

南开大学

本科生毕业论文（设计）

中文题目： Java 语言与应用期末大作业报告

外文题目： Java In Applications Final Project Report

学 号： 2111408

姓 名： 周钰宸

年 级： 2021 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 王子纯

完成日期： 2022.12.12

诚信声明

本人郑重声明：所呈交的论文，是本人在导师指导下，独立进行学习实践所取得的成果。除文中已经标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。有关观点、方法、数据和文献等的引用已在文中指出，并与参考文献相对应。本人完全意识到本声明的法律结果由本人承担。

摘 要

众所周知, Java 语言是一门面向对象的编程语言, 具有着简单易用和功能强大两个最为显著的特征。简单易用是因为它在设计时对 C++ 语言进行了一定程度的“取其精华, 弃其糟粕”, 不仅学习沿用了 C++ 语言中的各种优点, 又同时放弃了 C++ 里让人难以理解的多继承、指针等概念, 一定程度上方便了初学者的接触和学习的同时, 也让其具有了众多的优良特性。同时它作为静态面向对象编程语言的代表, 极好地实现了面向对象理论, 允许程序员以优雅的思维方式进行复杂的编程。在应用上, Java 可以编写桌面应用程序、Web 应用程序、分布式系统和嵌入式系统应用程序等, 所涉及的领域极广。

本学期我在初步了解了 Java 的各方面优势后, 本着既为了提高自己的编程硬实力, 也为了能够为将来工作和研究有着更为丰富的选择的目的, 努力地学习了一学期的 Java 语言, 最终以这份期末大作业呈现。

本次作业基于 eclipse 平台, 依托 Java 语言, 以 MVC 为基础, GUI 界面中的 Swing 包为主要工具设计了能够具有单机模式 (PVE) 和网络对战模式 (PVP) 的五子棋小游戏, 在实现了基本的图形化五子棋下棋功能 (选择颜色, 开局, 下棋, 判断输赢, 重新开始等) 和基本的网络对战功能 (在两台电脑上, 一方下棋, 传到另一方) 以外, 我还实现了悔棋、网络聊天、背景音乐的开始与停止、两种模式的切换、智能语音包嘲讽、游戏大厅、重开还有下棋落子、犹豫、游戏胜出和失败的多种丰富的音效添加等功能。

本次作业由于时间紧迫, 没有进一步在对 GUI 界面进行更好地美化和完善, 只是实现了基本的五子棋小游戏。但是在这个过程中, 通过自主学习, debug 等过程收获了无数的知识能力和解决问题的方法和思路。之后我会好好汲取本次项目的经验, 不仅为了日后对其进行进一步的完善和补充, 更是为了能够创作出更多功能多种多样且优秀的作品。

关键字: Java; eclipse; MVC 模型; 五子棋; 网络; 可视化; GUI 界面设计

Abstract

As we all know, Java language is an object-oriented programming language, with two of the most significant characteristics: easy to use and powerful. Easy to use is because it has carried out a certain degree of "take the essence, discard the dross" in the design of C++ language, not only learning to use a variety of advantages in C++ language, but also gave up the C++ in the difficult to understand the concept of multiple inheritance, pointer, etc., to a certain extent, it is easier for beginners to contact and learn at the same time, but also has many excellent features. At the same time, it as a representative of static object-oriented programming language, excellent implementation of object-oriented theory, allowing programmers to elegant way of thinking for complex programming. In the application, Java can write desktop application, Web application, distributed system and embedded system application and so on, involving a wide range of fields.

This semester, after I preliminarily understood the advantages of Java in various aspects, in order to improve my programming hard strength, but also to have more choices for future work and research, I worked hard to learn Java language for a semester, and finally presented with this final assignment.

This assignment is based on the eclipse platform, relying on the Java language, on the basis of MVC, the Swing package in the GUI interface as the main tool to design can have the single machine mode (PVE) and network battle mode (PVP) of GoBang small game, in the realization of the basic graphical GoBang chess function (choosing color, start, chess, Win or lose, restart, etc.) and the basic network battle function (on two computers, a party to play chess, spread to the other party) outside, I also realized the contrite, network chat, background music start and stop, two modes of switching, intelligent voice package mocking, Game Hall, restart and chess end, hesitation, the game win and failure of a variety of rich sound effects added and other functions.

This operation due to time constraints, no further in the GUI interface to better beautify and improve, just to achieve the basic GoBang small game. However, in this process, through independent learning, debug and other processes, I gained numerous knowledge and ability and methods and ideas to solve problems. Later, I will learn from the experience of this project, not only to further improve and supplement it in the future, but also to create more versatile and excellent works.

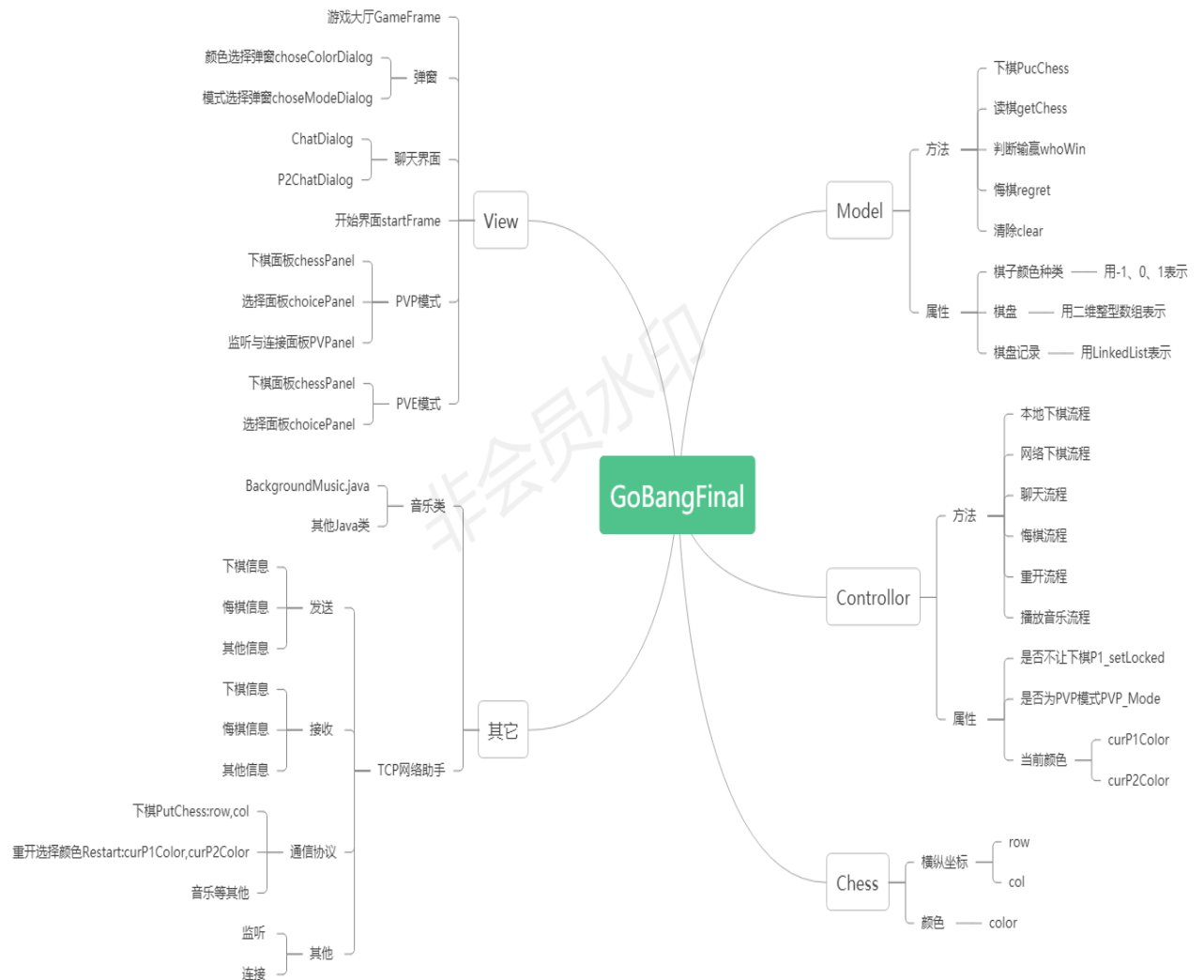
Keywords: Java;The eclipse;MVC model;GoBang;Network;Visualization;GUI interface design

第一章 概述

1.1 设计框架

本次项目主要是需要完成一个能够实现基本的图形化五子棋下棋功能（选择颜色，开局，下棋，判断输赢，重新开始等）和基本的网络对战功能（在两台电脑上，一方下棋，传到另一方）。我在此基础中还实现了悔棋、网络聊天、背景音乐的开始与停止、两种模式的切换、智能语音包嘲讽、游戏大厅、重开还有下棋落子、犹豫、游戏胜出和失败的多种丰富的音效添加等功能。

其中大体的设计框架是以 MVC 为模型，具体实现框架图如下：（如果看不太清，我放进了附件里）



其中主要核心为 MVC 模型下的 Model、Controllor、View。还有抽象的棋子类 Chess 和 TCP 网络助手，和一些音乐类比如 BackGroundMusic.java。

Model 主要分为属性和方法两部分，属性包括诸如棋子颜色、棋盘等，方法包括下棋、读棋和一些拓展功能;View 以 GUI 界面设计原理为基础，有 JFrame、JDialog、JPanel 等实现了可视化的游戏大厅和单独的聊天界面;Controllor 也分为属性和方法两种，属性包括 PVP 模式下双方的棋子颜色等，方法很多，主要汇集了来自其余各方的汇报，都通过 Controllor 与 Model 或 TCP 进行联系，其中包含网络 PVP 下棋流程和一些其他功能的相关流程；其他还有很重要的实现了网络相关功能的 TCP 网络助手，其中包含了负责发送、接收信息和规定交流的通信协议等；最后还有为了能够独立播放各种音乐和音效的许多音乐类。

1.2 设计方法

使用 Java 语言，基于 eclipse 环境和 MVC 模型，同时使用 GUI 界面设计原理、Swing 包等工具包，实现了并开发出了多功能网络版五子棋程序。

1.3 报告整体结构

本文一共由 5 部分组成，分别是：封面、摘要、Abstract、正文、致谢。其中正文部分内容如下。

第一章：概述；

第二章：工具选型；

第三章：系统设计；

第四章：系统实现与关键问题描述；

第五章：总结与展望；

第二章 工具选型

2.1 开发环境和语言

1.开发语言：Java

2.集成开发环境：eclipse

3.所用工具包：java.swing、java.awt、java.io.*、java.net、java.sound、java.util

2.2 工具简介

1. 集成开放环境 eclipse:

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括 Java 开发工具（Java Development Kit，JDK）。

Eclipse 是一个开放源代码的软件开发项目，专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台。它主要由 Eclipse 项目、Eclipse 工具项目和 Eclipse 技术项目三个项目组成，具体包括四个部分组成——Eclipse Platform、JDT、CDT 和 PDE。JDT 支持 Java 开发、CDT 支持 C 开发、PDE 用来支持插件开发，Eclipse Platform 则是一个开放的可扩展 IDE，提供了一个通用的开发平台。它提供建造块和构造并运行集成软件开发工具的基础。Eclipse Platform 允许工具建造者独立开发与他人工具无缝集成的工具从而无须分辨一个工具功能在哪里结束，而另一个工具功能在哪里开始。

2. 所用工具包

(1) java.swing 与 java.awt.Swing 是一个用于 Java 开发应用程序用户界面的开发工具包，是由纯 Java 实现的。以抽象窗口工具包（AWT）为基础使跨平台应用程序可以使用任何可插拔的外观风格，不依赖操作系统的支持，这是它与 AWT 组件的最大区别。Swing 开发人员只用很少的代码就可以利用 Swing 丰富、灵活的功能和模块化组件来创建优雅的用户界面。Swing 库替代了绝大部分的 AWT 组件，但需要使用 AWT 的事件处理机制通过使用 AWT 和 Swing 提供的图形界面组件库，Java 的图形界面编程可以变得比较简单，程序只要依次创建所需的图形

组件，并以合适的方式将这些组件组织在一起，就可以开发出不错的用户界面。

二者共同为 GUI 界面设计的必要工具。

(2) `java.io`:Java 的核心库 `java.io` 提供了全面的 IO 接口。包括：文件读写、标准设备输出等。

(3) `java.net`:几乎包含了所有的网络设施，提供了 `Socket` 套接字等多种功能，是连接远程主机，实现网络功能的必备 `package`。

(4) `java.sound`:从根本上涉及音频传输，是实现音乐和音效功能的必备 `package`。

第三章 系统设计

3.1 设计思路

1. 本地下棋（即 PVE 模式）：

（1）本地下棋经过分析和抽象后，得出需要以下几部分：对棋的抽象描述、对棋的颜色的抽象描述、对棋盘的抽象描述、对棋盘记录的抽象描述、下棋界面、其他功能界面、下棋方法、获取棋子颜色方法、判断输赢的方法、清除界面方法、悔棋方法等。

（2）采用整数-1、0、1 三者分别对应黑、空白、白三种颜色对棋的颜色进行抽象描述，其中将白和黑设为相反数的目的也是便于在下棋方法中进行颜色的翻转（采用取相反数的方式）；定义一个 Chess 的 Java 类对棋子进行抽象描述；定义一个二维数组（其中非 0 即为有棋）对棋盘进行抽象描述；采用一个 LinkedList<Chess>对棋盘记录进行抽象描述，其中存储着每一次双方所下的棋。采用 GUI 界面设计对相关下棋界面和其他功能界面进行设计，其中采用了单例模式即 instance 模式，之后再通过 BorderLayout 等布局方式对相关组件进行放置。完成界面绘制。

（3）下棋方法需要注意下的位置是否合法，包括坐标和是否已经有棋子在这个位置；获取棋子颜色方法也需判断位置是否合法；判断输赢的方法需要知道最后一个棋子下的位置，故先从进行棋盘记录的 LinkedList<Chess>获取最后一次放下的棋子，再获取其最后一次下的横纵坐标和颜色，进而进行判断；悔棋方法需要注意若从来没下过棋则不能悔棋，思路便是使上一个下棋的棋盘位置置为空白；清除界面即把所有位置置为空白，同时此时需要把棋盘界面进行及时的重绘，调用 ChessPanel 的 repaint 方法。

2. 网络下棋（即 PVP 模式）：

（1）网络下棋经过分析和抽象后，需要实现以下功能：是否要进行 PVP 模式、打开两个程序后将两个程序彼此连接、能够使二者的 Model 里面的棋盘和下棋

记录完成同步（即 P1 下一个棋在 P2 那里需要立刻显示）、当一方下完棋之前另一方不能再次下棋。

（2）采用一个 `Controllor` 中的两个 `boolean` 类型的变量即 `PVP_Mode` 和 `P1_setLocked` 二者对是否进行 PVP 模式和控制让棋盘封锁在对方下完棋后无法下棋；设计一个 `PVPanel` 对监听和连接的相关按钮和界面进行设计；为了使二者的棋盘完成同步，故需要将一边的下棋信息传输给另外的对方，设计一个非常关键的辅助类 `TCP` 进行操作，具体会在第四章的关键问题描述中阐述解决方案。

3. 其他功能（除悔棋和重开外）

（1）聊天功能：采用两个与棋盘不一样的独立聊天界面，将会在第四章关键问题描述中详细阐述解决方案。

（2）背景音乐：先通过 `java.sound` 的 `Clip` 类实例和 `AutioSystem`、`AutioInputStream` 等实现音乐的播放，又由于在 PVP 模式中需要与其他线程相互独立，故设计继承自 `Thread` 类的音乐类，比如 `backGroundMusic`，通过设置 `clip.loop()` 或不设置来进行音乐播放的循环或者单次选择。

（3）相关音效：对于下棋和“犹豫不决”音效，在 `ChessPanel` 的 `mousePressed` 和 `mouseReleased` 两个方法中设置相关音乐的播放，便可实现落子的音效；对于 PVP 模式中的胜利与失败音效，也将在关键问题描述中详细阐述。

（4）模式的切换与游戏大厅：游戏大厅的实现便是采取在刚开始打开程序时先生成一个 `StartFrame` 来进行是否开始游戏的选择，同时设计弹出的聊天窗口以此把游戏大厅独立出来，在 `StartFrame` 中绘制相关按钮实现进一步的控制和界面转换，采用 `setVisible` 相应不需要的界面设为不可见，而让游戏大厅可见等操作实现界面的转换；在 `startFrame` 后设计一个跳转而来的弹窗 `choseModeDialog`，在其中进行界面的绘制和相关按钮的设置，同时选定了某个模式后，对 `Controllor` 中的 `PVP_Mode` 的 `boolean` 值进行设定以达到目的。

（5）颜色的选择：在选择重开游戏后，可以进行游戏的选择，通过弹出一个 `choseColorDiaglog` 实现，由于涉及双方颜色的协调互反，故需要进行信息的传输，因此需要 `TCP` 参与其中，将在第四章关键问题描述中详细阐述。

（6）智能语音包嘲讽：受经典斗地主游戏启发，进行整活，在聊天界面设计一个 `JComboBox` 将可以发出的语音包对应的文字进行存储，由于需要让另一方收

到相应语音，因此需要 TCP 参与其中，将在第四章关键问题描述中详细阐述。

3.2 设计重难点

(1) 信息传输：网络下棋模式（即 PVP 模式）需要实现的两方棋盘同步更新引发了需要进行信息传输的问题，将在第四章关键问题描述中解决。

(2) 多种任务的同时进行：由于在网络模式中需要多种模式同时进行，故需要利用多线程来解决，对一些任务设为线程以让他们彼此相互独立运行来实现。

(3) GUI 界面：由于游戏需要实现的功能多，对相关 GUI 界面进行绘制和设计也格外重要。

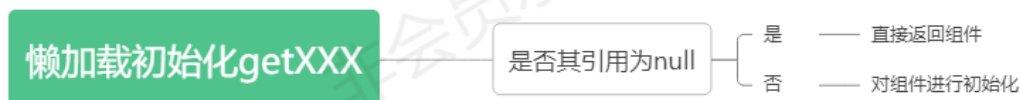
(4) 方法之间彼此的联系：由于功能彼此之间联系的复杂性，需要明确 MVC 模式和 TCP 的所有相关方法和对应职能，理清之间的关系进行设计。

第四章 系统实现和关键问题描述

4.1 系统实现

1. GUI 界面:

(1) 相关控件的初始化: 对于 GUI 界面中的几乎所有控件 (除 JLabel 之外, 比如 JButton 和 JomboBox 等), 在声明时声明为私有变量, 并初始化为 null, 之后通过懒加载的方式对所有空间进行初始化:



以这种懒加载的模式进行初始化可以同时实现通过公有方法对私有变量的访问和对私有变量的初始化, 因此达到提高性能和减少数据开销的目的。在每个界面的初始化中, 将控件添加时, 统一进行懒加载初始化, 方便理解与阅读。

(2) 相关控件的布局 and 界面的绘制: 采用 java.awt 中的布局变量如 BorderLayout 和 setLayout 方法进行界面绘制, 在一些较为简洁的页面中, 通过 setLayout (null) 将布局设为空后再对各个组件进行坐标及大小的设计, 已实现了画面的美观。界面等采用了 JFrame、JDialog、JPanel 等等容器进行设计。

2. 相关操作的提示: 通过 JOptionPane 的不同形式设计不同的弹出提醒或错误警告窗口, 并附上了有趣的文字进行提示, 在相应的部分方法中的特例处理部分调用使其生成, 同时也用在了对胜利者的恭喜和对失败者的嘲讽上, 也用到了弹窗。

4.2 关键问题描述: 信息传输

1. TCP 的设计和使用: 为了能够实现信息的传输, 设计一个新的 Java 类 TCP, 在其中使用 ServerSocket 和 Socket 两个分别进行监听和等待连接, 与 PVPanel 中的监听和连接按钮相对应, 当两个程序的监听和连接按钮被按下时, 通过 ip 地址和一个端口将两个程序实现 TCP 连接, 之后通过 java.io 包中 BufferedReader

类的 `in` 和 `PrintWriter` 类的 `out` 来进行通信。

为了使下棋和连接两个任务独立进行，将 TCP 中的重要方法设计为一个新 `Thread`，当两个程序实现彼此连接后，通过都去调用方法 `startReadThread` 获取对方所发来的信息，并且随时等待处理。

TCP 需要既能实现发送也要实现连接功能，由于 `in` 和 `out` 传输给彼此的只能是字符串，因此读出的也只能是字符串，因此不同的传输内容需要调用不同的指令以传达给另一方的程序，故设计通讯协议，即通讯的字符串所具有的固定格式模板。即传输的字符串以不同的单词开头，对应着不同的指令。并且涉及一些需要传输数据的字符串，还需要通过 `String` 类的 `substring` 和 `split` 方法进行获取相关数据。比如

`PutChess: row, col Restart: curP1Color, curP2Color`

`Chat Regret RushMusic` 等

在获取相关指令后，传输给 `Controllor` 最后进行集中的操作

- 2.棋盘的同步：利用上述的 TCP，在 `Controllor` 中定义相关的 `putP1Chess` 方法，在下棋成功后，通过 TCP 的 `sendChess` 方法，按照通讯协议的格式，读棋相关参数，再回到 `Controllor` 中使用不同的 `putP2Chess` 方法完成对另一方棋子的同步。
- 3.重开的同步：与棋盘同步同理，不过涉及颜色的重选，故将 P1 新选的颜色传个 TCP 让 P2 进行读取和相关的设置，并且通过一个 `JOptionPane` 的弹出进行提示。
- 4.聊天功能：与棋盘同步同理，在按下开始聊天按钮后，通过 TCP 发送讯息，使双方同时弹出聊天窗口，聊天窗口的设计与大作业 14 类似，通过一个新的端口和线程实现聊天。
- 4.智能语音包嘲讽：通过对 `JComboBox` 的不同便捷文本的选择，不仅可以在文本域中打出，同时通过传输 TCP，让对方的程序播出对应的音乐包以实现功能。
- 5.胜利和失败的音效：由于 `Controllor` 中并不相同的两个方法 `putP1Chess` 和 `putP2Chess`，在下棋后判断输赢时，若恰好产生了赢家，则能立刻确定赢家。（如若由于 `putP1Chess` 下了上一个棋，产生了赢家，那赢家一定是本方；`putP2Chess` 则一定是对方为赢家），依此特殊的原理，在两个方法的判断输赢时弹出 `JOptionPane` 和播放对应胜利和失败的音乐以实现功能。

第五章 总结与展望

5.1 总结

1. 项目优点:

(1) 通过懒加载等方式实现通过公有方法对私有变量的访问和对私有变量的初始化, 因此达到提高性能和减少数据开销的目的。同时增加了代码的可读性。

(2) 采用了 MVC 模式, 逻辑清晰, 各种方法之间联系紧密, 同时一些注释的添加, 更是增加了代码的可读性。

(3) 实现功能较多, 在单纯的下棋游戏以外, 增加了很多有趣的彩蛋比如提示弹窗, 智能语音包嘲讽等。实现音效的方式也是清楚明白, 巧妙的同时为下棋过程增加了趣味性。

2. 项目缺点:

(1) 由于时间原因, 没有进行进一步的 GUI 界面美化和相关背景格式的进一步修正, GUI 界面平淡没有亮点。

(2) 设计用来记录下棋内容的链表未实现复盘功能, 有些浪费。

(3) 游戏第一次开始后, 默认监听方为黑, 连接方为白, 没有进行初次的颜色选择。

(4) 未设置选择先手功能, 默认监听方先手。

3. 项目可进行改进的地方:

(1) 进一步进行界面的美化, 通过设置个性化背景和相应文字文本域增加整体界面美感。

(2) 通过记录下棋内容的链表实现复盘功能。

(3) 在初次进行游戏时, 提供选择颜色的功能。

(4) 在每次重开和初次进行游戏时, 提供选择先手的功能。

5.2 展望

在本次项目的设计和实现过程中，我不仅通过查阅 `csdn` 的许多相关资料丰富了自己的 `Java` 语言知识储备，同时面临着第一个如此大体量的项目，我学会了将项目拆成许多部分逐个分析设计并进行最终的实现。本次项目中 `MVC` 模式的再次利用，让我能够对其进行更为熟悉地应用。`TCP` 模式和通讯协议的设计思路巧妙至极，完美地实现了两个主机程序的信息传输和同步，相信在未来的许多复杂程序中，都会有 `TCP` 思路的应用。在修改程序的过程中，我也锻炼和提高了编程和 `debug` 的硬实力。

本次自我设计这个项目的经历给我留下了深刻的印象，它为我带来的不仅是编程能力的提升和对一门语言的熟练掌握，未来我会继承并多次将 `Java` 面向对象程序设计的思路应用在编程学习和未来工作的每时每刻，将 `Java` 语言的使用烂熟于心，设计出更多更优秀的作品，成为一名全面的编程高手。

致 谢

本学期我在选择了 Java 语言与应用课程后，在刘嘉欣老师和王子纯老师的悉心教导下，我不仅掌握了一门新的语言，并且在这个过程中，通过每周一个的应用大作业，让我明白了编程最重要的就是亲手实践去应用，去写程序实现各种功能。在每周作业的锻炼过程中，我认真对待每一次作业，力求能从其中收获最多的知识和对 Java 语言的熟悉。我更不能忘记的是刘嘉欣老师上课幽默的讲课风格，令本身枯燥的编程课堂也变得生动起来，还有王子纯老师每周对作业的认真批改，甚至有一次我收到了王老师“思路很独特”的批语，这些都让我受益匪浅。

最后一次辛苦两位老师的审阅。感谢您们这一学期对我的悉心指导。祝您们未来生活工作一切顺利！