黑白棋大作业实验报告

肖之屏 1200012850

基本情况：

黑白棋人机对战字符界面小游戏

1. 功能简介
2. 有基本的存档/读档/选择难度/选择先手后手的功能以及基本应战能力；

达到了各项基本要求指标。

1. 在游戏过程中通过输入一个负数随时可以进入选择菜单进行存盘/读盘/退出，也可以在该菜单下获得帮助——显示所有还可以落子的位置。
2. 附有详细规则说明，可供初学者使用。
3. 输入输出采用可以带空格的字符串全行读入并进行处理的方式，不会因为一次输入字符过多而影响下一次输入，也不会因为选项是数字而不小心键入字母（int型读入的话是有这个问题的）就陷入死循环之中。
4. 存档的时候如果已经有同名文件存在会给出提示询问是否要覆盖重写，以免误操作引起重要数据丢失的情况。
5. 虽然经过长期的debug应该不会再有这种情况发生，但是如果计算机搜索得到的落子地点其实是不合法的，棋局将被终止，返回错误提示，而不是陷入死循环或是继续错误。
6. 读盘的时候可以判断棋盘是否合法再读入，坚决不读入非法棋盘（非法依据：中心四颗有空位/所有棋子并未连成一片/存档的字符中存在非棋盘内容类非法字符）。
7. 中英文双语同时显示提示语,不仅可以适应中外友人的需要还可以在某种程度上帮助万国友人学习汉语。事实上，因为一些私人原因（鄙人的法国籍发小正在对门兄弟院校留学），本来一开始是设计成英文提示语给她们拿去玩的，在一些同学讨厌外语的呼声中才添置了中文提示。念及吾友人一行对学习汉语兴趣浓厚，私以为不如就中英文界面同步显示了。

其实设成可以全英文可以全中文的话也就是一个language全局变量的事，但是根据鄙人自身的经验，那样的话他们都会觉得中文界面用得别扭，毕竟不熟练。

应该说，这是考虑到客户群体的一个专项配置。虽然这个客户群体设置得很狭窄，但是在非商业目的下应该是可以理解的。

PS：所有英文都是编者，也就是在下，的纯脑力翻译结果，所以，如果有点小小的不地道，望海涵。

1. 设计思路
2. AI决策方式：

采用最为传统的博弈树搜索法，本质就是个递归+回溯+剪枝，其中剪枝是个不完全的alpha-beta剪枝，之所以说它是不完全的，是因为它并不是把可剪的枝子都剪干净了，某种意义上讲可以称之为“类alpha-beta剪枝法”，原理是相同的，但是实际操作上，每一个节点的第一个子节点都会被遍历一次，这其实是因为估值函数设计得有些夸张，只能用long long类型，inf设成0xfffffff都不够用，再大一点就默认是unsigned int 类型了，所以最终只能依据自己对alpha-beta剪枝原理的理解自创了一种看似alpha-beta 剪枝又不尽然的神奇剪枝法……（姑且称之half-a-b?）

1. 估值函数：主要有以下几个指标：

① 每个格点的静态权重，从始至终没有变化，角上值大，边上值大，紧挨着角的三（\*4）个格子赋值较小；

② 每个格子的动态权重，主要用于加强边角处理，在己方抢到角之后，不再避免落子在该角附近的三个格子，反之万一对手抢角则要避免紧挨着该角下子否则简直自寻死路；为了尽量拖延对手占边的速度，在对手落子在边的时候把其临位边位的值减小，等等。

③ 行动力，己方和对方分别有多少落子选择。这项指标据说是应该在游戏初期比较重要，但是个人观察发现它的价值体现一般都是到了终局让人无路可走……

④ 稳定性：

注意，这真的是我自己想出来的！但是之后却看见许多经典程序都是这么算稳定性的！我该说英雄所见略同，还是深深哀叹自己的出生为时过晚？！

思路：看四个穿过某棋子的直线上，如果该棋子前后直到空格或边角前都只有自己的子，或者两头已经都有敌方的子敌方一时半会翻不过来，就算稳定性+1，否则此直线上稳定性+0；累计己方每个子四个方向的稳定性并且求和为己方稳定性值，减去对方稳定性值再乘某个比例系数折合即可。

⑤ 先手后手：后手的时候调用随机函数取一个值来定第一步下哪里。

3) 界面设计：字符界面，虽然也听说图形界面引一个graphic头文件挺好做的，但是身为一个严重的病入膏肓的强迫症并且是学美术出身的人我不保证自己不会在“咿呀这里不好看”、“哎呀那里不好看”这种无聊的问题上耗掉太多时间。可以的话我希望自己的黑白棋是以AI的强大而取胜的，而不是徒有界面华而不实的花瓶（亲爱的同学们在此我发誓我没有鄙视任何人的意思，你们的程序都是好样的）。

4) 用户友好性：随时提供各种选择，并且到处都有提示语（中英文双语提示哟~），每一步的指令弄的还算是比较清楚的。

5) 输入随意性：在交战的时候，虽然本方落子采用的是原始的坐标输入法，但是：

① 坐标的输入横纵先后不拘，纵坐标（字母）大小写不拘，空格数不拘

② 输入格式错误会返回提示提醒并且要求重新输入

③ 不会因为输入非法字符而陷入无限死循环……我的这个程序很坚强，绝对不会因为一些恶意捣乱瞎敲字母的行为而崩掉

1. 优点
2. 保证基本的框架齐全功能齐全，麻雀虽小，五脏俱全。
3. 结构经典，方法经典，采用传统几大指标写估值函数，采用minimax搜索和类alpha-beta剪枝（至少算法思想是高度一致的）。
4. 把许多细节的情况列入考虑，比如：同名文件已存在、输入指令不合法甚至是乱来、计算机下棋系统判断故障、读盘的时候不是合法棋盘，等等。这些都不足以造成本程序死循环。
5. 有明确的用户群体目标（某对中国文化颇有兴趣的外国友人及其友人们），设计具有针对性。
6. 随时提供打开菜单的选择，可以随时存档，随时退出，随时获得提示。
7. 跟大部分同学的程序比，本人的AI还算比较强的了（这可不是吹牛，你们可以试试来）。
8. 人选后手的时候第一步计算机是随机下的，这个很有意思觉得么？
9. 缺点
10. 因为提示语行数过多，经常把棋盘挤到上边需要滚动一下才能看全。这一点如果是图形界面应该就能解决了，但是既然不是图形界面，做到如此已经可以算是尽力了。
11. 字符界面其实还是有点原始，也许应该改成图形界面吧，只可惜时间不是很够了，以后的大作业再说吧。这次把重点放在AI上我并不后悔。AI部分我已经尽力了。
12. 没有直接显示在棋盘上的可落子符号——这一点我要声明，这是纯粹的个人兴趣，我只是非常讨厌直接把可以下棋的位置都显示出来而已，我当初最开始玩黑白棋（和我的诺基亚），它的设置就是这样的——你要提示可以提示你哪里能下，但是不会一开始就显示在棋盘上。这才像话，不然是把棋手当傻子么？自己发现可落子的地方才会开心吧。我只是不想剥夺棋手的这项乐趣，当然我不是也设了help选项吗，没亏待大家，非得说这是缺陷的话我也是没办法。
13. 不能悔棋——这是我故意的。说老实话我一直很讨厌悔棋的行为，自己的决策为什么要出尔反尔呢，有句话说，覆水难收，如果给了大家悔棋的机会——虽然也是很容易的，只要存着之前的棋盘就好了，但是这样大家就不会用认真严肃的态度来下这盘棋了。可能会有人想“先下这里看一看，待会势头不妙再回来”，我不喜欢这种想法，人应该为自己的每个决定负起100%的责任。当然……这也可以说是功能缺陷……但是我的用户群是我多年的老朋友们……他们不会在意这种细节的。
14. 评估函数一根筋走到底，不区分前中后期。这也没有办法，因为事实上本人自己是黑白棋小白痴，做不来这么高端的分析，害怕贸然切分时期分期不当只会让AI越来越笨，所以没敢下这个手……
15. 代码放在一个CPP文件里面……这个是因为我自己没常识，以为就应该这样，本来分函数模块分得好好的结果都被自己全都拼到一个文件里边了……唉唉算了，这是细节，只是细节而已啦……
16. 收获感悟
17. 天下事有难易乎？为之，则难者亦易矣，不为，则易者亦难矣。诚然，黑白棋这项作业是我们目前编过的最长的代码，但是编完了之后觉得也不难，真的有种成为程序员的自豪感，想要大吼一声“yes we can”！
18. 千里之堤，溃于蚁穴。什么大风大浪都走过来了，回首看看自己在此程序中犯过的错：①变量名搞混，整个乌龙掉了

②加加减减的自己把自己搞晕了

③大括号只加一边忘加另一边

④数组设短了导致stack overflow

还有许多更弱智的错我就不在这里说了，倒是递归回溯剪枝基本都是一两下通关……这些都是平时作业也会犯的bug，但是出现在上千行的代码里，就一点也不可爱了……光锁定区间就要好半天设好几个断点分析原因……怎么说呢，还是不积小流无以成江海的感觉吧，基本功都是平时打的，没什么太多新花样。

1. 特别鸣谢：
2. 家父家母：谢谢你们帮我玩程序挑出那么多提出那么多改进建议，最后我的黑白棋能完成，你们二位功不可没。
3. 舍友黄文颖：作为我身边唯一的黑白棋高手，真是辛苦你和我的程序下那么多盘棋了，你提出的“几要几不要”原则指导我完成了了现在的AI，可以说，没有你就没有今天的它。
4. 李戈老师：谢谢您为我们打下如此坚实的编程基础，若非如此，这项大作业我们也不能如此轻松愉快地完成。
5. 亲爱的助教们：真是辛苦你们长期被我们骚扰了，不管是debug，还是问算法，我和各位同学们总也免不了麻烦你们，谢谢你们这么长时间的耐心指导，在各个方面，你们为我们提供了很大的帮助。
6. 丽滋里高：虽然你没有直接帮助我设计程序，也没有提出批评意见，但是我在编这个程序的时候，一直想着你，希望你会喜欢，这是我设计它的最强动力。我希望你会赞叹一下，以自己有一个优秀的程序员朋友而感到自豪。
7. 杨茜：我的第一大损友，就等着你来玩呢，快来找我吧，难度梯度这个最后加上的设定就是专门等你的，我要用最高难度虐你！
8. C++吧、C语言吧（百度贴吧）、CSDN、PUDN。没有你们，我还且一会才明白什么是alpha=beta剪枝。