

南 开 大 学

JAVA 大作业论文

中文题目：JAVA 版网络五子棋设计实验报告

学 号：2110803

姓 名：谢雯菲

年 级：2021

专 业：信息安全、法学双学位

系 别：信息安全、法学双学位

学 院：网络空间安全学院

指导教师：刘嘉欣

完成日期：2022 年 12 月 12 日

摘 要

本学期 java 课的大作业是设计一个网络版的五子棋小游戏，该论文详细介绍了实现该游戏用到有关工具和方法、系统的设计理念和系统重要部分的设计思想。

关键词 JAVA；五子棋；网络通信；GUI

目 录

一、概述	4
二、工具选型	4
(一) javax.swing 和 java.awt 包	4
(二) java.net 包	5
(三) java.io 包	5
三、系统设计	5
四、系统实现与关键问题描述	6
(一) 五子棋模型 (Model)	6
(二) 控制器 (Controller)	7
(三) 视图 (View)	8
(四) 网络助手	9
五、总结与展望	10

一、概述

该程序为基于 java 语言设计的五子棋小游戏。它包括本地版和网络版两个版本。在本地版中实现了简单的下棋、悔棋、重新开始等功能。网络版在本地版的基础上实现了双人联网对战、投降、求和、邀请参赛、聊天等功能。

游戏主要界面如图 1 所示。

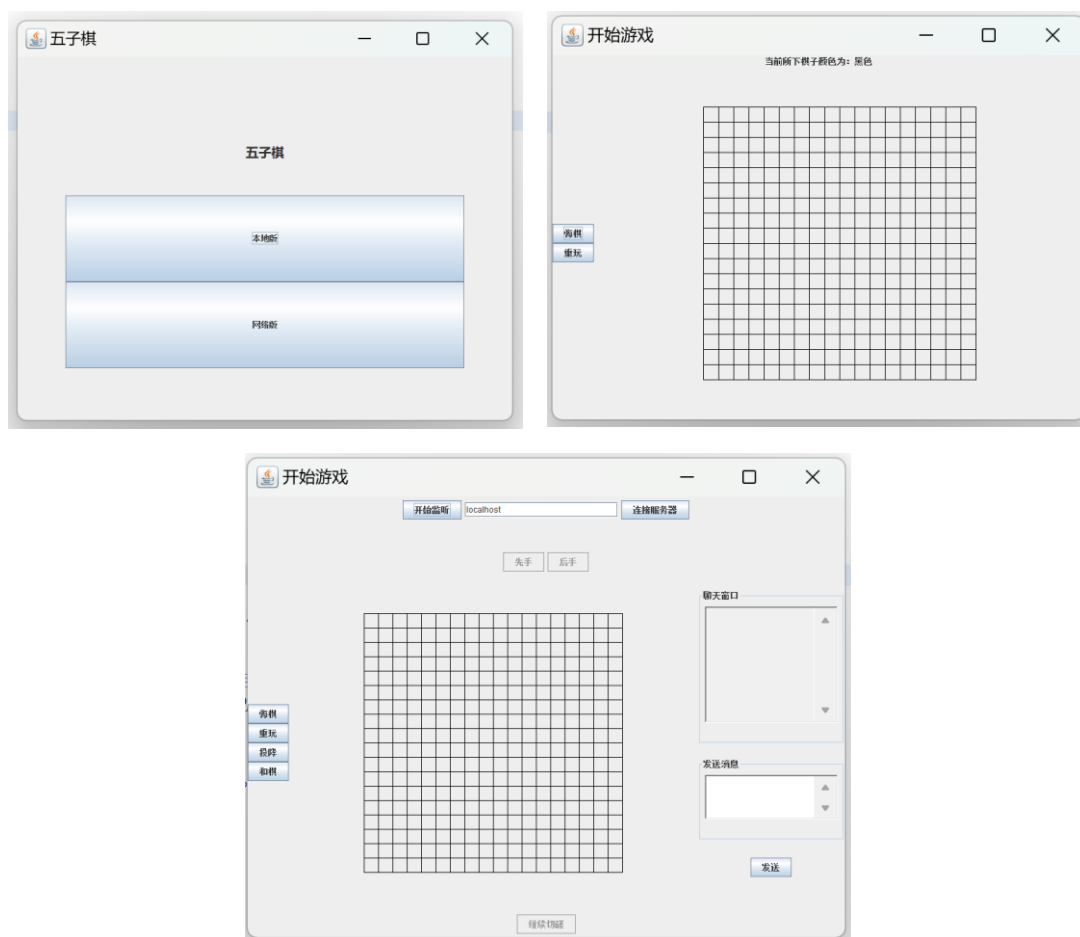


图 1 主要界面

二、工具选型

小游戏整体使用 java 语言编写，其中运用到了多种包。

(一) javax.swing 和 java.awt 包

Swing 是一个为 Java 设计的 GUI 工具包，包括了图形用户界面（GUI）的设计工具：组件（Component）、容器（Container）和布局管理器（LayoutManager）。

awt 是一个软件包，包含用于创建用户界面和绘制图形图像的所有分类。

该小游戏设计过程中大多数都使用了这两个包。

1. 组件

该小游戏中所使用到的诸如按钮、文本框、画图工、对话框等等，都是 swing 和 awt 包中的组件。组件上的监听器也是 swing 和 awt 包中的重要方法，小游戏中使用了这些方法来实现组件之间的互动。

2. 容器

该小游戏中所使用的窗口 (JFrame)、面板 (JPanel) 等都是 swing 和 awt 包中的容器。容器是外部的，只有有了容器才可以向里面添加组件和布局。小游戏中使用了这些容器来实现组件的分区。同时，小游戏中也采用了在容器中嵌套容器的做法使得布局更加的方便调整与整齐。

3. 布局管理器

swing 和 awt 包中的布局管理器可以更好地帮助我们对容器里的组件进行排列，小游戏中使用了流式布局 (FlowLayout)、表格布局 (GridLayout)、边框布局 (BorderLayout)、盒式布局 (BoxLayout) 等来对容器中的组件位置进行整理与美化。

(二) java.net 包

net 包是一个提供实现网络应用与开发的类。本实验中主要使用了它的 TCP 编程。小游戏中使用 socket 收发数据，实现网络对战的传输

(三) java.io 包

本游戏中主要使用 BufferedReader 类和 PrintWriter 类对网络传输的数据进行读和写操作，使得数据能够在服务器端和客户端传输，保证程序的正常运行。

三、系统设计

系统主要框架与结构如图 1 所示。

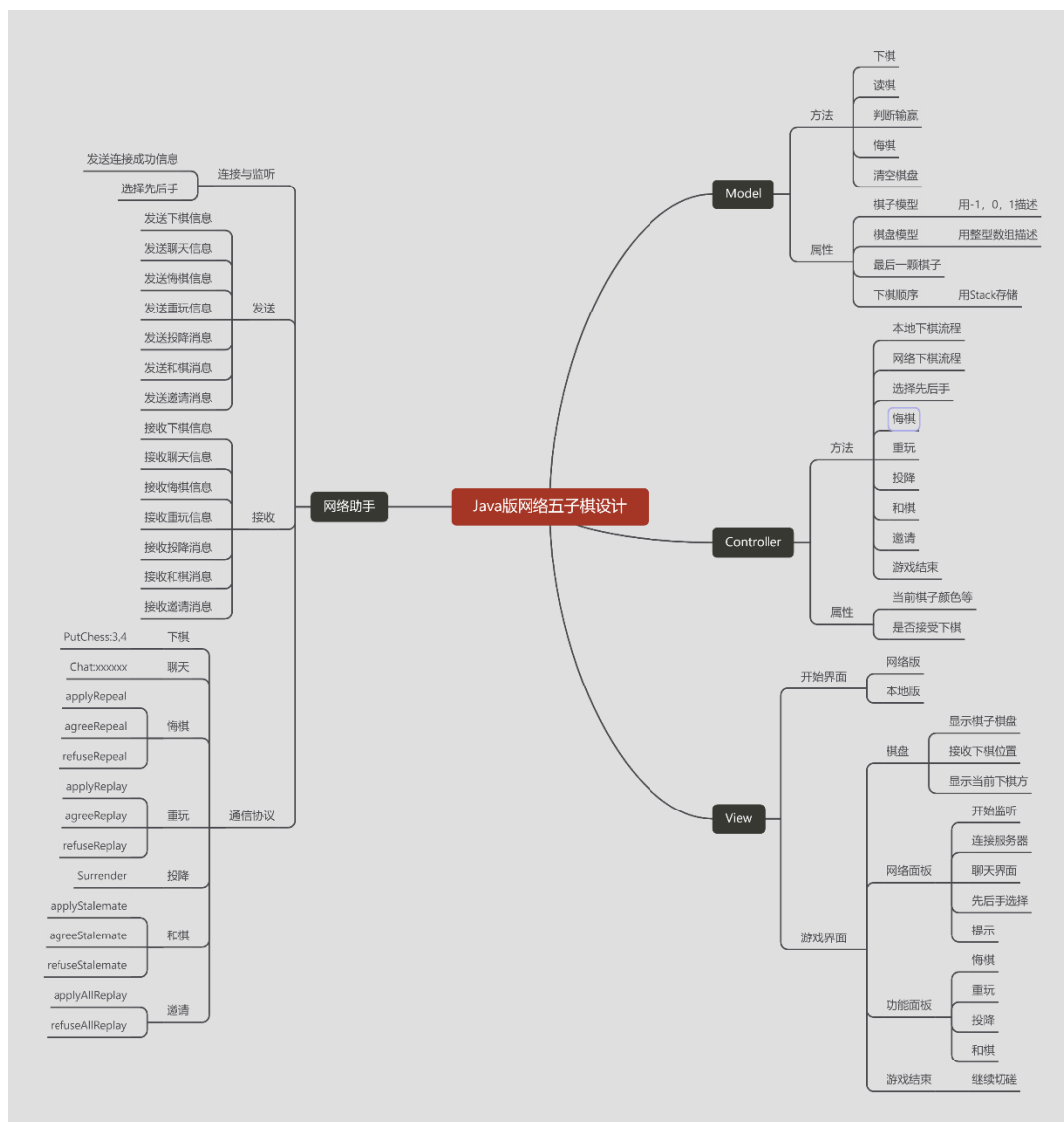


图 2 系统设计框架

四、系统实现与关键问题描述

(一) 五子棋模型 (Model)

1. 属性

(1) 棋子模型

用 3 个 int 整型存放棋子颜色。用 -1, 0, 1 分别表示黑棋、空和白棋。

(2) 棋盘模型

用整型数组存放棋盘的数据。data 数组为一个 19 行 19 列的 int 型数组，其中每一个数据代表对应行列的棋子类型。

(3) 最后一枚棋子

用两个 int 型数据分别存放最后落下的棋子的行和列数据，为后续判断输赢等等做好准备。

(4) 下棋顺序

考虑到后续将要实现悔棋等功能，就用一个栈来存放每次落下的棋子的位置。需要悔棋的时候，每次都取出栈顶的对象，做到先进后出，方便后续悔棋时存取。

2. 方法

(1) 下棋方法

每次下棋的时候读取到当前的行，列和颜色，把颜色存在 data 数组相应的位置。

(2) 读棋方法

返回指定行列位置的棋子颜色。

(3) 判断输赢

从横行、竖行、斜行三个角度判断输赢，其中斜行还要分为一三象限斜行和二四象限斜行。判断每一种行时都用两个循环进行，用一个 int 型数据记录除当前最后一个落下的棋子以外相同颜色的棋子个数。先从最后一个棋子落下的地方开始向一边进行判断，如果判断的位置超出棋盘范围或者不是与该棋子相同的就停止计数，向另一边进行判断。在过程中每判断一个棋子都要查看计数是否为 4，如果与 4 相等，则游戏结束。反之，继续从别的行进行判断。

(4) 悔棋

每次都取出存放下棋顺序的栈的栈顶的对象，将该对象对应的行列位置的棋子在 data 数组中的颜色改为 SPACE，则在面板重画时自动实现悔棋。

(5) 清空键盘

利用两层循环，将 data 数组的每个数据都变成 SPACE。

(二) 控制器 (Controller)

1. 属性

控制器中的属性包括当前棋子颜色、是否接受下棋等。

2. 方法

(1) 本地下棋流程

每次下棋调用本地下棋函数，判断当前点击的位置是否为空。如果为空，就将当前点击的新数据传入 Model 存储。每次下完棋判断当前是否决出胜负。

（2）网络下棋流程

在本地下棋的基础上接收对方的下棋数据，与本地下棋的流程基本一致。

（3）选择先后手

默认选择先手的一方是黑棋。设置选择先手的一方可以下棋，选择后手的一方的 allowPutChess 为 false，棋盘锁定。

（4）重玩

重玩时调用 Model 类中的清空棋盘函数，并调用选择先后手的函数重新设置游戏。

（5）邀请

游戏结束时双方可以邀请对手继续切磋，如果双方都同意，则直接调用重玩函数。

（6）游戏结束

将 allowPutChess 函数设为 false，锁定界面，游戏结束。并且打开 Model 中切磋函数的按钮，使得游戏可以重复进行。

（三）视图（View）

1. 开始界面

小游戏在开始时设置了一个窗口让玩家选择是进行本地版还是网络版，本地版和网络版分别对应了不同的游戏窗口。

2. 游戏界面

（1）棋盘

棋盘面板继承了 JPanel 类来实现面板的刷新和画图等功能。

其中，在显示棋盘时，面板读取 Model 中 data 的数据进行画图，将列与行分别转换为间距，在画图时呈现。

在接收下棋位置时，添加鼠标动作的监听器，在鼠标点击时接收鼠标点击的横坐标与纵坐标，进行数学运算得到所点击位置的行和列。

在本地版的游戏界面中，我还添加了显示当前落子的棋子颜色的 JLabel 给玩家更好的游戏体验。

（2）网络面板

网络面板中包括了连接服务器、聊天界面、先后手的选择等等内容，这些界面中都添加了监听器与 Controller 连接来实现他们具体的操作。

（3）功能面板

除了以上必备的功能以外，小游戏还设计了一排功能按钮，有悔棋、重玩、投降、和棋等功能。这些按钮也都和监听器连接来实现他们的功能。

（四）网络助手

由于网络版的存在，小游戏创建了一个网络助手类来帮助实现网络版的基本功能。

1. 连接与监听

在服务器端创建 ServerSocket 来监听客户端的连接，在客户端创建 Socket 连接服务器端。在连接成功的同时调用 Controller 中的选择先后手方法，使游戏正常进行。

2. 发送

在创建 Socket 时用 PrintWriter 创建了 out 输出流。以后每次界面的操作都会给对方传输相应的信息。

3. 接收

在创建 Socket 时用 BufferedReader 类创建了 in 输入流，在新的线程中读取对手传来的信息，用 while 死循环保证一直读取。判断接收到的信息类型是什么，然后根据不同的信息类型来进行进一步的操作。

4. 通信协议

在按动不同按钮的时候会触发不同的监听器，在触发不同监听器的时候，服务端和客户端都给对方传输了不同的信息。

（1）下棋信息

每次下棋时，一端都会给另一端传输 PutChess:row,col 形式的信息。小游戏利用 java 语言中 String 自带的去头方式 substring 把前面的” PutChess:” 去掉，得到后面有效的信息。再利用”,” 来分割 String，得到行和列的值，传给 Controller 进行后续操作。

（2）聊天信息

每次发送聊天内容时，一端都会给另一端传输 Chat:xxxxx 形式的信息。和下棋信息的处理方式一样，利用 substring 方法去掉” Chat:”，得到后面聊天内容的 String，传给 Controller 来进行页面的刷新。

(3) 悔棋信息

一方申请重玩时给另一方传输 applyRepeal 的信息，接收方收到信息后调用 JOptionPane.showOptionDialog 方法弹出对话框。用 int 型数据存储 JOptionPane.showOptionDialog 方法的返回值。0 表示同意，1 表示拒绝。通过此判断接收方是否同意。若同意，接收方调用 Controller 中的 netModeRepeal 方法实现悔棋，并且发送 agreeRepeal 的信息给申请方，申请方收到信息后也调用 Controller 中的 netModeRepeal 方法进行悔棋和棋盘刷新。若接收方不同意，则发送 refuseRepeal 的信息给申请方，申请方只能继续游戏。

重玩信息和和棋信息的处理与悔棋信息的处理类似。

(4) 投降信息

一方选择投降时，会向另一方传输 Surrender 的信息，对方接受以后弹出对手已投降的提示，游戏结束。

(5) 邀请信息

在输赢决出或者双方采用投降或和棋的方式结束游戏时，调用 Controller 中的 GameOver 方法使得“继续切磋”的按钮可用，双方都可以通过点击按钮调用监听器给对方发送 applyAllReplay 的继续切磋的邀请信息。接收方收到后会弹出对话框，接收方可选择同意或者拒绝。若选择同意，接收方调用 Controller 中的重玩方法并且传递 agreeAllReplay 的信息给申请方，申请方也调用 Controller 中的重玩函数实现重玩。进行先后手的重新选择后，双方重新开始切磋。

五、总结与展望

本游戏综合了本学期学习的 java 内容，综合运用了栈、java GUI、网络通信等内容，实现了游戏功能，进一步强化了我对 java 知识的运用与理解，使我更熟练了 java 语言的实际运用。

该小游戏还有不足之处，考虑到可以加入游戏大厅、观战功能、音效与音乐、加入图片使界面更加美观等，之前做过有关尝试，但是没有成功。希望通过日后

的学习可以使这个游戏更加完善，能利用 java 实现这个小游戏更多的功能。