****

**毕 业 论 文**

**题目：基于JAVA的五子棋游戏的设计**

**学 院 : 网络继续教育学院**

**专 业 : 计算机科学与技术**

**届 别 : 2015 届**

**学 号 : 15203140164000**

**姓 名 : 卢小平**

**指导老师 : 杨正老师**

**基于JAVA的五子棋游戏的设计**

网络与继续教育学院 计算机科学与技术专业

(卢小平 15203140164000 指导教师：杨正)

【**摘要**】五子棋作为一个棋类竞技运动，在民间十分流行，为了熟悉五子棋规则及技巧，以及研究简单的人工智能，决定用Java开发五子棋游戏。主要完成了人机对战和玩家之间联网对战2个功能。网络连接部分为Socket编程应用，客户端和服务器端的交互用Class Message定义，有很好的可扩展性，客户端负责界面维护和收集用户输入的信息，及错误处理。服务器维护在线用户的基本信息和任意两个对战用户的棋盘信息，动态维护用户列表。在人机对弈中通过深度搜索和估值模块，来提高电脑棋手的智能。分析估值模块中的影响精准性的几个要素，以及提出若干提高精准性的办法，以及对它们搜索的节点数进行比较，在这些算法的基础上分析一些提高电脑AI方案，如递归算法、电脑学习等。算法的研究有助于理解程序结构，增强逻辑思维能力，在其他人工智能方面也有很大的参考作用。

【**关键词**】：深度搜索；估值；电脑AI；五子棋；算法

**Gobang Java-based games design**

【**Abstract**】As a sport,gobang is very popular in civil,in order to become familiar with gobang rules and techniques, and the study of simple artificial intelligence, I decide to use the Java to develope gobang games and complete the two functions including man-machine war and man-man war. Network Connection is Socket Programming for some applications, client and server interaction is definited by Class Message, which is a very good scalability, Client interface is responsible for the collection and maintenance of user input information, and error handling. Server users maintain online basic information and arbitrary two-time users of the chessboard of information, dynamic maintenance user list. During the man-machine players, it improves intelligence of the computer players through depth search and valuation module. Analyzes Module valuation of the precise elements, as well as a number of increased precision, and compares their search for nodes, which raises some computer AI programs on the basis of analysis, such as recursive algorithm, computer learning. Algorithm of procedures contribute to the understanding of the structure, logical thinking ability, In other areas of artificial intelligence has great references.

【**Keywords**】Search depth; Valuation; Computer AI; Gobang ; Algorithm

目录

[1 引言(Introduction) 1](#_Toc504337303)

[1.1 课题背景 1](#_Toc504337304)

[1.2 本课题研究的意义 1](#_Toc504337305)

[1.3 本课题的研究方法 2](#_Toc504337306)

[2 课题相关基础（Subject foundation） 2](#_Toc504337307)

[2.1 五子棋 2](#_Toc504337308)

[2.1.1 棋盘和棋子 2](#_Toc504337309)

[2.1.2 规则及解释 3](#_Toc504337310)

[2.1.3 五子棋常用术语 3](#_Toc504337311)

[2.1.4 五子棋攻防 6](#_Toc504337312)

[3 Java 7](#_Toc504337313)

[3.1 Java简介 7](#_Toc504337314)

[3.2 Java开发环境 9](#_Toc504337315)

[4 课题详细研究方法（Detailed research method of the subject） 10](#_Toc504337316)

[4.1 程序结构说明 10](#_Toc504337317)

[4.2 棋盘及棋子的类 11](#_Toc504337318)

[4.2.1 棋盘 11](#_Toc504337319)

[4.2.2 棋子 13](#_Toc504337320)

[4.3 胜负判断条件 13](#_Toc504337321)

[4.4 网络对战 15](#_Toc504337322)

[4.5 电脑AI 17](#_Toc504337323)

[5 结论（Conclusion） 21](#_Toc504337324)

[6 致谢（Thank） 22](#_Toc504337325)

# 引言(Introduction)

## 课题背景

五子棋是起源于中国古代的传统黑白棋种之一。现代五子棋日文称之为连珠，英译为Renju，英文称之为Gobang或FIR(Five in a Row 的缩写)，亦有连五子、五子连、串珠、五目、五目碰、五格等多种称谓。

五子棋起源于古代中国，发展于日本，风靡于欧洲。对于它与围棋的关系有两种说法，一说早于围棋，早在 “ 尧造围棋 ” 之前，民间就已有五子棋游戏；一说源于围棋，是围棋发展的一个分支。在中国的文化里，倍受人们的青睐。本世纪初五子棋传入欧洲并迅速风靡全欧。通过一系列的变化，使五子棋这一简单的游戏复杂化、规范化，而最终成为今天的职业连珠五子棋，同时也成为一种国际比赛棋。

Java语言是当今最为流行的程序设计语言之一 作为一门非常优秀和极为健壮的编程语言，它同时具有的面向对象，与平台无关，分布式应用，安全，稳定和多线程等优良的特征，使用Java语言，不仅可以开发出功能强大的大型应用程序，而且Java语言本身突出的跨平台的特性也使得它特别适合于Internet上的应用开发，可以这样说，Java的出现使得所开发的应用程序“一次编写，处处可用”的实现成为了可能。

## 本课题研究的意义

近来随着计算机的快速发展，各种各样的电脑游戏层出不穷，使得我们能有更多的娱乐项目，而棋类游戏能起到锻炼人的思维和修身养性的作用，而且棋类游戏水平颇高，大有与人脑分庭抗礼之势。其中战胜过国际象棋世界冠军-卡斯帕罗夫的“深蓝”便是最具说服力的代表；其它像围棋的“手淡”、象棋的“将族”等也以其优秀的人工智能深受棋迷喜爱。

越来越多的具有智能的机器进入了人类的生活，人工智能的重要性如今显而易见。自己对人工智能比较感兴趣，而五子棋游戏程序的开发实现这个课题，正好提供给我这样一个研究的机会，通过对人工智能中博弈方面的研究（人机对弈），让我在简单的人机对弈全局设计，以及具体到相关算法上有了深入的了解。人工智能属于计算机科学的领域，它以计算机技术为基础，近几十年来，它的理论和技术已经日益成熟，应用领域也正在不断扩大，显示出强大的生命力。人工智能大致可以分成几个学科，它们每一个都是独特的，但是它们常常又互相结合起来完成设计任务，这时，这些学科之间的差别就变的很模糊。人工智能在专家系统，自然语言理解，自动定理证明，自动程序设计，人工智能在机器人学、模式识别、物景分析、数据库的智能检索、机器下棋（实质上是博弈论问题）和家用电器智能化等领域都有广泛的应用。而这个课题就是和人工智能中的博弈论领域紧密相关的。

这个题目核心是人工智能和Socekt编程，。并且人工智能中的博弈部分，由于采用了大量的搜索算法，其中很多被利用到各方面。它的概念、方法和技术，正在各行各业广泛渗透。智能已经成为当今各种新产品、新装备的发展方向。所以，趁着这个机会，对人工智能中比较容易实现的人机博弈进行了解研究学习，也是很实用且很有必要的。

## 本课题的研究方法

在进行游戏设计之前，首先决定的第一个问题就是，使用什么开发环境来编写环境? 自己虽然比较熟悉Java语言，但是实际开发项目经验很少，所以决定用Jcreator,其拥有高亮语法编辑、使用向导以及完全定制的用户界面，最主要的是能够自动查找文件于 Main 方法或 Html 文件以支持Java 小应用程序，然后启动相应的工具。

其次确定整个程序的结构框架。由于Applet运行速度较慢，如果在加上算法搜索时间，显然不符合程序的设计要求，决定用Java应用程序开发.整个程序的功能实现流程是这样的：网络对战涉及算法较少，所以先实现网络部分，实现基本的棋盘和棋子的类，添加判断胜负条件，这部是基础，也是很重要的，电脑AI也在这些基础上添加上来的。这个题目的2个功能包括2个重要算法，电脑AI和胜负条件，胜负条件运算量不大，有固定的模式。难点是人工智能，可以这样说，人工智能的好坏决定了这个题目的完成深度。所以，大部份时间花在AI算法的研究和改进上，对于算法我掌握的不多，研究了一些国内的五子棋算法，参考了一些游戏设计算法，详细比较各种算法的优缺点，而且参考了现代五子棋比赛的各种规则和技巧，尽量联系实际，努力提高电脑AI。

# 课题相关基础（Subject foundation）

## 五子棋

### 棋盘和棋子

现代五子棋棋盘，经过国际棋联的多次修改，最终定为15 X 15路，即棋盘由横竖各15条平行线交叉组成，共有225个交叉点；棋盘上共有五个星位，中间的星位称为天元，周围四点为小星，与围棋盘略有不同。见图1

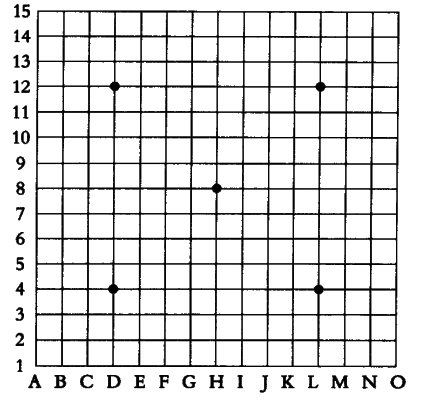


图1 棋盘

五子棋的棋子和围棋相同，分黑白2种颜色，通常为散圆形，有一面凸起或二面凸起等形状，一副棋子总数为225枚，其中黑子113枚，白子112枚。

关于计时，正规比赛按不同级别设置了不同的时间限制，一般的玩家都没有这个限制。

### 规则及解释

1：黑棋先行，白棋随后。从天元开始相互顺序落子。

2：最先在棋盘的横向、竖向、斜向形成连续的相同色五个棋子的一方为胜利。

3：黑棋禁手判负、白棋无禁手。黑棋禁手包括“三、三”；“四、四”；“长连”。黑方只能用“四、三”去取胜。

4：如分不出胜负，则定位平局。

5：对局中拔子、中途退场均判为负

6：五连与禁手同时形成，先五为胜。

7：黑方禁手形成时，白方应立即指出。若白方未发现或发现后不立即指正，反而继续落子，则不能判黑方负。

五子棋是由两个人在一盘棋上进行对抗的竞技运动。在对局开始时，先由执黑棋的一方将一枚棋子的落在“天元”上，然后由执白棋的一方在黑棋周围的交叉点上落子。如此轮流落子直到某一方首先在棋盘的直线 横线或斜线上形成连续的五子或五子以上，则该方就算获胜。但是五子棋的特点是先行的一方优势很大。因此，在职业比赛种对黑方做了种种限制，以利公平竞争。黑白双方的胜负结果必须按照职业五子棋的规则要求来决定。

### 五子棋常用术语

二：二是五子棋的一切进攻的基础；又分为活二和死二。

活二：即再下一子可形成活三的二，见图2。

死二：即再下一子可形成死三的二，见图3。

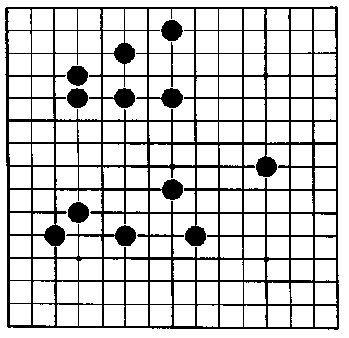
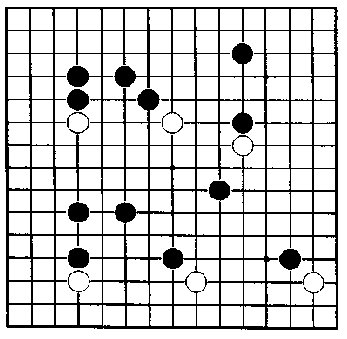


图2 活二 图3 死二

三：三是五子棋最常见的攻击手段。三分为活三、死三、眠三。

活三：为再下一子即可变成活四的三（活四介绍见后）。或三分为连三、跳三。见图4。

我们通常所说的“三”。就是指活三而言。如果是死三，将特别指出。

死三：即对方有棋子在同一条线相邻交叉点防守的三；死三分为三种。见图5。

眠三：是死三的一种特性，它看上去相邻点没有对方棋子防守，很像活三，但由于受空间限制，不论如何发展都不可能称为活四，故称为“眠三“。见图5。

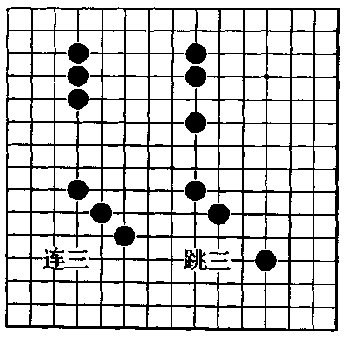
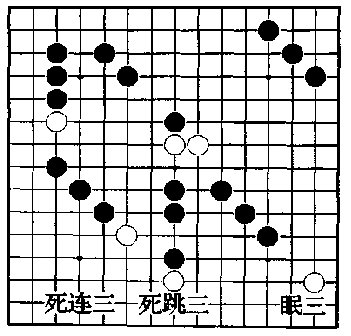


图4 活三 图5 死三

反三：反三不是指一种“三”的形状，而是指再阻止对方进攻的同时使自己称为一道活三的反攻，称为反三。

四：四的形状有三种，它分为：活四，冲四、跳冲四。

活四：在同一条线上相连的四个同色棋子成为“连五”，如图6所示。

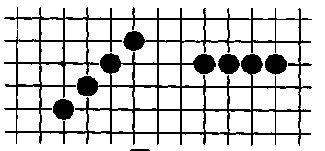


图6 活四

冲四：在同一条线上相连的相同颜色的四个棋子，它与活四不同的是，其中的一端有对方的棋子进行阻挡，它只有再在另一端下一手棋才可形成“连五”。

跳冲四：它的形状在“四”中是比较特殊的，一般分为两种，它的特点是同一条线上的同色的四个棋子，有一个或两个棋子与另外三个或两个棋子之间有一个交叉点的间隔，它的下一手棋只有下在间隔的交叉点上才能形成“连五”，因此，对方的棋子防守时，也只能阻挡在间隔的交叉点上，见图7。

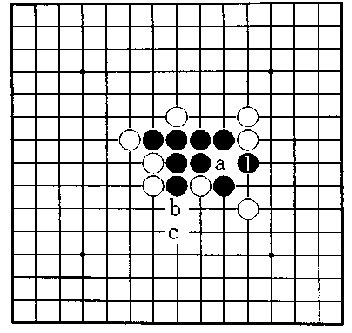
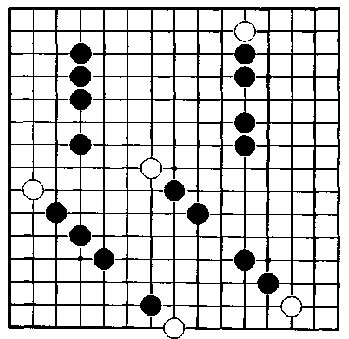


图7 跳冲四 图8 反冲四

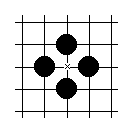
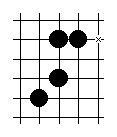
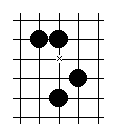
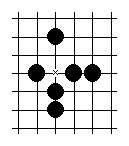
如图8，黑棋在1点进攻，形成一子双杀，这时白棋的妙手是在a位活三，它同时阻挡了黑棋的两种取胜方法。当黑棋仍在c位跳冲四活三时，白棋即在b位阻断黑棋冲四并形成反冲四，是黑棋的进攻功亏一篑。

一子双杀：指用同一手棋同时形成两个胜点的着法。

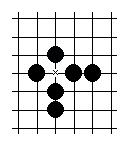
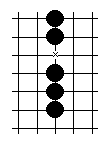
追下取胜：是指白棋逼迫黑棋形成禁手而取胜的方法。

自由获胜：除了追下取胜以外的获胜方法，称之为自由取胜。

禁手(Forbidden)：指黑方一子落下同时形成两个或两个以上的活三、冲四或形成长连的棋形，是对局中对黑棋禁止使用的战术或被判为负的行棋手段，见图15。白棋无禁手，如果黑子在落下的关键的第五子即形成五连的同时，又形成禁手。此时，因黑方已成连五，故禁手失效，黑方胜利。之所以这么规定也是为了规范比赛，其实我们业余的棋手，黑白棋子谁先下都不限制，只是先落子会有比较大的优势，另一方始终处于防守的被动状态。所以那一方先下子，他就有禁手，这是为了保持公平的一种手段。但是职业连珠五子棋虽然对黑棋采取了种限制，但是先行的一方优势依然很大。在高段位的比赛中还添加了一些规定，由于很少用到，就不必详细叙述了。

A B C D

E F

图9 禁手

**见图9。A、B、C 中的 x 点为三、三禁手。D、E 中的 X 点为四、四禁手。F中的 X 点为长连禁手。**

长连：相同颜色的连续六子或六子以上。

### 五子棋攻防

五子棋是一项对抗性很强的运动，在开局进入五手两打后，就开始进入了白刃战，相互争夺先手，任何一方都不能掉以轻心，要尽可能少犯错误，甚至不犯错误，否则将会导致速败。

众所周知，连珠五子，连五为胜。有四才能有五，有三才有四，以此类推。所以，在五子棋的对局众进攻和防守都是从“二”和“三”的争夺开始的。“好的开始是成功的一半”，这条格言用在五子棋里再合适不过了。

五子棋的点的选择十分关键。五子棋实际上是通过选择最佳的落点，加上正确的落子次序，一步一步地占领各个要点，最终获得胜利。

对局的早期，选点的着眼点主要是使本方的棋子保持联系，为以后创造尽可能多的成三、成四的机会，同时尽量限制对方成为好形。由于在五子棋对弈过程中，通过行棋落点来控制对方的落点是可能的，比如活三 冲四的应点是可以预知的，完全有可能通过不断走出这样的先手来控制对方的着点直至胜利。所以，在对局的后期，就要在精确计算的前提下，尽早发动攻势，以取得棋局的控制权，否则一旦贻误战机，被对方抢先发动攻势，就会成为被控制的一方。

当有多个攻击点可供选择时，要选择后续手段多，又不会被对方反先的着点。如果是黑方，还特别要注意进攻终被对方反击时出现的各种禁手点的可能性。

进攻分为单攻棋和双攻棋，单攻棋指单线即单行或单向的攻棋子，包括三子攻棋（活三、填四）和四子攻棋（冲四、填五）。而双攻棋指双向或双行的攻棋。包括三、三攻棋（双活三、双填四、填四活三）；四、三攻棋（冲四活三、冲四填四、填五活三、填五填四）；四四攻棋（双冲四、双填五、填五冲四）。

五子棋的取胜思路是由一个子开始，目标是运用各种方法在棋盘上发展出五连乃至长连而取胜。在这个发展过程中必然要经过由一子到两子 两子到三子 三子到四子的过程。因此说，把各种形状和各个方面上的二 三 四等子力结构烂熟于胸，做到举一反三，是学习五子棋的基本功，更是能灵活运用五种取胜技巧的基础。

# Java

## Java简介

Java是美国Sun公司开发的语言，它使用解释器执行代码，因此，无需对源代码进行任何更改即可在不同计算机上运行，是真正跨平台的编程开发语言。

Java有以下主要特点：

1：简单（Simple）

制定Java的原则之一，是要建立一种结构简单而且使用容易的系统，可以让用户不必接受很深的训练就可以开始设计程序，所以Java的语法尽可能与在当前许多程序设计师都采用的C及C＋＋语言相似。并且，Java删除了C及C++许多极少使用、不易理解或常被混淆的功能，多多重继承、指针等。

2：面向对象(Object-Oriented)

面向对象是近年来信息界广为使用的概念和技术。它有许多良好的特性。对象的封装性可以使对象的接口定义明确；继承性可以增加软件的可重用性，有助于分类及模版设计等。实现面向对象的概念及其各种良好的特性是Java的设计理念之一。

3：分布式（Distributed）

计算机网络的发展使得信息应用朝着分布式的环境发展，所以现代的信息开发语言及环境要有配合分布式的特性及功能。Java具有一个网络功能的程序库，其中包含与如HTTP和FTP等TCP/IP网络通信协议整合的能力。

4：强壮性(Robust)

由Java所编写的程序要能在各种情况下运行，而且必须具有高的稳定性。Java在制定时即加入了能防止存储器被覆写和数据损坏的相关处理机制。

5：安全性(Secure)

Java是被设计用于网络及分布式环境中的，所以安全性是一个很重要的考虑。Java拥有数个从简单到复杂的安全保护措施，能有效地防止病毒的侵入和破坏行为的发生。

6：结构中立性(Architecture Neutral)

在网络上存在许多不同类型的计算机，从中央处理器到操作系统的机构均有很高的差异性。因此要使应用程序在每一种机器上均能运行是相当困难的。针对这个目的，Java的编译器可以产生一种结构中立的目标码文件格式――字节码（Byte Code）。这种字节码可以在许多种不同的计算机上运行。

7：多线程(Multithreaded)

多线程是开发复杂和功能强大的程序所必须的手段之一，Java同样支持这个重要功能。

一个Java程序的开发过程如图10所示。



图10 Java程序的开发过程

源文件：使用一个文本编辑器，如Edit或记事本来编写源文件。不可使用Word编辑器，因为它含有不可见字符。将编好的源文件保存起来，源文件的扩展名必须是Java。

编译器：源文件要经过编译器（Javac.exe）的编译生成可扩展名为.Class的字节码文件。字节码文件是由与平台无关的二进制码组成的，执行时由解释器解释成本地机器码。

运行Java程序：Java程序分为两大类——Java应用程序（Application）和Java小应用程序（Applet）。Java应用程序必须通过Java解释器（java.exe）来解释执行其字节码文件；Java小应用程序可通过支持Java标准的浏览器来解释执行。

## Java开发环境

使用Jcreator开发环境必须安装JDK,我安装的版本是1.5.0，在环境变量里新建名为Path的变量名，变量值为D:\软件\JAVA\bin（根据JDK的安装目录而定）。在命令提示窗口输入Java -version如果能正确显示版本号，则表示环境配置成功。如图11所示。

****

图11 环境变量

Jcreator的安装很简单，这里不在详细叙述，把路径配置好就可以了。如图12。

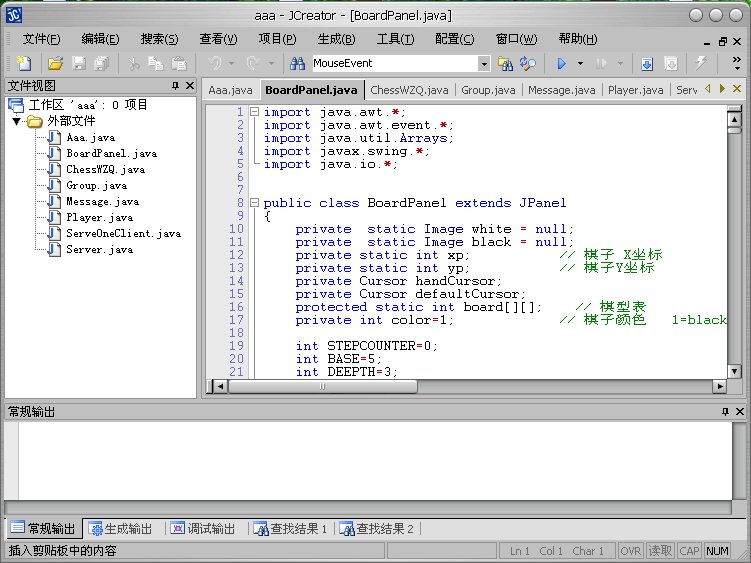


图12 Jcreator IDE

# 课题详细研究方法（Detailed research method of the subject）

## 程序结构说明

既然是Java Application,要实现网络对战，故采用C/S模式编写，程序包含7个独立的类文件－ChessWZQ.java、Group.java、Message.java、Player.java、ServerOneClient.java、Server.java、BoardPanel.java。

其中BoardPanel.java主要负责棋盘的初始化，鼠标事件的处理，以及判断胜负条件。ChessWZQ.java定义了面板上的其他元素，包括玩家列表，标题栏等。也包括了事件处理和人工智能。ServerOneClient.java则负责网络对战的处理。其他的类都是又这3个主类延伸出去的。在具体实现的时候再介绍其作用。见图13。



图13五子棋程序流程图

## 棋盘及棋子的类

### 棋盘

棋盘如图1，具体代码如下：

String line = "a b c d e f g h i j k l m n o";

char [] rowNum1 = {'1','2','3','4','5','6','7','8','9'};

char [] rowNum2={'1','0','1','1','1','2','1','3','1','4','1','5'};

这部分为棋盘的边界标识符，是必须要有的。用字符数组存储，最后确定位置放上去即可。

棋盘的绘制：

private static int xp; // 棋子 X坐标

private static int yp; // 棋子Y坐标

public void paint(Graphics gc){

super.paint(gc);

//this.setBackground(Color.gray);

//this.invalidate();

gc.setColor(Color.blue);

//gc.setColor(new Color(255, 255, 240));

//画横向标识符

gc.drawString(line,25,15);

// 画竖向标识符

for(int i=0;i<9;i++){

gc.drawChars(rowNum1,i,1,10,35+i\*30);

}

for(int i=9,j=0;i<15;i++,j+=2){

gc.drawChars(rowNum2,j,2,10,35+i\*30);

}

// 画棋盘

for (int i = 0; i < 15; i++) {

gc.drawLine(30, 30 + i \* 30, 450, 30 + i \* 30); //行

gc.drawLine(30 + i \* 30, 30, 30 + i \* 30, 450); //列

}

gc.drawLine(25, 25, 455, 25);

gc.drawLine(25, 25, 25, 455);

gc.drawLine(25, 455, 455, 455);

gc.drawLine(455, 25, 455, 455);

//面板初始化

for(int i=0;i<15;i++){

for (int j = 0; j < 15; j++) {

xp=16+i\*30;

yp=16+j\*30;

if (board[i][j] == 1){

gc.setColor(Color.black);

gc.fillOval(xp,yp,28,28);

//gc.drawImage(black, 16 + i \* 30, 16 + j \* 30, this);

}

if (board[i][j] == 2){

gc.setColor(Color.white);

gc.fillOval(xp,yp,28,28);

//gc.drawImage(white, 16 + i \* 30, 16 + j \* 30, this);

}

}

}

}

### 棋子

确定下子的坐标（xp,yp）画特定大小的椭圆，这里的坐标指棋子相对棋盘的绝对坐标。

private static int xp; // 棋子 X坐标

private static int yp; // 棋子Y坐标

if(col==1) g.setColor(Color.black);

else g.setColor(Color.white);

g.fillOval(xp,yp,10,28,28);

## 胜负判断条件

要判断四个方向，横向、竖向、以及2个斜向。思想还是比较简单，相同颜色连成五子即胜利，网上有些网友评论说完整的判断胜负条件包括连五和活四，我觉得完全没必要，活四还要检查两边的棋子，虽然运算量不大，但五子棋的标准就是连五即胜，一步之差，我们既要遵守规则，也要简化代码实现尽完整的功能。

我们要事先建立一个盘面数组board[ ][ ],即棋型表，数组的每一个元素对应棋盘上的一个交叉点，用‘0’表示空位，‘1’表示黑棋，‘2’表示白棋。由于代码太多，下面给出了一般状况的判断胜负函数,及以坐标（x,y）为中心的9X9矩形，只能在棋盘的内部，如果超过棋盘，就要另外考虑。下面的代码就是一般情况，整个矩形在棋盘内部的时候的判断胜负条件，如图14。

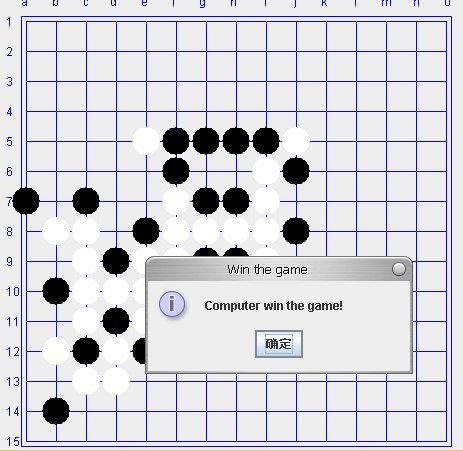


图14 获胜

胜负判断条件以下给出了X方向的代码：

protected boolean judge(int x,int y,int clr){

int i=0,j=0,count=0;

// x方向

for(i=0,count=0;x-i>=0&&i<5;i++){

if (clr==board[x-i][y]){

count++;

}

else{

break;

}

// System.out.println("( "+x+" , "+y+" )"+"count = "+count);

if(count==5)

return true;

}

for(i=1;x+i<15&&i<5;i++){

if(clr==board[x+i][y]){

count++;

}else{

break;

}

if(count==5)

return true;

}

return false;

}

为保证公平，先下子的就有禁手。但是我们一般没有这个规则限制，都是轮流先下子。理论上是这样的。但很多专家表明，先下子有很大的几率获胜，即使有禁手，先下子的一方还是有很大的优势，我觉得对于我们一般玩家而言，这些规定可以不考虑。

判断胜负的不管是单机还是玩家相互游戏，都必须开服务端，因为判断胜负是放在里面的。如果有一方获胜，弹出提示框，如果确认则清空棋盘继续新游戏。

public void getVictory(Message msg){

JOptionPane.showMessageDialog(null,

"You Win The Game",

"Congratulations",

JOptionPane.INFORMATION\_MESSAGE);

//继续新游戏

label3.setText("Player2");

newGame();

}

需要注意的一点是落下的棋子如果离任何一方的边界小于4，则以边界为限制判断是否有一方获胜，这样的话也要考虑多种方向，但原理还是和基本情况是一样的。

## 网络对战

这部分也属于网络套接字编程的经典应用，根据服务器地址连接特定端口。

网络部分很简单，但是我也做出了自己的特色，就是事件处理,做工可根据具体情况修改，虽然没有多少实用价值，但是也尽量使程序留有扩展性。下面给出了Message的代码。

/\*\*

\* 消息列表

\* type = 1 // 请求连接, msg = 连接者名字

\* type = 2 // 放棋子

\* type = 3 // 请求和其他人游戏

\* type = 4 // 拒绝游戏请求

\* type = 5 // 同意请求

\* type = 6 // 发送胜利消息

\* type = 7 // 断开连接请求

\* type = 8 // 保存游戏，但是不放在磁盘上，在下一局开始时将会丢失

\* type = 9 // 添加新的玩家到所有客户端的玩家列表

\* type = 10// 响应信息 type ==1

\* type = 11// 和玩家游戏,msg 保存玩家的名字 ,创建游戏的玩家设置游戏

\* type = 12//信息设置

\* type = 13// 设置玩家颜色

\* type = 14// msg＝ 接受请求的一方设置

\* type = 15// 服务端更新信息

\* type = 16// 发送控制或错误信息

\* type = 17// 游戏失败

\* type = 18// 服务端套接字关闭

\* type = 19// 玩家结束游戏及更新

\* tyep = 20// 电脑获胜/ 玩家获胜

\*/

SOCKET监听端口，如果有玩家则更新玩家列表，如果双击除自己以外的玩家，则可以开始游戏，发送游戏请求，如果同意则返回棋子颜色并开始游戏，同时清空玩家列表。每一下子，把坐标传给服务端，并在2端显示出来，并且判断胜负。如果，有一方获胜，则提示消息通知双方，确定则继续开始新游戏。

服务端开启服务监听线程和客户端，如图15，如果有玩家知道服务器IP地址，即可选择在线玩家进行联网游戏，如图16。



图23 服务端监听

图15 服务端监听

双击除自己以外的一个玩家，发出游戏请求，同时要确定自己棋子的颜色，用MSG发回服务端。如果被拒绝，则返回原来的状态。

if(msg.color==1){

ss = new String("white");

bpanel.setColor(2);

}

else{

ss = new String("black");

bpanel.setColor(1);

}

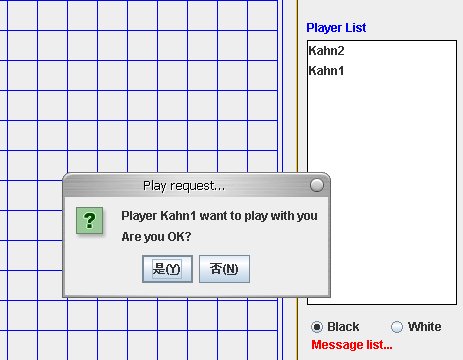


图16 邀请游戏

## 电脑AI

广义上来讲，博弈是指在一定的环境条件和一定的规则约束下，依靠自己所能够掌握的信息，从各自选择的行为或是策略进行选择并加以实施，并从各自取得相应结果或收益的过程。冯·诺伊曼（John von Neumann,1903-1957）和摩根斯坦恩（Oskar Margenstern, 1902-1977）在1944年出版了《博弈论与经济行为》（Theory of Games and Economic Behavior）一书中，最早地提出了关于博弈论的概念。但是，对于非合作、纯竞争型博弈，诺伊曼所解决的只有二人零和博弈。在这里所抽象化后的博弈问题是，已知参与者集合(两方)，策略集合(所有棋着)，和盈利集合(赢子输子)，最终是想去找到一个理论上的解或平衡，也就是对参与双方来说都最合理、最优的具体策略。

而在这里狭义的讲，博弈论主要是研究棋手们落子中理性化、逻辑化的部分，并将其系统化为一门科学。换言之，博弈就是研究个体如何在错综复杂的相互影响中得出最合理的策略，博弈论正是衍生于古老的游戏或曰博弈如象棋、扑克等。数学家们将具体的问题抽象化，通过建立自完备的逻辑框架、体系研究其规律及变化。

参考了很多五子棋算法，大部分思想差不多，就是搜索估值确定重要性，然后选取最大的一个点下子。具体做法如下：为电脑和玩家各建立一张表，用来存放棋型数据，比如“20”代表“冲四”的点，用“15”代表“活三”的点，那么在计算重要性时，就可以根据20>15得出前者比后者重要，下子时电脑便会自动选择“冲四”的点，这里还要说明一点的事，还要考虑四个方向。因为有可能有复合棋型，比如“四三”..从第一步起，不管是哪一方下子，电脑都有以这点为中心搜索9X9的矩阵内的所有空白点上棋子的重要性，一颗棋子对棋型影响的大小有9X9。重要看来虽然说进攻和防守的重要性一样的，但是我认为防守更重要。

在估值的时候，必须要考虑棋子的合法落子情况。不同的棋类博弈，其估值必定有极大的差别，各种因为规则而造成的不同因素影响估值的设计。不同的棋类游戏各有所谓的规则，规则中就有博弈双方都可以走哪些着法。某些博弈游戏很容易就找到合理着法，我所实现的五子棋，它就具有很简单的落子规则，即棋盘上所有的空位都可以落子，它们都是合理的着法。但是有些棋类游戏，比如在中国象棋和国际象棋中，情况就有些复杂了，每个棋子都有它特定的着法，

电脑下子要考虑自己和玩家的棋型，优先防守，如果没有要防的棋型，则搜索自己的棋型下子。

设置重要性,即估值模块，下面只给出横向的代码

private void setWeight(int x,int y,int tcolor){

int i=RectX1,j=RectY1,value=0,k=0,n=0,flag=0;

// '--' 方向

for(i=RectX1,j=y;i<=RectX2;i++){

if(BoardPanel.board[i][j]!=0){

continue;

}

value=0;flag=0;

for(k=1;i-k>=RectX1 && k<5;k++){

if(BoardPanel.board[i-k][j]==tcolor){

value++;

continue;

}

if(BoardPanel.board[i-k][j]==0){//black space

flag++;

break;

}

}

for(k=1;i+k<RectX2 && k<5;k++){

if(BoardPanel.board[i+k][j]==tcolor){

value++;

}

if(BoardPanel.board[i+k][j]==0){

flag++;

break;

}

}

n=weight(value,flag);

if(weightBoard[i][j]<n){

weightBoard[i][j]=n;

}

}

}

设定相应空位的重要值以后，选取最大的值下子，代码如下：

private void getBiggest(int [][] arr,int x,int y){

int [] temp=new int[2];

int swt=arr[0][0],tmp=0;

for(int i=0;i<15;i++){

for(int j=0;j<15;j++){

if(arr[i][j]>swt){

temp[0]=i;temp[1]=j;

swt=arr[i][j];

}

}

}

x=temp[0];

y=temp[1];

arr[x][y]=0;

}

对于特定的棋型，都有一个不同的估值，以此来区别不同棋型的优劣，也以此来决定最终的落子位置。毫无疑问，像已有四子连成一线且还可以继续落子的情况，明显要比只有三个子连成一线的情况要好，或者说优先级要更高，对弈双方对此种棋局，肯定都是把第一种情况放为首要分析的位置上。因此，要使棋手做出这种判断，就要把第一种情况的估值设置得高。

对不同的棋型设置重要值，比如：活四、死四、活三、死三、活二、死二。

同理，和判断胜负一样下子也要考虑边界特殊情况，矩形设置如下：

private void setRect(int x,int y){

if(x-4>0) RectX1=x-4;

else RectX1=0;

if(x+4>14) RectX2=14;

else RectX2=x+4;

if(y-4>0) RectY1=y-4;

else RectY1=0;

if(y+4>14) RectY2=14;

else RectY2=y+4;

if(RectX1>RectY1) RectX1 = x-(y-RectY1);

else RectY1 = y-(x-RectX1);

if(RectX2>RectY2) RectY2 = y+(RectX2-x);

else RectX2 = x+(RectY2-y);

}

努力提高电脑AI一直是五子棋游戏关键，第一步的人工智能只是估值和搜索算法的集合，要真正的提高电脑AI还有很多步，例如，我的电脑AI只是片面的分析了双方的器型，没有前瞻性。如果玩家多想几步，电脑就发现不了。不过即使没采用递归算法，要是让电脑先下子的话，你的大部分时间也是花在防守上，可能是我的棋力太菜了，我自己还没有下赢过电脑。

目前有2种方法提高电脑棋力：一是递归算法，二是增加细致的特定棋形的判断，鉴于自己对算法方面欠缺甚多，没有办法完成这个功能，实在是遗憾。我大体说一下这2种算法的思路：递归算法的意思可以说成“今后几步预测法”，首先让电脑分析一个可能的点，如果在这儿下子将会形成对手不得不防守的棋型（例如：“冲四”、“活三”）；那么下一步对手就会照你的思路来防守你，如此一来便完成了第一步的预测。这时候在调用盘面分析模块对预测后的棋型进行分析，如果出现了“四三”、“双三”或“双四”等制胜点，那么己方就可以获胜了。否则照同样的方法向下分析，就可以预测出多步。如果盘面上没有对手必须防的棋型，进攻不成的话就得考虑防守了，将自己和对手调换一下位置，然后用上面的方法来预测对手的棋重要防守和攻击都可以平衡，不过缺点是预测的算法量比较大。关于增加细致的特定棋形的判断，前面已经说过，以为不同的棋型，例如“死四”就有不同的几种情况，如果对每一种设置不同的重要值，也可以提高电脑AI,但是要设置合适的分数就要在实践中检验了，因为这个不是大小的问题，而是相差多少。正如前面所说，增加对细致棋型的判断也会提高电脑AI，虽然没有递归算法明显，但的确是一种途径，不过考虑的因素较多。

关于电脑学习，这听起来似乎是算法无法实现的功能。但是在对弈中却是非常有用的，但还只是些理论上的东西，比如棋局结束后，反向搜索，在自己的棋库中设置相关记忆。但我并不以为这是很好的方法。因为用这种方法，很有可能它没有找准原因，又或者进行学习的时候反而把劣等的学习进去；并且这种学习是很片面的，它只会认准一种极相似的情况(或者说一模一样的棋局情况)，而不会辨识出相似的棋局情况。

# 结论（Conclusion）

通过本次课题的研究，用JAVA实现了五子棋人工智能和网络游戏。知道了这个课题的关键是电脑AI算法，在研究和编程其间，有了很多新的想法，同时对JAVA套接字编程也有了更新的认识。对五子棋相关规则及技巧有了新的收获，最重要的是自己动手解决问题的能力得到了提高。

其中人工智能部分，由于采用了大量的搜索算法，其中很多被利用到各方面。它的概念、方法和技术，正在各行各业广泛渗透。智能已经成为当今各种新产品、新装备的发展方向。随着新的算法和理论的研究，人工智能必定会在人们生活中扮演重要的角色。

# 致谢（Thank）

这次论文的完成，不止是我自己的努力，同时也有老师的指导，同学的帮助，以及那些无私奉献的前辈，正所谓你知道的越多的时候你才发现你知道的越少，通过这次论文，我想我成长了很多，不只是磨练了我的知识厚度，也使我更加确定了我今后的目标：为今后的计算机事业奋斗。在此我要感谢我的指导老师——杨正老师，感谢您的指导，才让我有了今天这篇论文，您不仅是我的论文导师，也是我人生的导师，谢谢您！我还要感谢我的同学，虽然我未必记得住每分每秒，但是我记得每一个有你们的精彩瞬间，我相信通过大学的历练，我们都已经长大，变成一个有担当，有能力的新时代青年，感谢你们的陪伴，感谢有你们，这篇论文也有你们的功劳，我想毕业不是我们的相处的结束，它是我们更好相处的开头，祝福你们！我也要感谢父母，这是他们给我的，所有的一切；

通过这次毕业设计，我学习了很多新知识，也对很多以前的东西有了更深的记忆与理解。漫漫求学路，过程很快乐。我要感谢学院的老师，我从他们那里学到了许多珍贵的知识和做人处事的道理，以及科学严谨的学术态度，令我受益良多。同时还要感谢学院给了我一个可以认真学习，天天向上的学习环境和机会。

即将结束福师大的学习生活，我感谢福师大提供了一次接受教育的机会，感谢院校老师的无私教导。感谢各位老师审阅我的论文。