**程序设计——五子棋大作业**

**人工智能2283班 522030910158 宋源祎**

**一、程序说明：**

本程序是一个简易的五子棋游戏，包含以下功能：开始游戏、读取最后一次保存棋局、打印棋盘格、玩家落子、确认或取消落子、判断落子合法性、获取下棋提示、判断输赢、中途退出游戏并保存棋局，悔棋（每一回合、每一玩家），棋盘格满后自动退出游戏。以上功能会在下文分别详细阐述。

本程序在windows系统可以成功运行。由于涉及<windows.h>库（设置棋子颜色）的引用,其他系统应该无法使用。（可以利用ifdef相关语句处理，出于实际需要考虑，我并未做此处理）

**二、程序使用方式：**

编译代码需要切换到main函数界面。

如果您在您的IDE上编译并运行代码，请注意换成GBK模式，因为我是使用GBK模式保存的，否则您可能无法看到我的代码注释，如果控制台输出中文仍为乱码，建议您使用可执行文件运行程序。

该程序使用非常简便（以.exe文件为例），打开可执行文件，页面自动输出游戏说明，按照程序要求执行每步操作即可。

游戏开始，页面输出游戏规则以及操作方式。输入s开始新的游戏，输入i读取最后一次保存记录。双方轮流下棋，系统会提示目前是哪位玩家执棋。每轮中，玩家需要先决定是否打印棋盘格p/n，然后选择是否悔棋r/n（注意每次重启程序后需要下够两步才能悔棋）或者退出游戏q/n（此时自动保存），是否获取落子提示t/n（仅提示可以成三、四、五的棋），接下来，玩家输入落子索引（0～14之间的整数），系统自动判断合法性，如果输入数字超出索引范围，或该点已有棋子，则返回错误，重新输入索引，每次输入索引后，系统会询问玩家是否确认落子，如果确认落子输入y，否则输入n，重新选择落子位置。每一步下棋后，系统会自动判断棋局是否结束，一局游戏结束后，玩家可以选择保存棋局并退出q，或者悔棋r（此时最多可以悔两轮，但仍满足开局前两步不能悔棋的条件），输入其他的字母将退出游戏，不会保存该局棋局记录。注意，如果保存了已分胜负的棋局并退出，读取记录后并不能再悔棋，该功能主要用于战后分析棋局。

**三、代码思路：**

在实现代码中，我创建了两个类，chessnode棋子类和chessboard棋盘类。

chessnode棋子类，共有五个成员数据，分别表示横纵坐标、落子情况、value1和value2分别为该点为玩家1或玩家2落子时的所连棋子最大长度，成员函数用于实现设置和获取私有成员。

chessboard棋盘类，共有五个成员数据，使用成员函数将在棋盘上实现的功能分解，对棋盘直接进行操作来实现打印棋盘格、检验落子合法性、下棋、悔棋、判断输赢、获取建议等功能。

函数具体的功能在代码中均有详细注释，下面单独说一下私有成员函数value，其余的函数在介绍程序功能时会介绍。私有成员函数value函数负责计算该点下棋后，能够连成的棋子最大长度。分别计算横向、纵向、主对角线、副对角线方向上的连子长度，取最大值为该函数的返回值。

play函数：游戏中每一个步骤被分解为单独的函数，定义在print.h头文件中，print.h中定义的函数用于实现系统与玩家交流的大部分内容，在play函数中按顺序调用这些函数，实现玩家游戏过程中的全部流程（除开局和结束外）。该函数在main函数中调用。如果棋局分出胜负，或者棋盘已满，或者玩家选择中途退出，play函数都会退出。

**四、功能解析以及相关函数实现**

共有两位玩家：玩家 1（Player 1） 和 玩家 2（Player 2）

程序说明：本程序是一个简易的五子棋游戏，包含以下功能：

**（一）基本功能**

1.开始游戏：

此部分写在main函数中，使用ifelse语句实现，如果保存文件能够成功打开，询问玩家开始新游戏（s）还是读取保存记录（i），否则玩家输入s开始新游戏。通过while循环语句，如果输入字母非要求的值，程序将要求玩家重新输入。

2.打印棋盘格：

打印棋盘格的功能由chessboard类里的display成员函数实现，数字0～14表示棋盘坐标，棋盘为空的地方用o表示，玩家一的棋子设置为%（天蓝色），玩家二的棋子设置为#（亮黄色），便于区分。系统询问玩家是否打印棋盘格并调用display的函数为print\_display，定义在print.h文件中，在play函数中调用。

3.玩家落子：

玩家落子要经历以下过程，选手输入落子索引（0～14之间的整数），系统自动判断合法性，如果输入数字超出索引范围，或该点已有棋子，则返回错误，重新输入索引，每次输入索引后，系统会询问玩家是否确认落子，如果确认落子输入y，否则输入n，重新选择落子位置。整个过程的实现在print\_gochess和print\_judge函数中，在这些函数中调用相关的函数实现功能。具体落子操作的函数定义在chessboard类里的go\_chess函数中。

4.判断落子合法性：

chessboard类里的avail函数。合法性中判断两部分，一是落子索引不超过范围，二是落子位置没有其他棋子。

5.判断输赢：

chessboard类里的judge函数，judge里调用了该类里的私有成员函数value函数，如果value的返回值大于等于5，判断为该局结束。

6.异常情况退出：

代码中整型变量num用于实时记录棋子个数，如果num=225，程序会输出某玩家获胜或该局平局，棋盘已满并退出棋局，由玩家选择悔棋或者结束游戏。

7.中途退出游戏并保存棋局：

在print\_retract函数中，系统会询问玩家是否退出游戏，如果退出，通过play函数返回false，如果main函数中捕捉到返回的false值，则调用chessboard的save函数保存记录，并结束游戏。

**（二）增加功能：**

1.保存棋局：

chessboard类里的save函数，负责将当下棋盘的数据存入到save\_file文件中，存入棋盘中每个位置的落子情况，以及整个棋盘上的棋子数，用整型变量num表示。（由于没有将下棋顺序保存到文件中，在读取记录后的前两步时无法悔棋）,使用方式，退出时输入q会自动保存。

2.读取最后一次保存记录：

chessboard类中的in函数，负责将保存的数据输入到棋盘中，改变棋盘上的棋子，改变代表当前棋子数的num变量的大小，使用方式，在开局时输入i 。

3.悔棋：

在print\_retract函数中询问玩家是否悔棋，若悔棋，再调用chessboard类里的retract函数，悔棋的实现使用了<vector>库，定义了一个chessnode类的vector变量record，用于储存下棋顺序，每次悔棋退回两步，即退到上一回合，删去record顶层的两个数据，num减2，悔棋后，自动打印棋盘格，便于玩家选择落子位置。分出胜负后，玩家仍有一次选择悔棋的机会，回到上一步继续游戏，为了实现这一功能，且不破坏原有程序的结构，我大胆的使用了一处goto语句。使用方式，在询问是否悔棋后输入r。

4.提供下棋建议：

在print\_tip函数中询问玩家是否需要建议，调用chessboard类里的evaluate函数，功能的实现思路是，遍历棋盘里所有的空点，假设该位置下棋，计算满足的连子最大长度，该功能通过value函数实现，返回连子的最大长度值，类中的两个chessnode类型的vector对象value1、value2负责储存满足连子长度大于等于3的可落子位置，接下来，对两个数组冒泡排序，使其按从大到小的顺序排列，根据使用tip的玩家输出建议的攻击或防守落子位置。此功能只是一个简单的实现，很多情况并未考虑到，提供的建议只是一个参考，但能够满足简单的要求。使用方式，在询问是否需要建议时输入t。

**五、其他：**

1.关于代码中经常出现的num%2，是借助棋盘上的棋子数判断当下是玩家一还是玩家二；

2.一些函数的返回值被定义为布尔型，是便于区分两种不同的状况；

3.该程序中，给出tip的条件我设置的是大于等于3，可以在chessboard类中的evaluate函数中修改（tmp>=?）；

4.代码中有对于每个函数的详细注释，为防止不同编译器中文乱码的情况，我在另一个文件中附上完整代码。