

**软件工程pk项目**

**详细设计说明书**

院 系： 软件学院

组 员： 尤煜

编 制 人： 尤煜

编制日期： 2015-5-25

审 核 人： 尤煜

**2015年5月**

[1．引言 2](#_Toc12409)

[1.1编写目的 2](#_Toc21308)

[1.2项目背景 2](#_Toc9214)

[1.3定义 3](#_Toc15944)

[1.4参考资料 3](#_Toc25255)

[2．总体设计 3](#_Toc4360)

[2.1需求概述 3](#_Toc5084)

[2.2软件结构 4](#_Toc41)

[3．程序描述 4](#_Toc16601)

[3.6程序逻辑 8](#_Toc562)

[3.8存储分配 12](#_Toc14168)

[3.9限制条件 12](#_Toc25090)

[3.10测试要点 12](#_Toc8866)

五、详细设计说明书

# 1．引言

## 1.1编写目的

中国象棋系统将计算机知识和中国象棋知识结合起来的一种新型的游戏方式。本次实习的中国象棋系统是在传统人人对弈的基础上，实现人与机器对弈，突破了以往传统象棋游戏只能人与人对战的限制，使中国象棋这一古老的游戏形式焕发出蓬勃朝气。

本文结合在中国象棋机器博弈方面的实践经验，在分析了中国象棋游戏需求基础上，设计并实现了中国象棋系统。该系统包括人人对战、人机对战等功能模块。人人对战规则明确，包含了中国象棋所有的招法；人机对战中电脑棋力，便了不同水平人群的不同选择。

本系统的实现满足了人们对中国象棋的基本需求，解决了传统象棋游戏学习性差、不易演示等问题。

## 1.2项目背景

在人类文明发展的初期，人们便开始进行棋类博弈的游戏了。近几十年来，随着计算机硬件和软件技术的不断发展，人们开始对计算机能否战胜人脑这个话题产生了浓厚的兴趣。从1980开始，电脑博弈便开始逐渐大规模地向人类智能发起了挑战，到了1997年，IBM超级电脑Deeper Blue 击败了当时的国际象棋冠军卡斯帕罗夫，成为了人工智能挑战人类智能发展的一个重要里程碑。

而当今对中国象棋的研究也正如专家们所期望的那样在蓬勃地发展着。中国象棋不仅是中国传统智慧的体现，同时也具有着比国际象棋更高的复杂度，如何实现网络对是本课题研究的一个重要问题。通过本课题的研究,掌握C/S网络模式并对所学知识熟悉锻炼。

## 1.3定义

将军：对局中一方的棋子要在下一着棋将对方的帅（将）吃掉，称为"将军"。

将死：如果被"将军"而无法"应将"，就算"将死"。

河界线：构成河界的两条横线。

对局：双方下棋称"对局"，奕叫"对弈"

## 1.4参考资料

1.中国象棋游戏规则。参考文档：

<http://wenku.baidu.com/link?url=mw0GTIqFoBA1TBTCB1pvM9M0OjfOIigsMeQ3FebwXNn8Dh0DwUySCV5QbwtR7cmUP-DTCxxtyhr76YFbsQlFkCNPHAJ6ioJBilqsjECCSrG>

2阿尔法贝塔搜索算法。参考文档：

<http://www.xqbase.com/computer/search_alphabeta.htm>

3迭代加深算法。参考文档：

<http://www.xqbase.com/computer/search_iterative.htm>

4需求规格说明书；

5概要设计说明书；

# 2．总体设计

## 2.1需求概述

**棋盘模块的算法设计；**

**棋子模块绘制算法设计；**

**棋种走棋规则模块算法设计；**

**声音控制模块算法设计；**

**AI模块算法设计;**

## 2.2软件结构

用户选择界面

人机对战

人人对战

# 3．程序描述

【逐个模块给出以下的说明：】

1. **棋盘模块的算法设计**

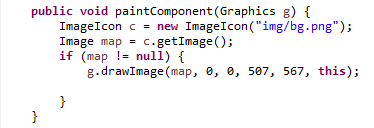
该部分讲述关于中国象棋棋盘采用怎样的方法实现的，以及如何判定棋盘交界处有棋子。

1.



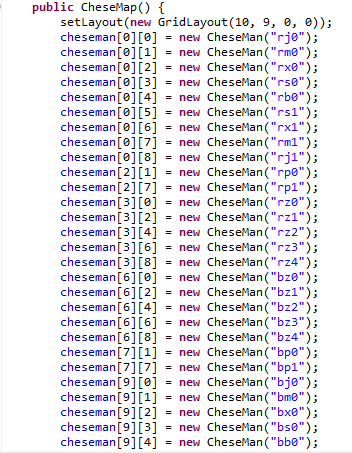
观察中国象棋的布局，可以发现这是一个10x9的棋盘，所以我们可以构造一个二维数组cheseman[10][9]表示棋子,若其中的值为空值，则说明该交界处没有棋子。

2.



本中国象棋有限的棋盘和棋子采用的都是图片，采用图片比较简单，不用绘制那么多条线，用图片来代替棋盘和棋子，只要用Graphics对象g调用drawImage()方法就可以把图片加载进去。如上图所示。

3.



对弈开始时，布置棋盘(将各种棋子”摆”在上面)，实例化棋子对象。

1. **棋子模块绘制算法设计**

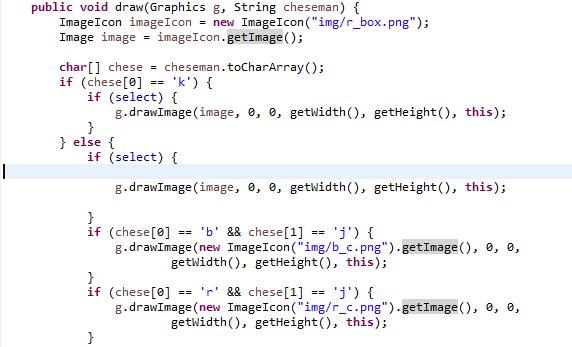


首先无论哪一种棋子我们都让他继承JButton，这样的话做成按钮会比较方便。



这里构造一个paint的函数，只要用Graphics对象g调用drawImage()方法就可以把图片加载进去。

**3.**

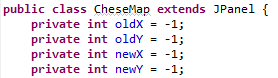


这一部分是将图片绘制在按钮上。Char[]chese = cheseman.toCharArray()这一句是将cheseman[][]转化为字符串存储在一维数组中。然后根据该数组中存储的值作为条件语句的判定，如果为真，调用drawImage()方法就可以把图片加载进去。

1. **棋种走棋规则模块算法设计**

象棋的每种棋种都有它们自己的走棋规则的，例如，车走直线，中间不能有棋子；马走“日”；象走“田”等等。下面我们就拿象棋棋子车，马，炮，象、士、卒、将的走棋规则算法分别来进行详细的说明：

**首先在程序里声明几个重要的变量，如下图：**



这几个变量分别代表上一步棋子的落点坐标和下一步棋子的落点坐标，分别初始化为-1。

**然后我们再来熟悉一下中国象棋棋种的移动和吃子规则算法，如下**：

1卒的移动及吃棋规则

 移动规则：根据本系统的详细情况来说，黑卒向上移动，红兵向下移动，并且不能回头，但可以左右移动。通过相关算法得到合法坐标，执行移动、吃棋和过河等操作。

吃棋规则：不管是黑卒还是红兵都可以吃掉正前方和左右方向上的任何棋子，但不能隔子吃棋。

2炮和车的移动及吃棋规则

走棋规则：首先判断起点和终点是否有棋子，炮和车只能上下和左右移动并吃棋，

起初要指定所有模糊的Y（X）坐标，移动的Y（X）坐标是否有指定坐标相近的所有棋子，找出在同一条竖线（横线）的所有棋子，并不包括自己，从起点到终点判断中间是否有棋子，如果中间有棋子就不可以从这条竖线（横线）过去，从起点到终点找起点和终点的棋子，起点和终点没有旗子就可以移动了。

吃棋规则：找出起点和终点是否有棋子，找出所有棋子，找出同一条线上的棋子，不包括自己，自己是七点被吃的终点。起点和终点没有棋子是车的吃棋方法，不能吃己方棋子，起点和终点有一个棋子是炮的吃棋方法，也不能吃己方棋子。

3马的移动和吃棋规则

走棋规则：新建集合用来保存坐标和障碍，找到合法的Ｘ和Ｙ坐标，判断正前方、正后方、正左方、正右方是否有别的棋子，判断移动该棋子。来实现马飞日的走法。

吃棋规则：也要有集合存放障碍棋子，首先判断正方是否棋子，若有则不能吃正方前方的两个棋子，若没有则可以移动到指定位置及吃棋，不能吃自己的棋子，吃棋后，停止闪烁，不显示被吃棋子。

4相（象）的移动和吃棋规则

 走棋规则：新建集合用来保存坐标和障碍棋子，用相关算法实现飞田算法，并且

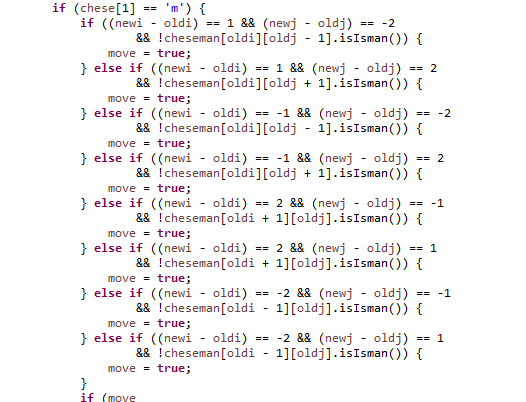
红（黑）棋不能过楚河汉界。

吃棋规则：吃棋时也是按照移动规则，进行判断吃子，判断被吃的棋子是否与士（仕）相近，如果相近则吃掉，且不能吃掉己方棋子。

5将（帅）的移动和吃棋规则

帅和将是棋中的首脑，是双方竭力争夺的目标。它只能在‘九宫‘之内活动，可上可下，可左可右，每次走动只能按竖线或横线走动一格。帅与将不能在同一直线上直接对面，否则走方判负。

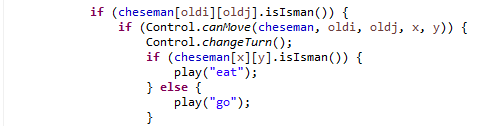
针对以上的各种情况，决定采用条件判断来规范棋子的行为，这一部分在controll实现的，举例说明马的走法，马走日：



如果上一步和下一步的横坐标相差一，纵坐标相差二或者横坐标相差二，纵坐标相差一，则可以移动。其中各棋子行为的具体规范请参考control.java，代码量过多，不宜粘贴。

1. **声音控制模块算法设计**

这一部分主要控制移动棋子时的声音，吃子时的声音以及胜利时的声音



判定是否有吃子现象，如果有，则play(“eat”)；否则play(“go”)。

1. **AI模块算法设计**

阿尔法贝塔搜索算法。参考文档：

<http://www.xqbase.com/computer/search_alphabeta.htm>

　　 Alpha-Beta 同“[最小-最大](http://www.xqbase.com/computer/search_minimax.htm)”非常相似，事实上只多了一条额外的语句。最小最大运行时要检查整个博弈树，然后尽可能选择最好的线路。这是非常好理解的，但效率非常低。每次搜索更深一层时，树的大小就呈指数式增长。

迭代加深算法。参考文档：

<http://www.xqbase.com/computer/search_iterative.htm>

有一个思想，就是一开始只搜索一层，如果搜索的时间比分配的时间少，那么搜索两层，然后再搜索三层，等等，直到你用完时间为止。这足以保证很好地运用时间了。如果你可以很快搜索到一个深度，那么你在接下来的时间可以搜索得更深，或许你可以完成。如果局面比你想象的复杂，那么你不必搜索得太深，但是至少有合理的着法可以走了，因为你不太可能连1层搜索也完不成。

　　这个思想称为“迭代加深”(Iterative Deepening)，因为你在迭代搜索，每次都比一次前一次加深1层(多1层没有什么奥妙的，当然你可以试试多两层，但是1层比较好)。

　　代码如下：

for (depth = 1; ; depth ++) {

　val = AlphaBeta(depth, -INFINITY, INFINITY);

　if (TimedOut()) {

　　break;

　}

}

这里面首先要为各种棋子赋值，因为计算机无法知道哪一种棋子优先级高，当一个棋子发现可以同时吃两个棋子的时候，计算机只能根据棋子的value来判断,比如帅或者将的value设为无穷，这样就优先“将军”

## 3.6程序逻辑

【详细描述模块实现的算法，可采用：

核心模块流程图：



“车”的走棋规则算法设计流程图



“马”的走棋规则算法设计流程图



“卒”的走棋规则算法设计流程图



“将”的走棋规则算法设计流程图



“炮”的走棋规则算法设计流程图



“象”的走棋规则算法设计流程图



“士”的走棋规则算法设计流程图

## 3.8存储分配

系统自动分配内存

## 3.9限制条件

无限制

## 3.10测试要点

1. 测试选择模式模块是否能正常运行。

2. 测试每一个棋子的行为是否规范；

3.测试AI的智能度；

4.测试声音模块是否能正常运行；