**南开大学**

JAVA编程课程结课项目

中文题目：基于JAVA的五子棋交互游戏

外文题目：**Interactive GOBANG Game Based on Java**

学 号： **2211757**

姓 名： **赵元鸣**

年 级： **2022级**

系 别： **计算机系**

专 业： **计算机科学与技术**

完成日期： **2023年12月21日**

指导教师： **刘嘉欣**

摘 要

本五子棋交互游戏以 JAVA 编程语言为基础，融合了图形化编程、计算机网络等多方面的知识，是一个综合性的开发项目。它采用了 Swing 和 AWT 组件库，利用了 Java 标准库中的丰富资源，包括 java.net 和 java.io 等经典类库，旨在扩展个人的编程技能和知识面。

在项目开发初期，立足于提升编程能力的目标，本人将经典的 Swing 和 AWT 组件库作为开发的基石。利用这些强大的图形化编程工具，开始构建游戏的用户界面，设计并实现了游戏所需的各种元素，例如按钮、文本框等，同时借助这些组件实现了用户友好的交互体验。

此外，还运用了 Java 标准库中的网络类和输入输出类，如 java.net 和 java.io，实现了基于网络的玩家交互功能。通过这些类库提供的方法和功能，本项目成功地建立了两个玩家之间的通信渠道，实现了游戏玩家间的交互式游戏体验。

在这个项目中，游戏界面设计涵盖了多项核心功能。首先，实现了完整的棋盘功能，玩家可以在棋盘上自由下棋，规则清晰，游戏流畅。其次，我加入了玩家准备功能，让玩家在游戏开始前有充足的准备时间。同时，悔棋功能的加入为玩家提供了更多的策略性选择，增添了游戏趣味。

除此之外，通过实现聊天功能，玩家可以在游戏过程中进行交流，增强了游戏的社交性和互动性。重新开始游戏的功能则使得玩家可以在一局结束后迅速重新开始新的对局，提高了游戏的连续性和娱乐性。

这些功能的完善不仅丰富了游戏的体验，也展现了项目的多样性和全面性。每项功能的实现都经过了精心设计和充分测试，为玩家提供了更加流畅、便捷、富有乐趣的游戏体验。

在项目开发中，着重实现了棋盘绘制和界面的分部分类等核心功能。棋盘的精准绘制和设计是项目的基础，确保了游戏规则的准确执行和玩家良好的游戏体验。同时，界面分部分类的设计让不同功能模块清晰可见，使得代码结构更为清晰和可维护。

此外，项目中的网络连接与线程交互也是关键。借助java.net类库，成功搭建了游戏玩家之间的网络连接，使得玩家可以在不同主机之间实现游戏互动。通过线程间的合理交互，实现了多线程下的游戏运行和数据传输，确保了游戏的稳定性和高效性。

这些功能的实现不仅突显了项目的技术深度和复杂性，同时也为项目的稳定运行提供了可靠保障。经过不断优化和调试，确保了项目的性能和用户体验，使得游戏在网络环境下能够稳定、流畅地运行。

总的来说，本五子棋项目不仅仅是一个简单的游戏开发，更是一个基于 Java 编程语言所涉及知识综合运用的项目。通过使用 Swing、AWT、java.net 和 java.io 等类库，本人图形化编程、网络通信等方面得到了丰富的实践经验，同时也进一步拓展了对 Java 标准库的深入理解和应用能力。这个项目不仅是对编程技能的锻炼，更是对创造性思维和个性化编程的实践。尤其是目前互联网经济依然庞大，网络游戏行业经久不衰，作为计算机专业的学子更该了解游戏的开发流程、设计理念、用户体验等内容。此也是本项目的意义之所在

**关键字**：JAVA编程能力、开发实践、创新能力

目 录

摘要.............................................1

目录..............................................3

第一章 引言.......................................5

第二章 游戏界面概览...............................6

2.1棋盘界面.............

..........................7

2.2 信息显示界面...................................7

2.3聊天界面........................................7

第三章 实现类简介...............................9

3.1 Start类.......................................9

3.2 Vars类.......................................10

3.3 Chess类.......................................10

3.4 ChessModel类..................................11

3.5 ChessBoard类..................................11

3.6 PaintControl类................................12

3.7 InfoPanel类...................................12

3.8 NetHelper类...................................13

3.9 TimerUpdater类................................13

第四章 绘制棋盘和下棋的实现......................15

4.1 用户操控棋盘..................................15

4.2 棋盘在UI界面的绘制...........................16

第五章 信息显示模块的实现........................17

5.1 显示信息以及对应的域..........................17

5.2 显示信息获取的实现............................18

5.3悔棋功能的实现.................................19

5.4倒计时模块的实现...............................20

5.5显示玩家模块的实现.............................22

第六章 网络交互连接的实现........................24

第七章 聊天模块的实现............................28

7.1 NetHelper类中的相关方法.......................28

7.2 ChatPanel类...................................29

第八章 实践经验总结和改进........................31

第一章 引言

当代计算机科学与技术领域中，图形化开发不仅仅是技术实践，更是一种创造性的表达方式。本文所述的五子棋交互游戏项目是在这种理念的引导下诞生的。作为一项结合了图形化编程、网络通信以及游戏设计的综合性项目，它旨在展现我在编程领域的技术熟练程度和创造力。

通过使用Java编程语言，本项目力求代码的简洁有效，并将其融合到一个高度交互性和趣味性的游戏中。在实现这一目标的过程中，本项目充分运用了Java的核心类库，如Swing和AWT组件库，以及java.net和java.io等用于网络通信和输入输出的类库。这些类库不仅为我提供了丰富的功能和工具，还让本项目能够在项目中灵活应用并实现各种功能模块。

游戏界面的设计和功能实现是项目的核心。本项目努力确保棋盘功能、玩家准备、悔棋、聊天以及重新开始游戏等功能都能顺利实现。同时，为了保证游戏的稳定性和互动性，我致力于完善网络连接和线程交互部分，以确保玩家能够流畅地进行多主机游戏。

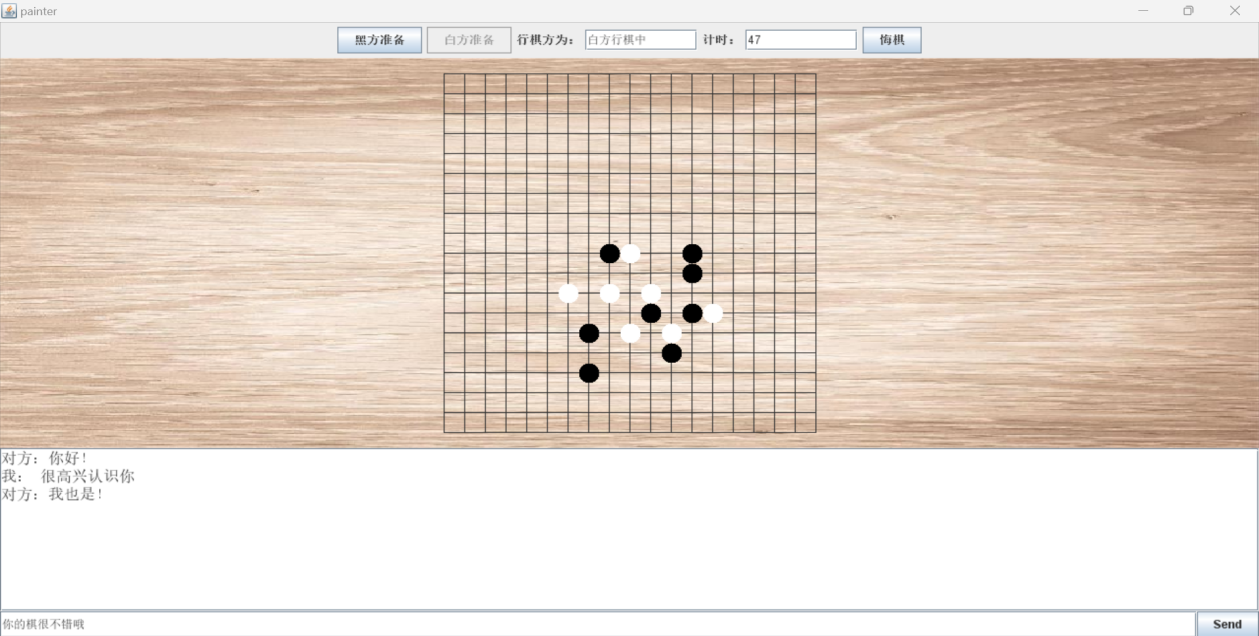
在完成这一项目的过程中，本项目注重了程序的结构和设计。通过棋盘绘制、界面的分部分类以及网络连接与线程交互等功能的实现，我更深入地理解了面向对象编程的思想，并通过模块化设计和逻辑清晰的代码结构，提升了代码的可维护性和可拓展性。

本文将详细阐述五子棋交互游戏项目的开发过程、设计思路和技术实现。通过这一项目的完成，本项目得以展现学生在软件开发方面的技术水平和创造力，并对面向对象编程、图形化编程和网络通信等领域有了更加深入的理解

和实践

第二章 游戏界面概览

这个游戏界面本着逼近现实、方便用户的理念，设计了总共三个界面，信息显示模块、棋盘模块、聊天模块，以及提示悔棋、胜利、重开的提示框模块等。效果图如下。



*图2.1：界面效果展示图*

五子棋游戏的设计遵循了一个核心理念：逼近现实、注重用户体验。它巧妙地划分为三个主要界面：信息显示模块、棋盘模块以及聊天模块。这些界面不仅以简洁直观的方式展示信息，还将游戏元素完美地融入其中。信息显示模块清晰地呈现了当前游戏状态，包括玩家轮次、倒计时、以及各种游戏提示。棋盘模块是游戏的核心，它提供了一个规范的五子棋棋盘，让玩家们在其中落子、竞技。而聊天模块则为玩家提供了交流互动的空间，使得游戏不仅仅局限于棋局，还增添了一份社交的趣味。

同时，游戏还配备了各种提示框模块，提供了对于悔棋、胜利和重新开始等操作的友好提示。这种设计理念不仅使得游戏界面更加简洁易懂，还使得玩家能够在游戏过程中获得及时的反馈和指导。这些提示框以清晰的方式向玩家传达重要信息，增强了游戏的可玩性和互动性。

整体来看，这三个界面相互配合、交互链接，为玩家呈现了一个富有活力且易于操作的游戏环境。游戏的界面设计注重用户体验，力求让玩家在游戏中感受到简单直观、信息丰富、操作便捷等特质。通过这种设计理念，游戏在展现五子棋的魅力的同时，也提升了用户参与度和沉浸感。效果图所呈现的界面设计，展现了这款游戏注重细节、致力于提供愉悦游戏体验的态度。

2.1 信息显示界面

其中信息显示模块是五子棋游戏界面的关键部分，旨在清晰、准确地展示游戏状态与关键信息。其设计包含游戏状态信息和提示内容，以便玩家随时了解游戏进展和操作需求。游戏状态信息区域包括当前玩家轮次、计时情况等，确保玩家了解自身和对手状态；而游戏提示信息则提供倒计时、悔棋操作和胜利提示等，帮助玩家更好地参与游戏。这一模块的设计旨在简洁明了地展现游戏信息，为玩家提供更流畅的游戏体验。

2.2棋盘界面

棋盘模块是五子棋游戏界面的核心组成部分，以精美的图形呈现游戏棋盘。其设计基于行列交叉的网格结构，清晰划分了可落子的区域。每个交叉点都标识了玩家可下棋子的位置，以便玩家进行操作。这个模块通过优雅的设计和醒目的标识，有效地传达了游戏规则和操作需求。棋盘模块的实现不仅保留了传统五子棋的基本规则，还在视觉上提供了简洁明了的界面，使玩家能够专注于游戏本身，享受到流畅的游戏体验。

2.3聊天界面

聊天模块是五子棋游戏的一个亮点，为玩家提供了实时交流和互动的功能。设计上，这个模块在游戏界面的侧边或底部位置，以文本框形式展现出来，方便玩家输入和显示聊天内容。它能够接收玩家的键盘输入，支持文字消息的发送和接收。这个模块不仅仅是一个简单的聊天框，更是游戏社交和互动的核心。玩家可以在其中发送消息、交换策略、表示友好，甚至通过互相激励或称赞来增强游戏体验。

这三个模块共同构成了游戏的整体界面设计，注重了信息传递的清晰性、游戏操作的便捷性和互动交流的实时性。整体设计风格简洁大方，使得玩家能够更专注于游戏本身，享受到愉悦的游戏体验。

第三章 实现类简介

3.1 Start类

Start 类是这个五子棋游戏的启动点。它实例化了 JFrame 对象作为游戏的主窗口，并设置了关闭窗口时的行为以确保可以正常退出游戏。通过 BorderLayout 布局管理器，chatPanel 被放置在窗口的南部，infoPanel 放置在北部，chessBoard 放置在中央。通过设置各个面板的首选大小，确保它们在窗口中占据合适的空间。这个类的目的是组织并展示游戏的基本界面布局，为游戏提供了起点。它搭建了一个容纳游戏功能模块的框架，让用户可以在界面上交互并享受游戏乐趣。此外，这个类也是游戏整体逻辑的入口，为后续游戏过程提供了必要的视觉和交互支持。

3.2 Vars类

Vars 类是这个五子棋项目的核心管理类之一。它通过静态属性存储了游戏中各个关键组件，为整个项目提供了方便的访问入口。control 属性实例化了 PainterControl 对象，管理着游戏的绘画控制。infoPanel 属性表示信息面板，用于展示游戏的相关信息。chessBoard 属性是五子棋棋盘，采用 ChessBoard 类的实例，并接收一个背景图像作为参数，提供了棋盘的基本绘制和显示。chatPanel 属性代表聊天面板，通过 ChatPanel 类实现了聊天功能。model 属性是 ChessModel 类的实例，管理着游戏中的棋局模型。net 属性则是 NetHelper 类的实例，处理了与网络通信相关的操作。

这个类充当了游戏中各个关键组件的管理者，提供了统一的访问点，方便其他模块之间的交互和数据传递。通过集中管理游戏中的重要组件，Vars 类使得整个项目更易于维护和扩展，并能够更加灵活地实现各种功能。

3.3 Chess类

Chess 类是五子棋项目中的一个关键类，用于描述棋盘上的棋子。在这个类中，p1 和 p2 表示棋子所在的位置坐标，这些坐标信息确定了棋子在棋盘上的放置位置。

Chess 类还包含了描述棋子的其他属性。size 属性表示棋子的大小，而 color 属性代表了棋子的颜色。通过设置不同的 size 和 color 属性，可以改变棋子的外观和显示效果。

该类的构造函数接受位置坐标、大小和颜色等参数，创建一个具有指定属性的棋子对象。通过这些属性，Chess 类为游戏中的棋子提供了基本的特征描述，使得每个棋子对象能够准确地表示在棋盘上的位置、大小和颜色。

3.4 ChessModel类

ChessModel 类是五子棋项目中用于管理棋盘状态和棋子的类。该类具有重要的属性和方法，用于跟踪棋盘上棋子的状态、记录最后下棋的位置以及实现棋子的放置和重置。

在 ChessModel 类中，通过一个二维数组 chess 来表示棋盘的状态，使用数字表示不同颜色的棋子。其中，WIDTH 常量定义了棋盘的宽度，约束了棋盘的大小为 19x19。BLACK 和 WHITE 分别表示黑子和白子的颜色。

类中的 addShape 方法用于向棋盘添加新的棋子，而 putChess 方法负责在指定位置放置棋子，并记录最后下棋的位置信息。通过 resetChessArray 方法可以重置整个棋盘，清除所有棋子状态，让游戏重新开始。

通过 getChess 方法可以获取指定位置的棋子颜色信息。getData 和 getChessArray 方法允许获取当前棋盘状态和所有棋子的信息。

ChessModel 类作为管理棋盘状态和提供棋盘操作的核心部分，为整个游戏提供了必要的功能支持。通过它，实现了对棋盘状态的有效管理，为棋局的进行提供了关键的数据结构和操作接口。

3.5 ChessBoard类

ChessBoard 类是五子棋项目中负责绘制棋盘和棋子的核心部分。这个类继承自 JPanel，通过 paintComponent 方法实现了自定义绘制，可以绘制背景图、棋盘格和棋子。

构造函数 ChessBoard 接收背景图片作为参数，初始化了鼠标监听器和组件监听器。鼠标监听器实现了鼠标释放事件，捕捉玩家的鼠标操作，将其转换为棋盘上的行和列，然后通知 PainterControl 对象处理用户的鼠标点击事件。组件监听器用于监控窗口的大小变化，动态调整棋盘和棋子的大小和位置。

在 paintComponent 方法中，通过 Graphics 对象绘制了背景图片、棋盘格和棋子。drawChessPanel 方法用于绘制棋盘的格子线，而 drawChess 方法则负责根据 ChessModel 中存储的棋子信息绘制相应的黑白棋子。黑棋使用黑色填充，白棋使用白色填充，通过 getChess 方法获取当前位置的棋子状态。

ChessBoard 类作为棋盘绘制的关键部分，实现了在界面上绘制棋盘、监控鼠标点击事件并将其转化为对应的棋盘位置。它与 ChessModel 和 PainterControl 等类相互配合，构成了实现五子棋游戏功能的重要组成部分。

3.6 PaintControl类

PaintControl 类是五子棋项目中负责控制棋盘绘制和游戏规则的核心部分之一。该类包含了处理玩家鼠标点击事件、检查游戏胜利条件以及控制游戏流程的方法。

通过 reportUserPressMouse 方法，PaintControl 捕捉玩家鼠标点击事件，并将其转换为棋盘上的行和列。接着，它使用 Vars.model.putChess 方法在指定的行列位置落子，然后调用 checkVictory 方法检查游戏是否达到胜利条件，若达成则弹出胜利提示框并重启游戏。

checkVictory 方法用于检查胜利条件，它根据当前棋子的位置和颜色，在横向、纵向以及斜向检查是否有五颗相同颜色的棋子连成一线，若满足条件则返回 true。

此外，PaintControl 类还包含一些其他方法，比如 askForRestart 方法用于询问玩家是否重新开始游戏。整体上，PaintControl 类在游戏过程中起到了控制游戏逻辑和处理用户操作的重要作用，实现了游戏的流畅进行和胜负判断。

3.7 InfoPanel类

InfoPanel 类是五子棋项目中负责展示游戏信息和操作按钮的面板部分。该类包含了用于展示当前行棋方和计时、悔棋按钮、连接服务器按钮、监听服务器按钮的组件。

在构造函数中，InfoPanel 设置了标签标识当前行棋方和计时，并创建了用于显示和操作这些信息的文本框、按钮等组件。监听服务器按钮(listenBtn) 和连接服务器按钮(connBtn) 分别启动和连接服务器，同时设置了相应的按钮禁用状态以防止重复连接。regretBtn 用于执行悔棋操作，通过添加监听器实现了点击按钮时向服务器发送悔棋消息的功能。

3.8 NetHelper类

NetHelper 类是一个用于处理五子棋游戏中网络通信的辅助类。它包含了一系列方法，负责建立连接、发送消息、接收消息和处理游戏逻辑等功能。

该类通过 startListen 和 connect 方法分别实现了作为服务器端和客户端的连接建立，并且根据不同的连接状态设置了相应的游戏逻辑。在 startReadThread 方法中，通过多线程实现了对网络消息的实时监听和处理，根据收到的消息类型执行相应的操作，如悔棋、聊天信息接收、计时控制等。其中的 restart 方法用于处理游戏重启的逻辑，通过协调双方玩家的同意状态，重置棋盘并重新开始游戏，或者在双方不同意时选择退出游戏。

此外，该类还包含了发送棋子、聊天信息、计时控制等功能的方法，以及一些用于通知等待中的线程的同步处理逻辑。

3.9 TimerUpdater类

TimerUpdater 类是一个用于更新计时器并与文本框交互的辅助工具。它基于 Java 的 Timer 和 TimerTask 类实现了一个计时器，用于在图形界面中显示游戏的计时信息。

这个类通过定时器每秒执行一次的任务，实时更新游戏的计时文本框。在每次执行任务时，计时器会自动更新时间并将其反映在文本框中，以便玩家可以清晰地看到游戏进行的时间。

另外，它还包含了一些功能：reset() 方法用于重置计时器。stopTimer() 方法用于停止计时器，在游戏超时时，弹出对话框提示哪一方超时，并触发游戏的重启操作。通过这些功能，TimerUpdater 类为五子棋游戏提供了一个可靠的计时器工具，可以及时准确地显示游戏的计时情况，并在必要时触发游戏超时的处理逻辑。

本章介绍了五子棋游戏的核心实现类。Start类是游戏的入口，Vars类是全局变量存储器。Chess类代表棋子，ChessModel管理棋子逻辑，ChessBoard负责棋盘绘制。PaintControl游戏过程与绘制控制器。InfoPanel用于显示游戏信息，而NetHelper处理网络通信。TimerUpdater用于实时更新游戏计时。这些类相互协作，构建了一个完整的五子棋游戏系统，管理着游戏流程、UI显示、游戏规则和网络交互等多个方面，为玩家提供了完整的游戏体验。

第四章 绘制棋盘和下棋的实现

4.1 用户操控棋盘

当用户通过鼠标在 ChessBoard 上点击时，会触发 mouseReleased 方法。在这个方法中，会根据鼠标点击的位置计算出对应的行和列，然后调用 Vars.control.reportUserPressMouse(row, col) 方法。

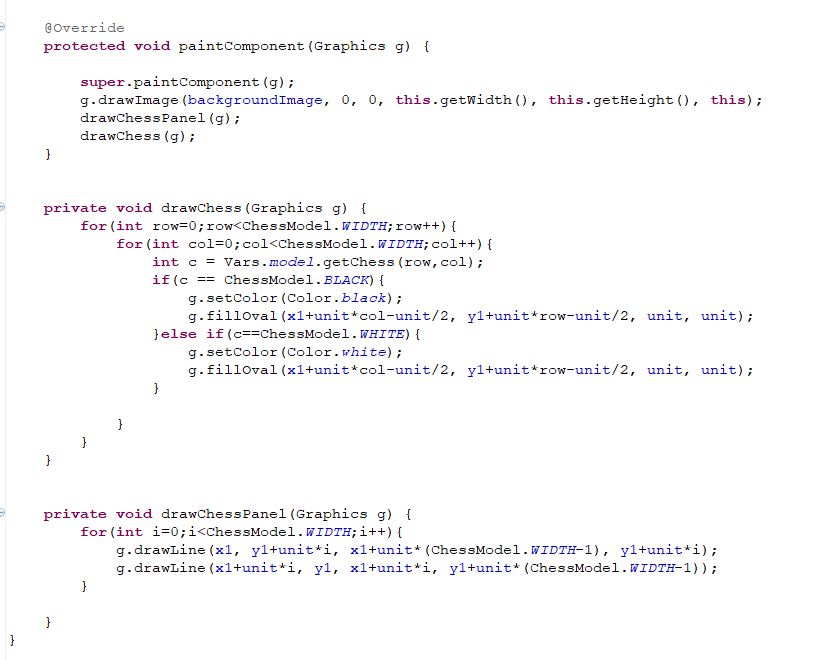


*图4.1 paintControl的reportUserPressMouse（）函数*

在 reportUserPressMouse 方法中，首先检查是否可以落子，如果可以，就执行 Vars.model.putChess(row, col, chessColor) 方法，在棋盘上放置对应颜色的棋子。接着检查是否达成了胜利的条件，若是则弹出胜利提示框，并触发重新开始游戏的操作。如果当前没有胜利，就切换到对方下棋的状态，并更新信息面板显示。

4.2 棋盘在UI界面的绘制

在 ChessBoard 类中的 paintComponent 方法中，首先调用 super.paintComponent(g) 方法清空绘图区域，然后绘制背景图并调用 drawChessPanel(g) 和 drawChess(g) 方法分别绘制棋盘和棋子。其中 drawChessPanel(g) 通过画线的方式绘制棋盘网格，drawChess(g) 则根据 Vars.model 中棋子的信息，在对应位置绘制黑色或白色的圆形棋子。



*图4.2 绘制组件、棋子的函数*

总结一下，ChessBoard类中的paintComponent方法用于绘制棋盘和棋子。首先，通过super.paintComponent(g)清空绘图区域，绘制背景图，然后分别调用drawChessPanel(g)和drawChess(g)绘制棋盘和棋子。drawChessPanel(g)通过画线绘制棋盘网格，而drawChess(g)根据Vars.model中的棋子信息在对应位置绘制黑色或白色的圆形棋子。

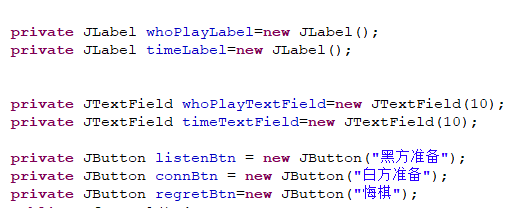
第五章 信息显示模块的实现

InfoPanel 类是游戏界面中负责显示信息的部分，通过不同的 Swing 组件实现了信息的展示和用户交互。这个类利用了 JLabel、JTextField 和 JButton 这些 Swing 组件来呈现不同类型的信息和实现按钮功能。

5.1 显示信息以及对应的域

首先，JLabel 被用来展示静态文本标签。whoPlayLabel 和 timeLabel 分别标识了“行棋方为”和“计时”这两个信息。这些标签通过 setText 方法设置文本内容，以便清晰地显示给用户。

其次，JTextField 用于动态展示文本内容。whoPlayTextField 和 timeTextField 分别显示当前行棋方和计时信息。这些文本框也可通过 setText 方法更新显示的文本内容。



*图5.1 信息显示盘类的域*

最重要的是 JButton 组件，它们是与用户交互的关键。listenBtn 和 connBtn 分别是“黑方准备”和“白方准备”按钮。点击这些按钮会触发 ActionListener 监听器中定义的操作，比如开始监听或发起连接。同时，regretBtn 是“悔棋”按钮，点击它将触发发送悔棋请求的操作。

5.2 显示信息获取的实现

每个按钮都被设置了相应的监听器，当用户点击按钮时，监听器会捕获该事件并执行相应的操作。例如，点击 listenBtn 将启动监听器并为黑方设置准备状态，同时禁用按钮，确保了游戏开始前的准备工作。

InfoPanel 类是游戏界面的一个组成部分，负责展示一些信息和提供用户交互的功能。它包含了用于显示当前行棋方和计时的文本框，以及用于用户操作的按钮。同时，它也与网络连接相关，具体表现在以下几个方面：

listenBtn 和 connBtn 分别对应“黑方准备”和“白方准备”按钮，点击这些按钮会触发网络连接的一系列操作。当用户点击 listenBtn 按钮时，ActionListener 监听器会调用 Vars.net.startListen() 方法，这表示用户想要作为“黑方”准备开始游戏。同时，它还设置了黑方的棋色并禁用了按钮，避免了重复准备。connBtn 的点击事件则调用了 Vars.net.connect() 方法，尝试与指定 IP 和端口进行连接，并设置了白方的棋色。连接成功后，按钮被禁用，防止重复连接。 当用户点击 regretBtn（悔棋）按钮时，触发了 ActionListener 监听器中的操作。在该监听器中，Vars.net.sendRegretMessage() 方法被调用，向对手发送悔棋请求。



*图5.2 获取信息的监听函数的实现*

通过 InfoPanel 类，用户可以方便地参与游戏的交互和准备阶段。这些信息显示组件和按钮为用户提供了直观的界面操作，让用户能够轻松准备、连接，甚至请求悔棋，为游戏体验增添了便利和趣味。

5.3悔棋功能的实现

实现悔棋功能涉及多个类之间的协作。在游戏中，NetHelper 类扮演了关键角色。当玩家选择悔棋时，NetHelper 的 sendRegretMessage() 方法被调用，向对手发送悔棋请求。对手接收到请求后，可以选择同意或拒绝。

如果对手同意悔棋，消息将通过网络传输到玩家端。NetHelper 接收到消息后，会触发 dealWithotherSideRegret() 方法。在这里，会弹出一个确认对话框，询问玩家是否同意对方悔棋。玩家的选择决定了是否执行悔棋操作。

若同意悔棋，调用 undoMyLastMove() 方法，该方法会撤销玩家的上一步棋操作，更新游戏状态并在界面上反映这一变化。反之，若玩家不同意，系统会显示拒绝悔棋的提示信息。

这个过程中涉及了界面的交互、消息的传递和游戏状态的管理。NetHelper 作为网络通信的核心，负责发送和接收悔棋请求。玩家在界面上作出选择后，会触发相应的操作，这些操作会影响游戏状态并在界面上呈现出来。

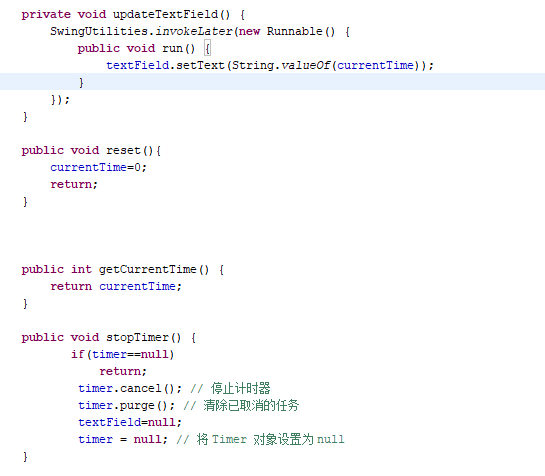
整个流程需要确保通信的准确性和及时性，以及正确地反映玩家的选择和游戏状态。这样，玩家可以方便地与对手沟通、协商，并在游戏中进行悔棋操作，提升游戏体验。

5.4倒计时模块的实现

倒计时模块的实现主要由 TimerUpdater 类完成。该类通过 java.util.Timer 实现定时器功能，每秒更新游戏界面上的倒计时文本框。在每次计时器任务执行时，当前时间会自增，达到设定的时间阈值时触发相应的事件。

当时间到达设定的限制（比如60秒）时，会触发游戏中的超时判断逻辑。这个模块跟踪并管理游戏的时间，并在超时情况下触发相应的游戏结果提示，比如显示哪一方获胜。

该倒计时模块确保准确计时，与游戏状态同步。它通过 Timer 定时执行任务并更新界面，同时也需要处理时间到达阈值的情况，触发对应的游戏结果。这个模块负责游戏中时间的计算与显示，是游戏进行过程中的重要辅助组件。



5.5 显示玩家模块的实现

在这种游戏中，使用一个信息面板（Info Panel）来显示当前是哪一方行棋。这个信息面板可以包含一个标签或文本框，用来显示“行棋方为：黑方”或“行棋方为：白方”这样的信息。

1. 显示行棋方信息： 在信息面板中设置一个标签或文本框来展示当前是哪一方的回合。这个信息会根据当前行棋方的变化而更新。

2. 更新行棋方信息： 在游戏中，当一方完成了下棋操作，你可以调用相关的方法来更新信息面板的内容，以显示轮到哪一方行棋。例如，在点击棋盘上的格子时，如果下了一步棋，就可以根据下棋的颜色来更新信息面板显示的内容。

***代码：***

// 设置为黑方行棋

Vars.infoPanel.setWhoPlayTextField("黑方行棋中");

// 设置为白方行棋

Vars.infoPanel.setWhoPlayTextField("白方行棋中");

3. 界面显示更新触发： 在用户完成下棋动作后，你可以调用相应的函数来更新界面显示。这可以在用户与界面交互时触发，例如当点击棋盘格子后，根据下棋操作的完成情况和当前的行棋方来更新显示信息。

这样的机制允许玩家清楚地知道哪一方当前可以进行操作，增强了游戏的可玩性和交互性。

第六章 网络交互连接的实现

NetHelper 类是实现两个界

面之间通信的关键。这个类中使用了 Socket 编程，它可以建立客户端和服务器之间的连接，允许两个界面之间发送和接收消息。



*图6.1 网络交互类的域*

类中的 Listen() 和 connect() 方法分别用于服务器端和客户端建立连接（即分别是黑方棋手与白方棋手）。Listen() 创建一个 ServerSocket，并通过 accept() 方法等待其他端口的连接请求。而 connect() 方法是在客户端发起连接请求到指定的 IP 地址和端口。



*图6.2 连接函数*

接着，startReadThread() 方法创建了一个线程，不断地读取来自连接的输入流中的消息。这些消息包括棋步信息、聊天消息、重启游戏的请求、计时器的状态等等。收到消息后，根据不同的消息类型进行相应的处理，比如更新棋盘状态、显示聊天内容、处理重启游戏等。

startReadThread() 方法创建了一个新的线程，该线程通过网络连接读取来自另一端的数据。这个方法使用了匿名内部类创建了一个新的线程对象，然后在这个线程中执行数据读取操作。主要的逻辑在这个新线程的 run() 方法里，使用了一个无限循环来持续读取数据。在循环中，它通过 BufferedReader 的 readLine() 方法读取输入流中的数据，并根据读取到的内容来进行不同的处理。

这个方法是一个核心的部分，负责监听和处理来自其他连接的消息。根据接收到的不同消息类型，会进行一系列的操作，如更新界面状态、显示消息、处理棋步、处理计时器状态等。这种在单独线程中监听网络连接的方式允许程序在等待消息到达时继续执行其他任务，而不会被阻塞。这样可以确保程序在等待消息时仍然保持响应性。



*图6.1 监听函数（1）*



*图6.2 监听函数（2）*

此外我们还设置了十余种不同的函数，如sendChess() 和 sendChat() 方法用于向对方发送棋步信息和聊天信息。这样，当一个界面执行棋步操作或者发送聊天消息时，会通过 NetHelper 类将这些信息发送给另一个界面。最后，callForRestart() 和 restart() 方法用于处理游戏重新开始的请求。callForRestart() 发送一个重启的请求消息给对方，而 restart() 则根据对方的回应来决定是否重新开始游戏。以及还有restart(), notifyRestartMessageReceived(), sendRestartMessage(), callForRestart(), sendRegretMessage()等负责交流沟通和重启等的函数，这里就不一一赘述了。

第七章 聊天模块的实现

7.1 NetHelper类中的相关方法

通过 NetHelper 类的 sendChat() 方法，用户可以发送消息，而 NetHelper 的 startReadThread() 方法用于接收对端发送的消息。当接收到消息时，若消息以 "chat:" 开头，NetHelper 将提取出聊天内容并将其显示在 ChatPanel 中。这样，用户可以在界面上输入消息并发送，而对端收到消息后同样可以在自己的界面中看到接收到的聊天内容。

相关域有：

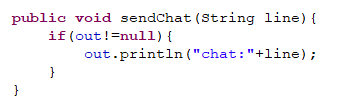
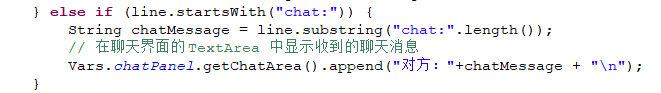
1. private BufferedReader in; 和 private PrintStream out;：用于处理输入输出流，实现消息的接收和发送。

2. protected String oTherSidelastMove; 和 private String lastMove;：用于存储最后一次的聊天消息内容。

相关函数：

public void sendChat(String line)：发送消息的方法。将消息内容作为参数传递给该方法，该方法会将消息发送给对方。

private void startReadThread()：启动读取线程的方法，用于监听对端发送的消息。当接收到消息时，根据消息的前缀，判断是否是聊天消息，并将聊天内容显示在相应的聊天界面中。

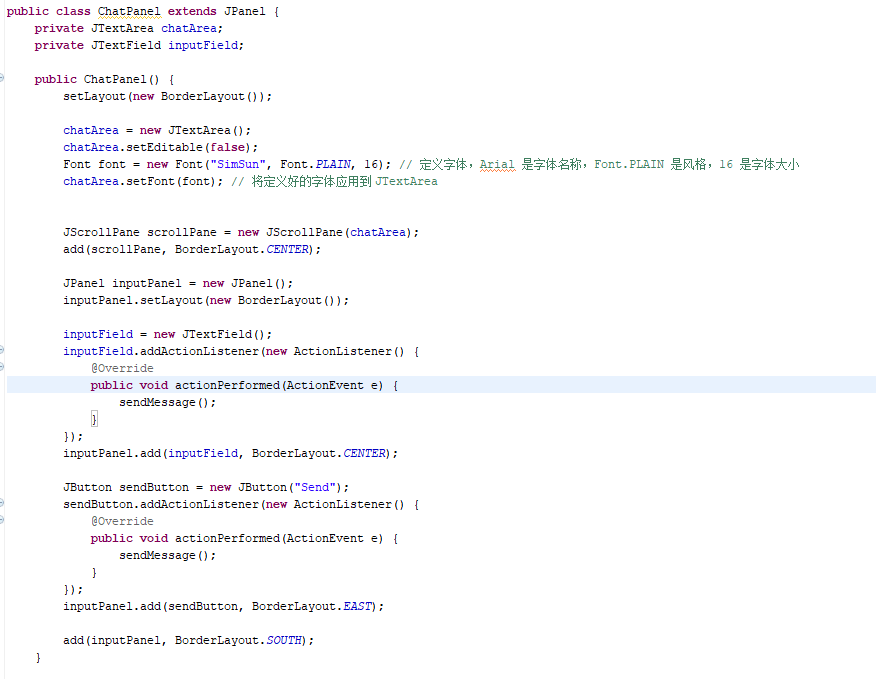
 

*图7.1.2 startReadThread()函数中监听聊天的部分*

*图7.1.1 发送聊天函数*

7.2 ChatPanel类

ChatPanel类用于实现游戏中的聊天功能，让玩家在游戏进行过程中进行交流。 ChatPanel包含一个文本区域，用于显示聊天消息，以及一个输入框或者文本域，让玩家输入发送消息。在接收到其他玩家发送的消息时，ChatPanel会实时更新并显示这些消息。同时这个类会调用NetHelper类中的方法来实现消息的发送和接收，通过网络连接将消息传递给其他玩家。



*7.2.1 ChatPanel类（1）*

***7.2.2 ChatPanel类（2）***

这个类包含了一个不可编辑的文本区域 chatArea，用于展示用户发送和接收到的消息，并且设置了字体和滚动面板。另外，还有一个文本输入框 inputField 和一个发送按钮，用户可以在输入框中输入消息，并通过按钮发送。当用户输入消息并发送时，消息会显示在 chatArea 中，同时也会通过 Vars.net.sendChat(message) 发送给对方。收到对方的消息后，会以 "对方: " 开头显示在 chatArea 中。

第八章 实践经验总结和改进

在这次项目中，我完成了一个完整的JAVA图形化界面的五子棋游戏，项目功能比较完整，但也有许多改进的地方。

有些方法和类可以进一步拆分或组织，以提高可读性和可维护性。可能需要更多的注释来解释复杂的逻辑或功能；游戏界面布局可以更进一步地优化界面布局和视觉设计，使界面更具吸引力和易用性；在网络通信和用户输入方面，可能需要更完善的异常处理机制。例如，在网络连接出现问题或输入不合法时，提供更友好的用户提示和处理方式；部分代码可以进行优化以提高性能或简化逻辑。例如，可能有些重复的逻辑可以抽象成函数或方法，减少代码冗余；也可以增加更多的用户，启用更多的线程。

虽然实现了基本的功能，但可以考虑增加一些额外的功能或特性，例如游戏设置、游戏规则介绍等，以提升用户体验，增加背景音乐、点击棋子的声音、登录界面等，但由于精力问题，我会在后面思考这些想法的实现。

总的来说，这次项目的实践对我个人是一次锻炼也是一次挑战，其中遇到了各种各样意想不到的问题，调了不知多少个bug，最后也是收获满满。临近学期末，感谢老师一个学期的教导与陪伴，希望我以后能写出和老师一样好的java代码，也祝南开的java课越办越好！