



吉林大学 2019-2020 学年第 2 学期

《数据结构》课程设计

C 题 Normal Track

六子棋锦标赛

五子棋起源于中国，发展在日本，是一类经典的棋类博弈项目。随着传统的五子棋被证明存在先手必胜的公平性问题，六子棋逐渐发展起来。六子棋规则与五子棋类似，且能保证公平性，是“中国大学生计算机博弈大赛暨中国计算机博弈锦标赛”的正式比赛项目。在本题中，你的任务是编写六子棋 AI 程序，即让程序自动下棋，并与其他同学对战博弈，进而决出冠军及名次。

游戏规则：

六子棋的规则与五子棋非常相似，玩家有黑白两方，各持黑子与白子，黑方先行。采用 19×19 的棋盘。具体玩法：除了第一次黑方下一子外，之后白黑双方轮流每次各下两子（即第一步黑方下一子、然后白方下两子、黑方下两子、白方下两子……）。直的、横的、斜的连成 6 子（或以上）者获胜。若全部棋盘填满仍未分出胜负，则为和局。不允许在棋盘同一位置重复下子，若一方在已有棋子的位置下子，则视为非法操作，直接判负。图 1 为黑方获胜。

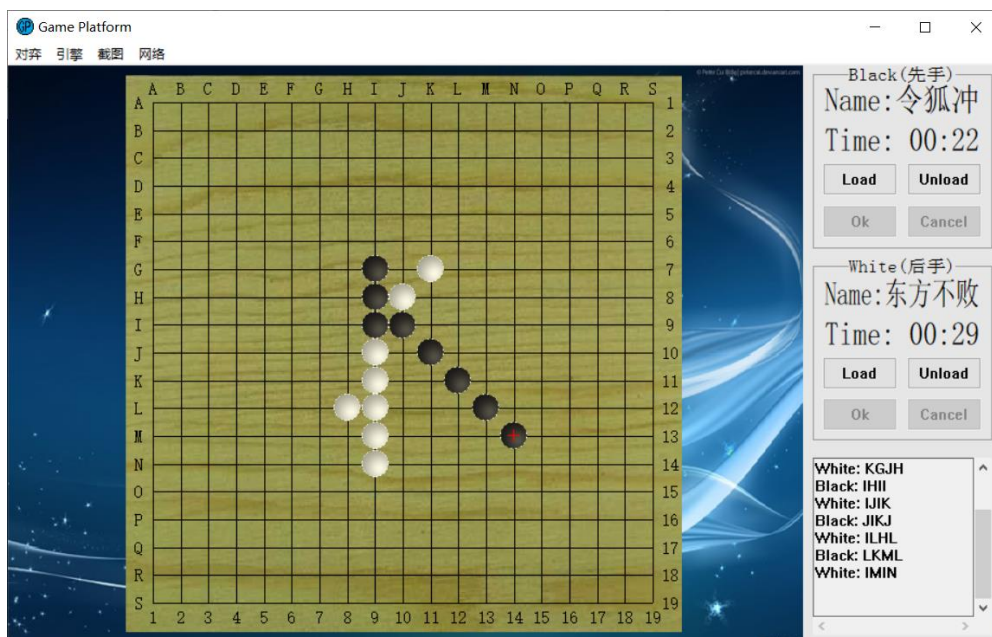


图 1

代码实现:

学生无需掌握图形界面编程技术, 只需使用C/C++语言, 在给定的“框架程序” chess.cpp 中填写核心代码即可, 对战平台负责图形显示。本题采用中国大学生计算机博弈大赛官方对战平台。允许使用STL。

(1) 棋盘坐标设定

19×19的二维棋盘分为横轴和纵轴两个维度, 以左上角为坐标原点。坐标系如图2所示。在框架程序中, 棋盘信息存储在数组`int Board[19][19]`中, 下标从0开始, 元素`Board[x][y]`有0、1、2三种可能取值, 分别表示棋盘(x, y)处为黑子、白子、空白(没有任何棋子)。

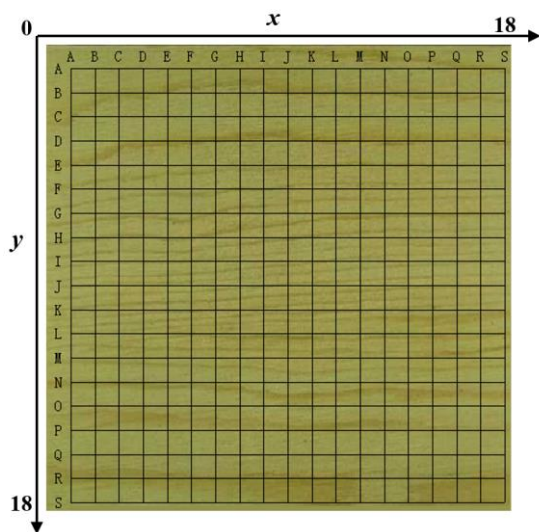


图2 棋盘坐标系

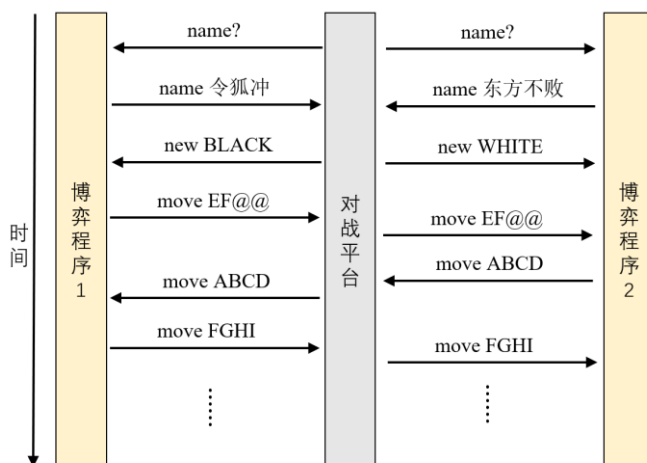


图3 对战平台与博弈程序通信示例

(2) 学生程序与对战平台的通信原理

了解本小节内容有助于理解框架程序的运行原理和流程。若不想或无法理解本节内容, 可直接跳过本节看(3), 对完成本题没有影响。

学生程序与对战平台程序是两个不同的进程, 两个进程间的通信通过匿名管道技术与标准输入输出重定向来实现。学生程序只需调用标准输入输出函数即可与对战平台进行交互, 具体地, 通过`scanf`函数接收对战平台的命令, 调用`printf`函数向对战平台发送命令及信息。

我们将学生编写的六子棋AI程序称为“博弈程序”。对战平台与博弈程序间通信如图3所示。

对战平台加载博弈程序后会向博弈程序发送“name?”命令询问队名, 此时博弈程序要发送“name 队名”命令予以回应, 否则对战平台将认为博弈程序没有为对弈做好准备。

当对弈开始时, 平台首先向博弈程序发送“new side”命令宣布对局开始并给博弈程序分配执棋颜色。当博弈程序接收到“new side”命令时, 若side值为BLACK, 则本方为先手, 此时应向平台发送“move”命令(具体格式稍后详述), 将第一子的落子位置告知平台。

在对弈进行过程中, 平台向博弈程序发送“move”命令, 将对手的行棋信息告知博弈程序, 然后博弈程序向平台发送“move”命令, 将自己的行棋策略告知平台。

对于“move”命令, 由于一步行棋着法包含两枚棋子位置2个坐标, 所以“move”命令采用“move X₁Y₁X₂Y₂”格式, 参数X₁Y₁X₂Y₂为4个大写字母, 每个字母在A-S之间, A表示0、B表示1、...、S表示18。其中(X₁, Y₁)表示第一枚棋子坐标; (X₂, Y₂)表示第二枚棋子坐标。例如图1中, 黑棋向平台发送“move MLNM”命令, 表示黑方第一个棋子坐标是(M, L), 第二个棋子坐标是(N, M)。但对于第一手着法, 因为只落一子, 所以X₂Y₂人为设定为@@, 表示无效信息。

(3) 博弈程序编写

你需要实现chess.cpp中的3个部分。

第一部分：设置队名（在chess.cpp第39行处），如下图

```
35     if (strcmp(message, "name?") == 0) //向对战平台发送队名
36     {
37         fflush(stdin);
38         /*****将"令狐冲"改为你的队名, 不超过6个汉字或12个英文字母, 否则无成绩*****/
39         /*****/ printf("name 令狐冲\n"); /**只修改令狐冲, 不要删除name空格****/
40         /*****将"令狐冲"改为你的队名, 不超过6个汉字或12个英文字母, 否则无成绩*****/
41     }
```

第二部分：生成第一步棋的走法（第 59 行处），将落子的坐标保存在 step.first 结构体中，如下图

```
42     else if (strcmp(message, "new") == 0) //建立新棋局
43     {
44         int i, j;
45         scanf("%s", message); //获取己方执棋颜色
46         fflush(stdin);
47         if (strcmp(message, "black") == 0) computerSide = BLACK; //执黑
48         else computerSide = WHITE; //执白
49
50         for (i = 0; i < 19; ++i) //初始化棋局
51             for (j = 0; j < 19; ++j)
52                 Board[i][j] = EMPTY;
53         start = 1;
54
55         if (computerSide == BLACK)
56         {
57             /*****生成第一手着法, 并保存在step结构中, 落子坐标为(step.first.x, step.first.y)*****/
58             /*****在下方填充代码, 并替换我的示例代码*****/
59
60
61             step.first.x = 9;
62             step.first.y = 9;
63
64
65             /*****在上面填充第一步行棋代码*****/
66
67             Board[step.first.x][step.first.y] = computerSide; //处理己方行棋
68             printf("move %c%c@@\n", step.first.x + 'A', step.first.y + 'A'); //输出着法
```

第三部分：生成每步棋的走法（第 85 行处），因为每步下两子，故将第 1 子的坐标存于 step.first 中，第 2 子的坐标存于 step.second 中，如下图

```
81     /*****生成落子的坐标, 保存在step结构中, 第一子下在(step.first.x, step.first.y), 第二子下在(step.first.x, step.first.y)*****/
82     /*****在下方填充代码, 并替换我的示例代码*****/
83
84     //生成第1子落子位置step.first.x和step.first.y
85     int x, y;
86     x = rand() % 19; y = rand() % 19;
87     while (Board[x][y] != EMPTY)
88     {
89         x = rand() % 19; y = rand() % 19;
90     }
91     step.first.x = x;
92     step.first.y = y;
93     Board[step.first.x][step.first.y] = computerSide;
94
95     //生成第2子落子位置step.second.x和step.second.y
96     x = rand() % 19; y = rand() % 19;
97     while (Board[x][y] != EMPTY)
98     {
99         x = rand() % 19; y = rand() % 19;
100     }
101     step.second.x = x;
102     step.second.y = y;
103     Board[step.second.x][step.second.y] = computerSide;
104
105     /*****在上面填充代码*****/
106     /*****
```

目前提供给大家的示例程序，即可直接运行，其行子方式是随机下棋。

(4) 博弈程序的调试及对战平台的使用

将 chess.cpp 编译运行，生成 exe 文件。打开对战平台 SAU Game Platform.exe，选择“六子棋”，如图 4 所示。点击图 5 右侧的 Load 按钮加载博弈程序，选定博弈程序 exe 文件。点击“对弈开始”菜单，如图 6 所示，即可进行对弈。退出时需先点击 Unload 按钮卸载两个博弈程序。

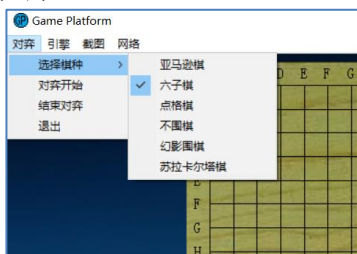


图 4



图 6



图 5

提交方式:

本题以团队方式完成，组队方式另行通知。每队通过超星作业的方式提交 chess.cpp 源文件和课设报告。chess.cpp 源文件不要改名。课设报告每队 1 份，格式模板另行发布，没有命名要求。每队只由队长提交作业，成员不用提交。

我们希望绝大多数同学选择本题。

评测方法:

以淘汰赛的方式进行评测，比赛分为两阶段。

- 第一阶段：每班内部采用淘汰赛方式，决出 16 强、8 强、4 强、2 强、冠军。比赛前每班进行种子队排序，按每队所有成员中上学期《数据结构》期末成绩的最高分进行排序。假如某班有 16 队，则淘汰赛时，种子排序第 1 位的队对阵第 16 位的队，第 2 位对阵第 15 位，以此类推。如果某班队数不足 2^k ，则可能出现某队轮空情况。
- 第二阶段：诸神之战。第一阶段产生的各班冠军（计算机学院 15 个班，软件学院 10 个班）+7 个成绩最好的亚军，共 32 队，再进行淘汰赛，决出两院总冠军。第二阶段仅供娱乐，与本课程成绩无关。
- 每队成员的成绩依据该队取得的名次、队内各成员的贡献等决定，我们会尽最大努力客观公正地评判每名同学的成绩。只要能正常完成对弈者，无论输赢都可及格。若程序编译出错或无法完成对弈，则记 0 分。
- 每班决赛采用 3 局 2 胜制；两院总决赛采用 5 局 3 胜制；其余比赛 1 局定胜负，以随机抽签方式决定先后手。因为六子棋具有公平性，所以先手并无太大优势。
- 每局对战用时采用包干制，每方总共有 3 分钟（注意不是每步 3 分钟，而是一局总共 3 分钟），超时判负。

➤ 上述规则可能在题目进行过程中根据同学们的具体情况进一步优化、修改。

提示:

本题没有标准答案，同学们可以充分发挥想象力给出自己的解法。任何基础、任何层次的学生都有能力给出解决方案。只要程序能正常完成对弈，即可及格。

对于基础较弱的同学，至少可以随机下子。再高级一点，可以构思一些规则，比如贪心规则，让本方尽可能多的棋子连起来，或者尽可能阻止对方棋子连线。更高级一点，可以使用博弈树+剪枝。再高一点，可以尝试机器学习方法，例如训练神经网络。甚至将上述几种方法相结合。

诚信要求:

我们允许并鼓励大家查阅资料、文献、学术论文，甚至自学一些高级技术完成本题。可以借鉴思路，但绝不允许抄袭其他队的代码，或者抄袭网络上的开源和非开源代码，改头换面后作为自己开发的程序参赛。老师已经收集了网络上与本题相关的所有代码作为查重对比母板，与网络或其他队代码雷同者，均视为抄袭。抄袭者与被抄袭者双方同论，不做区分。**某队一旦被判为抄袭，该队所有成员 C 题记 -300 分，这意味着这门课程基本上将不及格，无论之前 A、B 题取得什么成绩。**

奖励措施:

获得两院总冠军和亚军的队伍，在其自愿的前提下，老师将资助其代表吉林大学参加 2020 年中国大学生计算机博弈大赛（若受疫情影响，也可改为 2021 年大赛）。

