# C2022五子棋实验报告

李治林 2022080908011

一、功能：拥有基本五子棋的功能，可以悔棋，可以中途重来，可以选择是否先手，无禁手，可以调整棋盘或棋子的颜色，可以调整字体，可以显示作者名，在游戏结束后不会立即退出由玩家选择重玩或退出

二、使用工具：easyx图形库，vs

三、如何实现：

1.主体思想：

//开局

void begin(struct point p[25][25])

{

board(); //棋盘图形背景加载

while (1)

{

if (/\*按下鼠标左键\*/)

{

if (/\*鼠标坐标在悔棋区域\*/)

{

//悔棋

}

if (/\*鼠标坐标在重开区域\*/)

{

//重开

}

if (/\*鼠标坐标在投降区域\*/)

{

//投降

}

if (/\*鼠标坐标在退出区域\*/)

{

//退出

}

if (/\*鼠标坐标在棋盘上\*/) //玩家执棋

{

fillcircle(m.x, m.y, 20); //在该位置画红圆

p[i][j].state = -1; //修改该位置的状态

/\*判断胜负\*/

robot(p); //电脑执棋，在某位置画白圆

/\*判断胜负\*/

}

}

}

}

1. 关键代码部分：评估函数与极大值极小值搜索。该评估函数能够给出当前局面对于电脑的优势程度，并以具体数字呈现，数字越大该局面对电脑越有利，越小则对玩家越有利。并且对算法进行优化后以加深搜索深度加入了博弈树α-β剪枝。在这里我采用的是用if语句对不同的情况进行赋分。对每个点进行赋分，空点价值为0，红子和白子取相反数，每个点有8个方向的延伸价值，每个方向为10分，被敌方或边界挡住即为0；每两个相同的子连在一起则他们的分数乘一定的倍数，还有距离中心的优势分等等，在对每个点位进行评估后还会汇总为对整体局势进行评估的最终分数。部分代码：

评估函数：

for (x = 0, n = j; p[i][j].state != -p[i][n + 1].state && n + 1 < 20 && x < 4; x++, n++) //右

{

}

if (x < 4)

{

v1 = 0; //被挡住为0

}

else {

for (k = 0, n = j; p[i][j].state == p[i][n + 1].state && k < 5; k++, n++) //右

{

v1 \*= VALUE; //连续分数翻倍

}

}

剪枝：

double recursion(struct point p[25][25], int sex, int depth, long double a, long double b)

{

int m, n, i, j, t = 0, br = 0;

long double va = 0;

if (depth == 0 || end(p) != 0) //达到深度或棋局结束

{

return all\_value(p); //返回分数

}

else if (sex % 2 == 1) //玩家回合

{

for (m = MIN; m < MAX; m++)

{

if (br == 1) { //b剪枝

break;

}

for (n = MIN; n < MAX; n++)

{

if (p[m][n].state == 0) //为空

{

if (num < 80) {

t = 0;

for (i = -1; i < 2; i++) {

for (j = -1; j < 2; j++) {

if (p[i + m][j + n].state != 0) {

t++; //偏僻点剪枝

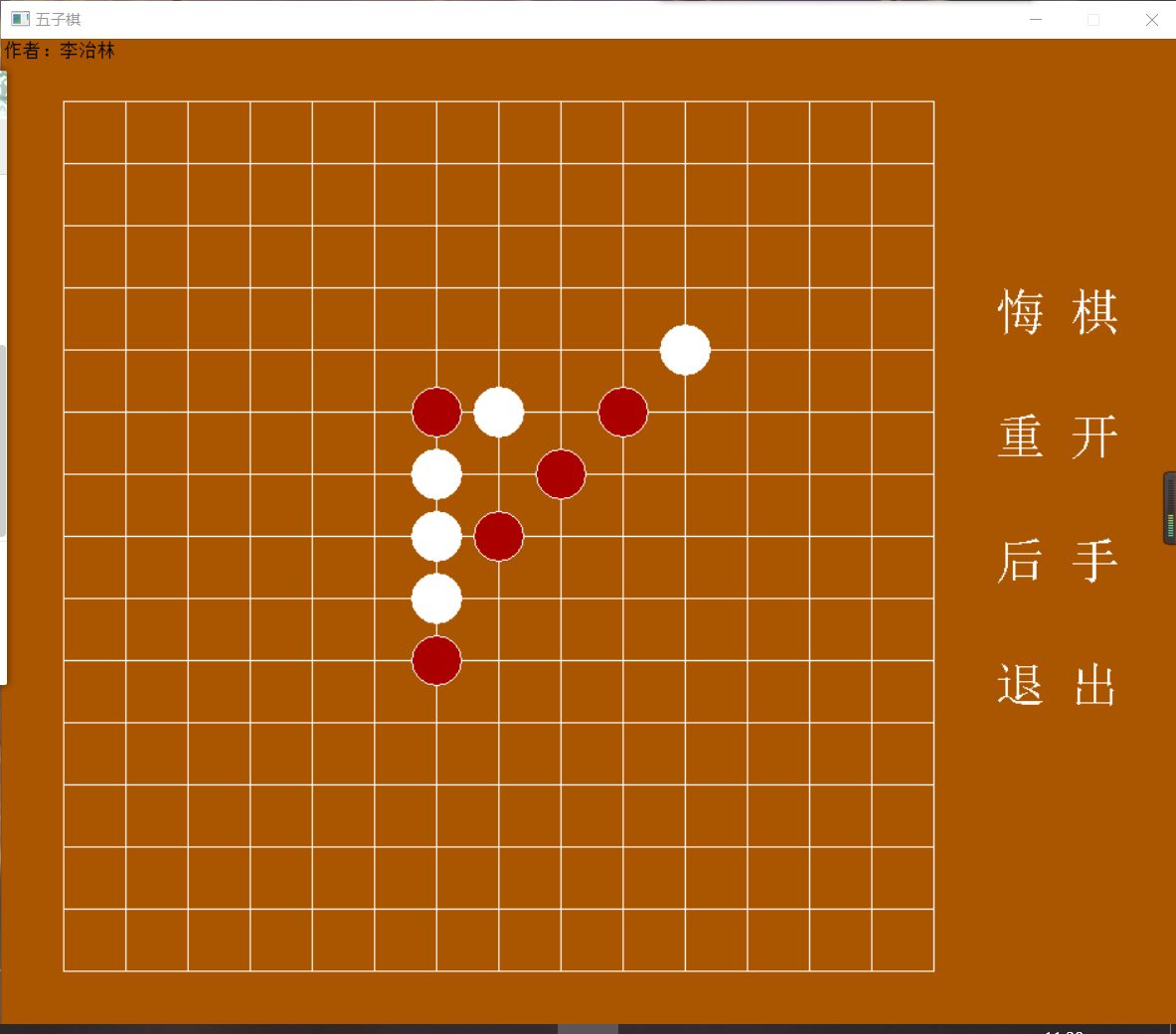
}

}

}

}

1. 运行：

程序大小约9M，运行无卡顿，截图如下：



1. 棋力评价：

抓禁技术：无

算力与大局观：可以评估全局局势算出当前的棋局分数，三层算法保证了其高于普通人的算力

定式储备：无

防守能力：能够比较精确地防住对方的无杀连攻

### 六、参考文献：《五子棋ai算法》作者：**lihongxun945 https://blog.csdn.net/lihongxun945/category\_6089493.html**