Java大作业期末论文

南开大学 计算机科学技术

姓名 唐明昊

学号 2113927

班级 0982

2022年12月9日

Java期末大作业

实验报告

# 摘 要

Java大作业——五子棋的开发设计采用MVC的设计模式。View模块主要负责系统的图形用户界面设计，完成于用户的交互，获取交互所产生的数据并传递给Control进行处理。Model模块实现游戏过程中棋子玩家等具体模型，是数据传递和运输的具体对象。Control模块实现游戏的流程控制，对具体的模式以及游戏过程中的数据进行管理。

关键字：Java；五子棋；MVC

# Abstract

Java big job - Gobang Chess development design using MVC design pattern. The View module is mainly responsible for the design of the graphical user interface of the system, complete the user interaction, obtain the data generated by the interaction and transfer it to the Control for processing. The Model module realizes the concrete model of chess pieces and players in the game process, which is the specific object of data transmission and transportation. Control module to realize the flow control of the game, the specific mode and the game in the process of data management.

**Keywords:** Java; Gobang Chess; MVC

目 录

[摘要 2](#_Toc121425508)

[Abstract 3](#_Toc121425509)

[第一章 引言 4](#_Toc121425510)

[第二章 工具选型 4](#_Toc121425511)

[第三章 系统设计 5](#_Toc121425512)

[第四章 系统关键实现描述 10](#_Toc121425513)

[1. 人机下棋，获取最佳下棋位置算法实现 10](#_Toc121425514)

[2. DataSocket包装处理数据 11](#_Toc121425515)

[3. 自定义对话框联机时交互 13](#_Toc121425516)

[第五章 总结与展望 14](#_Toc121425517)

1. **引言**

Java期末大作业旨在让学生运用所学到的程序设计基本知识与基本技能，帮助学生建立良好的计算机语言思维方法，培养运用计算机语言进行分析问题和解决问题的能力，将所学理论知识应用于本专业及相关领域实际工作中，从而提高实际动手能力、分析问题及综合处理问题能力，为后续专业课程的学习打下良好基础。

本文记录Java大作业——五子棋的开发设计过程，并对系统整体框架设计和系统开发过程中的关键模块实现与关键问题解决进行描述，总结设计开发遇到的困难以及收获，在此基础上对以后的面向对象程序设计开发做出展望。

1. **工具选型**
2. Java虚拟机版本：

Java version "1.8.0\_341"

Java(TM) SE Runtime Environment (build 1.8.0\_341-b10)

Java HotSpot(TM) Client VM (build 25.341-b10, mixed mode, sharing)

1. Java编译器版本：

Javac 1.7.0\_06

1. 集成开发环境：

Eclipse IDE for Java Developers (includes Incubating components)

Version: 2022-09 (4.25.0)

1. **系统设计**

采用MVC的设计模式，分为View，Model，Control三个模块进行设计与开发。各Package之间相互独立，但又通过数据传输进行耦合。图1为系统的整体类图。图2为程序流程流程图。

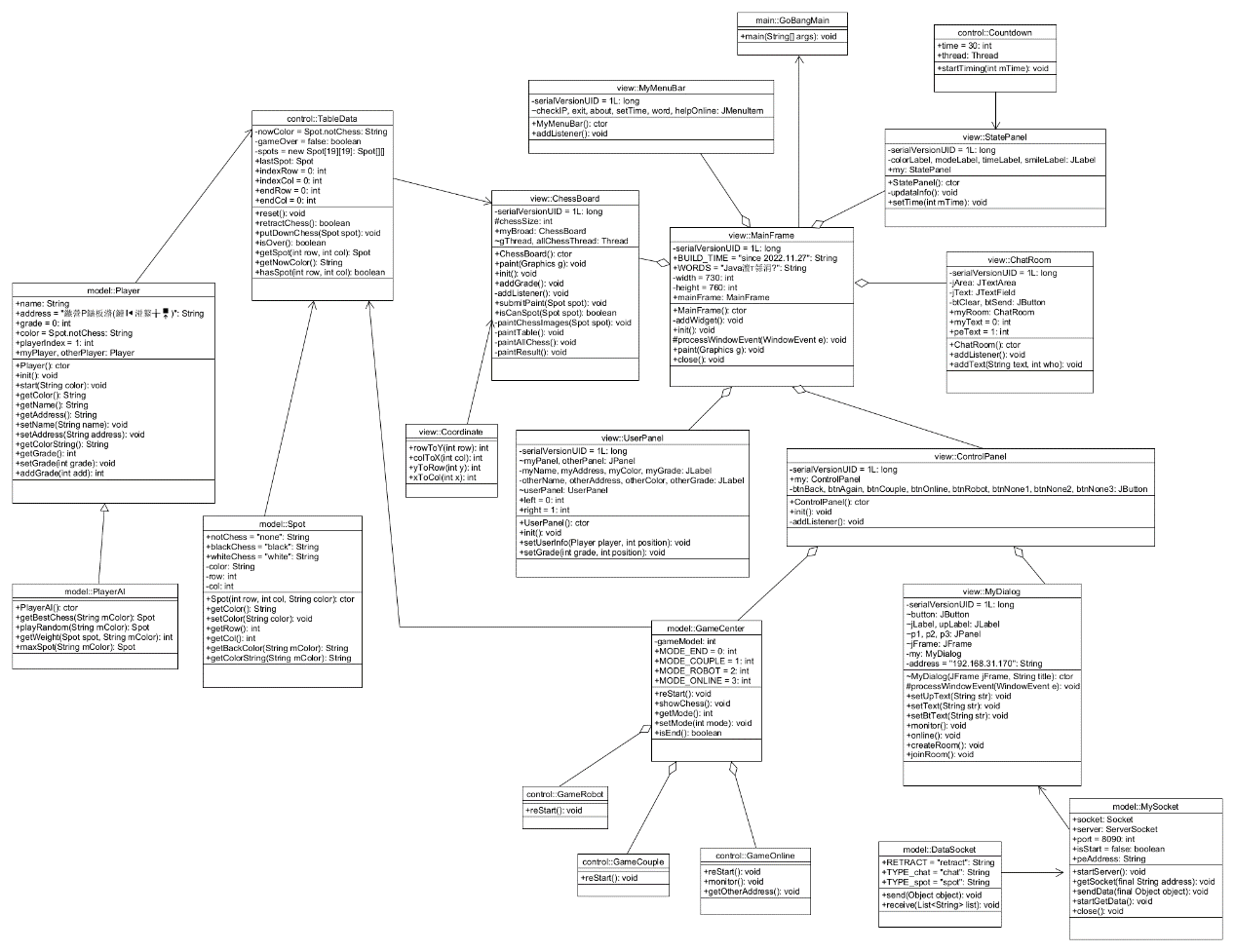


图1 程序类图

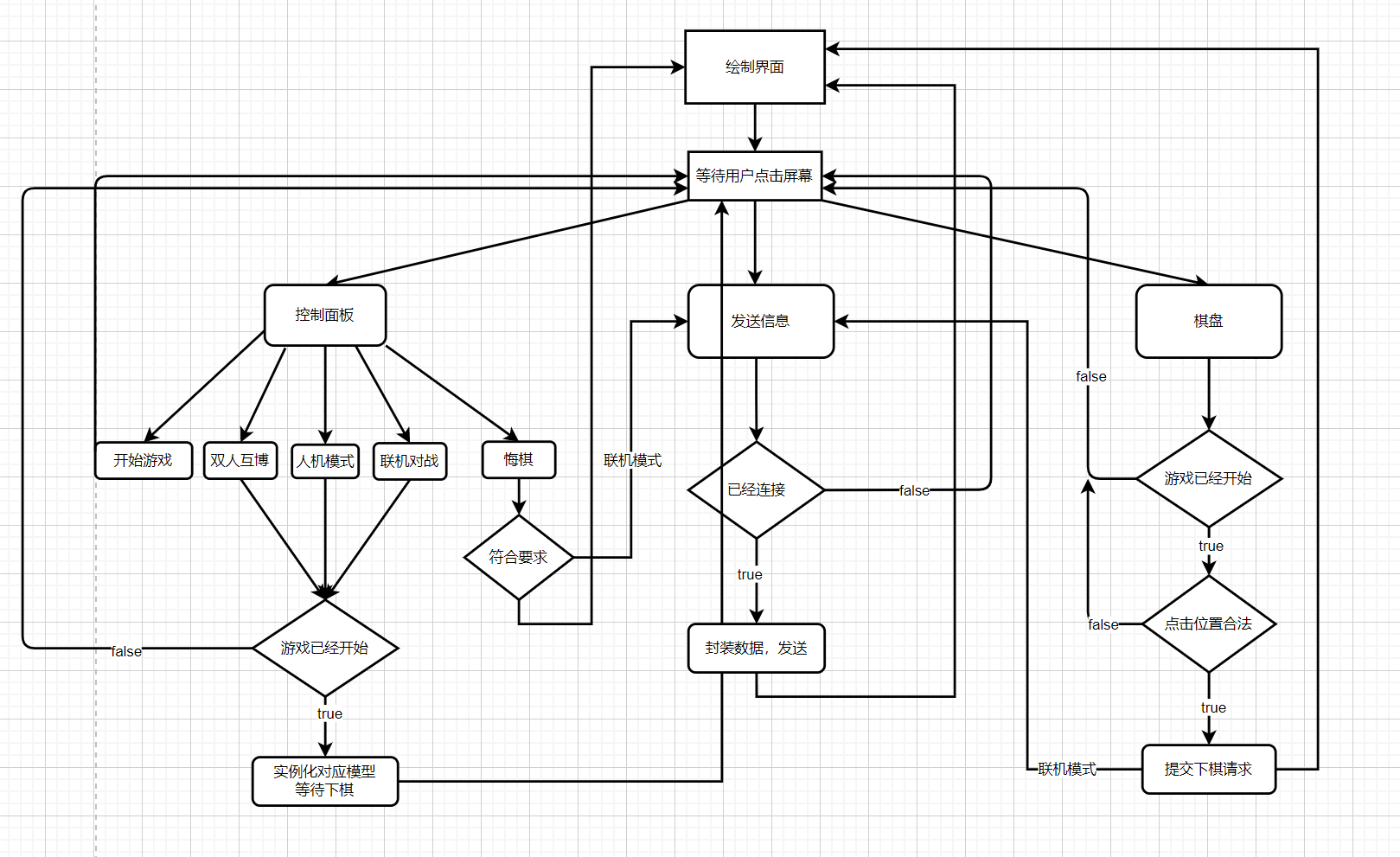


图2 流程图

1. View模块主要负责系统的图形用户界面设计，完成于用户的交互，获取交互所产生的数据并传递给Control进行处理。首先搭建主框架MainFrame，直接继承自JFrame，再往里面添加各样的自定义控件并挨个调用初始化init()函数。图3为系统最终实现的用户界面。



图3 系统GUI界面

MyMenuBar继承自JMenuBar，为系统添加菜单栏。设置有菜单，设置，帮助三个菜单栏，并在下面分别设置有相应的菜单项。MyMenuBar类相对独立，除了设置倒计时功能会在状态栏StatePanel上显示时间文本外，其余按钮均不涉及与其他类进行功能上的交互与信息传递。

UserPanel继承自JPanel，GUI界面下部的玩家信息面板，显示两名玩家信息，必要时更新信息。在init()函数中调用setUserInfo()函数，根据传入的Player对象参数设置面板上的信息。设置有setGrade()函数，用于给ChessBoard类在游戏结束后对获胜玩家进行加分。

ChessBoard继承自JPanel，棋盘类，是系统的核心部分之一。在棋盘面板上设置有监听器，监听鼠标事件，完成游戏核心功能——下棋。定义isCanSpot(Spot)函数，从该处是否已有棋子，游戏是否已经结束，联机模式下是否轮到自己下棋三个方面考虑当前棋子能否放置；submitPaint(Spot)函数，用于在判定玩家可以下棋后提交下棋请求绘制棋子，向TableData传递下棋数据，并在下棋后调用接口判断游戏是否已经结束；paintChessImages(Spot)，paintTable()，paintAllChess()，paintResult()四个函数，分别实现绘制棋子，绘制棋盘，绘制所有棋子，绘制游戏胜利后五子连线的功能。

StatePanel继承自JPanel，游戏的状态面板，设置有几个标签，实时刷新显示当前选择的游戏模式以及当前下棋玩家。updataInfo()函数建立一个独立的线程，无限循环，从TableData类中获取当前下棋棋色并跟新到标签上，另一方面从GameCenter类获取当前所选游戏模式，更新到相应的标签。

ChatRoom为聊天区，由一个JTextArea，一个JTextField，以及两个按钮组成。两个按钮设置监听器，清空按钮将Area区域的文字全部清空；发送按钮检查完MySocket连接状况以后，从TextField获取字符串并调用DataSocket类的send(String)函数处理发送信息。另定义有addText(String,int)接口来设置Area的文本内容。

ControlPanel为游戏右侧按钮面板，由多个按钮组成，是游戏功能的重要部分。每个按钮设置有音效与监听器，并在按钮间设置分割线。开始游戏按钮点击后唤醒GameCenter以及ChessBoard，并将MySocket的连接处于断开状态；三个游戏模式按钮在点击后会将开始游戏按钮置红，更矮其文本为重新游戏，并将其余几个按钮设置为无法点击，最后再让对应的游戏模型开始允许；悔棋按钮调用TableData.retractChess()悔棋接口，再让ChessBoard进行重新绘制，如果是联机模式还将发送悔棋信息给对方玩家。

另外定义Coordinate类用来将行列位置与面板的XY坐标互相转换。MyDialog类继承JDialog用于选择联机模式后弹出进行交互提示和数据传递。

2. Model模块实现游戏过程中棋子玩家等具体模型，是数据传递和运输的具体对象。

GameCenter类存储当前的游戏模式并提供对外接口。

Player玩家类，存储玩家的姓名，IP地址，得分以及棋色，定义有两个静态对象myPlayer和otherPlayer，在选定模式后对两对象进行赋值操作。

PlayerAI直接继承自Player类，用于实现简单的人机对战。定义算法实现下棋功能，getWeight(Spot,String)函数计算一个棋子的权值；maxSpot(String)返回当前传递的棋色参数下最大权值点；getBestChess(String)是对外接口，获取自己棋色与对手棋色的最大权值点后选择最优下棋点返回；playRandom(String)是在获取最大权值点失败后随机找一个空位置点返回。

Spot棋子类，定义三个静态变量对应棋子三种状态：none、black、white，另外定义了私有变量储存棋子对象的行列值和颜色。

MySocket负责两名玩家的网络连接。变量isStart作为连接标志，peAddress储存对方的IP地址。函数startServer()作为Server创建socket，等待client的连接；getSocket(String)作为Client连接到指定的IP地址上去；sendData(Object)发送数据，在异常时将标志置零；startGetData()建立线程，无限循环监听输入，并调用DataSocket类的接口处理数据，同样在异常时将标志置零。

DataSocket是网络连接核心部分之一，进行数据的包装与处理。发送信息时send(Object)对发送对象进行判断，并将信息封装成字符串的形式，最后调用MySocket.sendData()发送；receive(List<String>)负责解析数据，判断信息的类型后对其进行拆分，传递。

3. Control模块实现游戏的流程控制，对具体的模式以及游戏过程中的数据进行管理。

TableData类是游戏核心部分，记录了当前应下棋色，最后一次下棋的棋子，游戏结束时五子连珠始末位置，棋盘数组spots[][]等重要数据。定义reset()函数负责初始化所有数据；retractChess()函数负责执行悔棋操作，更新棋盘数组；putDownChess(Spot)下棋，对棋盘数组进行更新；isOver()遍历棋盘，根据算法判断游戏是否结束，并返回标志；另外定义了getSpot(int,int)，getNowColot()，hasSpot(int,int)等接口。

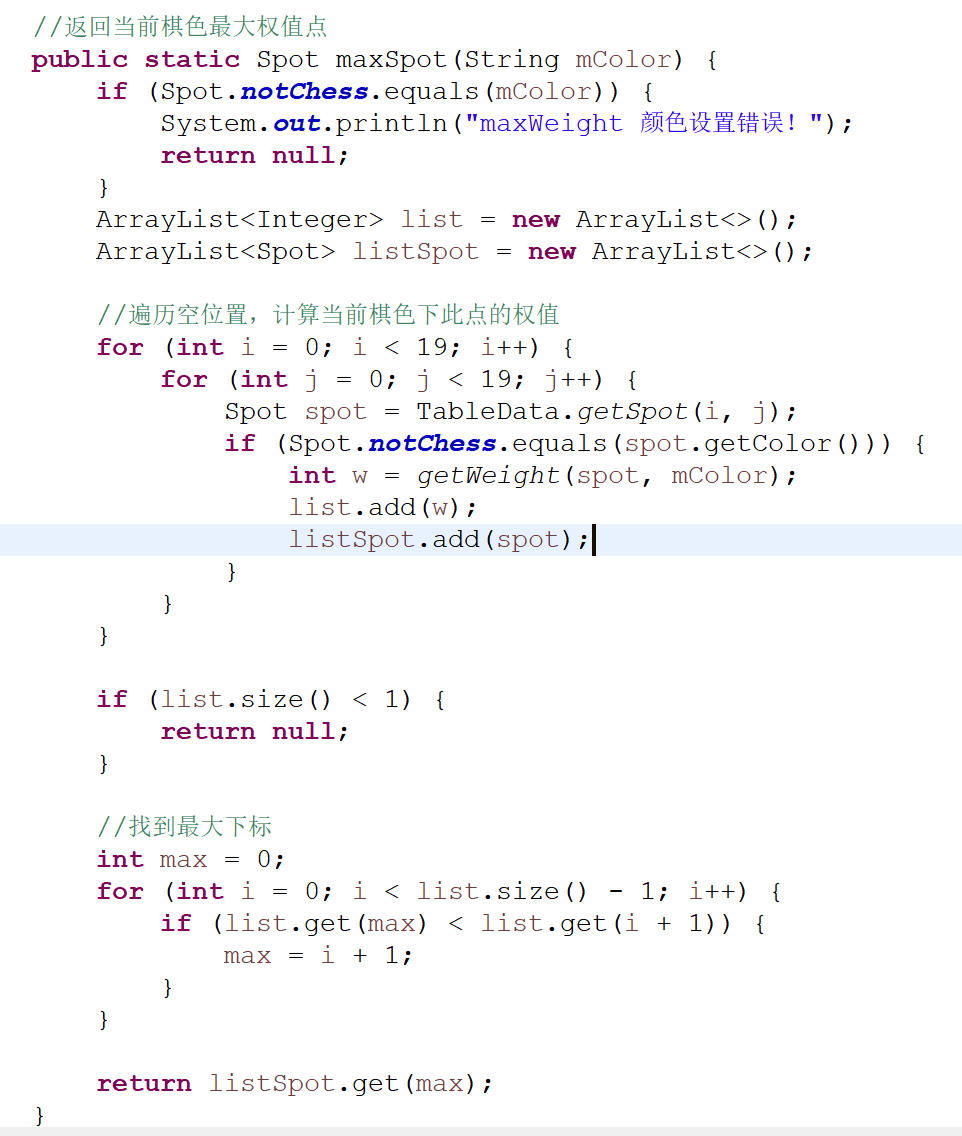
三个游戏模式类GameCouple，GameRobot，GameOnline。在ControlPanel被选择后刷新GameCenter数据，更新用户面板上的信息。GameRobot专门为AI建立下棋线程，每隔一秒进行一次尝试，轮到AI下棋则使用算法获取最佳下棋位置并调用ChssBoard.submitPaint(spot)下棋。GameOnline定义有监控线程，监控对方是否下线，一旦MySocket断开连接则调用MyDialog的接口弹出提示对话框。

1. **系统关键实现描述**
2. 人机下棋，获取最佳下棋位置算法实现

对于棋盘上一个位置，默认权值为0，遍历其上下左右四个方向各五个棋子，若有棋子符合要求的棋色则权值增加，离当前点越近增加的权值越多，得到最终的权值。



对于某一个特定棋色，遍历棋盘上的所有空位置点，获取它们的权值，找到在当前棋色下的最大权值位置。



在getBestChess()中调用两次maxSpot(String)函数，分别获得两个棋色的最大权值点，比较两个权值，权衡是进攻还是围堵，返回最佳下棋点。



1. DataSocket包装处理数据

需要发送的数据类型分两种，棋子和聊天信息，一个为Spot类，一个为String类，通过instance of判断参数object的类型再封装成需要发送的字符串。如果是Spot类，则提取出行列值放在关键字后，合成字符串链表。如果是String类则直接添加进入字符串链表。



解析数据时通过关键字进行判断，若是棋子信息则提取出行列值再调用ChessBoard的submitPaint(Spot)函数下棋；若是聊天信息则在ChatRoom添加上字符串。而悔棋操作是通过发送一个特定的字符串实现，是在解析聊天信息时进行判断，若符合则执行悔棋操作。



1. 自定义对话框联机时交互

选择联机模式后弹出自定义对话框，提供两个选择，创建房间与加入房间。选择创建房间，设置完用户面板信息后，作为Server创建连接线程，监控线程。另一方面，需要维护窗体。若连接成功，则修改对话框的标签，按钮的文本，确定开始游戏。在等待过程中，实时修改对话框的文本，营造窗体动画效果；若遇窗体被异常关闭，则需要及时终止socket。加入房间与创建房间流程相同，只需在调用MySocket的接口时以Client的身份调用。

监控功能通过建立线程，无限循环监控MySocket的标志实现。当信息发送接收出现问题时标志会被修改，此时监控循环被打断，弹出提示对话框。





1. **总结与展望**

本次Java大作业让我完整的体会了面向对象设计的过程，对开发，对知识以及知识的应用有了更深刻的理解。

在开发过程中需要不断学习知识，查阅资料，对于一个功能的实现往往还需要涉及到开发工具的更新。学会了采用MVC开发模式对问题进行适当拆分，对具体步骤进行合理安排，避免过度耦合。大型项目的开发也让我的编程习惯进一步的提高：规范的命名，必要的注释，提高打字速度。对于线程的理解更进一步，特别是在播放音乐，绘制棋盘时，如果不使用新建线程实现会导致系统卡顿，绘制不完全的现象；在进行监听时也需要线程进行不断循环的查看。初步理解了Server和Client的通信机制，加深了之前在作业中体会到的浅显的认识，也从查阅来的资料中学习到了封装处理数据的很好的思路。对Try Catch机制有了更深刻的了解，学会了对其进行利用。

本次的大作业设计开发一定会对我往后的学习以及项目开发提供很好的经验：学习将更加具有目标性和规划性，对需要额外补充学习的资料能够较短时间内在网上进行查询和获取；项目开发也会更具经验，不会出现不知从哪入手，设计出来的程序过分耦合的现象。

最后不得不说，Java课是我大学目前以来拿的最“充实”的2.5学分，确实是很累，但代码能力提升也是实实在在的，寒假我也将继续学习精进！