

五子棋 AI 算法设计

【主要思路】

算法主要分为游戏逻辑实现和 UI 设计两大部分。

一、 游戏逻辑实现部分：

核心思想是对棋盘上尚未落子的点进行打分，判断下一步落在哪个位置时获胜的可能性最大。

具体操作包括：

1. 设置一个 15*15 的二维数组用于记录棋盘上每一点的状态，数组中每一元素应包括该点的坐标及落子情况，0 表示该位置为空，1 表示该位置上是黑棋，2 表示该位置上是白棋。

2. 定义 set 和 get 函数分别用于选择落子位置和获取某一位置的状态。

3. 定义四个函数，分别对某一位置的南北方向、东西方向、西北-东南方向、东北-西南方向的落子情况进行分析，判断是否已有一方获胜（连续五个同色棋子），记录当前结果及棋手。

4. 游戏的初始化（将记录棋盘状态的 15*15 的二维数组置空），输出“游戏开始”；若已有一方获胜，则游戏结束，输出该局结果，选择退出游戏或者进行第二局。

5. AI 部分：

能够得分的模型（以黑棋为例）：

0—空，1—黑棋，2—白棋

①连五—10 分：

11111

五个同色的棋子相邻，则已有一方获胜，该局结束。

②活四—6 分：

011110

已经有四个同色棋子相邻，且两端均无对方棋子，即有两个位置可以形成连五。

③冲四—5 分：

211110 / 11011 / 11101

只有一个位置可以形成连五

④活三—4 分：

0011100 / 011010

可以形成活四

⑤眠三—3 分：

211100 / 211010 / 210110 / 11001 / 10101 / 2011102

只能形成冲四

⑥活二—2 分：

001100 / 01010 / 1001

可以形成活三

⑦眠二—1 分：

211000 / 210100 / 210010 / 10001

只能形成眠三

1) 设置两个 15*15 的二维数组 ScoreA[]和 ScoreB[]，分别用来记录己方和对方的棋面评分情况。

2) 对每一个空位置进行评分，已落有棋子的位置分数均为 0。假设该空位置

上已落有黑棋，分别对该点的南北、东西、西北-东南、东北-西南四个方向上相邻的 8 个位置上的落子情况进行分析（以该空位置为中心，两侧各测 4 个位置），判断棋型，从而得出该空位置的得分。白棋同上。

不同棋型的具体判断过程（以黑棋为例）：

若含有中心棋子的连续棋子个数为 5，则为连五；

若连续棋子个数为 4：

若两端均为空，则为活四；

若一端为空，另一端为白棋，则为冲四；

若两端均为白棋，则该位置得分为 0；

若连续棋子个数为 3：

设相邻位置棋子分布为：AB111CD

若 $A=B=C=D=0$ ，则为活三；

若 $C=0, D=1$ ，或 $B=0, A=1$ ，则为冲四；

若 $B=2, C=D=0$ ，或 $A=B=0, C=2$ ，或 $A=D=2, B=C=0$ ，则为眠三；

否则该位置得分为 0；

若连续棋子个数为 2：

设相邻棋子分布为：ABCD11EFGH

若 $B=C=1, D=0$ ，或 $E=0, F=G=1$ ，则为冲四；

若 $D=E=G=0, F=1$ ，或 $B=D=E=0, C=1$ ，则为活三；

若 $D=2, E=G=0, F=1$ ，或 $E=2, B=D=0, C=1$ ，或 $B=2, C=1, D=E=0$ ，
或 $G=2, F=1, D=E=0$ ，或 $E=F=0, G=1$ ，或 $C=D=0, B=1$ ，则为眠三；

若 $C=D=E=F=0$ ，则为活二；

若 $D=2$, $E=F=G=0$, 或 $E=2$, $B=C=D=0$, 则为眠二 ;

否则该位置得分为 0 ;

若连续棋子个数为 1 :

说明该位置周围无相邻的同色棋子, 则该位置得分为 0。

3) 比较己方和对方的最高分 (分别记为 maxScoreA 和 maxScoreB ,

若 $\text{maxScoreA} \geq \text{maxScoreB}$, 进攻 :

若 maxScoreA 只对应一个位置, 则在该位置落子 ;

若有多个位置得分都为 maxScoreA , 则从中选择 ScoreB 最大的位置落子。

若 $\text{maxScoreA} < \text{maxScoreB}$, 防守 :

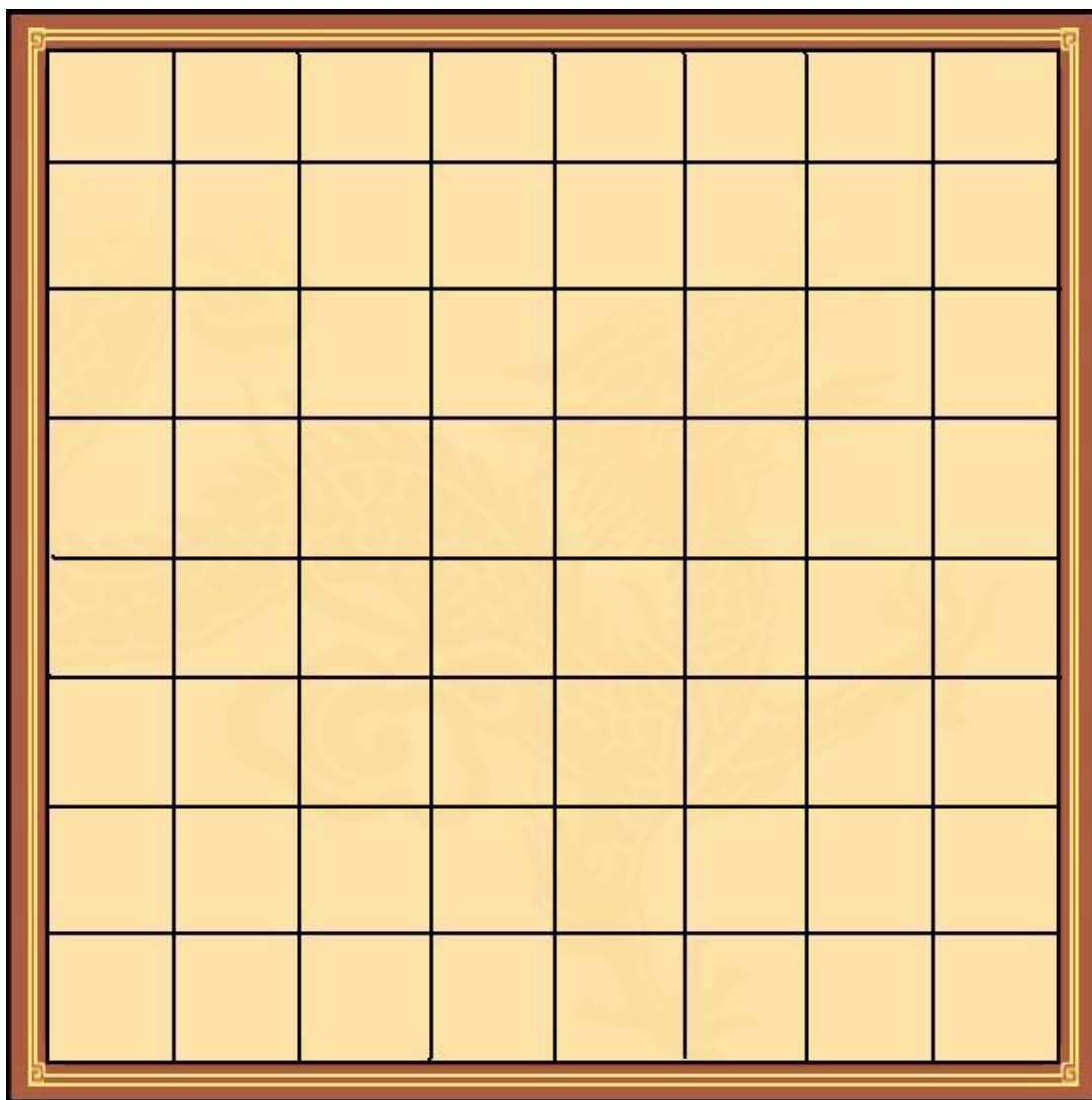
若对方的 maxScoreB 只对应一个位置, 则在该位置落子 ;

若对方有多个位置得分都为 maxScoreB , 则从中选择 ScoreA 最大的位置落子。

二、UI 设计部分(暂时未完成) :

1) 准备棋盘和黑白棋子的图片, 记录像素





(6

00*600)

2)