

---

# 基于 Java 的黑白棋游戏的设计与实现



学 院 软件学院

专 业 软件工程

年 级 2007 级

姓 名 刘程

指导教师 张坤龙

2011 年 6 月 11 日

## 摘 要

黑白棋又称为奥赛罗棋，是一项历史悠久的棋类游戏，同时也是一项很好的益智对弈游戏。论文运用 Java 编程语言设计和实现了一个名为 Haroro 黑白棋游戏软件。

根据需求分析，基于 Java 的黑白棋游戏的设计与实现应有的功能主要分为三大类。第一类是基本的落子功能，即能在棋盘面板上下子，并完成棋子的反转，以及在对方玩家的棋盘面板上绘制出同样的棋子。第二类功能包括进入游戏大厅、建立游戏、加入游戏、放弃游戏等这些基本功能。第三类功能是游戏辅助功能，包括游戏面板变色、聊天等等。

基于 Java 的黑白棋游戏的设计与实现运用到了很多 Java 方面的知识，包括进行图形用户界面设计所需的 Java GUI 类、事件驱动程序设计、异常处理、多线程编程、网络通信等等一些技术。

论文将详细介绍了基于 Java 的黑白棋游戏的设计与实现所运用到的技术和相关知识，并对此次设计开发的过程和思路进行了论述。

**关键词：**黑白棋；Java；套接字；多线程编程

## **ABSTRACT**

Reversi is also called Othello. Reversi is a board game with a very long history, it is also a good educational two-player game. The goal of this project is to accomplish a Othello which can provide two players compete with each other in LAN with the use of Java.

Before starting programming, there are a lot of things should be done, first, Software Requirements Analysis, then according to the analysis, learn the tools and skills that needed to accomplish the project. After that, design the use interface and make it simple and attractive. And at last code and develop the project.

This project involves a lot of Java skills, including Java GUI class, event-driven programming, exception handling, multithread programming and network communications.

This thesis will describe the relative skills and the whole design and programming procedure in details.

**Key words:** Othello; Java; socket; multithread programming

# 目 录

第一章	绪论.....	1
1.1	黑白棋 .....	1
1.2	JAVA 语言 .....	3
1.3	论文的组织结构.....	5
第二章	HARORO 黑白棋系统的需求分析 .....	6
2.1	功能需求 .....	6
2.1.1	服务器端功能需求.....	6
2.1.2	客户端功能需求.....	8
2.2	软硬件需求.....	12
第三章	HARORO 黑白棋系统的设计 .....	13
3.1	概要设计 .....	13
3.2	界面设计.....	15
3.2.1	服务器端界面设计.....	15
3.2.2	客户端界面设计.....	17
第四章	HARORO 黑白棋系统的实现 .....	22
4.1	服务器端实现 .....	22
4.2	客户端实现.....	25
4.3	棋盘面板实现.....	27
第五章	总结与展望.....	29
5.1	总结.....	29
5.2	展望.....	29

参考文献.....	31
外文资料	
中文译文	
致谢	

## 第一章 绪论

### 1.1 黑白棋

棋牌在中国是一项悠久的运动，具有广泛的群众基础。棋类游戏包括围棋、四国军棋、中国象棋、国际象棋、飞行棋、五子棋、黑白棋等。对中国人来说，有些棋类游戏已经远远超出了它的娱乐功能。例如围棋，它早已成为一种理念，一种生活态度，一种生命的哲学；又如中国象棋，它同样具有悠久的历史，早在战国时期就有了象棋的相关记载。

在棋类游戏中，黑白棋受到很多人的喜爱，它界面简单，且很容易就地取材制作简易棋盘和棋子，加上简单的规则，不管男女老少都对其特别喜爱。

黑白棋，也叫苹果棋、奥赛罗棋，是一个经典的策略性游戏<sup>[8]</sup>。它使用  $8 \times 8$  的棋盘。棋盘如图 1-1 所示。

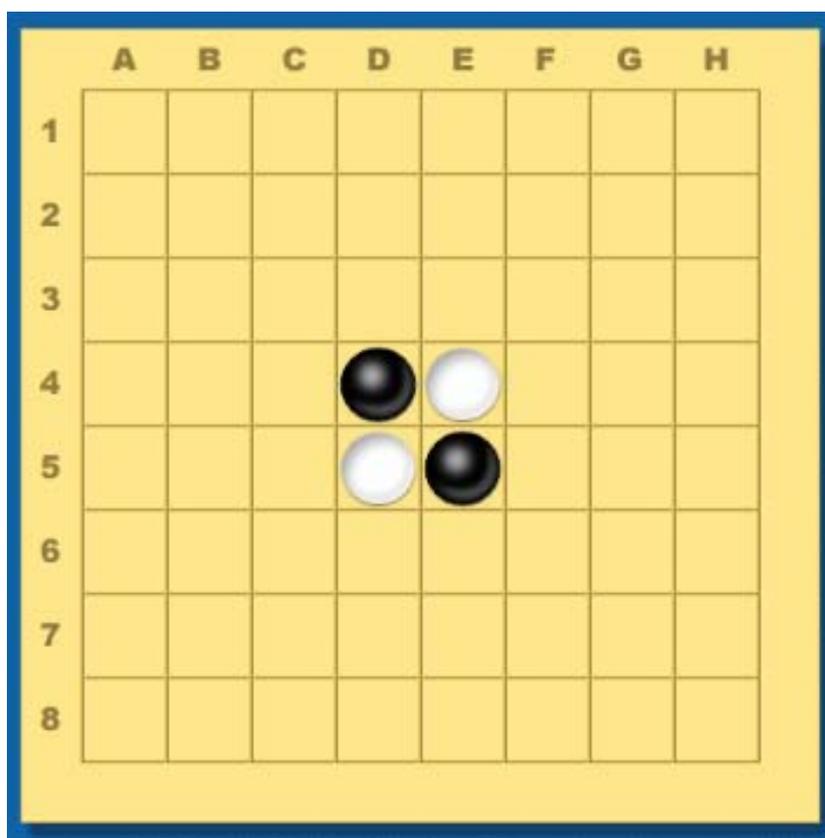


图 1-1 黑白棋棋盘示例

由两人分别执黑子和白子下棋，最后子多方为胜方。黑白棋是由英国人在 19 世纪发明的，黑白棋是借用著名文学家莎士比亚的一部作品中的男主角奥赛罗命名的。黑白棋中的黑棋子和白棋子分别指的是那部作品中的相互斗争的一

个黑人和一个白人。

黑白棋的规则很简单。首先它的棋盘是由  $8 \times 8$  的方格组成，棋子为白子和黑子两种。黑白棋由两方对弈，一方为白方，执白棋，另一方为黑方，执黑棋，其中，执黑棋的先手。

在一盘棋开始前，黑白双方各有两个棋子在棋盘中央放置，且放置的位置交错。黑白棋的赢棋判定是根据当棋盘已放满双方棋子或双方都没有位置可以落子时，哪方在棋盘上的棋子数，则哪方为胜利方。

黑白棋是一种双人完全信息游戏，在这样的游戏中，游戏双方在游戏进行中的任何时刻都能完全地获悉游戏中的所有信息，一方知道的信息，另一方也一定知道，没有信息是可以只让其中一方独自占有的<sup>[13]</sup>。因此战术很重要，同时每走一步都要非常小心，要想到这一步对自己和对方游戏玩家会各自产生怎么样的影响，玩家需要注意棋盘的每个位置，并尽量想到对方可能选择的落子位置，这样才能做出较好的判断，并选择较好的位置落子。

玩家的目标是使自己的棋子最多，而使对方游戏玩家的棋子数最少。那怎样才能使自己一方的棋子更多地出现在棋盘上呢<sup>[14]</sup>？这涉及到反转的规则。黑白棋的下法为黑方先下第一步，然后双方交替下子，落子的地点必须是能够将对方的若干棋子反转成自己一方颜色的棋子的位置，否则无法落子，当一方没有落子的位置时，则由另一方继续落子，直到对方有落子的位置。反转棋子是指当一方落下一个棋子后，自己一方有两个棋子在横、竖、斜四个方向上的任何方向上夹住了对方的一个或多个棋子时，对发的这些被本方棋子夹住的棋子将会反转成代表本方颜色的棋子。在进行游戏时，当一方有位置可以落子时，不可以跳过，一定要落子。

落子到黑白棋棋盘上的各个格子所带来的效益是不同的，比如落子到边角会带来非常大的益处<sup>[15]</sup>。黑白棋的玩法中边角的下法非常重要，是胜负的关键。绝大多数情况下，占领了边角的玩家将会在游戏中取得最后的胜利，因为首先，占住棋角后，棋角上的棋子是不可能以后的游戏中被对方玩家反转成对方的棋子的；其次，边上的棋子也不易被反转，相反，这些边角上的棋子很容易为翻转对方玩家的棋子提供便利。因此，占领棋盘边角很重要，玩家要尽量努力控制边角，制约对方。

需要注意到是玩家自己一定要避免没有落子位置的情况出现，因为这样会让自已非常被动，对方玩家将获得连续落子的机会，进而更好地控制棋局；相反，本方玩家要尽力使对方游戏玩家没有落子的位置，让自己能够拥有更多次数的落子机会，进而从中获利。

还有几点是游戏玩家需要注意的。首先，如果一个角落是空着的，那么在紧挨着这个角落的方格处下子是致命的。因为下在此处会提供给对手占领角落的机

会。其次，那些邻接着空着的方格的棋子是变幻无常的，因为它们很可能因对方在空格处下子而翻转成为对方的棋子。所以，棋手要尽量减少自己产生这样的棋子。

简而言之，在黑白棋中，每一步都可能对双方的棋子数产生巨大的变化，从而影响最终的胜负，所以黑白棋的每一步都很重要。

黑白棋是一个很好的益智游戏，能够使娱乐时开动脑筋，在轻松中学习，在学习时放松。

中国最早出现黑白棋是在 80 年代的任天堂游戏机和苹果 II 个人电脑游戏里，黑白棋真正发展起来还是在互联网普及以后，黑白棋作为一种经典的策略性游戏，受到了广大网友，特别是得到了希望锻炼智力的网友的喜爱。

## 1.2 JAVA 语言

此次的黑白棋游戏的设计与实现是基于 Java 编程语言的。选择使用 Java 语言的一个重要原因是用 Java 语言实现的程序可以很好地移植到其他计算机上，并能很好地运行<sup>[2]</sup>。

Java 语言是由 Sun Microsystems 公司开发出来的。Java 语言最初的名字是 Oak，Oak 是一种橡树的名字。到了 1995 年，才正式地更名为 Java 语言<sup>[1]</sup>。Java 语言是一种面向对象的程序设计语言，同时它是一种高级编程语言，它是基于 C++ 语言进行开发的一种新型编程语言。

最初开发 Java 语言的原因是在 Sun Microsystems 公司开发一个叫做 Green 的项目时，这个项目组起先准备用 C++ 语言来进行编码开发，但是开发成员们发现 C++ 语言有很多不足，对于这个项目来说，用 C++ 语言太复杂，并且其安全性不能保障<sup>[2]</sup>。所以最终决定开发出一种全新的编程语言，并在当时取名叫 Oak 语言。

Java 已被广泛地接受，在计算机产业的很多大型公司都购买了 Java 语言的使用许可证，并且很多的软件开发商都开发支持 Java 语言的软件产品，加上 Java 语言自己良好的性能和特点，以及自身的不断发展，Java 语言在软件开发中的地位不断提升<sup>[3]</sup>。

经过多年的发展，Java 语言已经称为了一个功能强大的比较完整的计算机编程语言体系。Java 语言已被成功地运用在诸多领域，如 Web 程序设计，跨平台的独立应用程序的开发，以及大型网站、电子商务等应用系统和手机等等一些嵌入式的移动设备中。

在已经公布的 Java2 标准中，规定了 Java 语言的一些应用领域<sup>[4]</sup>。这些领域主要包括一下四个方面：

- ①网络服务器应用程序；
- ②一般计算机应用程序；
- ③嵌入式设备应用程序；
- ④智能卡应用程序。

Java 语言的迅速流行以及被广大开发者接受运用的一个主要原因是它的独立性。Java 语言并不依赖于某种特定的计算机系统或软件平台。用 Java 语言编写的程序可以在任何地方运行使用<sup>[6]</sup>。Java 语言能很好地运用到网络和各种嵌入式设备中。下面将介绍一下 Java 语言的一些特点。

#### ①面向对象编程

在前面已经提到过，Java 语言是基于 C++ 语言开发出来的全新编程语言。它是比 C++ 更彻底的、完全的面向对象的高级编程语言<sup>[11]</sup>。

#### ②简单

Java 语言简单明了，和 C++ 有很多相似之处，一个掌握了 C++ 的编程人员可以非常快地学会并掌握 Java 语言<sup>[7]</sup>。同时 Java 语言比 C++ 在很多方面要简单得多，非常易于学习。其中，一个很明显的不同是，在 Java 语言中，没有指针这一概念。C++ 语言中的指针以及涉及内存管理的部分很容易引起程序发生错误。以为内存管理和指针在高级编程语言中并不是必不可少的，所以在 Java 中，去掉了这些易引发错误的部分，使其更简单。

#### ③分布式程序设计语言

Java 是一个分布式程序设计语言，它支持网络应用程序。Java 语言中有 java.net 包，它是一个基于 TCP/IP 协议的子库，用于支持 HTTP 和 FTP 等等<sup>[12]</sup>。此次项目开发就运用到了 Java 语言的这一特性，创建了分布式的黑白棋游戏服务器端和客户端程序。

#### ④安全性

Java 语言是一个安全性很高的高级编程语言。在前面已经提到了在 Java 语言中没有指针等这样一些涉及到内存管理的内容，这样极大地提高了 Java 语言的安全性。同时，Java 语言程序在防止病毒入侵上也做得很好。Java 语言的字节代码的验证方式也保证了网络上的 Applet 小程序是安全无毒的。

### ⑤独立性

Java 不依赖于某种特定的操作系统或软件平台。Java 语言易于移植，并且移植时不需要再进行编译。在一台计算机上编写的 Java 程序可以在其他任何实现了 Java 虚拟机的计算机上得以运行。

### ⑥支持多线程

Java 语言支持多线程开发，即能使多个程序、线程并行执行，实现线程同步。此次项目开发也多次运用到 Java 语言支持多线程的这一特点。

下面介绍一下 Java 程序的主要开发工具以及此次开发所选用的开发环境和工具。

Java 程序的开发工具主要有 Java 开发工具箱——JDK、Borland 公司的 JBuilder、JCreator 以及 IBM 公司的 Eclipse<sup>[5]</sup>。这些 Java 工具的作用是为 Java 语言程序的开发提供一个集成开发环境——IDE。

此次的基于 Java 的黑白棋游戏开发所选用的工具是 JDK 和 IBM 公司的 Eclipse。

## 1.3 论文的组织结构

本篇论文首先对黑白棋的历史和规则进行简要说明，接着描述了 Java 语言的相关概念。然后逐步讨论并描述此次基于 Java 的黑白棋游戏的设计与实现的整个过程和具体细节。（注：本文将实现的黑白棋游戏命名为 Haroro 黑白棋系统。）

第二章将描述 Haroro 黑白棋系统的需求分析，包括服务器端和客户端的功能需求分析，以及软硬件需求分析；

第三章将展示 Haroro 黑白棋系统的设计，包括整体概要设计，服务器端和客户端的界面设计；

第四章会展示 Haroro 黑白棋系统的实现细节；

第五章会对 Haroro 黑白棋系统进行总结和展望。

## 第二章 HARORO 黑白棋系统的需求分析

本章将描述 Haroro 黑白棋系统的设计与实现的需求分析。其中包括功能需求和非功能需求，硬软件需求等等。通过本章的描述，读者可以对此次基于 Java 的黑白棋游戏的设计与实现的目标和前景有一个大致的了解和初步认识，并将弄清楚此次工程最终大概将呈现出的游戏运行结果。

进行需求分析是非常必要的。因为只有做出正确的完善的需求分析，才能更好地指导接下来的开发过程，为整个项目的开发提供了指示和规范作用<sup>[9]</sup>。

### 2.1 功能需求

在 Haroro 黑白棋系统的设计与实现的工程中主要涉及两类用户：服务器端用户和客户端用户。下面将分别对这两类用户进行功能需求分析。

#### 2.1.1 服务器端功能需求

(1) 服务器端用户用例图。

服务器端用户的用例图如图 2-1 所示。

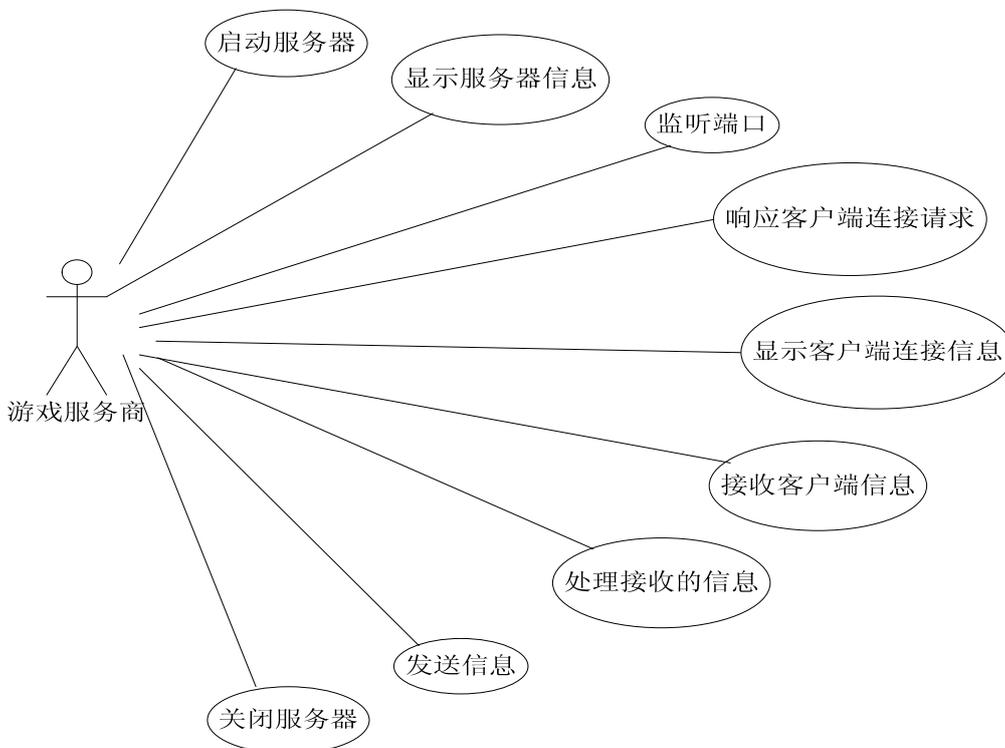


图 2-1 服务器端用户（游戏服务商）用例图

此类用户充当黑白棋游戏的服务商，为游戏玩家提供服务器支持。此类用户将要在计算机上运行服务器端应用程序，让游戏玩家可以通过游戏客户端连接在同一局域网内的服务器。

服务器端用户在运行服务器端应用程序后，服务器运行结果框将显示服务器信息并等待响应来自客户端应用程序的连接请求。当监听到很客户端程序请求连接，即请求进入游戏大厅时，服务器端会给予回应，允许其进入黑白棋游戏大厅。待客户机进入大厅后，服务器会继续监听等待来自客户端程序发来的信息，然后做出相应的回应。

由于客户端会根据游戏玩家的操作，向服务器发送不同类型的消息。因此，在接收到不同类型的信息时，服务器会采用不同的应对措施，做出不同的有效回应。

当黑白棋游戏玩家想要建立一盘游戏时，在点击相应的按钮后，服务器端会接收到表明游戏玩家想要建立游戏棋局的信息。服务器会进行相应反馈，让客户端成功建立一盘棋局；

而当服务器端接收到客户端发来的表明某一游戏玩家申请加入由另一游戏玩家已建立的棋局中时，服务器会向客户端传递加入游戏成功，游戏开始的信息给客户端；

若服务器接收的是一方游戏玩家的落子信息，则会把这个信息传给另一游戏玩家，方便其在本方棋盘上显示对方的落子位置；

如果服务器接收到了游戏玩家放弃游戏的信息，则判定此局游戏结束；

当服务器接收的信息是游戏玩家关闭游戏客户端的申请时，服务器将在服务器端显示出退出消息。

## (2) 服务器端功能说明

### ①启动服务器

首先游戏服务商要启动服务器，才能让游戏玩家的客户端连接到服务器，进而进行游戏。

### ②显示服务器信息

在游戏服务商启动服务器后，服务器面板上将会显示出服务器端的信息，包括服务器 IP 和监听的端口号。

### ③监听端口

服务器会监听某一指定的端口，本文指定的端口为 9001，当然也可以选择其它的端口号，只要不是已被占用。服务器监听此端口，检测是否有客户端提交

连接服务器的请求。

④响应客户端连接请求

当服务器监听到有客户端请求连接服务器时，服务器会作出相应的响应，并创建一个新的线程来处理以后和这个客户端的信息传递。

⑤显示客户端连接信息

当有客户端连接到服务器或者有客户端关闭时，服务器显示信息的面板除了显示服务器的信息外，还将显示客户端连接或退出服务器的信息。

⑥接收客户端信息

在客户端连接到服务器后，客户端即可与服务器进行信息沟通。服务器会接收来自客户端的信息。

⑦处理接收的信息

服务器会接收到来自客户端的信息。而这些信息分为很多种类，对不同类型的信息，服务器会采用不同的处理方式。

⑧发送信息

服务器处理来自客户端的信息后，会将处理结果反馈到指定的客户端。

⑨关闭服务器

要停止游戏服务时，游戏服务商可以选择关闭服务器。

## 2.1.2 客户端功能需求

(1) 客户端用户用例图。

客户端用户的用例图如图 2-2 所示。

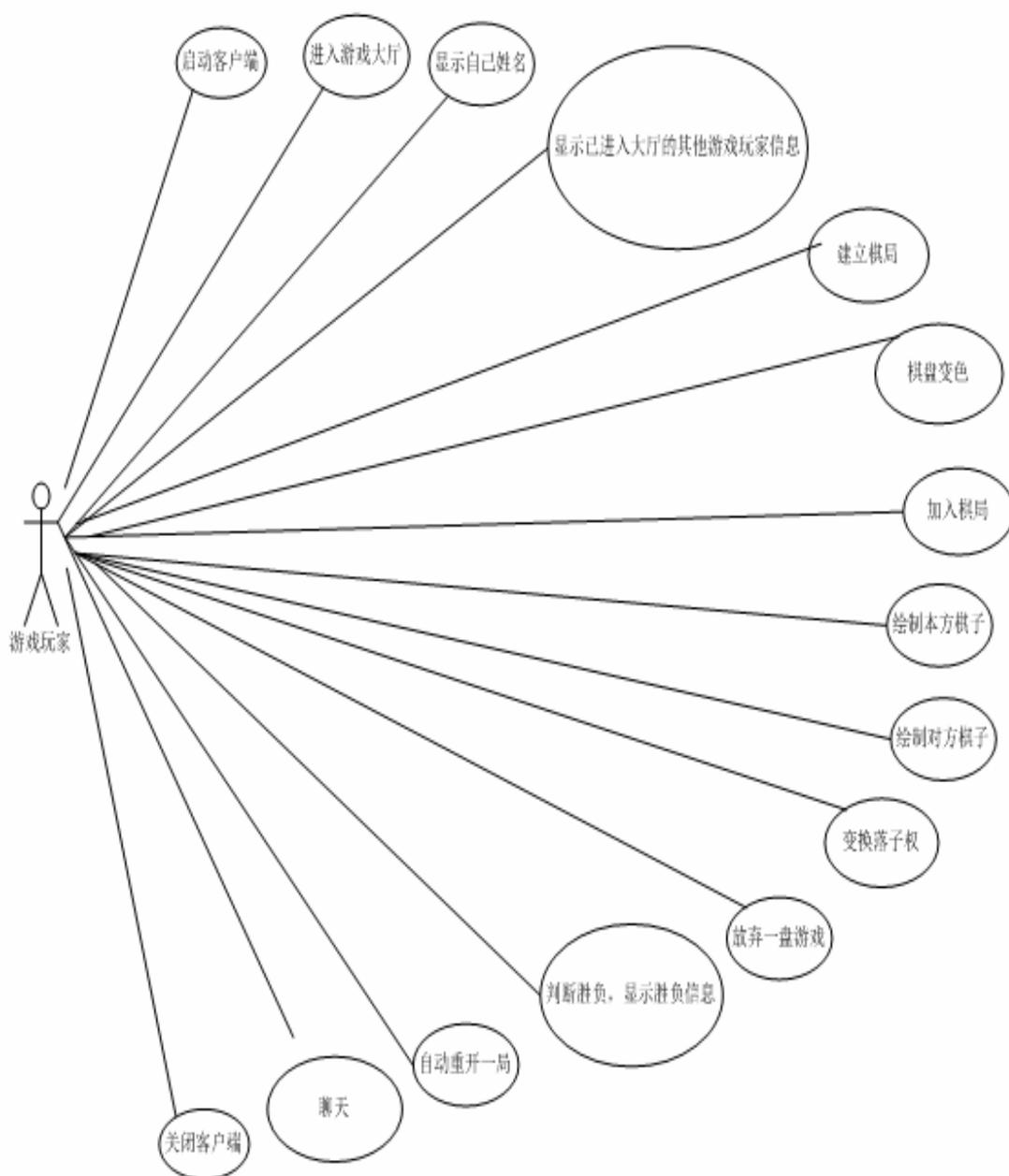


图 2-2 客户端用户（游戏玩家）用例图

此类用户在此次项目中充当的角色是黑白棋网络游戏玩家。客户端用户运行黑白棋游戏客户端应用程序后，可申请进入网络黑白棋游戏大厅。当服务器端用户已开启服务器，即已运行网络黑白棋服务器端应用程序时，服务器会给予客户端响应，让客户端成功进入游戏大厅。

在一游戏玩家进入游戏大厅后，他可能选择自己建立一盘棋局，然后等待其他游戏玩家加入他所建立的棋局中，也可选择加入一由其他游戏玩家建立的棋局中，而对于这些游戏玩家的操作，服务器都将给予适当响应，以支持游戏玩家的操作。

当一盘棋成功开始后，游戏玩家可以在轮到自己落子时，进行落子操作，并且本方的落子消息会通过服务器的传递到对方的客户端，从而在对方游戏面板上进行落子的绘图。

在游戏进行途中，游戏玩家可选择放弃本盘游戏。并且，在游戏玩家开启客户端后的任何时刻，都可以选择关闭客户端。

## (2)客户端功能说明

### ①开启客户端

玩家要进行黑白棋游戏首先要启动黑白棋游戏客户端，然后才能连接服务器，进而进入游戏大厅、进行游戏。

### ②进入黑白棋游戏大厅

当玩家启动客户端后，必须进入黑白棋游戏大厅，才能进行游戏。点击进入游戏大厅的按钮后，在以服务器已启动的前提下，玩家将顺利进入游戏大厅，进入游戏大厅后，客户端会显示玩家的相关信息并且玩家将可以进行相关游戏操作。

### ③显示玩家自己的名字

玩家进入游戏大厅后，系统应显示出玩家自己的名字。显示的区域是在游戏客户端的左上角，显示在游戏名称的旁边的。显示玩家自己的名字能够方便玩家进行相关的后续操作。

### ④显示已进入游戏大厅的其他游戏玩家的名字

在游戏玩家进入游戏大厅后，除了在客户端左上角显示出玩家自己的名字外，系统还将在游戏面板的右侧以列表的形式显示出其他已进入游戏大厅的玩家的名字。方便玩家加入由其他游戏玩家建立的棋局或者与指定游戏玩家进行聊天。

### ⑤建立游戏棋局

当玩家进入游戏大厅后，想要玩黑白棋游戏，可以自己建立一盘游戏，等待其他游戏玩家加入自己建立的游戏棋局，当有其他游戏玩家加入自己建立的游戏

时，游戏将自动开始，建立游戏的一方执黑子，先下子，加入游戏的一方执白棋，后下子。

#### ⑥加入游戏棋局

在游戏玩家进入游戏大厅后，玩家想要进行黑白棋游戏的另一个选择是加入由其他游戏玩家建立的棋局中，加入由其他玩家建立的游戏棋局后，游戏将自动开始，建立游戏棋局的一方执黑子，先下子，而加入游戏棋局的一方执白棋，后下子。

#### ⑦棋盘变色

为了实现界面的美观，当一盘棋局成功开始时，游戏双方的棋盘和部分按钮的颜色都将改变，并且建立游戏的玩家的棋盘和按钮的颜色将不同于加入游戏的玩家的棋盘和按钮的颜色。棋盘颜色的变化能使游戏界面更加美观新颖，提高游戏的趣味性。

#### ⑧绘制本方棋子

当轮到自己下子时，游戏玩家点击棋盘上自己选择的可落子的位置后，应绘制出棋子，并反转对方部分棋子。

#### ⑨绘制对方的棋子

当本方落子后，要将本方的落子信息通过服务器发送到对方客户端。对方根据所接收的信息在自己棋盘上绘制出棋子。

#### ⑩变换落子权

游戏有黑方先下子，然后白方后下。此后一方落子后，原则上落子权将转交给游戏的另一方。但是如果对方在棋盘上没有落子的位置，则本方依然持有落子权，直到当对方在下一步可落子时，方转交落子权给对方。

#### ⑪放弃一盘游戏

当游戏玩家觉得一盘游戏没有了获胜的希望或因为其他原因，可选择放弃这盘游戏，视为主动认输。

#### ⑫判断胜负，显示胜负信息

每当游戏双方中的任何一方走一步棋后，系统都将自动检测此局游戏是否已分出胜负或战成平局。若分出胜负或者战成平局，则显示胜负信息。

### ⑬ 一盘游戏结束后自动重开一局

当系统判断一局游戏已分出胜负或者战成平局时，除了显示胜负信息外，还会自动重开一局新的游戏。

### ⑭ 聊天

为了方便游戏双方的沟通，以便更好地进行游戏，游戏中应有聊天功能。当某一游戏玩家想和另一游戏玩家进行聊天时，应先在游戏面板右侧显示已进入游戏大厅的游戏玩家的列表上选择想要进行聊天的对象。然后，在聊天信息输入框中输入要传达给对方的信息，接着，点击发送按钮，则会成功发送聊天信息，发送的信息将会显示在聊天信息面板上。

### ⑮ 关闭游戏客户端

当游戏玩家想离开时，可以在任何时刻选择关闭游戏客户端。

## 2.2 软硬件需求

根据上面的需求分析以及此次基于 Java 的黑白棋游戏的设计与实现工程的特点，本次开发所需要的硬软件设施如下。

运行此游戏的硬件设施：普通的 PC 机即可，一台 PC 机作为服务器，运行服务器端应用程序。

而运行黑白棋游戏客户端时，可以有两种选择。

第一种选择是，在运行服务器端应用程序的 PC 机同时运行客户端程序，既充当服务器，又充当一个客户端。这样，还需要一台 PC 机来运行另一客户端，以实现局域网内双人黑白棋网络对战。

第二种选择是，还需要两台 PC 机来分别充当一个客户端，实现局域网内双人黑白棋网络对战。

开发此游戏所用的系统平台：Windows 7（注：此游戏同样可以在 Windows XP 的环境下正确运行）。

开发环境和开发工具：在安装了 JDK 的前提下，运用 Eclipse 开发工具进行开发。

开发游戏所选用的编程语言：Java 语言

运行此游戏所需的网络环境：服务器和客户端应处于同一局域网内。

### 第三章 HARORO 黑白棋系统的设计

#### 3.1 概要设计

本游戏拥有独立的服务器和客户端，并且服务器可以接收多个用户同时连接。由于需支持双人对战，则游戏进行需要开启一个服务器和两个客户端。

Haroro 黑白棋系统基本框架如图 3-1 所示。

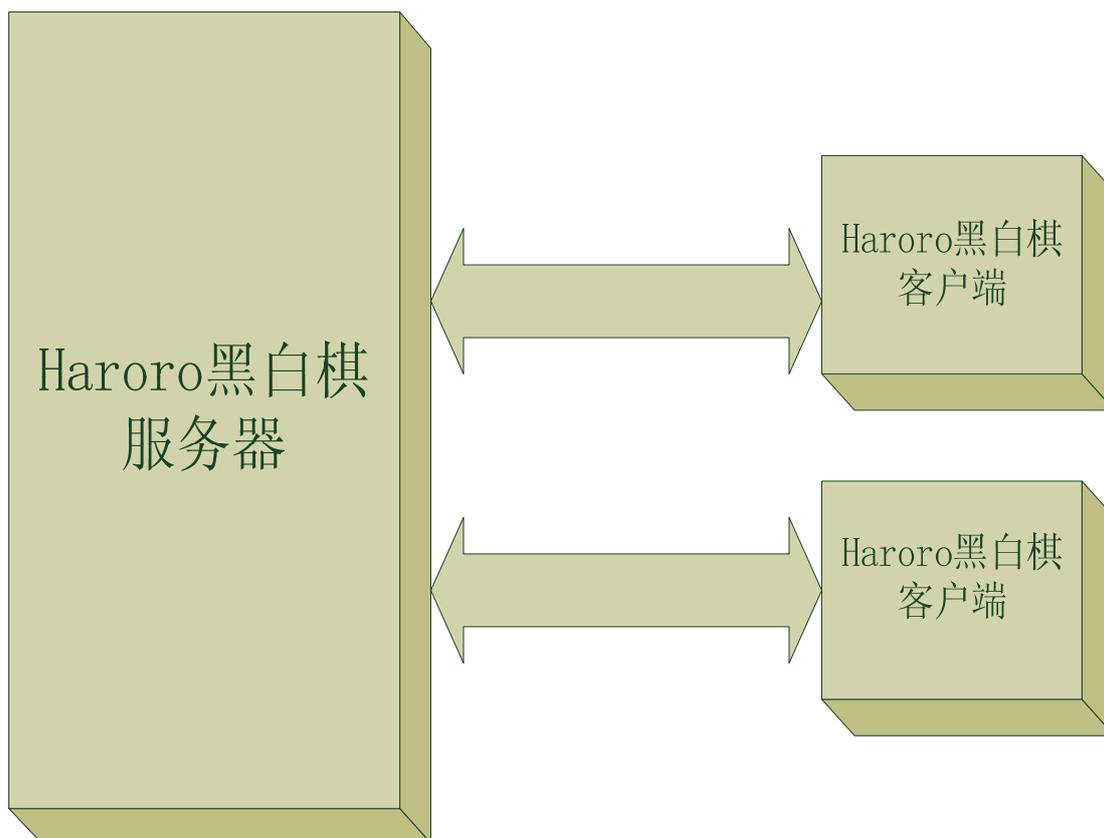


图 3-1 Haroro 黑白棋系统基本框架

由于有多个客户端与服务器连接，并且每个客户端都与服务器有信息的传递，加上网络传输数据的特点，Haroro 黑白棋系统将采用多线程机制对系统进行控制，以达到同步的效果，实现多个客户端与服务器信息的正确传递。

服务器开启后，将对服务器端某一指定端口进行监听，监听是否有客户端请求连接服务器。在服务器已开启的前提下，运行客户端后，要尝试连接服务器，从而进入游戏大厅。而每当一个客户端连接到服务器后，此客户端会建立一个线程来监听来自服务器端的信息，读取来自服务器端的信息，并进行处理后反馈相应信息。而在服务器端，服务器也会随时监听来自客户端的消息，每当接收到一

个新的连接时，都会建立一个新的线程来处理与客户端的通信，处理并传递落子、聊天等等一些信息。

本游戏的通信模式如下，Haroro 黑白棋系统采用了 Socket 套接字来处理数据交流与传输。服务器启动后，采用 ServerSocket 来监听某一指定的端口，Haroro 黑白棋系统里这个端口为 9001。当然，也可指定其他的端口作为监听端口，只有是可以随意使用的<sup>[10]</sup>。

开启客户端后，要进行游戏先得连接到服务器，客户端也将使用 Socket 来对服务器上的某一指定的端口发送连接请求，本程序指定的端口为 9001。连接成功后客户端即开始了与服务器的会话，会话将一直持续到客户端退出为止，那时将关闭 Socket。

在服务器方面，会创建一个 ServerSocket，通过运用 accept()方法来等候来自客户端的连接请求。当有客户端请求连接服务器时，accept()方法会返回一个相应的 Socket，然后用这个 Socket 建立一个新的线程，用于与那个连接到服务器的特定客户端进行通信服务。然后会无限循环调用 accept()方法来等待下一个新的连接请求。

本游戏拥有独立的服务器和客户端。为了使进行游戏的对战双方进行通信，客户端会首先将信息发送到服务器，然后服务器会根据接收到的信息的类型进行相应的处理，然后将处理后的信息返回到此客户端或者发送到另一客户端，以实现客户端与客户端，以及客户端与服务器之间的通信。

Haroro 黑白棋系统设计主要可分成三个部分，分别是服务器的设计、客户端的设计，以及棋盘面板的设计。

其中，服务器的功能是支持两个客户端的连接，并对各个客户端分别进行监控和信息传递，为客户端的信息传递提供通道，实现双人黑白棋联机对战。同时服务器会显示相应的客户端连接信息。

客户端的功能是实现与服务器的连接并实现网络通信。除此之外，还包括界面的布局、界面变色等等。客户端分为四个面板：棋盘面板、控制面板、玩家面板、聊天面板等等。其中，棋盘面板将单独进行重点设计。

控制面板包括如下部分：服务器的 IP 地址、进入游戏大厅按钮、建立棋局按钮、加入棋局按钮、放弃按钮、退出按钮。

玩家面板上会显示已进入游戏大厅的游戏玩家的名单，方便进行游戏相关操作和与指定游戏玩家进行聊天。

启动客户端后，玩家点击进入游戏大厅的按钮，以服务器已启动为前提，玩家将顺利进入游戏大厅，进入游戏大厅后，客户端会在玩家面板上和客户端的左上角显示玩家的相关信息。当玩家进入游戏大厅后，想要玩黑白棋游戏，可以自己建立一盘游戏，等待其他游戏玩家加入，当有其他游戏玩家加入自己建立的游

戏时，游戏将自动开始，建立游戏的一方执黑子，先下子，加入游戏的一方执白棋，后下子。进入游戏大厅后，玩家想要进行黑白棋游戏的另一个选择是加入其他游戏玩家建立的棋局中，加入由其他玩家建立的游戏后，游戏将自动开始，建立游戏的一方执黑子，先下子，加入游戏的一方执白棋，后下子。当一盘棋局成功开始时，改变棋盘和部分按钮的颜色，使游戏界面更加美观新颖。

聊天面板实现聊天功能，游戏玩家可以通过鼠标点击指定某一已进入游戏大厅的玩家进行聊天。棋盘面板主要包括棋盘的绘制和落子的绘制以及落子信息的传递。

## 3.2 界面设计

总的来说，对界面的要求是简洁明了，且操作方便，在此前提下，尽量提高游戏界面的美观程度。

下面将描述对界面的具体要求。

### 3.2.1 服务器端界面设计

首先，在服务器端，应有两个区域：

第一是显示当前连接到服务器的客户端的数量的区域；

第二是显示服务器信息和客户端连接服务器的信息。

服务器最初运行时的界面如图 3-2 所示。

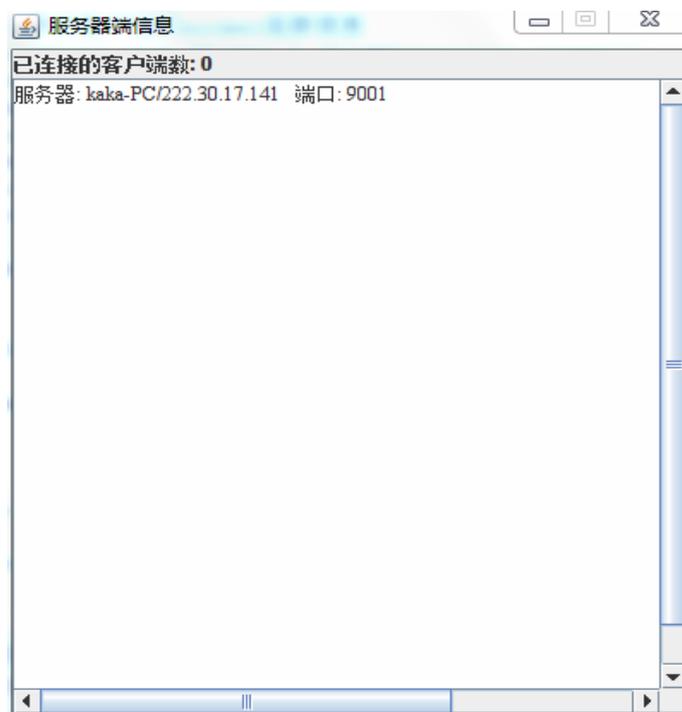


图 3-2 服务器启动后界面

注意到在服务器窗口的最上面部分是服务器的标题：“The Server Information”。而紧挨着标题下面的就是上面说到的第一个区域，这个区域用来显示当前已连接到服务器的客户端数量。下面的面板区域即上面说的第二个区域，显示服务器信息和来自客户端的连接信息。读者还可以看到服务器端窗口存在滚动条，这样的设计可以避免当信息长度超过了面板的范围时看不到完全信息的情况出现。

下面来看一下当有客户端连接到服务器后，出现的变化。

客户端连接到服务器后服务器端界面如图 3-3 所示。

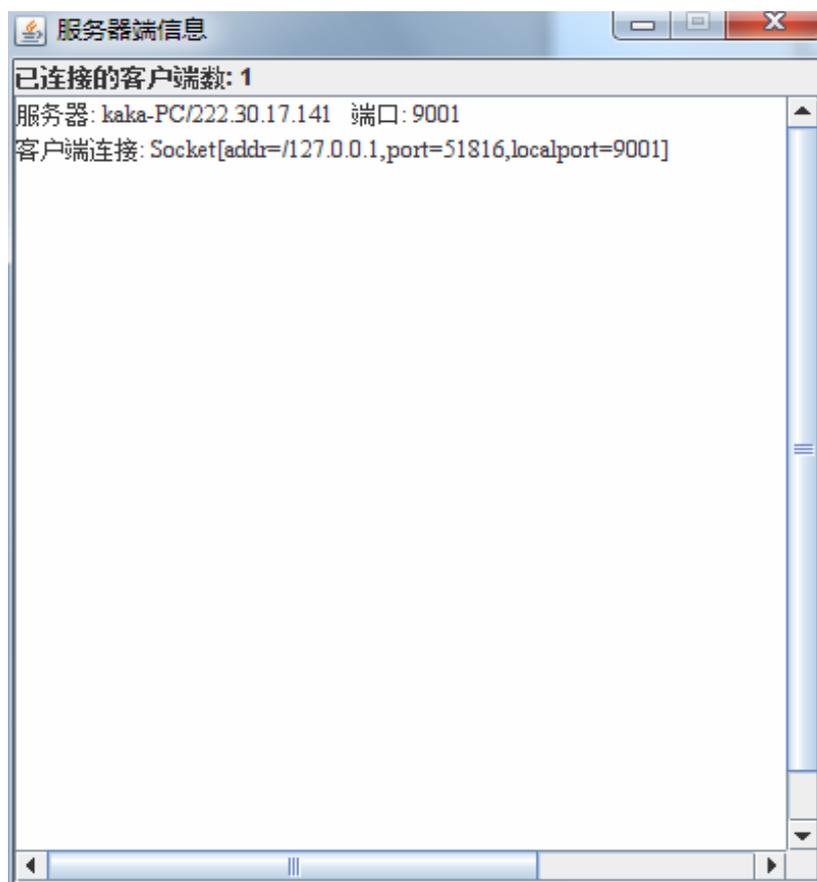


图 3-3 一个客户端连接到服务器

当有客户端连接到服务器后，当前已连接到服务器的客户端数量会显示为 1。并且显示出了客户端的连接信息。

而客户端关闭后，服务器端界面显示同样会发生变化。

当有客户端关闭时，服务器端界面如图 3-4 所示。

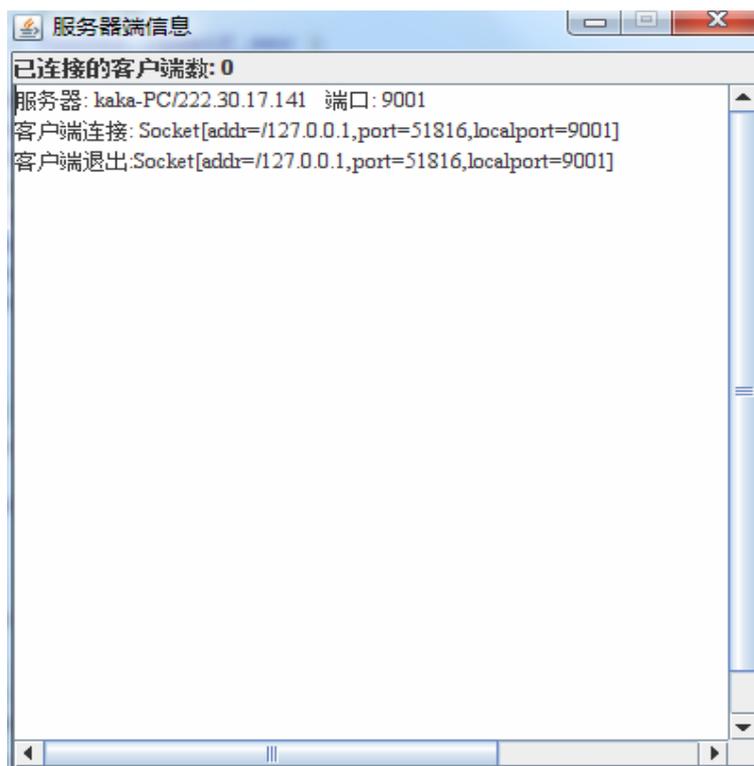


图 3-4 客户端退出

当有客户端关闭，即断开了与服务器的连接后，服务器端会显示出客户端关闭的消息。

### 3.2.2 客户端界面设计

其次，在客户端，这部分是界面设计的重点。客户端即玩家的游戏界面，此部分的界面的好坏将直接影响游戏玩家对本游戏的喜爱程度，因此，应设计出简洁清新的游戏界面，方便游戏玩家进行游戏操作。

游戏客户端界面应包括如下几方面：

①实现各种功能的按钮

包括：进入游戏大厅、建立游戏、加入游戏、放弃本盘游戏、离开等这些功能按钮。

②显示本方名字的区域，用于显示自己的名字。

③棋盘

这是最重要的界面部分，应做到简洁美观，最好能有色彩的变化功能，吸引游戏玩家。

④显示当前已进入黑白棋游戏大厅的全部游戏玩家的名字的区域，方便游戏玩家加入由其他游戏玩家建立的游戏以及聊天功能的顺利实现。

⑤服务器 IP 地址区域

游戏玩家在此处输入游戏服务器 IP 地址，以实现连接服务器。

⑥聊天区域

聊天区域包括用于显示聊天信息的显示区域，以及编辑聊天信息的聊天编辑框与发送聊天信息的按钮。

下面来通过界面图示图 3-5 来具体介绍：

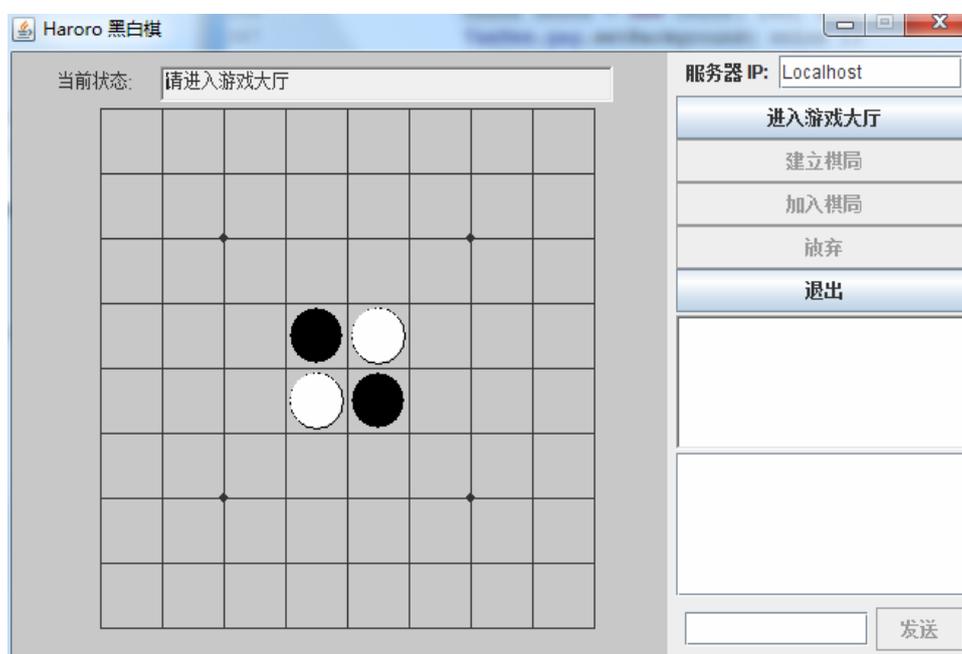


图 3-5 启动客户端后的最初界面

客户端启动后，可以看到，在客户端的左上角会显示此游戏的名称，即“Haroro 黑白棋”。而在名称下面是描述玩家当前状态的状态栏，在刚启动客户端是，状态栏上会显示“请进入游戏大厅”。状态栏下方则是棋盘面板，棋盘的初始状态是有四颗棋子放在棋盘中央，其中黑白双方各有两颗，且双方棋子交错摆放。在客户端右侧界面上，最上方是服务器 IP 栏，下面接着是五个功能按钮。这五个按钮依次是：“进入游戏大厅”按钮、“建立棋局”按钮、“加入棋局”按钮、“放弃”按钮、“退出”按钮。

在功能按钮面板下面是显示已进入游戏大厅的玩家信息的面板。在游戏玩家进入游戏大厅之后，系统会将其信息显示在此区域。

而在显示已进入游戏大厅的玩家信息的面板的下方是聊天信息面板和聊天发送面板。在棋局开始后，玩家可以选择其他某个游戏玩家作为进行聊天的对象。选定聊天对象后，在聊天输入框编辑信息，然后点击发送按钮即完成聊天。

游戏玩家须点击“进入游戏大厅”按钮，并在游戏服务器已启动的前提下，则可进入游戏大厅。

进入游戏大厅后的界面如图 3-6 所示：

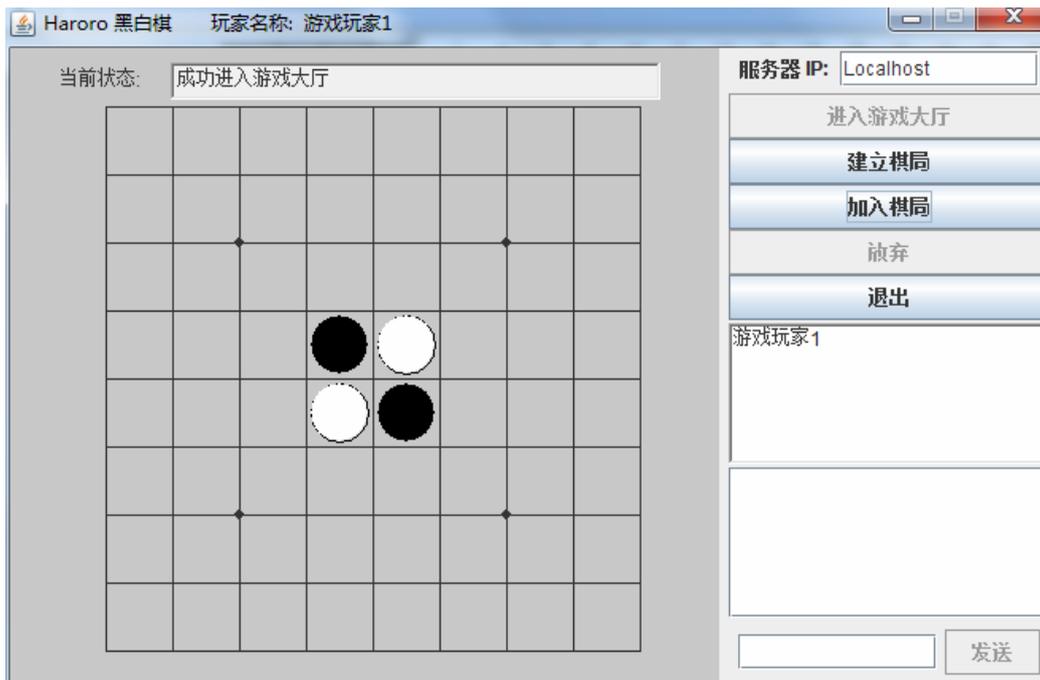


图 3-6 进入游戏大厅后的界面

从上图可以看到，在玩家进入游戏大厅后，在客户端左上角的游戏名称旁，将显示出玩家自己的名字。而状态栏的文本变成“成功进入游戏大厅”，让玩家了解当前自己的状态。在右侧，几个功能按钮的可点击状态将发生改变。之前的“进入游戏大厅”按钮将变为不可点击状态，而“建立棋局”按钮、“加入游戏”按钮的则转换成可点击状态。在功能按钮下方的显示已进入大厅的游戏玩家信息的面板上出现此玩家名字，即“游戏玩家 1”。

当某一游戏玩家建立一盘棋局，并且有另一游戏玩家加入了此棋局时，此局游戏将成功开始。同时游戏双方的客户端上的棋盘面板以及部分功能按钮的色彩将发生变化。并且游戏玩家可以选择对方进行聊天，聊天的信息的颜色也将是不同的。

游戏进行阶段界面如图 3-7、图 3-8 所示：

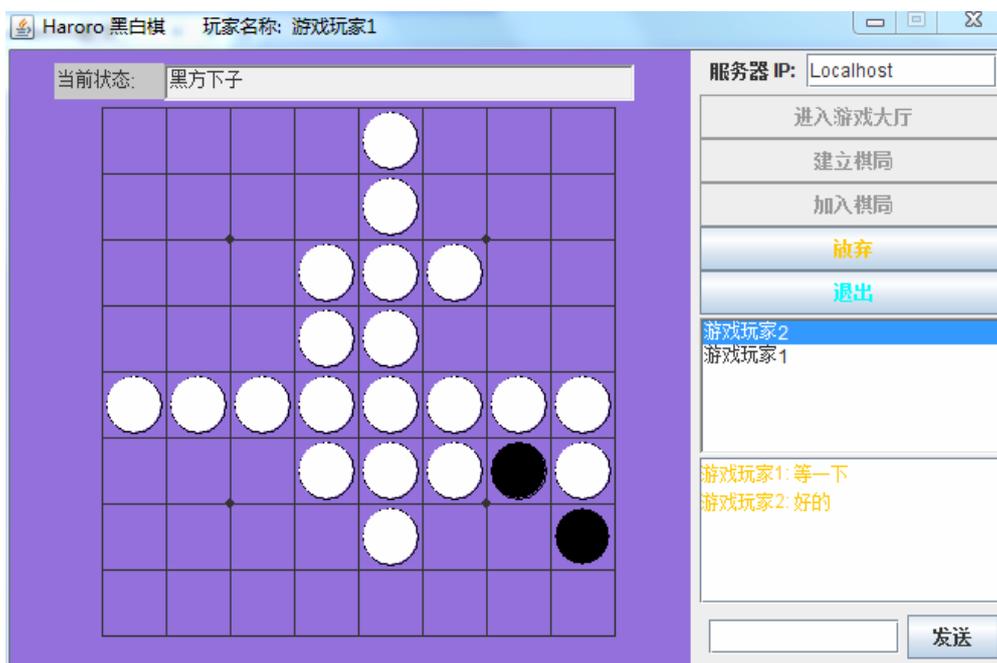


图 3-7 游戏进行中（黑方）

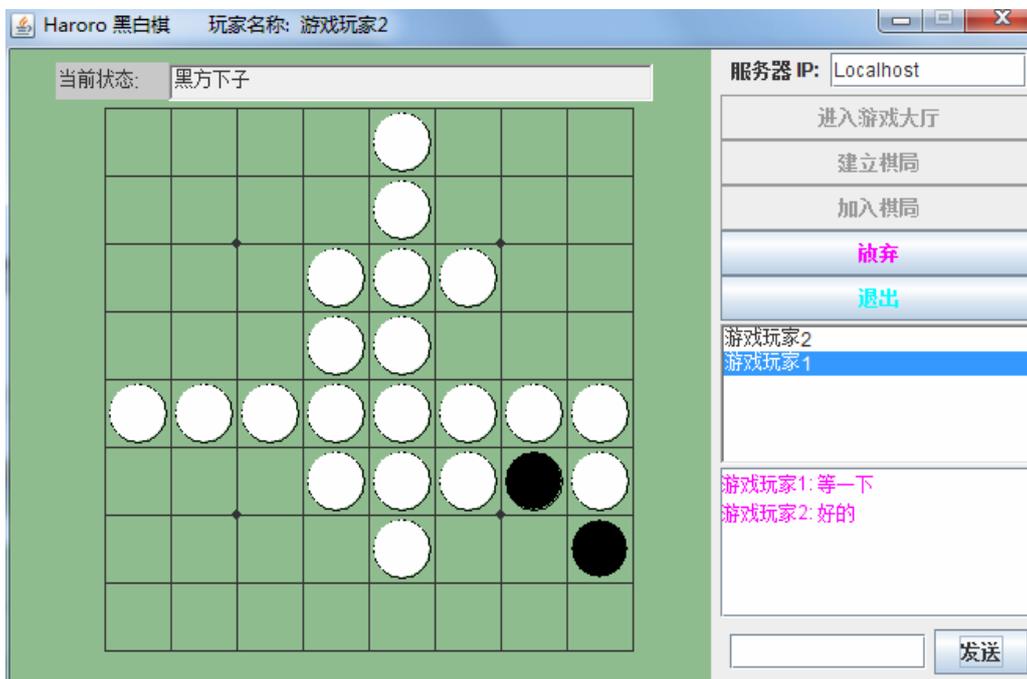


图 3-8 游戏进行中（白方）

由图示可看到，在游戏双方各自客户端的状态栏上都显示着下一步该谁落子。建立游戏的一方为黑方，执黑子。棋盘颜色为紫色，聊天信息的颜色为金黄色。而加入游戏的一方为白方，执白子。棋盘颜色为绿色，聊天信息的颜色为紫

红色。

而当一盘棋局分出胜负或者战成平局后，系统将自动重开一局。界面和前面所讲的一局棋局刚开始时是一样的。

## 第四章 HARORO 黑白棋系统的实现

本章将详细介绍此次基于 Java 的黑白棋游戏实现的细节。让读者能更好地理解此项目。

此次项目的开发实现可以分为如下三个部分：

- ①黑白棋服务器端实现；
- ②黑白棋客户端实现；
- ③黑白棋棋盘面板实现。

### 4.1 服务器端实现

首先，将介绍 Haroro 黑白棋系统的服务器端的设计情况。服务器端要能响应客户端的连接请求，且服务器端的界面面板上要显示出服务器信息和客户端的信息。

服务器面板上显示有用信息的两个地方：

```
JLabel NumOfClients = new JLabel( "已连接的客户端数量: " );
```

```
JTextArea Information = new JTextArea( "", 23, 60 );
```

NumOfClients 这个标签用来表示已进入游戏大厅的游戏玩家的总人数；Information 这个文本域用于显示服务器的信息和来自客户端的请求连接或关闭信息。

服务器运行后，要能监听来自客户端的请求连接服务器的信息。并且能够接收、处理并回馈从客户端发来的信息。

下面展示服务器端的详细设计情况。

服务器端的设计于实现含有三个类：

ServerInformaton 类、HeiBaiServer 类以及 JiYeonThread 类。

(1) ServerInformaton 类

ServerInformaton 类是一个 JPanel 的扩展类。作用是用于显示服务器信息和客户端的请求连接信息或关闭信息。

在上面已提到，ServerInformaton 类里设置了一个 JLabel 变量 NumOfClients 与一个 JTextArea 变量 Information。它们分别用来显示已进入游戏大厅的游戏玩家的总人数和服务器的信息和来自客户端的请求连接或关闭信息。

ServerInformaton 类中只有一个方法，即构造函数 ServerInformaton()。此方法首先设置了面板的布局格式，格式为 BorderLayout()。接着设置面板的大小以显示相关信息的字体格式，具体的实现语句为：

```
setSize( 350, 300 );
```

```
Information.setFont( new Font( "Serif", Font.PLAIN, 12 ) );
```

在这个显示相关信息的面板上，设置了滚动条，方便了信息的显示，成功避免信息过长超出面板显示范围。

设置滚动条的具体实现如下：

```
JScrollPane scrollPane = new JScrollPane( Information );
add( scrollPane, BorderLayout.CENTER );
```

## (2)HeiBaiServer类

HeiBaiServer类是一个JFrame的扩展类。它是服务器的框架。

HeiBaiServer类含有的成员变量包括：

### ①checkConnection

此变量为ServerSocket类型，用于监听和接收来自客户端的socket连接请求信息。

### ②IU

IU是ServerInformaton类的一个实例。

### ③clientsok\_name, clientsok\_exchangeinfor, HeiBai\_twosides

这三个变量均是Hashtable类型。其中，clientsok\_name是为套接字和相应的游戏玩家建立的映射，clientsok\_exchangeinfor是为套接字和用来发送消息的输出流建立哈希映射，而HeiBai\_twosides是为游戏双方建立哈希映射。

HeiBaiServer类含有三个方法，分别是：构造函数HeiBaiServer(), Respond( int )和main()。

其中，HeiBaiServer()会调用 Respond ( ) 方法。Respond ( ) 方法的作用是监听 9001 号端口，通过 Commnuni\_with\_Client = checkConnection.accept() 这句即是实现了监听的功能。当监听到有客户端请求连接时，生成一个用于与该客户端进行信息交流的 socket。并且将客户端的连接信息显示到信息面板上。接着，服务器会创建一个新的线程来处理以后与该客户端的信息沟通。同时，服务器会不断地监听来自客户端的连接请求，所以服务器监听运用了无限循环 while ( true ) {}。

## (3)JiYeonThread类

下面来详细介绍一下处理与客户端通信的线程类 JiYeonThread。此线程类的设计分为多个部分，其中最主要的是如何处理来自客户端的消息并做如何的反馈的部分。

当服务器接收到来自客户端的消息后，会调用方法

```
interpret_infor_andoperate( String infor_from_client ){}
```

来自客户端的消息分很多种，所以要对不同类型的信息做不同的处理。

当信息是以“&”开头时，则信息可能分如下几种：

①游戏玩家请求建立一盘游戏；

这种情况下，信息以"&setup [unavailable]"开头。

处理以"&setup [unavailable]"开头的信息的方法如图 4-1 所示。

```
if ( infor_from_client.startsWith( "&setup [unavailable]" ))
{
    String game_founder = infor_from_client.substring( 20 );
    synchronized ( clientsok_name )
    {
        clientsok_name.put(Commnuni_with_Client,
            infor_from_client.substring( 7 ));
    }
    synchronized ( HeiBai_twosides )
        HeiBai_twosides.put( game_founder, "vacant" );
    show_users_opponentinfor( game_founder, "&OK" );
    show_users( all_usenames());
}
```

图 4-1 处理以"&setup [unavailable]"开头的信息

其中，show\_users\_opponentinfor()方法的作用是向指定的游戏玩家所在客户端发送相关消息；show\_users( all\_usenames())的作用是在客户端显示所有已进入游戏大厅的玩家的姓名。

②游戏玩家请求加入一盘游戏；

此情况下的信息是以"&participate "开头的。

```
else if ( infor_from_client.startsWith( "&participate " ))
```

当某一游戏玩家想要加入到由另一游戏玩家建立的游戏时，如果另一游戏玩家存在且建立了一盘棋局并正在等待其他游戏玩家加入，则允许申请加入游戏的玩家加入，否则，回馈失败消息给申请加入游戏的玩家。

③游戏玩家的落子信息；

此情况下的信息是以"&[unavailable]"开头的。

```
else if ( infor_from_client.startsWith( "&[unavailable]" ))
```

如果传过来的消息是游戏玩家的落子信息，则提取相关信息并转发给游戏玩家的对手。

④游戏玩家的聊天信息；

此情况下的信息是以"&[unavailable]"开头的。

```
else if( infor_from_client.startsWith( "&Q" ))
```

如果传过来的消息是游戏玩家的聊天信息，则提取相关信息并转发给游戏玩家的对手。处理方面和上面的处理游戏玩家落子信息的方式很相近。

⑤游戏玩家放弃本盘游戏的信息。

此情况下的信息是以"&abstain "开头的。

```
else if ( infor_from_client.startsWith( "&abstain " ))
```

当接收到的信息是游戏玩家放弃此盘游戏时，做相应的清理工作。

此外，JiYeonThread 类还包括 send\_infor\_to\_client()、enter\_hall()等方法。其中，send\_infor\_to\_client()方法的作用是将信息发送给客户端，enter\_hall()方法是在当游戏玩家进入游戏大厅时调用的，目的是显示玩家自己的名字和已进入游戏大厅的玩家的信息。

## 4.2 客户端实现

其次，下面将介绍 Haroro 黑白棋系统的客户端的设计情况。同样，客户端设计的重点是信息的发送和对所接收到的信息的处理。

客户端的设计与实现包括两个类：HeiBaiClient 类和 HongSooA 类。

在客户端，主要通过监听鼠标的点击来生成所以传递的信息。这些操作主要由 HeiBaiClient 类涉及。

①如果点击“进入游戏大厅”按钮

```
if ( ChaeYoung.getSource() == buttonnel.enter_hall )
```

客户端会请求连接服务器，成功后改变部分按钮的可点击状态，并提示游戏玩家已进入游戏大厅。

②如果点击"建立棋局"按钮

```
if ( ChaeYoung.getSource() == buttonnel.setup )
```

当游戏玩家点击"建立棋局"按钮时，请求建立游戏，向服务器发送相关信息。

③如果点击"加入棋局"按钮

```
if ( ChaeYoung.getSource() == buttonnel.participation )
```

当游戏玩家想加入的游戏确实可加入时，将被允许加入，否则，无法加入。

④点击发送聊天信息的按钮

当点击聊天信息发送按钮时，处理方法如图 4-2 所示。

```
if( ChaeYoung.getSource() == nana.send )
{
    String chosenplaye = player_pan.players.getSelectedItem();
    if ( chosenplaye.equals( thisplayername )|| chosenplaye == null )
    {
    }
    else
    {
        String JuYeon = nana.write.getText();
        nana.messa.append( thisplayername + ": " + JuYeon + "\n" );
        nana.write.setText( "" );
        gag.VictoriaSong.traninfoerto_server( "&Q" +
        player_pan.players.getSelectedItem() + " "
        + "&S" + thisplayername + ": " + JuYeon );
    }
}
}
```

图 4-2 点击聊天信息发送按钮时系统的处理方法

当所选择的聊天对象符合要求时，即聊天对象存在，不为空，且不是玩家自己时，聊天可以顺利进行。

游戏玩家还可能点击功能为放弃游戏或离开的按钮，具体操作与上面所介绍的比较类似，故在此不再一一详细描述。

同样，客户端也会接收到很多来自服务器的信息，客户端将对这些信息进行处理。这些操作主要由 HongSooA 类涉及。

①接收到的信息以"&selfna"开头

客户端会在游戏面板的最上方显示出玩家自己的名字，方便游戏玩家进行聊天和其它操作。

②接收到的信息以"&allnames"开头

客户端将在显示所有游戏玩家名字的面板上显示所有游戏玩家名字。

③接收到的信息以"&opponen"开头

表示游戏成功开始，若本方为黑方，则先走棋；若为白方，则后走棋。

④接收到的信息以"\$"开头

表示此信息是聊天信息，将在聊天面板上将这些信息显示出来。

⑤接收到的信息是"&OK"

表示建立游戏成功。

⑥接收到的信息是"&yousucceed"

表示自己获得胜利。

⑦接收到的信息是"&dec"

表示自己加入由其他游戏玩家建立的游戏失败。

⑧如果接收到落子信息，会调用到 `gagBoard` 类里的方法，进行作图。

### 4.3 棋盘面板实现

最后，将介绍基于 Java 的黑白棋游戏的设计与实现的棋盘面板的设计情况。棋盘面板主要处理的是落子、棋子的绘制以及相应棋盘信息的处理。

首先进行棋盘的绘制，调用绘制棋盘的方法 `paint()`。一盘棋局开始后，游戏玩家即可执棋落子。当轮到自己落子时，监听鼠标点击，判断鼠标点击的位置是否可以落子，当判断可以落子后，绘制棋子。

判断是否可以落子的方法为：

```
boolean MC_onedirec( Point liu, int a, int e, int q )
```

上面的方法只是根据某一方向判断是否可以落子，程序实现时要判断 8 个方向，而不只是一个方向。

通过运用此方法，检测鼠标点击位置的八个方向上是否存在允许落子的情况。当判断鼠标点击的地方可以落子后，调用 `draw_self()` 方法绘制棋子并反转相应的棋子。然后把落子信息发到对方，让对方也在他的棋盘上绘制相应的棋子。

如下是根据对方落子信息绘制棋子的方法：

```
public void draw_stepofopponent( int i, int j, int colo )
```

而每次落子，都伴随着胜负的检查。判断游戏是否结束的方法为：

```
public void getsuccess()
```

若判断出没有分出胜负，则游戏继续；若胜负已分，则显示结果，并自动重新开始一局。

以上就是对基于 Java 的黑白棋游戏的设计与实现中主要的设计细节的详细描述，希望能使读者更好地理解此次游戏的设计。

## 第五章 总结与展望

### 5.1 总结

基于 Java 的黑白棋游戏的设计与实现通过运用 Java 语言最终实现了局域网内的黑白棋双人联机对战。

程序设计出了简洁漂亮的用户界面，在实现游戏基本功能的同时，还添加了棋盘变色这个特色和聊天功能。当一盘棋局正式开始时，棋盘界面会从原来的灰色变为蓝紫色或青绿色。色彩的改变起到的作用是吸引游戏玩家，避免对游戏界面的视觉疲劳。此外，辅助游戏的聊天功能增加了游戏的完善度，提供了游戏双方及时沟通的功能。

Haroro 黑白棋系统拥有独立的服务器和客户端，并且服务器可以接收多个用户同时连接。支持双人对战，游戏进行需要开启一个服务器和两个客户端。由于网络传输数据的特点，Haroro 黑白棋系统将采用多线程机制对系统进行控制，以达到同步的效果，为了使进行游戏的对战双方进行通信，客户端会首先将信息发送到服务器，然后服务器会根据接收到的信息的类型进行相应的处理，然后将处理后的信息返回到此客户端或者发送到另一客户端，以实现客户端与客户端，以及客户端与服务器之间的通信。

Haroro 黑白棋系统采用了 Socket 套接字来处理数据交流与传输。服务器启动后，采用 ServerSocket 来监听某一指定的端口。开启客户端后，要进行游戏先得连接到服务器，客户端也将使用 Socket 来对服务器上与前面说到的相同的端口发送连接请求。而每当一个客户端连接到服务器后，此客户端会建立一个线程来监听来自服务器端的信息，读取来自服务器端的信息，并进行处理后反馈相应信息。而在服务器端，服务器也会随时监听来自客户端的消息，每当接收到一个新的连接时，都会建立一个新的线程来处理与客户端的通信，处理并传递落子、聊天等等一些信息。

基于 Java 的黑白棋游戏的设计与实现运用到了很多 Java 方面的知识，包括进行图形用户界面设计所需的 Java GUI 类、事件驱动程序设计、异常处理、多线程编程、网络通信等等一些技术。

### 5.2 展望

通过不断的努力和改进，最终 Haroro 黑白棋系统成功完成。游戏可以正确的运行，拥有正确的算法，并且具有良好的用户界面，操作简便，实现了当初的计划和需求。

在 Haroro 黑白棋系统中，实现的是局域网内的双人联机对战，并且拥有棋盘变色和聊天功能。但系统还存在可改进的地方，今后计划添加一些诸如棋局记录、保存复盘等一些功能，并计划将对黑白棋的人工智能方面进行学习和研究，

争取能够找出较好的人工智能算法并将其运用于 Haroro 黑白棋系统。

## 参考文献

- [1]陈洁. 零基础学 Java(第 2 版)[M]. 北京: 机械工业出版社, 2010. 1.
- [2]高宏静等. Java 从入门到精通[M]. 北京: 化学工业出版社, 2009.
- [3]李刚. 疯狂 Java 突破程序员基本功的 16 课[M]. 北京: 邮电出版社, 2010.
- [4]李兴华. Java 开发实战经典[M]. 北京: 清华大学出版社, 2009.
- [5]刘亚滨, 杨红等. 精通 Eclipse[M]. 北京: 电子工业出版社, 2004.
- [6]埃史尔. Java 编程思想[M]. 陈昊鹏, 译. 北京: 机械工业出版社, 2007. 6.
- [7]孙一林, 彭波. Java 编程技术全接触[M]. 北京: 清华大学出版社, 2008. 6.
- [8]张鹏. J2ME 手机游开发教程[M]. 北京: 京华出版社, 2010. 3.
- [9]Dean Leffingwell, Don Widrig. 软件需求管理用例方法（第二版）[M]. 蒋慧, 译. 北京: 中国电力出版社, 2004.
- [10]James F.Kurose, Keith W.Ross. 计算机网络 自顶向下方法[M]. 陈鸣, 译. 北京: 机械工业出版社, 2009. 1.
- [11]Y.Daniel Liang. Java 语言程序设计 基础篇[M]. 万波等, 译. 北京: 机械工业出版社, 2008. 6.
- [12]Y.Daniel Liang. Java 语言程序设计 进阶篇[M]. 万波等, 译. 北京: 机械工业出版社, 2008. 6.
- [13]MJ Korman. *Playing Othello with Artificial Intelligence*[R].  
<http://mkorman.org/othello.pdf>. 2003.
- [14]Patrick Henry Winston. *Artificial Intelligence*[M]. Addison-Wesley, 1992.
- [15]Thomas Dean, James Allen, and Yiannis Aloimonos. *Artificial Intelligence: Theory and Practice*[M]. Addison-Wesley, 1995.

## 致 谢

此次论文的完成离不开老师和同学的关心和帮助。

首先要感谢张坤龙老师对我的指导，不仅仅是这次毕业设计的指导，同时还有平时学习生活的指导。张老师为我此次的论文写作提供了很多建议和帮助，让我获益匪浅。

同时，我要感谢那些帮助过我的同学，让我能更好地进行毕业设计和论文写作。

最后衷心感谢我的家人，谢谢每时每刻对我的支持和关心。