1、需要复审的源代码

#include<stdio.h>

#include<stdlib.h>

#define N 14

char state[N][N];

void init();//初始化棋盘

void printState();//打印棋盘

bool isWin(bool isBlack,int x,int y);//判断谁赢

bool isLevelWin(bool isBlack,int x,int y);//判断水平线线上是否五个棋子

bool isVerticalWin(bool isBlack,int x,int y);//判断竖直线上是否五个棋子

bool isLeftInclinedWin(bool isBlack,int x,int y);//判断左斜线是否五个棋子

bool isRightObliqueWin(bool isBlack,int x,int y);//判断有斜线是否五个棋子

int main()

{

int x,y;

bool isBlack=true;

init();

printf("五子棋小游戏\n\n@代表黑子，0代表白子，\*代表键盘空白\n");

printf("------------------------------------------------\n");

printState();

while(1)

{

printf("请%s方走棋:\n",(isBlack?"黑":"白"));

printf("请输入所下棋子的坐标，如1-2\n");

scanf("%d-%d",&x,&y);

if(state[x][y]=='@'||state[x][y]=='0')

{

printf("此位置已经有棋子了\n");

continue;

}

state[x][y]=isBlack?'@':'0';

system("CLS");

printf("五子棋小游戏\n\n@代表黑子，0代表白子，\*代表键盘空白\n");

printf("------------------------------------------------\n");

printState();//重新打印棋盘使下的棋出现

if(isWin(isBlack,x,y))//判断是否有人赢

{

printf("胜利方是:%s\n",isBlack?"黑":"白");

break;

}

isBlack=!isBlack;

}

return 0;

}

void init()

{

int i,j;

for(i=0; i<N; i++)

for(j=0; j<N; j++)

state[i][j]='\*';

}

void printState()

{

int i,j;

printf("%3c",' ');

for(i=0; i<N; i++)

printf("%3d",i);

printf("\n");

for(i=0; i<N; i++)

{

printf("%3d",i);

for(j=0; j<N; j++)

printf("%3c",state[i][j]);

printf("\n");

}

}

bool isWin(bool isBlack,int x,int y)

{

return isLevelWin(isBlack,x,y)

||isVerticalWin(isBlack,x,y)

||isLeftInclinedWin(isBlack,x,y)

||isRightObliqueWin(isBlack,x,y);

}

bool isLevelWin(bool isBlack,int x,int y)//判断水平线线上是否五个棋子

{

int count;

char ch=isBlack?'@':'0';

while(y>0&&state[x][y]==ch)

y--;

count=0;

if(state[x][y]==ch)count=1;

y++;

while(y<N&&state[x][y]==ch)

{

count++;

y++;

}

if(count==5)

return true;

else

return false;

}

bool isVerticalWin(bool isBlack,int x,int y)//判断竖直线上是否五个棋子

{

int count;

char ch=isBlack?'@':'0';

while(x>0&&state[x][y]==ch)

x--;

count=0;

if(state[x][y]==ch)count=1;

x++;

while(x<N&&state[x][y]==ch)

{

count++;

x++;

}

if(count==5)

return true;

else

return false;

}

bool isLeftInclinedWin(bool isBlack,int x,int y)//判断左斜线是否五个棋子

{

int count;

char ch=isBlack?'@':'0';

while(x>0&&y>0&&state[x][y]==ch)

{

x--;

y--;

}

count=0;

if(state[x][y]==ch)count=1;

x++;

y++;

while(x<N&&y<N&&state[x][y]==ch)

{

count++;

x++;

y++;

}

if(count==5)

return true;

else

return false;

}

bool isRightObliqueWin(bool isBlack,int x,int y)//判断有斜线是否五个棋子

{

int count;

char ch=isBlack?'@':'0';

while(x>0&&y<N&&state[x][y]==ch)

{

x--;

y++;

}

count=0;

if(state[x][y]==ch)count=1;

x++;

y--;

while(x<N&&y>=0&&state[x][y]==ch)

{

count++;

x++;

y--;

}

if(count==5)

return true;

else

return false;

}