特点及其技术应用. 收藏, 2019, 21.
- [4] 胡伏湘, 雷军环. Java 程序设计实用教程. 清华大学出版社有限公司, 2005.
- [5] 朱文偉. Visual C++ 2013 從入門到精通: 影片教學版. 清華大學出版社, 2018.
- [6] Robert C Martin. Java and C++ A critical comparison. Technical Note, Object Mentor, 1997.
- [7] 陈刚. Eclipse 从入门到精通 (Java 开发利器)(配光盘). 清华大学出版社有限公司, 2005.
- [8] 曹大有, 赵韬. Java 基于可观察者技术的 MVC 编程的实现方式. 计算机系统应用, 2007, 16(5): 115–119.

## 致 谢

在这个学期即将结束之际，我想向刘嘉欣老师表达我最深切的感激之情。在您的耐心教导和悉心指导下，我对 JAVA 这门强大而灵活的编程语言有了深刻的理解和掌握。您的讲解不仅清晰透彻，而且总能激发我们对编程的热情和兴趣。在您的课堂上，我不仅学到了 JAVA 的理论知识，更重要的是，您教会了我们如何去发现问题、解决问题，以及如何勇于探索新的知识领域。您的教学方法使我受益匪浅，特别是在编程实践中，我学到了许多实用的技巧和宝贵的经验。这些技能和经验将伴随我未来的学习和职业生涯，成为我宝贵的财富。

同时，我也要感谢我的同学们，感谢你们在学习上给予我的热心帮助和支持。我们一起讨论问题、共同进步，这种团队合作的精神让我深受感动。

我还要感谢我的父母，感谢你们无条件的爱和辛勤养育。在我追求知识和成长的道路上，你们始终给予我最大的支持和鼓励，是我不断前进的动力源泉。在我遇到困难和挑战时，是你们给予我力量，让我有勇气面对一切。

在未来的日子里，我将带着这份感激之情，继续努力学习，不断进步。我会将所学的知识运用到实践中，用行动来回报你们的教导和支持。再次感谢刘嘉欣老师、我亲爱的同学们以及我敬爱的家人，你们是我人生旅途中最宝贵的财富。