

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

1.程序系统的结构

1.1 软件基本处理

软件基本处理流程见图 1.1。

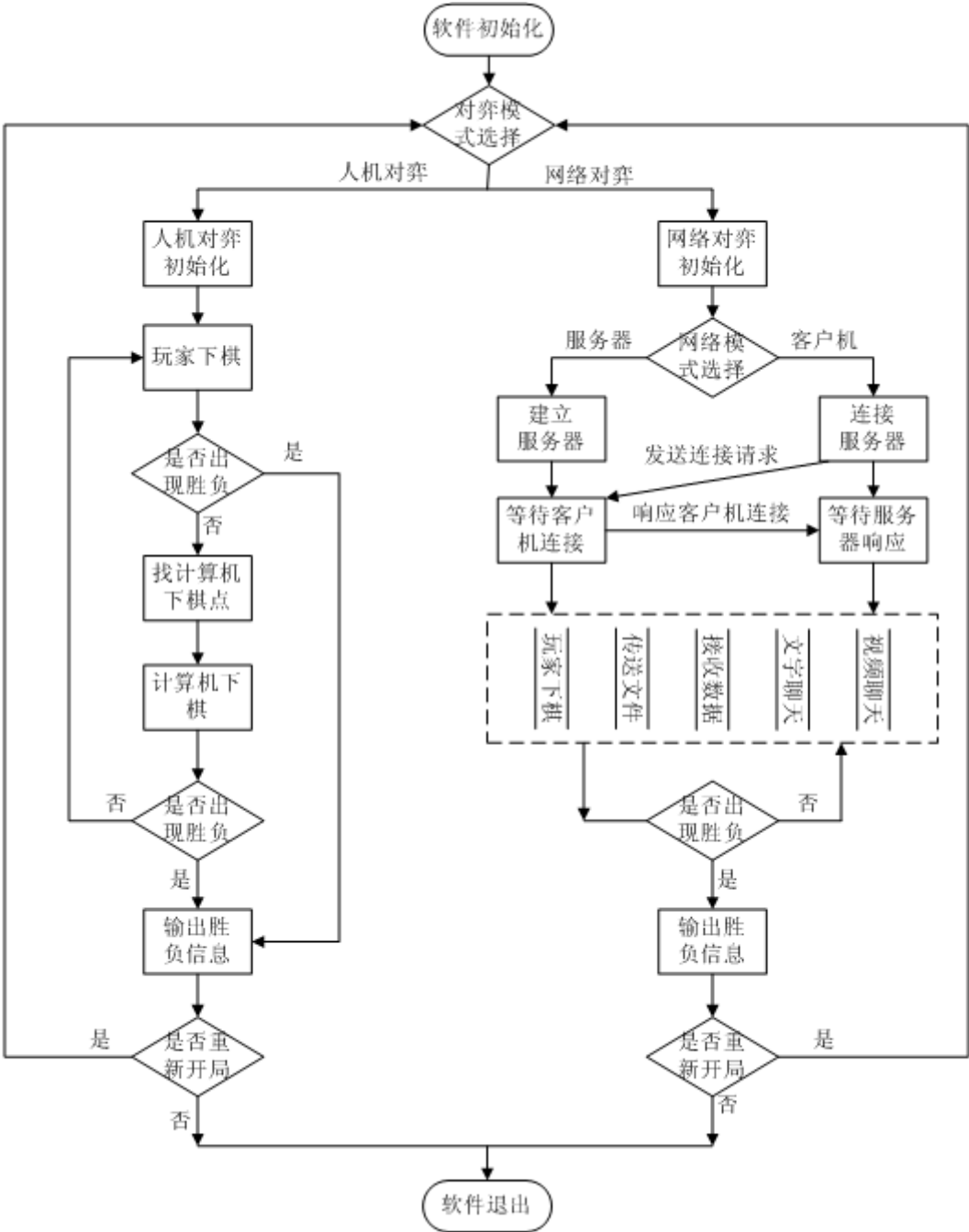


图 2.1 软件基本处理流程

本系统各个模块之间的控制关系图 1.2。

1.2 模块控制关系

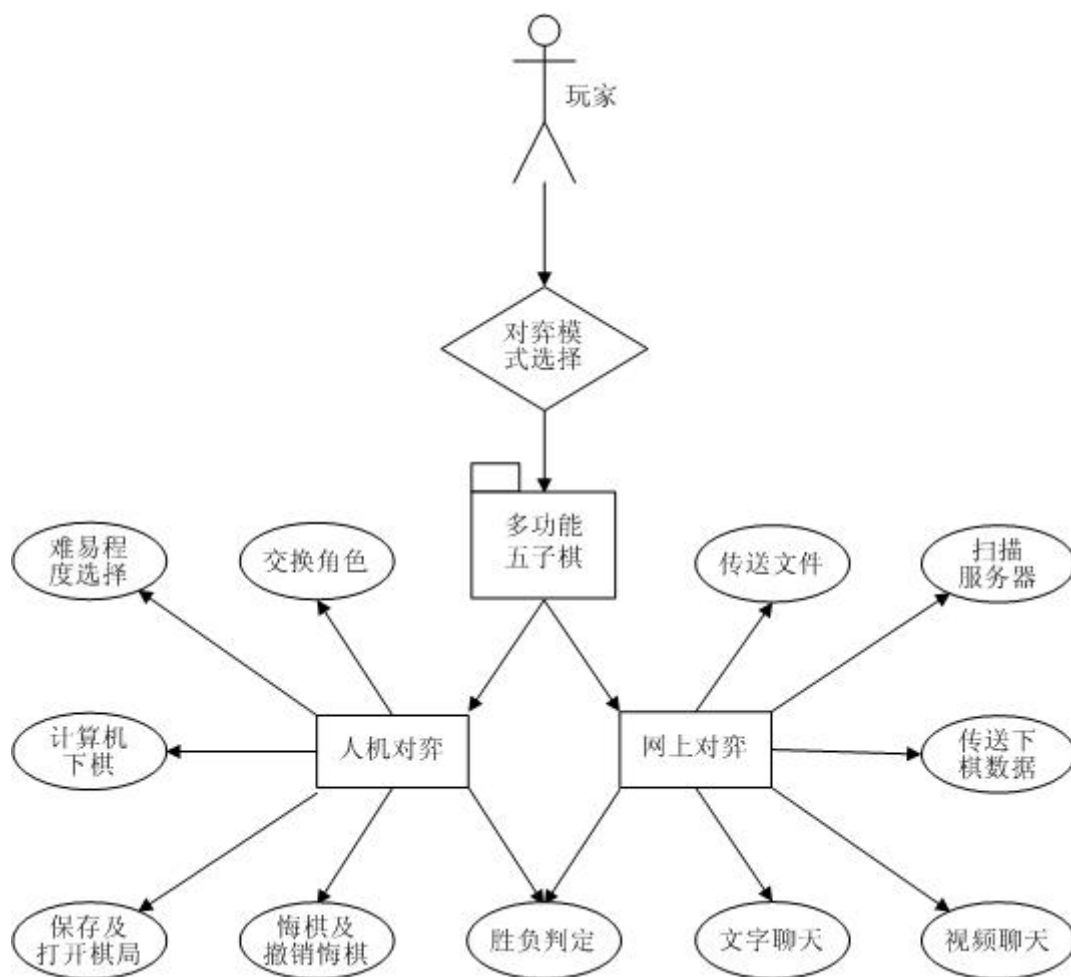


图 1.2 系统模块之间的控制关系

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。

2.4 性能

为保证能无限次悔棋及撤销悔棋，直到棋局初始状态为止，堆栈变量的大小应该足够大。这里，设置为棋盘总棋子数目。

2.5 算法

用面向连接的 TCP 连接控制方式，保证数据传输的可靠性。

主要过程包括：读取、解析下棋数据包，更新相应数据结构，更新棋局显示，发送下棋者下棋后封装的下棋数据包。

分别计算黑白棋在各空位置下棋时的棋局评分，对各评分进行比较，取最高者为当前下棋点。

悔棋时，需经过以下步骤：

1. 对棋步堆栈进行弹栈到临时变量（需弹两步，包括计算加下棋的一步）
2. 若上一步操作不是悔棋，则对悔棋堆栈进行清零
3. 将临时变量压栈到悔棋堆栈
4. 清除当前棋局数组中的临时变量表示的棋步，并更新显示棋局

撤销悔棋时，需经过以下步骤：

1. 对悔棋堆栈进行弹栈到临时变量（需弹两步，包括计算加下棋的一步）
2. 在当前棋局数组中增加临时变量表示的棋步，并更新显示棋局

2.6 程序逻辑

悔棋和撤销悔棋功能的程序逻辑如图 2.1 和图 2.2 所示：

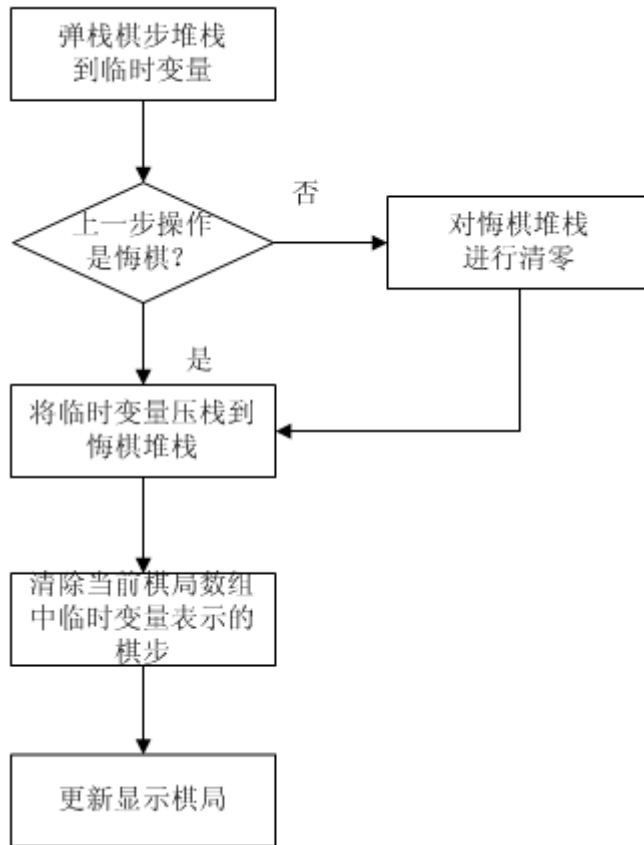


图 2.1 悔棋功能的程序逻辑

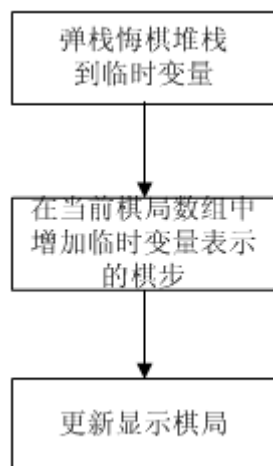


图 2.2 撤销悔棋功能的程序逻辑

3.对战包

3.1 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。

3.2 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。

3.3 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。

3.4 性能

(1) 人人博弈

网络传输延迟应小于 0.5 秒，否则玩家感觉到的延迟会比较明显

(2) 人机对战

为实现该功能，需要较高的人工智能，既要使棋局朝有利于计算机的方向发展，又要尽量堵死对方的“四连”、“冲四”、“活三”等棋局。

(3) 胜负判断

要求必须能准确无误地进行胜负判定，否则软件将出现重大错误。

胜负判定应在 0.1 秒内完成。

3.5 算法

(1) 人人博弈

用面向连接的 TCP 连接控制方式，保证数据传输的可靠性。

主要过程包括：读取、解析下棋数据包，更新相应数据结构，更新棋局显示，发送下棋者下棋后封装的下棋数据包。

(2) 人机对战

分别计算黑白棋在各空位置下棋时的棋局评分，对各评分进行比较，取最高者为当前下棋点。

(3) 胜负判断

分别从水平、垂直、斜线、反斜线方向搜索黑白连珠个数，若至少有一个方向连珠个数大于或等于 5，则判定为出现胜负，否则判定为未出现胜负。

3.6 程序逻辑

(1) 人人博弈

实现人人博弈的程序逻辑见图 2.1。

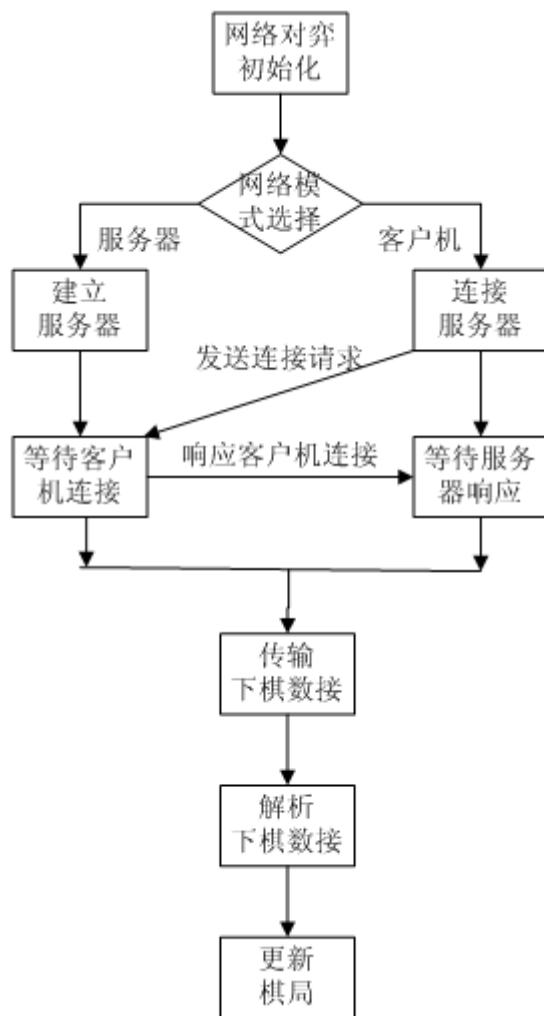


图 2.1 人人博弈功能的程序逻辑

(2) 人机对战

实现人机对战的程序逻辑见图 2.2。

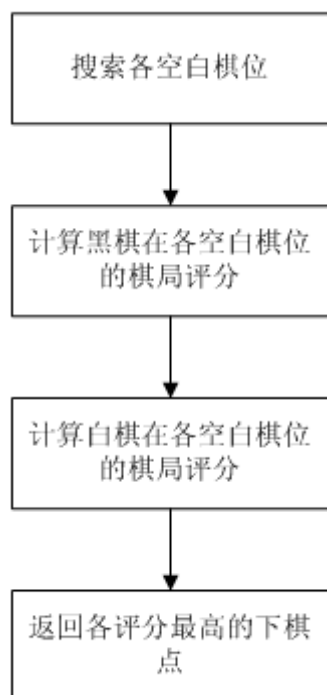


图 2.2 人机对战功能的程序逻辑

(3) 胜负判定

实现胜负判定的程序逻辑见图 2.3。

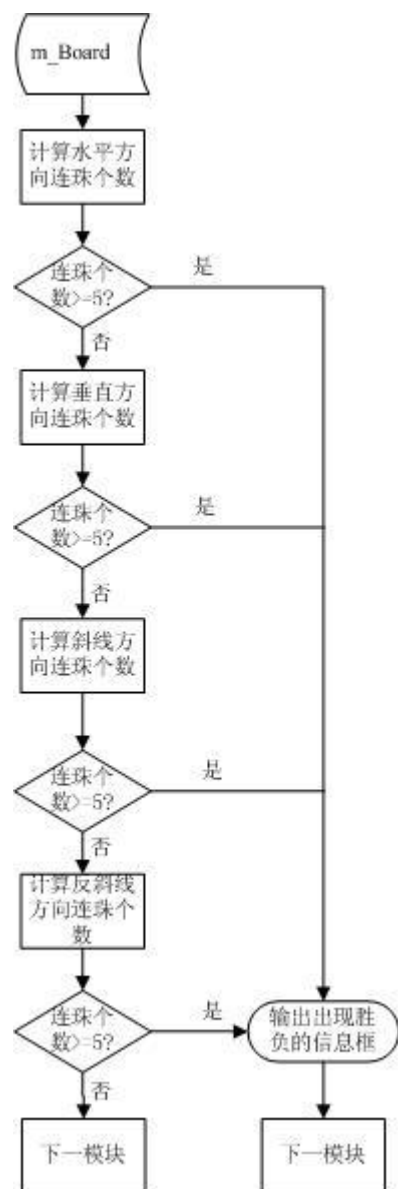


图 2.3 胜负判定功能的程序逻辑

4.存储包

4.1 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。

4.2 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。

4.3 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。

4.4 性能

无特殊性能需求。

4.5 算法

对于保存棋局，直接将当前棋局数组保存到指定文件即可。

对于打开棋局，直接将读取指定文件到当前棋局即可。

4.6 程序逻辑

保存棋局模块的流程逻辑如下：

```
ofstream fout(strName);  
int i, j;  
for (i=0; i<15; i++)  
    for (j=0; j<15; j++)  
        fout << m_Board[i][j] << endl;
```

打开棋局模块的流程逻辑如下：

```
ifstream fin(strName);  
  
int i, j;  
  
for (i = 0; i<15; i++)
```

```
for (j=0; j<15; j++)
```

```
    fin >> m_Board[i][j];
```