计算概论 A 大作业报告

王瞿骋 2300012260

一、程序功能

该程序支持新游戏、暂停、存档、读档、退出游戏等基本功能。除此以外，该程序还支持了 pvp 和 pve 两种模式，并允许在读档前对存档进行预览。

运行程序后将会显示菜单，可以通过输入对应字母选择新游戏、读档、查看规则或者退出。在本程序所有需要输入的地方都考虑了非法输入，会予以提示并重新回到等待输入状态。

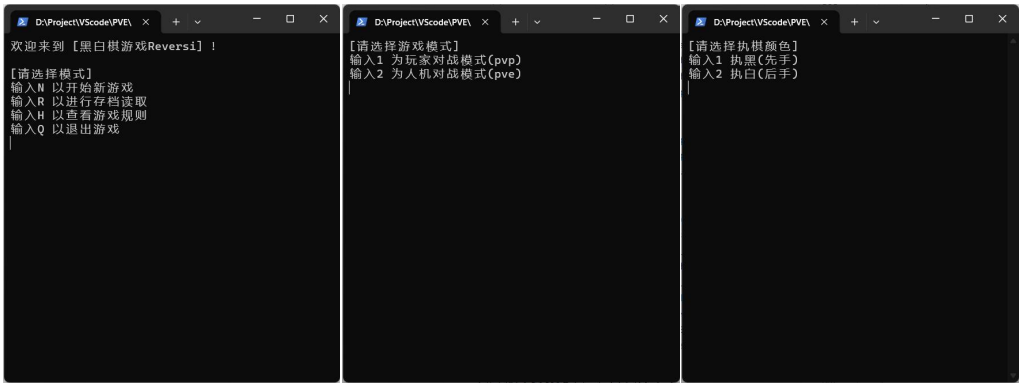

图 1 主界面、模式选择和执棋颜色选择

选择新游戏后，可以选择 pvp 或 pve 模式，如果选择 pve 后续将需要选择执棋颜色。接下来进入游戏，首先显示棋局（包含棋盘、棋子和可落子点）、黑白双方棋子数量和需要进行的操作；在 pvp 模式中，双方交替使用键盘输入落子点（例如 A1,D5），已有棋子、不可落子和非法输入会分别被拒绝；在 pve 模式中，玩家落子回合与 pvp 基本相同，电脑落子回合则显示“电脑正在落子”，当玩家按下任意按键后进入自己的回合。在游戏过程中输入 P 可以暂停游戏，并在暂停界面中选择返回游戏、存档、返回首页、退出。

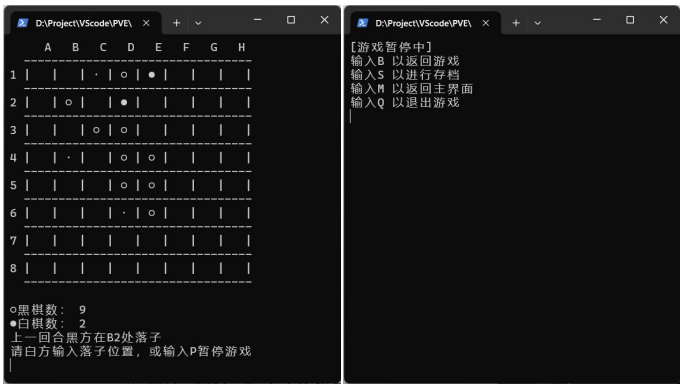

图 2 游戏过程界面和暂停界面

选择存档后，将会显示 9 个存档槽，已有存档的槽位会限时为[存档 N]，没有则显示[空]；输入 1-9 数字后可以将现有棋局存入存档（如果指定非空存档，会询问是否覆盖）；将一个初始棋局存入已有存档可以将该槽位恢复为空。返回首页后则可以选择新游戏、返回暂停界面、读档、查看规则和退出，选择新游戏会询问是否放弃现有棋局。

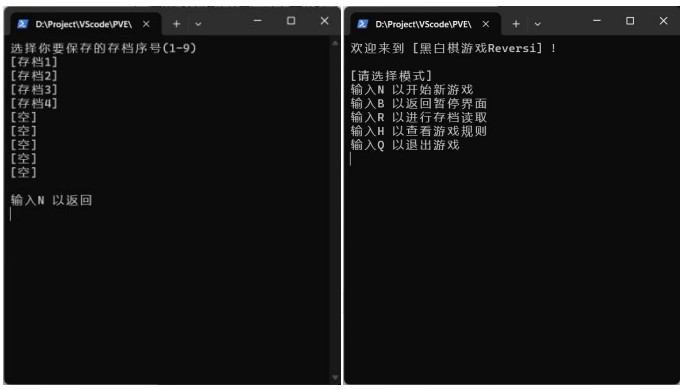

图 3 存档界面和暂停时主界面

选择存档读取后，与存档界面类似，会显示 9 个存档槽位，选择空槽位会提示并要求重新输入；选择非空槽位则进入存档预览（仅显示棋盘、棋子），并询问是否读档并覆盖当前棋局；如选择是，则进入 pvp/pve 选择，后同正常游戏流程。

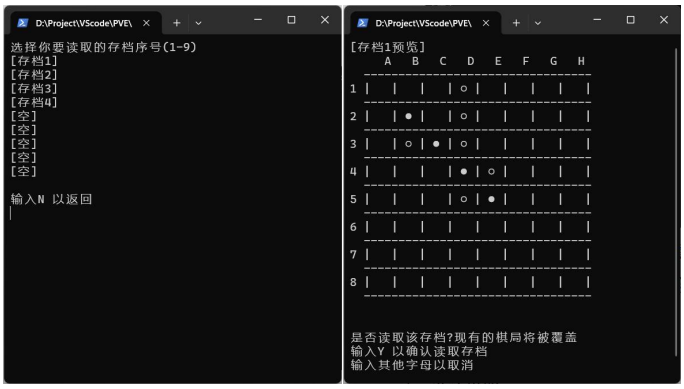

图 4 读档界面和存档预览

选择查看游戏规则，会简要介绍黑白棋的游戏规则，然后按任意键跳回；选择退出游戏，会提示未保存的棋局将会丢失并确认是否退出，如是则结束程序运行。

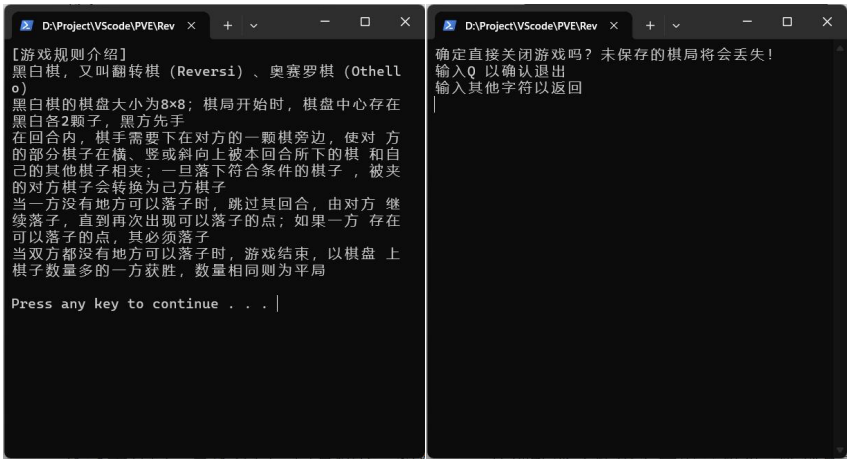

图 5 游戏规则介绍和退出界面

二、设计思路

游戏使用二维数组的方式存储棋盘（0 为空格，1 为黑子，-1 为白子，2 为可落子点），以及另一个用于辅助落子点搜索的二维数组。几个重要 int 变量，存储本回合颜色、棋局数、上回合落子点、是否需要初始化、pvp 和 pve 模式等。

可执行部分主要包含三部分：基础功能函数、综合功能函数和外接.h 头文件。基础功能函数用于实现画棋盘、检测某点是否可落子、标记落子点、落子并翻转旗子、黑白子计数、输出旗子数、规则介绍、存档、退出等功能相对独立、不太需要嵌套调用其他函数的功能。综合功能函数用于实现玩家落子操作界面、主界面、暂停时主界面、暂停界面、回合操作、新游戏、主程序等需要大量调用基础

功能函数、实现与玩家交互的功能。外接头文件 reversi_ai.h 是本人用于参加 botzone 的 AI 的修改版，将原程序的输入输出改为与本程序适配的参数和返回值形式（主要是将输出变为单个数值，即 1-64 表示落子位置），主函数改为 mainAI 函数，然后以头文件形式引入本程序，并在 pve 模式中调用。

三、实现方式

棋盘的初始化、图形化，棋子计数和输出，规则介绍，退出等原理较为简单，不做赘述。

检查某点是否可以落子（judge 函数）的方式是对 8 个方向分别进行搜索（使用 dx、dy 两个-1~1 的变量），若相邻格为异色旗子，则继续向该方向略过所有异色棋子，直到遇到同色棋子时即返回 1，表明该点可以落子。在该函数基础上，用另一函数将一回合内所有可落子点存在 tempBoard 数组中，然后再将其标到 Board 上（search 函数）。在得到落子位置信息后，reverse 函数将以与 judge 类似的 8 向搜索方式向 8 个方向检测，区别是遇到同色棋子后会返回并将该方向上的异色棋子全部翻转为同色。

newgame 函数同时承担新游戏和读档继续游戏的功能。新游戏时，mode 变量值设为 0，此时 newgame 函数会执行初始化操作并将 mode 设为 1；如果是读档游戏，mode 值为 1，直接进入回合循环。游戏结束的检测方式是：有一个数组 flag 存入白棋上一局是否落子和黑棋上一局是否落子。一旦连续两个颜色都无法落子则结束游戏，并对棋子进行计数，多的一方为胜（或平局）。游戏结束后，会回到主界面。

存档功能使用<fstream>头文件的 ofstream 来进行写入。存档为 txt 文件，保存在 exe 文件旁的 saves 文件夹中。txt 中的数据为：回合数-白棋是否可落子（0/1）-黑棋是否可落子（0/1）-64 格内的棋子情况（不存入可落子点，只存入棋子和空格，因此在预览中只显示棋盘和旗子），每个数之间由空格隔开。

读档过程并未写成单独的函数（原本写了，但是后面忘记调用，故删除），大致思路是先将 txt 文件中的数据存到单独的变量和数组中，然后用 printBoard 函数以存档数据为参数实现预览。如果选择读档，会将 color，round，flag，Board 等数据用存档数据覆盖，然后将 mode 值设为 1，然后返回值，进入 newgame 函

数直接开始游戏。

综合功能函数是主体部分。由于函数不能嵌套定义，每个函数设计为返回某些特定值的 int 型，在需要调用该函数的函数中只需对该函数的返回值做处理（例如，回合操作 round 函数中调用了暂停界面 pause 函数，而 pause 中如果要返回游戏，则不能再调用 round，否则成为嵌套定义；在该程序中，是在将 pause 放入一个死循环中，当 pause 返回对应继续游戏的值时不跳出循环，而其他返回值则跳出）。

在程序需要输入的地方，在死循环中声明一个字符数组，用 cin.getline 读取，只有当 strlen 和字符都符合时才执行对应操作并跳出。其他输入方式会被提示非法操作并重新开始循环。

此外，pve 中使用的 AI 使用的是 minmax 方法加 α - β 剪枝。在该方法中对后面几步棋进行模拟（目前设为 7 步），用估价矩阵对终末状态的棋盘进行评分，并往回追溯，在己方回合取估值最大的枝，对方回合取估值最小的枝，最终判断出最佳的落子位置。

四、作业亮点

本程序实现了 pvp 和 pve 两种模式的结合，可以在新游戏和读档时任意选择，具有较强的可玩性。在多数交互方面充分考虑了玩家的操作可能，进行了一些人性化的设计，例如对所有输入处设计了对非法输入的处理、在退出时需要再次确认、游戏中开始新游戏或读档会提示现有棋局会被覆盖等等。读取存档过程中，允许玩家进行存档预览，使玩家对存档棋局有直观的了解，利于不同玩家直接游玩。