

课程设计报告



设计题目： 五子棋对弈——概要设计

学 院： 人工智能学院

专 业： 智能科学与技术

班 级： 1602052

学号姓名： 16020520006-王梓豪

16020520022-张亦弛

16020520024-王云峰

日 期： 2018 年 12 月 23 日

成 绩：

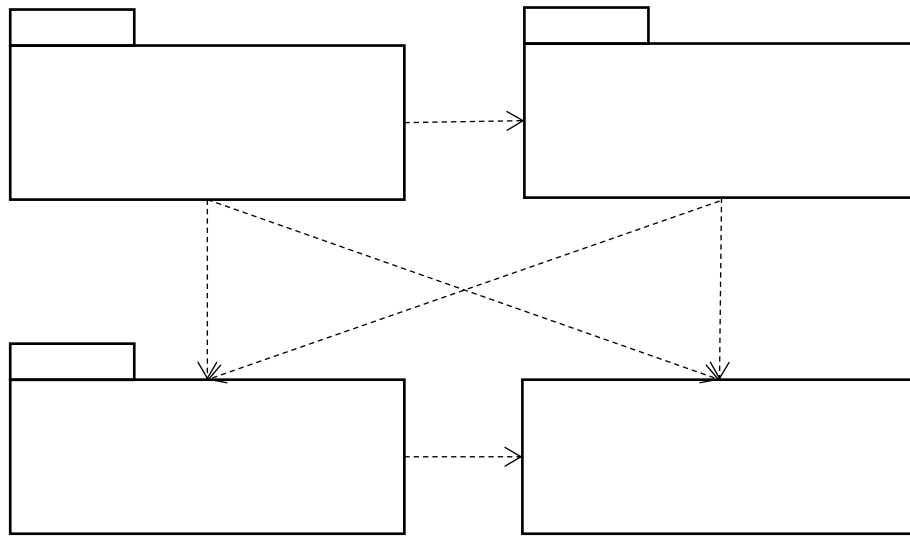
指导教师： 林杰

西安电子科技大学

目 录

1 模块结构图	2
2 人机界面	2
2.1 功能说明	2
2.2 输入接口设计	2
2.3 输出接口设计	3
3 用户包	5
3.1 功能说明	5
3.2 输入接口设计	5
3.3 输出接口设计	5
4 对战包	5
4.1 功能说明	5
4.2 输入接口设计	6
4.3 输出接口设计	6
5 存储包	7
5.1 功能说明	7
5.2 输入接口设计	7
5.3 输出接口设计	7
6 数据结构设计	7
6.1 估值核心数据结构和程序的关系	7
6.2 搜索引擎数据结构和程序的关系	8

1.模块结构图



2.人机界面

2.1 功能说明

负责响应用户在 PC 端的输入输出选择,展现用户使用界面,具体功能如下:

- (1) 用户人机交互平台。
- (2) 提供菜单栏选项,包括游戏模式选择选项、用户操作选项以及用户帮助。
- (3) 显示棋局。
- (4) 显示系统提示。

2.2 输入接口设计

(1) 人人博弈选项

①输入条件

用户选择人人博弈模式。

②传递消息定义

用户单击窗口按钮。

③传输方式

函数调用读取。

(2) 人机对战选项

①输入条件

用户选择人机对战模式。

②传递消息定义

用户单击窗口按钮。

③传输方式

函数调用读取。

(3) 悔棋选项

①输入条件

用户选择悔棋。

②传递消息定义

用户单击窗口按钮。

③传输方式

函数调用读取。

(4) 重新开始选项

①输入条件

用户选择重新开始。

②传递消息定义

用户单击窗口按钮。

③传输方式

函数调用读取。

(5) 认输选项

①输入条件

用户选择认输。

②传递消息定义

用户单击窗口按钮。

③传输方式

函数调用读取。

2.3 输出接口设计

(1) 棋盘绘制

①输出条件

用户打开程序。

②传递消息定义

绘制棋盘。

③传输方式

函数调用读取。

(2) 显示提示对话框

①输出条件

当用户事件发生时显示。

②传递消息定义

文字信息提示。

③传输方式

函数调用读取。

(3) 画黑棋

①输出条件

黑棋方落子。

②传递消息定义

绘制黑棋。

③传输方式

函数调用读取。

(4) 画白棋

①输出条件

白棋方落子。

②传递消息定义

绘制白棋。

③传输方式

函数调用读取。

3.用户包

3.1 功能说明

负责用户管理，用户战绩数据查询，用户信息显示，具体功能如下：

（1）记录与用户相关所有信息。

3.2 输入接口设计

（1）获得用户信息

①输入条件

用户开始连接。

②传递消息定义

管理用户信息数据文件。

③传输方式

函数调用读取。

3.3 输出接口设计

（1）显示用户信息

①输出条件

用户选择显示信息。

②传递消息定义

读取用户信息数据文件。

③传输方式

函数调用读取。

4.对战包

4.1 功能说明

负责用户选择对战模式，判断用户落子合理性以及判断胜负，电脑智能对战等；该模块主要提供人人博弈和人机对战功能，具体功能如下：

（1）实现人人博弈。

（2）实现人机对战。

（3）胜负判断。

4.2 输入接口设计

(1) 开始下棋

①输入条件

用户选择一种对战模式，开始下棋。

②传递消息定义

选项匹配。

③传输方式

驱动程序系统调用。

(2) 悔棋

①输入条件

用户选择悔棋。

②传递消息定义

选项匹配，并撤销最近两步棋子落子位置。

③传输方式

函数调用读取。

4.3 输出接口设计

(1) 记录用户落子

①输出条件

用户落子。

②传递消息定义

记录用户落子，便于悔棋程序及电脑 AI 执行。

③传输方式

函数调用读取。

(2) 电脑落子

①输出条件

人机对战中，用户落子后电脑计算落子。

②传递消息定义

电脑 AI 计算落子。

③传输方式

函数调用读取。

5. 存储包

5.1 功能说明

负责存储用户信息，存储用户棋局战况，具体功能如下：

- (1) 实现用户战绩存储。
- (2) 读取用户战绩。

5.2 输入接口设计

(1) 记录用户战绩文件

①输入条件

用户开始下棋，且对弈双方（包括 AI）有一方获胜。

②传递消息定义

记录用户战绩并生成相应文件。

③传输方式

驱动程序系统调用。

5.3 输出接口设计

(1) 读取用户战绩

①输出条件

用户读取。

②传递消息定义

打开生成的战绩记录文档。

③传输方式

函数调用读取。

6. 数据结构设计

6.1 估值核心数据结构和程序的关系

为了提高估值核心的效率，估值的计算不是通过一次操作来完成的，而是分散到估值核心的多个操作中来完成的。每次落子或悔棋时，都会重新分析落子点或提子点所影响的那四行（横行、纵行和两个斜行）上的棋子组合状态，然后更

改整个棋盘的棋子组合状态，这样做就不必到返回估值时在对整个棋盘进行全部扫描，因此极大的提高了估值核心的效率。

6.2 搜索引擎数据结构和程序的关系

根据历史得分调整节点顺序是通过排序过程完成的，因为排序也需要时间，所以不宜把排序放到搜索算法中去，但是这样调整节点顺序的机会就少了很多，为了能够更好的调整节点顺序，采用迭代深化是一个好的方法，它将仅进行一次 n 层搜索转化为从 1 层到 n 层渐进的进行 n 次搜索，每次搜索前都根据上一次搜索得到的历史得分来调整节点顺序以便进行更深一层的搜索。这样表面上看是进行了更多的搜索，而实际情况却是深层搜索比浅层搜索的节点多的多，多进行的那些浅层搜索对于深层的搜索可以忽略不计，而浅层搜索得到的历史得分更好的调整了节点的顺序，使得深层搜索的剪枝效率大大提高，因此实际搜索的节点数大大地减少了，搜索的速度也就提高了。