**南 开 大 学**

Java结课论文

中文题目：双人连网五子棋

外文题目：Online two people Gobang

学号： 2010699

姓名： 曹维伦

年级： 2020级

专业： 信息安全

系别： 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： 2021/12/23

**摘 要**

Java语言是一种优秀的程序语言。它最大的优点就是与平台无关，在Windows 9x、Windows NT、Solaris、Linux、MacOS以及其它平台上，都可以使用相同的程序代码。“一次编写，到处执行”的特点，使其在因特网上广泛采用。

Java语言有许许多多有效的特性，吸引着程序设计师们，最主要的有以下几个：简洁有效、高可移植性、面向对象、解释型、适合分布式计算、拥有较好的效能、健壮且防患于未然、具有较高的安全性等等。

学习了一个学期的Java语言，算是掌握了基本的线程、图形化用户接口设计以及联网功能等知识，并将课程所学用于设计一款双人连网五子棋游戏。

实现的功能大致包括了开局、悔棋、复盘、倒计时、投降以及聊天等等功能，亦可单机游玩。

关键词：

Java游戏；联网对战；聊天；五子棋

**Abstracts**

Java is a wonderful programming language. Platform-independent is the most significant advantage of Java, that is to say, you can use the same code on every different operation-systems and platforms. The features that "Design one time, use it everywhere." makes Java widely used on the internet.

Java has multiple features, attracts lots of programmers, mainly features following below: clean, easy to move, object-oriented, suitable for distributed computing, works more efficiently, stronger, and higher security.

After learning a whole semester of Java, I somewhat understand the basic knowledge of thread, graphical user interface and internet, and then put everything I learn into designing an online two people Gobang.

Basic function including start, redo, view last game, count down, surrender, and chat. Single player mode is also available.

Key words: Java games, internet gaming, chat, Gobang.

目 录

[摘 要 2](#_Toc91272955)

[Abstracts 3](#_Toc91272956)

[第一章：绪论 6](#_Toc91272957)

[一、 互联网趋势与Java： 6](#_Toc91272958)

[二、 五子棋简介： 6](#_Toc91272959)

[第二章：Java与开发软件的介绍 8](#_Toc91272960)

[**一、** Java具体介绍 8](#_Toc91272961)

[Java的关键特性包括： 8](#_Toc91272962)

[Java的缺点也包括： 9](#_Toc91272963)

[二、 IntelliJ IDEA开发环境简介 9](#_Toc91272964)

[1. 程序员意图支持 10](#_Toc91272965)

[2. 智能的选取 10](#_Toc91272966)

[3. 丰富的导航模式 10](#_Toc91272967)

[4. 历史记录功能 10](#_Toc91272968)

[5. 对重构的优越支持 10](#_Toc91272969)

[6. 编码辅助 10](#_Toc91272970)

[7. 动态语法检测 10](#_Toc91272971)

[8. 代码检查 10](#_Toc91272972)

[9. 对JSP的完全支持 10](#_Toc91272973)

[10. 智能编辑 10](#_Toc91272974)

[11. 预置模板 10](#_Toc91272975)

[12. 完美的自动代码完成 11](#_Toc91272976)

[13. 版本控制完美支持 11](#_Toc91272977)

[14. 不使用代码的检查 11](#_Toc91272978)

[15. 智能代码 11](#_Toc91272979)

[16. 正则表达式的查找和替换功能 11](#_Toc91272980)

[第三章：程序框架 12](#_Toc91272981)

[一、 Main (主程序) 12](#_Toc91272982)

[二、 North Panel (联网用北面板) 12](#_Toc91272983)

[三、 Chess Panel (棋盘面板) 12](#_Toc91272984)

[四、 East Panel (装载按钮与倒计时的东面板) 12](#_Toc91272985)

[五、 Btn Panel (东面板内装载按钮的面板) 12](#_Toc91272986)

[六、 Time Thread (用于计时的线程) 12](#_Toc91272987)

[七、 Replay Chess Frame (复盘时跳出的窗口) 12](#_Toc91272988)

[八、 Replay Chess Panel (复盘时重现上一局所用的棋盘) 12](#_Toc91272989)

[九、 Chat Panel (聊天用面板) 12](#_Toc91272990)

[十、 Chess Point (纪录棋子属性) 12](#_Toc91272991)

[十一、 Controller (操控所有操作的操控类) 12](#_Toc91272992)

[十二、 Model (存放重要数据的棋盘模型) 12](#_Toc91272993)

[十三、 Net (用于联网与交互的助手) 12](#_Toc91272994)

[十四、 Vars (所有类的代理) 12](#_Toc91272995)

[第四章：程序详细介绍 13](#_Toc91272996)

[一、 Main – 主程序： 13](#_Toc91272997)

[二、 North Panel – 北面板： 13](#_Toc91272998)

[三、 Chess Panel – 棋盘面板： 13](#_Toc91272999)

[四、 East Panel – 东面板： 14](#_Toc91273000)

[五、 Btn Panel – 按钮面板： 14](#_Toc91273001)

[1. 单人开局： 14](#_Toc91273002)

[2. 悔棋： 15](#_Toc91273003)

[3. 复盘： 15](#_Toc91273004)

[4. 结束此局： 15](#_Toc91273005)

[5. 投降： 15](#_Toc91273006)

[六、 Time Thread – 定时器线程 15](#_Toc91273007)

[七、 Replay Chess Frame – 复盘界面： 16](#_Toc91273008)

[八、 Replay Chess Panel – 复盘棋盘： 16](#_Toc91273009)

[九、 Chat Panel – 聊天面板： 16](#_Toc91273010)

[十、 Chess Point – 棋子属性： 16](#_Toc91273011)

[十一、 Controller – 控制器： 16](#_Toc91273012)

[1. 连接(listen, connect)： 16](#_Toc91273013)

[2. 下棋： 17](#_Toc91273014)

[3. 单人开局： 17](#_Toc91273015)

[4. 悔棋： 17](#_Toc91273016)

[5. 复盘： 17](#_Toc91273017)

[6. 投降： 18](#_Toc91273018)

[7. 聊天： 18](#_Toc91273019)

[十二、 Model – 模板： 18](#_Toc91273020)

[1. 下棋： 18](#_Toc91273021)

[2. 判断输赢： 18](#_Toc91273022)

[3. 悔棋： 18](#_Toc91273023)

[4. 判断是否能毁棋： 18](#_Toc91273024)

[5. 复盘： 18](#_Toc91273025)

[6. 清除棋盘： 19](#_Toc91273026)

[十三、 Net – 联网与交互类： 19](#_Toc91273027)

[1. Listen： 19](#_Toc91273028)

[2. Connect： 19](#_Toc91273029)

[3. 处理交互事件的线程： 19](#_Toc91273030)

[十四、 Vars – 代理： 19](#_Toc91273031)

[第五章：总结与展望 20](#_Toc91273032)

[一、总结： 20](#_Toc91273033)

[二、展望： 20](#_Toc91273034)

[参考文献 20](#_Toc91273035)

[致谢 20](#_Toc91273036)

# 第一章：绪论

1. 互联网趋势与Java：

如今我们处在一个互联网时代，计算机已经成为家家户户不可缺少的物品，与计算器、互联网挂钩的专业与知识也越来越受欢迎，计算器相关的专业在考大学中也愈来愈多人选择，如计算器科学与技术、信息安全、物联网等等。市场上许多公司对计算器、软件开发的人才亦是供不应求。不只是学生，许许多多的社会人士也争相学习相关知识。因此，对开发语言的选择也相对重要。

而Java作为一款面向对向的编程语言，最早其实是太阳(Sun)公司发明的一种跨平台的语言。他和C语言不同的是，Java是一种面相对象的程序语言。他最大的特点就是通过Java虚拟机实现跨平台效应。Java语言可以说是当今公司用的最多的一种语言。基本很多大中小型公司的提供数据的后台数据接口都是用这种语言编写的。而世界上最大的移动操作系统安卓操作系统的开发也是用Java语言来开发。同时Java语言也是非常安全的，由于Java通常用于网络环境，因此Java提供了一种防止恶意代码攻击的安全机制。除了Java语言的许多安全功能外，Java还在通过网络下载的类中具有安全机制(Class Loader)。所以说，如果你想找一份好一点的工作，或者找工作更容易一点，学习Java是一个不错的选择，不仅具有分布式、可移植姓、高性能与多线程等特点，还有提供自动的废弃物收集，因此程序设计师不必担心内存管理，且与其他语言相比也更加通俗易懂且操作性也更强，也除去了C/C++复杂的指针，更有宽阔的就职市场，诸多优点使其成为当今相当热门的一款编程语言。

1. 五子棋简介：

而这次使用Java设计的五子棋游戏作为一款老少皆宜的游戏，最早可以追溯到上古时期有所谓的”尧造围棋”， 有关早期五子棋的文史记载与围棋多有重合，因为在古代，五子棋的棋具与围棋是完全相同的。在上古的神话传说中有「女娲造人，伏羲作棋」一说，《增山海经》中记载：「休舆之山有石焉，名曰帝台之棋，五色而文状鹑卵。」，又《新唐书·李泌传》中道，「方若棋局，圆若棋子」。宋代，人们将棋盘上的线道谓之「秤」，线道间谓之「方」即「格」《博弈论》中道：「然其所志不出一枰之上，所务不过方之间，即指此棋也。」

不仅在中国盛行，经由高丽在日本元禄时代传到日本的，马上就成为在日本受到人们欢迎的棋类。自1899年日本棋士黑岩泪香证明了原始规则的五子棋先下必胜后，五子棋迈入一条不断改良的道路，经过数十年的修改、验证、再修改，最终发展出加入禁手的五子棋，并经过公开征名，称为连珠，因此规则在日本成型，又称为日式规则或连珠规则。

原始规则依然有人在玩，也被称为无禁规则、自由规则，但已有AI人工智能验证了黑棋(先手者)有一定的必胜下棋方法，至于詳細方法则离题太远，这边就不再赘述。

**第二章：Java与开发软件的介绍**

1. Java具体介绍

Java是目前世界上最流行的计算机编程语言，是一种可以编写跨平台应用软件的面向对象的程序设计语言，也是当今使用率最高的编程语言。Java有着自己独特的优势：语言简单、是一个面向对象、分布式应用并且安全、体系结构中立并且可移植，最重要，它是一个动态语言 。Java自1995年问世以来，因其卓越的通用性、高效性、平台移植性和安全性等特性，成为全球范围内应用范围最广的开发语言，而且即使历经十余年发展仍然在行业内保持着“常青树 ”的地位。Java有着较广泛的市场，几乎市面上你能想到的软件或者硬件都离不开Java的开发使用，比如我们熟悉的购物类网站、社交类网站或APP，PC端的客户端的比如学校、医院、大型企业等等客户管理系统等等，只要后台或前台要

涉及到代码的，都可以通过Java开发实现。目前正处在大数据时代，Java语言依然是大数据开发的主要编程语言。所以伴随着大数据技术的逐渐落地应用，Java语言在大数据时代依然有大量的发展机会

Java的关键特性包括：

1. 高可移植性：

对于一个程序设计师而言，写出来的程序如果不需修改就能够同时在Windows、MacOS、LINUX等平台上执行，简直就是美梦成真的好事！而Java语言就让这个原本遥不可及的事离我们越来越近。使用Java语言编写的程序，只要做较少的修改，甚至有时根本不需修改就可以在不同平台上执行了，Java绝对会是跨平台的一门程序语言。

举个例子，不同平台上对资料型别的大小有不同的规定。在有的平台上，整型（int）是用一个16位二进制制数表示，而有的平台上则是使用一个32位二进制制数表示的。C/C 语言并没有遮蔽掉这个由不同平台造成的不同。这样如果在一个使用16位二进制制数表示整型的平台上开发的C语言源程序，要在一个使用32位二进制制数表示整型的平台上执行，简直就是噩耗。

而在Java语言中，就遮蔽掉了这种由平台造成的不同。Java中的整型数是一定是由32位二进制制表示的，Java虚拟机(Java virtual machine 简称JVM)将这个差别消化了。当我们想将一个使用16位二进制制数表示整型的平台上开发的Java语言源程序，移植到一个使用32位二进制制数表示整型的平台上执行，就不用修改了。Java语言具备这种高移植性是一种必然，因为它的设计理念就是“一次编写，到处执行”嘛。

1. 具有较高的安全性：

由于Java语言在设计时，在安全性方面考虑很仔细，做了许多探究，使得Java语言成为目前最安全的一种程序设计语言。尽管Sun公司曾经许诺过：“通过Java可以轻松构建出防病毒、防黑客的系统”，但“世界上没有绝对的安全”这一真理是不会因为某人的许诺而失灵验的。就在JDK (Java Development Kit)1.0释出不久后，美国Princeton（普林斯顿）大学的一组安全专家发现了Java 1.0安全特性中的第一例错误。从此，Java安全方面的问题开始被广泛关注。不过值得庆幸的是，至今所发现的安全隐患都很微不足道，而且Java开发组还宣称，他们对系统安全方面的Bugs非常重视，会对这些被发现的进行立即修复。而且由于Sun公司开放了Java直译器的细节，所以有助于通过各界力量，共同发现、防范、制止这些安全隐患，因此Java的安全性还是十分值得信任的。

1. 发展时间悠久：

周边相关的套件较多元、完善。套件可以理解为工具：就像建筑师盖房子，工具越多也就意味者相对越方便。很多时候开发一个新的应用程序，把套件引进来再加几行code就完成了，不仅选择多元，开发速度也能大幅缩短(因为很多东西早已都有套件了，不用自行开发)。

且时间已然长到足够其形成了一个庞大且稳定的应用生态。对于企业而言，采用Java方案的风险相对较小

但Java当然也不是没有缺点的，相较于其他语言

Java的缺点也包括：

1. 效能问题

由于java是靠虚拟机运行，同时使用着大量的内存，使的运行速度与启动时间相对较慢。

1. 不支持底层操作

这项缺点其实来自于优点，因为Java是跨平台的，使用虚拟机运行，因此无法接近操作系统，也就没法和操作系统的底层打交道了。

1. 灵活性

因为Java删除了指针，改用其他方式替代，因此较不如C/C++等语言来的那般灵活。

1. IntelliJ IDEA开发环境简介

IDEA是java编程语言开发的集成环境。IntelliJ在业界被公认为最好的java开发工具，尤其在智能代码助手、代码自动提示、重构、JavaEE支持、各类版本工具(git、svn等)、JUnit、CVS整合、代码分析、 创新的GUI设计等方面的功能可以说是超常的。IDEA是JetBrains公司的产品，这家公司总部位于捷克共和国的首都布拉格，开发人员以严谨着称的东欧程序员为主。它的旗舰版本还支持HTML，CSS，PHP，MySQL，Python等。免费版只支持Java,Kotlin等少数语言。

IDEA所提倡的是智能编码，是减少程序员的工作，IDEA的特色功能涵盖了以下

1. 程序员意图支持

程序员编码时IDEA时时检测你的意图，或提供建议，甚至直接帮你完成代码。

1. 智能的选取

在很多时候我们要选取某个方法，或某个循环或想一步一步从一个变量到整个类慢慢扩充着选取，IDEA就提供这种基于语法的选择，在默认设置中Ctrl + W，可以实现选取范围的不断扩充，这种方式在重构的时候尤其显得方便。

1. 丰富的导航模式

IDEA提供了丰富的导航查看模式，例如Ctrl + E显示最近打开过的文件，Ctrl + N显示你希望显示的类名查找框（该框同样有智能补充功能，当你输入字母后IDEA将显示所有候选类名）。在最基本的project视图中，你还可以选择多种的视图方式。

1. 历史记录功能

不用通过版本管理服务器，单纯的IDEA就可以查看任何工程中文件的历史记录，在版本恢复时你可以很容易的将其恢复。

1. 对重构的优越支持

IDEA是所有IDE中最早支持重构的，其优秀的重构能力一直是其主要卖点之一。

1. 编码辅助

Java规范中提倡的toString()、hashCode()、equals()以及所有的get/set方法，你可以不用进行任何的输入就可以实现代码的自动生成，从而把你从无聊的基本方法编码中解放出来。

1. 动态语法检测

任何不符合java规范、自己预定义的规范、累赘都将在页面中加亮显示。

1. 代码检查

对代码进行自动分析，检测不符合规范的，存在风险的代码，并加亮显示。

1. 对JSP的完全支持

不需要任何的插件，完全支持JSP。

1. 智能编辑

代码输入过程中，自动补充方法或类。

1. 预置模板

预置模板可以让你把经常用到的方法编辑进模板，使用时你只用输入简单的几个字母就可以完成全部代码的编写。例如使用比较高的public static void main(String[] args){}你可以在模板中预设pm为该方法，输入时你只要输入pm再按代码辅助键，IDEA将完成代码的自动输入。

1. 完美的自动代码完成

智能检查类中的方法，当发现方法名只有一个时自动完成代码输入，从而减少剩下代码的编写工作。

1. 版本控制完美支持

集成了市面上常见的所有版本控制工具插件，包括git、svn、github，让开发人员在编程的工程中直接在intellij idea里就能完成代码的提交、检出、解决冲突、查看版本控制服务器内容等等。

1. 不使用代码的检查

自动检查代码中不使用的代码，并给出提示，从而使代码更高效。

1. 智能代码

自动检查代码，发现与预置规范有出入的代码给出提示，若程序员同意修改自动完成修改。例如代码：String str = "Hello IntelliJ " + "IDEA"; IDEA将给出优化提示，若程序员同意修改IDEA将自动将代码修改为：String str = "Hello IntelliJ IDEA";

1. 正则表达式的查找和替换功能

查找和替换支持正则表达式，从而提高效率。

**第三章：程序框架**

由于Java是完全面向对象的，因此在设计前心中需要先有一个完整的结构图，不同的模块不同的类(class)分别实现不同的功能，使的任何一个类和方法的代码都不要过长。清晰调用之必须，明确分工之必要，这种结构才能使程序员在编写程序时思路清晰并容易找错。

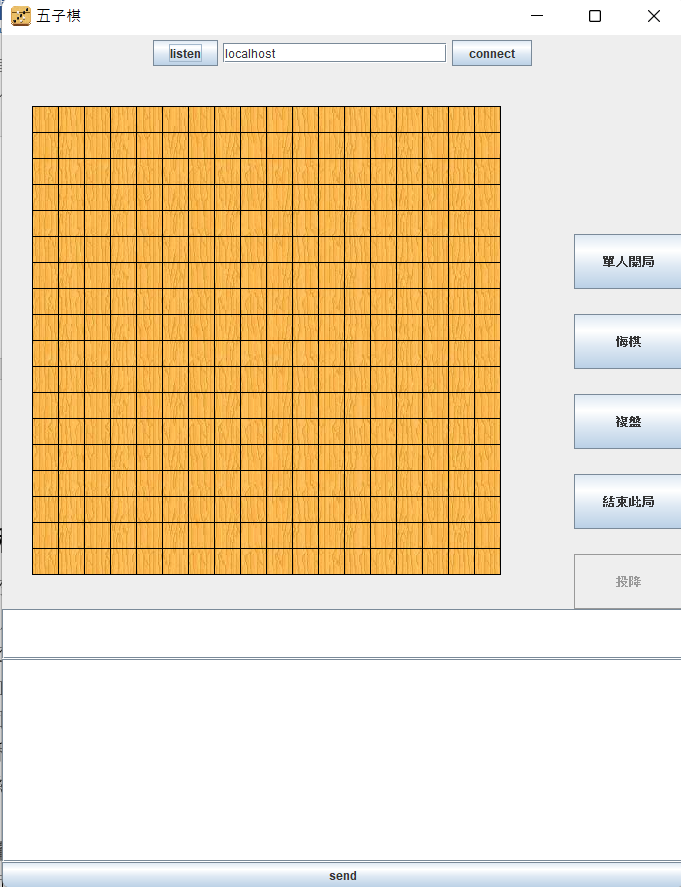
这次的五子棋游戏中总共包含十四个类，分别为：

* + 1. Main (主程序)
    2. North Panel (联网用北面板)
    3. Chess Panel (棋盘面板)
    4. East Panel (装载按钮与倒计时的东面板)
    5. Btn Panel (东面板内装载按钮的面板)
    6. Time Thread (用于计时的线程)
    7. Replay Chess Frame (复盘时跳出的窗口)
    8. Replay Chess Panel (复盘时重现上一局所用的棋盘)
    9. Chat Panel (聊天用面板)
    10. Chess Point (纪录棋子属性)
    11. Controller (操控所有操作的操控类)
    12. Model (存放重要数据的棋盘模型)
    13. Net (用于联网与交互的助手)
    14. Vars (所有类的代理)

**第四章：程序详细介绍**

1. Main – 主程序：

用于建立最外面那层Frame 并定义其大小、位置、窗口名字、关闭时的操作以及图标等等基本信息，并将其他面板包括Chess Panel, North Panel, East Panel, Chat Panel等绘制上去



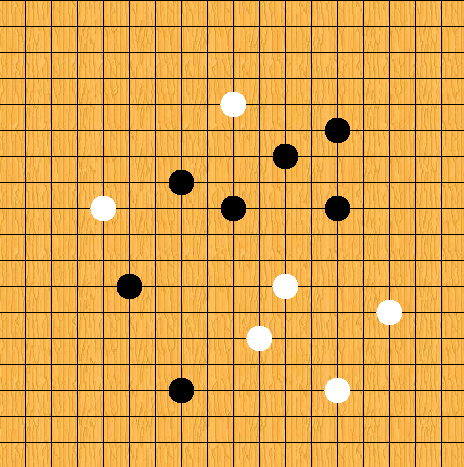
1. North Panel – 北面板：

装载listen button 与 connect button以及输入地址栏，并且监听按钮点下事件，点击后便调用Controller执行接收或连接服务。



1. Chess Panel – 棋盘面板：

设定棋盘大小，增加窗口大小监听事件，使棋盘可以随者窗口增大或缩小；同时添加鼠标事件，监听鼠标点击的XY坐标以便下棋，调用方法paint Component根据给予的特定XY来画棋盘、背景、黑棋白棋等等



1. East Panel – 东面板：

用Border Layout 分别将定时器与Btn Panel 加在北边与南边，并启动定时器 Time Thread 线程以便随时可以开始计时。



1. Btn Panel – 按钮面板：

采用Grid Layout放置按钮，几个按钮分别为单人开局、悔棋、复盘、结束此局、投降。

1. 单人开局：

点击后同时将listen、connect和投降按钮置成不可点击，并将结束此局按钮设置成可点击，开始单人游玩。

1. 悔棋：

若为单人模式则直接将最后一个棋子从history中删除，然后将color = -color

若为双人联机模式则先把两边的定时器都暂停计时，然后透过net类在对方的窗口跳出J Option Dialog，询问对方是否能悔棋，若对方回答可以则删除最后下的棋，并且开启我方(关闭对方)open Door，同时我方时间继续计时，对方时间暂停计时；不行则我方暂停计时同时关闭(对方开启)open Door，对方继续计时，我方暂停计时。

1. 复盘：

一局游戏结束后才能有效复盘，否则跳出”无上一局资料”的Message Dialog。

游戏结束后点击便跳出一新窗口(Replay Chess Frame)，仿照Chess Panel 的方式画出背景图片以及棋盘(Replay Chess Panel)，窗口底部有上一步与下一步的按钮，点击操控要画出哪一次下的棋(透过history)，关闭时隐藏此窗口。

1. 结束此局：

仅在单人模式下可用，透过clean与repaint两方法立刻清空棋盘并开始新的一局。

双人模式则无法点击。

1. 投降：

点击时立刻调用who Win方法，并且将赢家设成remote Color，who Win方法则会跳出黑棋(白棋)胜利的Message Dialog，并询问是否继续下一局。

若双方都点击同意则透过clean与repaint两方法清空棋盘，先后互换(若第一局黑棋先，第二局便改成白棋先)并开始下一局。

若其中一方点击”不玩了”便立刻退出程序，同时在另一方的屏幕上显示”对方已离开”的Message Dialog，并重新亮起listen 与 connect 按钮以便下一个连接。

1. Time Thread – 定时器线程

定义一个Time Thread 继承 Thread ，当running为true时，使其每休眠(sleep)一秒，就将时间减一，达到计时的功能，且当时间<=零时自动调用投降方法。

其下还包含个begin方法，可以开启一个新的线程并start，方便停止计时后调用这个方法便可以直接继续计时。

1. Replay Chess Frame – 复盘界面：

当点击复盘时构造此方法，将Replay Chess Panel与按钮绘制上此接口，

点击上(下)一步便透过history获取该次下的棋再绘制上去，达到客观的复盘效果。

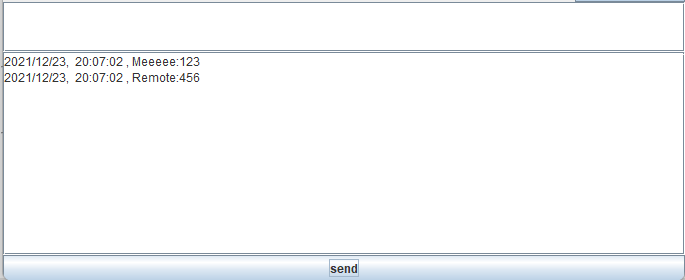
1. Replay Chess Panel – 复盘棋盘：

仿造Chess Panel绘制的方法，透过paint Component方法将棋盘、棋子与背景绘制上此Panel，并将此Panel 放上Replay Chess Frame完成复盘。

1. Chat Panel – 聊天面板：

使用Border Layout将输入栏放在此面板的北面，将显示栏放在此面板中间，将send按钮放在南面。

监听send按钮，当其被按下便在自己的显示栏输出当前时间与”Me:”与内容，并清空输入栏以便下一次输入，同时传送Message给对方，让对方的显示栏也显示当前时间”Remote:”与内容，达成聊天功能



1. Chess Point – 棋子属性：

纪录棋子的行、列与颜色，继承自Serializable

1. Controller – 控制器：

一切操作的根源，所有操作都必须经由此类，让我逐一介绍所有操作：

1. 连接(listen, connect)：

调用net以开始监听连接或联机，成功便将listen 与 connect按钮置成不可按，同时联机的那方将本地颜色置成白，对方置成黑，接者开始下棋。

1. 下棋：

若open Door为false则代表此轮是对方该下不是自己，因此直接return，否则透过模板的put Chess方法下棋。

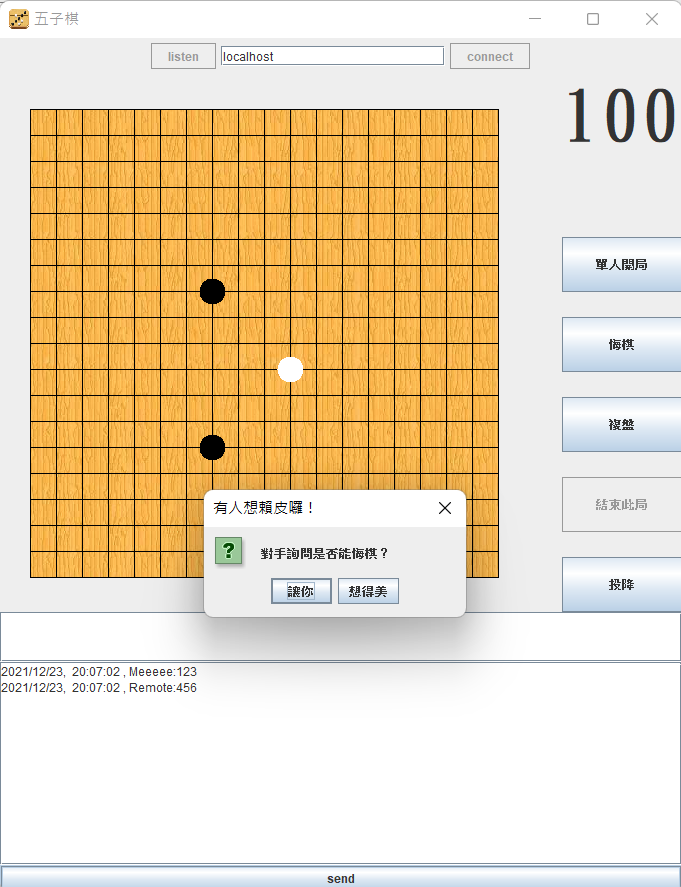
若下棋成功则repaint棋盘使其刷新进而显示棋子的位置，接着将本地悔棋按钮亮起来表示可以悔棋，同时侦测是否连五子产生赢家，若有则调用who Win方法，否则将open Door 与 线程 Time Thread 的 running 置成 false，再来发送一个下棋的信号给net，net再发送信息给对方，对方此时执行remote Put Chess，此方法使自己的open Door 与Time Thread 的 running 置成 true 并重新开启一个 Time Thread 以继续倒计时。

1. 单人开局：

调用start方法将open Door 置成true，同时将listen 、 connect 与投降按钮置成不可按，接者开始单人下棋。

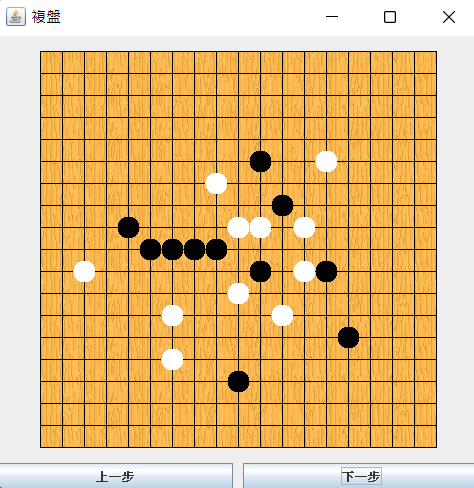
1. 悔棋：

调用ask Can Redo方法通过net询问对方是否能悔棋，若可以则通过模板将history里最后面一个纪录删掉，并将本地running置成 true，对方置成 false，本地悔棋按钮置成不可按，对方置成可按，接者repaint刷新棋盘；同时对方执行net Redo方法执行类似的操作。



1. 复盘：

点击复盘通过模板调用replay方法，模板调用Replay Chess Frame使其出现并将 history 传给他，以Replay Chess Frame为基底，其上绘制棋盘与上、下一步按钮开始重现棋局，点击关闭后将其隐藏。



1. 投降：

点击投向立刻调用who Win方法，赢家为remote Color，

1. 聊天：

若监听到send按钮被按下，则调用send Chat Message方法，透过net将输入栏内的信息发送给对方，同时将信息在自己的字段也打印一份；net再调用对方Controller内的remote Say Something方法，将收到的信息打印在显示字段。

1. Model – 模板：

定义棋子的颜色(Black, White, Space)、棋盘的宽度(Width)、存放下棋历史的history与棋盘的状态data。其内分别有以下这些功能：

1. 下棋：

接收参数row, col, color，并在data中的对应位置下棋，同时记录到history内。每下一次棋就把row, col, color都记录下来以便判断输赢

1. 判断输赢：

从八个方向数共有多少个同颜色的棋子，当凑满五个便调用who Win方法。

1. 悔棋：

在data中抹除最后一个history的棋子。

1. 判断是否能毁棋：

从history中查看上一个颜色的棋子是否跟local Color一样，一样才将按钮亮起(可悔棋)，否则使其不可按。

1. 复盘：

调用Replay Chess Frame从而开始回放棋局。

1. 清除棋盘：

将data中所有数据清除(恢复成SPACE)。

1. Net – 联网与交互类：

负责listen与connect还有执行剩下所有交互。

1. Listen：

监听到点击事件便初始化一个Server Socket，并使其accept客户端s，接者创建Buffer Reader与 Print Writer来随时准备处里交互行为。

1. Connect：

点击connect的客户端new 一个 Socket并创建Buffer Reader与 Print Writer来随时准备处里交互行为。

1. 处理交互事件的线程：

创建一个死循环不断读取Buffer Reader看有什么交互请求，有的话便处理，没有的话则继续循环。

所有交互事件包括下棋、询问是否能悔棋、悔棋、发送聊天信息、接收聊天信息、投降以及离开等等都在这个线程中读取并调用相应方法处理。

1. Vars – 代理：

所有类都在这里new一个简称，更统一且方便调用。

**第五章：总结与展望**

一、总结：

首先，这一学期的Java课程让我从一个只会些微C++的计算机小白，变成也可以使用Java设计一款小游戏的菜鸟。能多学会一门语言真的令人非常开心的一件事，非常感谢老师的细心教导，让我能在计算机的高塔中更上一层楼

这次的Java五子棋游戏设计，让我对Java各种功能的熟系度大幅的提升，包括线程的运用、网络的交互、圖形化用戶介面的版面設計、尽量使耦合性降低的分工模式与图形化用户接口的配置等等，也深刻的见识到了Java的强大，更强化了我学好编成的渴望与心念，期待将来也能利用我的所学设计一向较大规模且更加完整的工程。

二、展望：

这款五子棋小游戏虽然具备了最基本的功能，但还是有许多不足，像是无法实现多人联机，最多只能双人游玩；没有游戏大厅，几乎是程序一执行便直接开始；没有AI模式，单机游玩还是得有两个人一起在同一台计算机下棋；画面不够精美，许多按钮与面板都没有多加设计；没有音乐，总感觉少了点甚么；只能打字聊天没有多加语音聊天功能；也没有自己的用户名与密码等等。

希望将来更加熟悉掌握网络编程、AI设计方式与JAVA的各种库的运用后可以再更完善这款五子棋游戏，实现这次没实现的功能。

参考文献

[1] Bruce Eckel, Java编程思想，2007/6

[2] Kevin Tung, 为什么你应该选择Java作为你的起始语言 — 学习Java的各种好处, 2021/5/26

[3] 程序前沿, Java的特点和优点, 2018/6/30

[4] 维基百科, 五子棋-维基百科, 2021/10/24

[5] 每日头条, 上学时风靡全国的「五子棋」，在中国竟然有这么悠久的历史！, 2016/10/13

致谢

本次游戏设计在刘嘉欣老师的教学与指导下完成，因此我想向刘嘉欣老师表达真挚的感谢，谢谢您平日的细心的教导与详细的讲解，让我得以完善此次的课程设计。

再来，我想感谢Larry Page与Sergey Brin创办Google让我可以查找各种数据與資料来解决我遇到的问题。

接着，我也想感谢我的父母，无怨无悔的生我养我直到现在，才能让我上这门JAVA语言，进而设计这款小游戏。

最后，我最感谢的就是共产党与中国，提供我一个这么优质的环境学习JAVA语言，我爱永远与人民站在同一阵线的共产党、我爱中国。