程设大作业实验报告

实验名称：飞行棋小游戏		实验人：	杨凯盛
助教 gg 好帅求少扣分秋梨膏			

实验操作过程及配置说明：

Phase1：核心代码的实现

实验环境：Clion 2022 64bit，MinGW64，CMake

思路：通过 玩家->棋子->棋盘 的调用方式来进行走棋，事实证明其实这个思路在判断是否踩到或是重叠时有一定缺陷，让运算量较大，不过也能够轻易克服

1. 准备工作：头文件 prepare.h(由于内容较少直接粘贴在这里)(好奇怪编译器里没有下划线的这里有了)

```
struct board

{

    int SNj;

    int SNi;

};



struct chess

{

    char ID;

    board chess_place;
```

```
    bool start;
    bool arrived;
    chess* next;
};

struct player
{
    char ID;
    chess A;
    chess B;
    chess C;
    chess D;
    int unstart;
    int approched;
    player* next;
    int starti;//出发起始点的坐标
    int startj;
    bool human_gamer;
};
```

2. 部分核心代码展示：

<1> 生成随机数函数

```
srand(time(0));  // 使用当前时间作为种子初始化随机数生成器
```

```
r = rand() % 6+1;  // 生成 1~6 之间的随机数

printf("The number of points to roll the dice is: %d\n",r);

return r;
```

<2>链表式切换玩家

初始化部分

```
RED=(player*) malloc(sizeof (player));//写一半才想到能用 RWBY 致敬一下，然而懒，先申请内存
YELLOW=(player*) malloc(sizeof (player));

GREEN=(player*) malloc(sizeof (player));

BLUE=(player*) malloc(sizeof (player));



player*Player=RED;//初始化阶段


RED->next=BLUE;//轮流关系

BLUE->next=GREEN;

GREEN->next=YELLOW;

YELLOW->next=RED;
```

Main 函数 while 内部分

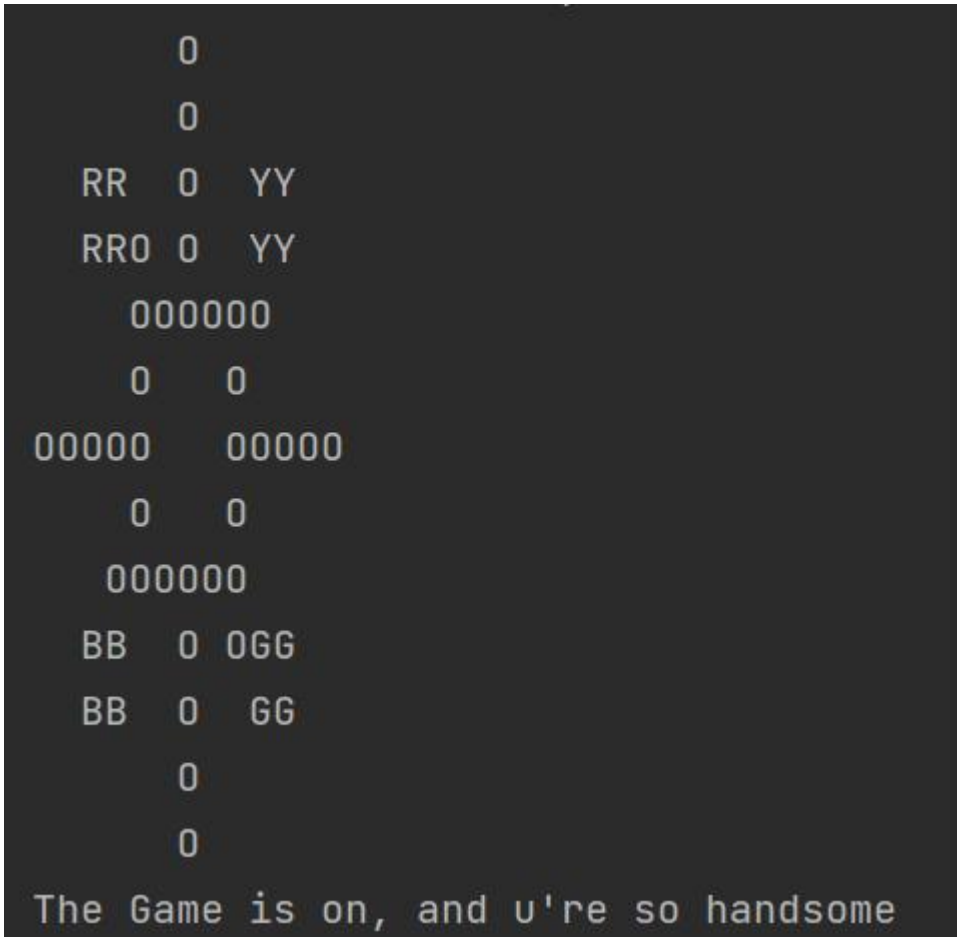
```
Player=Player->next;//切换玩家
```

<3>move 函数：一坨很暴力的屎山，技术含量较低，但是能够实现功能，帅气的助教 gg 有时间可以看下，没时间建议不看

3. 终端内效果展示

展示内容：<1>基础玩法

首先展示棋盘



然后红色的回合出了个新棋子

```
Round 1
It's your turn, R
wanna to cheat? Y/N
Y
how much step u want?(1~9)
6
first look at your chess guys
A is at (2,2)
B is at (2,3)
C is at (3,2)
D is at (3,3)
wanna a new chess to start? Y/N
Y
      O
      O
  OR  O  YY
  RRR O  YY
    OOOOOO
    O   O
OOOOO   OOOOO
    O   O
   OOOOOO
  BB  O OGG
  BB  O  GG
      O
      O
```

2. 在蓝色的回合里，我们试试错误的输入

```
Round 2
It's your turn, B
wanna to cheat? Y/N
Y
how much step u want?(1~9)
10
1~9 please
how much step u want?(1~9)
6
first look at your chess guys
A is at (2,9)
B is at (2,10)
C is at (3,9)
D is at (3,10)
wanna a new chess to start? Y/N
I
Y/N please
wanna a new chess to start? Y/N
N
But Y?,Please think more
wanna a new chess to start? Y/N
Y
      O
      O
  OR  O  YY
  RRR O  YY
    OOOOOO
    O    O
OOOOO    OOOOO
    O    O
   BOOOOO
  OB  O OGG
  BB  O  GG
      O
      O
```

另：此处没有考虑输入一个字符串，这不是正常发生的情况，可能会导致一下刷新很多行，一般对游戏没有影响，故不考虑使用 scanf_s 替代

如果投出的不是 6 直接跳过回合

```
Round 4
It's your turn, Y
wanna to cheat? Y/N
Y
how much step u want?(1~9)
1
first look at your chess guys
A is at (9,2)
B is at (10,3)
C is at (9,2)
D is at (10,3)
      O
      O
  OR  O  YY
  RRR O  YY
    OOOOOO
    O   O
OOOOO   OOOOO
    O   O
   BOOOOO
  OB  O GOG
  BB  O  GG
      O
      O
```

3. 我们快进到下一轮看看红色的走棋和蓝色其他的错误输入

```
Round 5
It's your turn, R
wanna to cheat? Y/N
Y
how much step u want?(1~9)
9
first look at your chess guys
A is at (4,3)
B is at (2,3)
C is at (3,2)
D is at (3,3)
look at your movable chess and make a choice:
A is at (4,3)
A
Now A is at (8,8)
      O
      O
  OR  O  YY
  RRO O  YY
    OOOOOO
    O   O
OOOOO   OOOOO
    O   O
   BOOOOR
  OB  O GOG
  BB  O  GG
      O
      O
```

字母 R 前进了 9 格

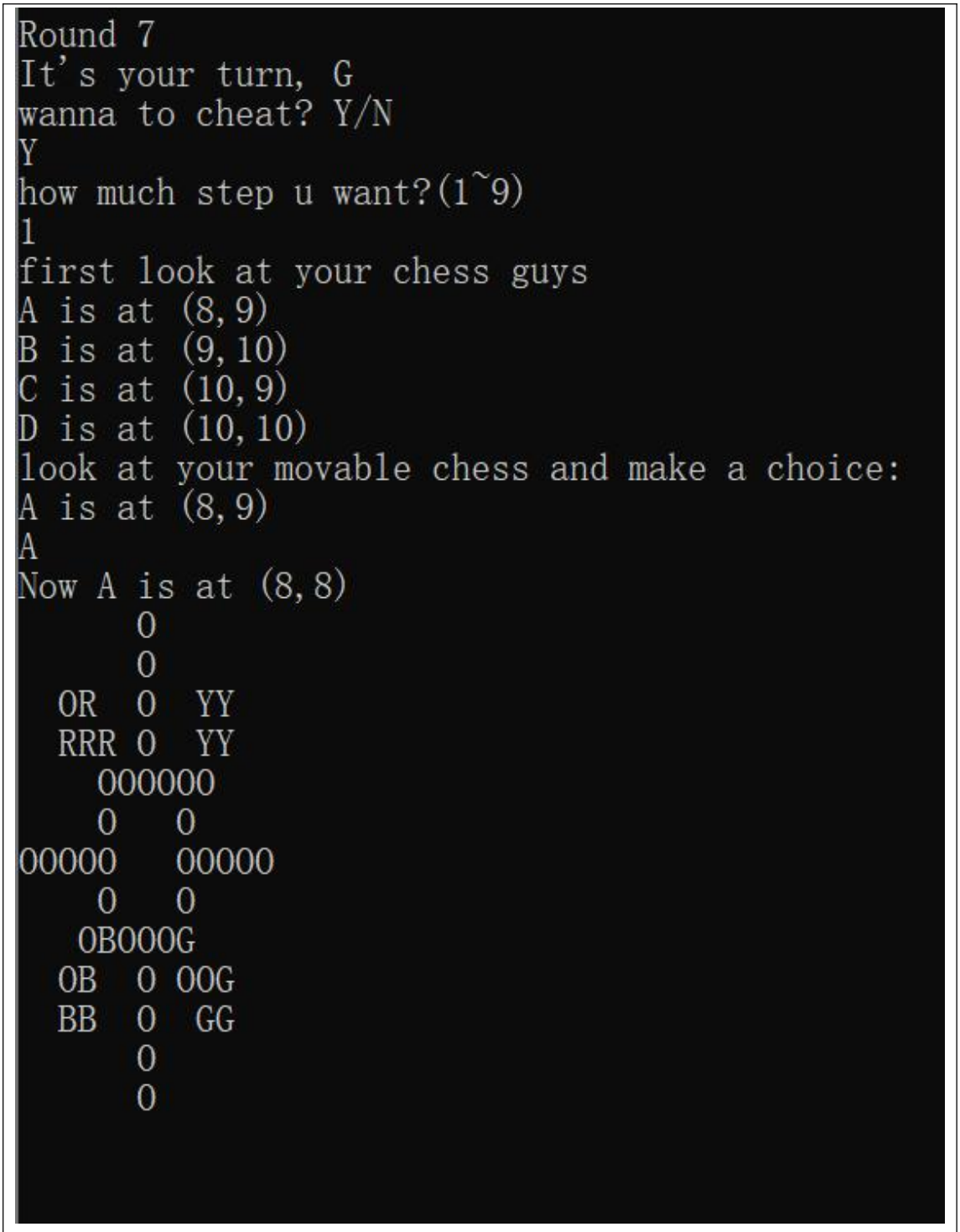
```
Round 6
It's your turn, B
wanna to cheat? Y/N
Y
how much step u want?(1~9)
10
1~9 please
how much step u want?(1~9)
1
first look at your chess guys
A is at (3,8)
B is at (2,10)
C is at (3,9)
D is at (3,10)
look at your movable chess and make a choice:
A is at (3,8)
O
don't do fucking illegal input please~ try again
look at your movable chess and make a choice:
A is at (3,8)
B
movable chess please~ try again
look at your movable chess and make a choice:
A is at (3,8)
A
Now A is at (4,8)
      O
      O
  OR  O  YY
  RRO O  YY
    OOOOOO
    O   O
OOOOO   OOOOO
    O   O
   OBOOOR
  OB  O GOG
  BB  O  GG
      O
      O
```

尝试了其他错误输入（同时这里判断以及走棋还用了 switch 的特性简化）（他太爱链表环了）

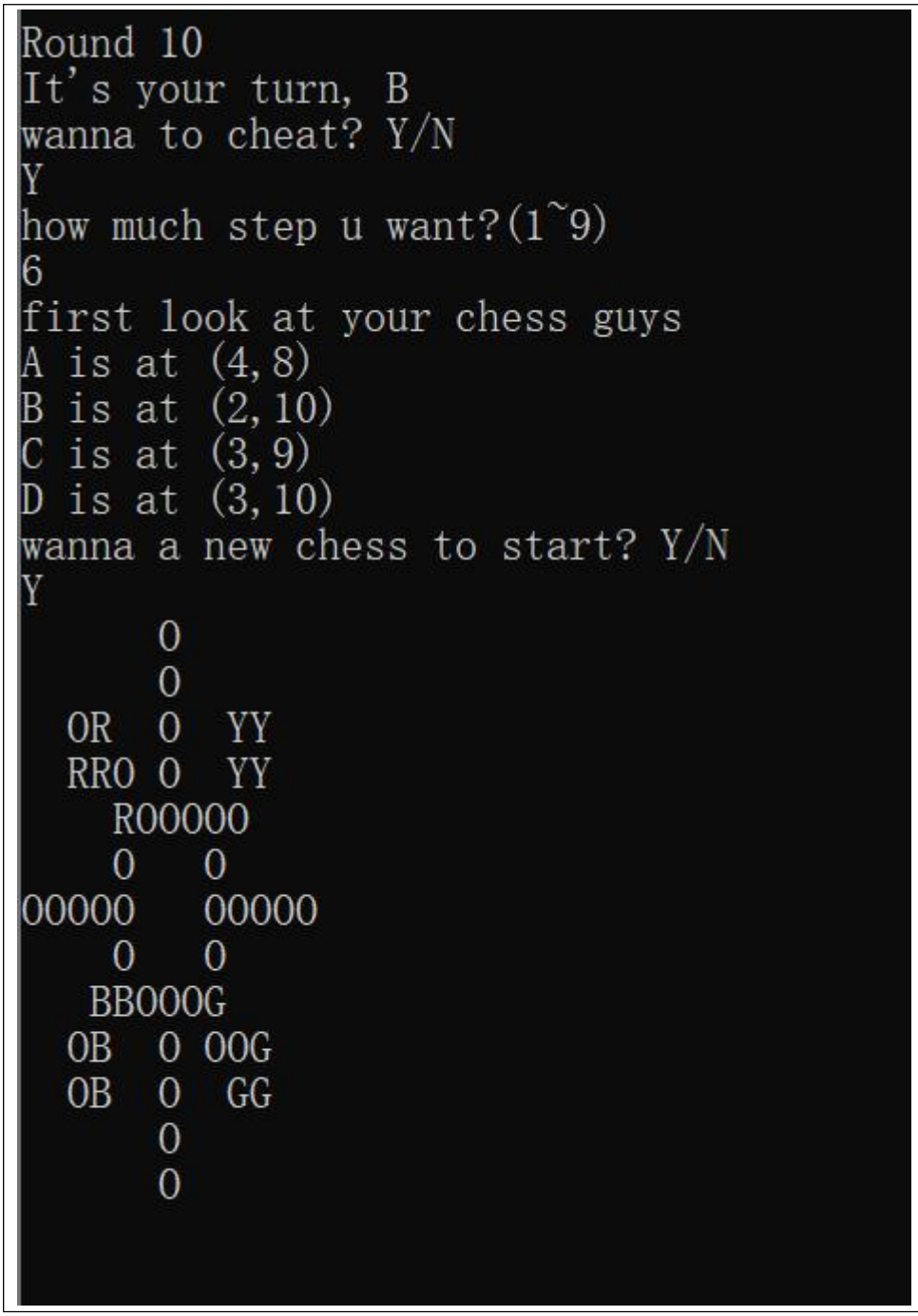
```
switch (chmove_chess)//错误输入类型：1 还未出发的棋子；2 已经到达的棋子；3 甚至不是一个棋子
{
    case 'D':
        movechess = movechess->next;
    case 'C':
        movechess = movechess->next;
    case 'B':
        movechess = movechess->next;
    case 'A'://这里是红色的移动
        //printf("DEBUG2\n");
        if (!movechess->start || movechess->chess_place.SNj == 0)//到达了或者还没出发
//修改 5
        {
            printf("movable chess please~ try again\n");
            switch (chmove_chess)
            {
                case 'B':
                    movechess=movechess->next;
                case 'C':
```

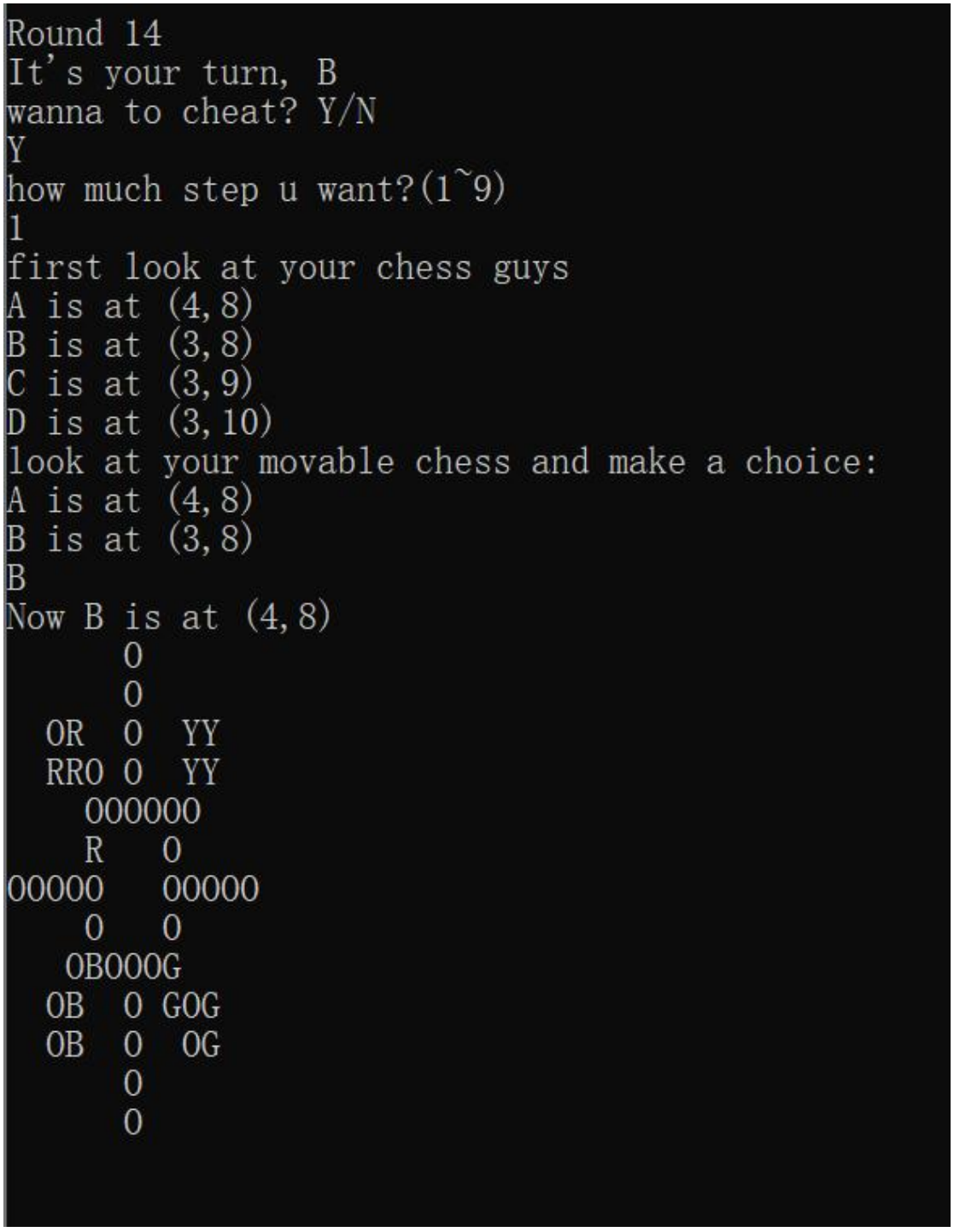
```
                movechess=movechess->next;
            case 'D':
                movechess=movechess->next;
            case 'A':
                break;
        }
        goto Move;
    }
```

4. 用绿色棋子把红棋子踩回出发点

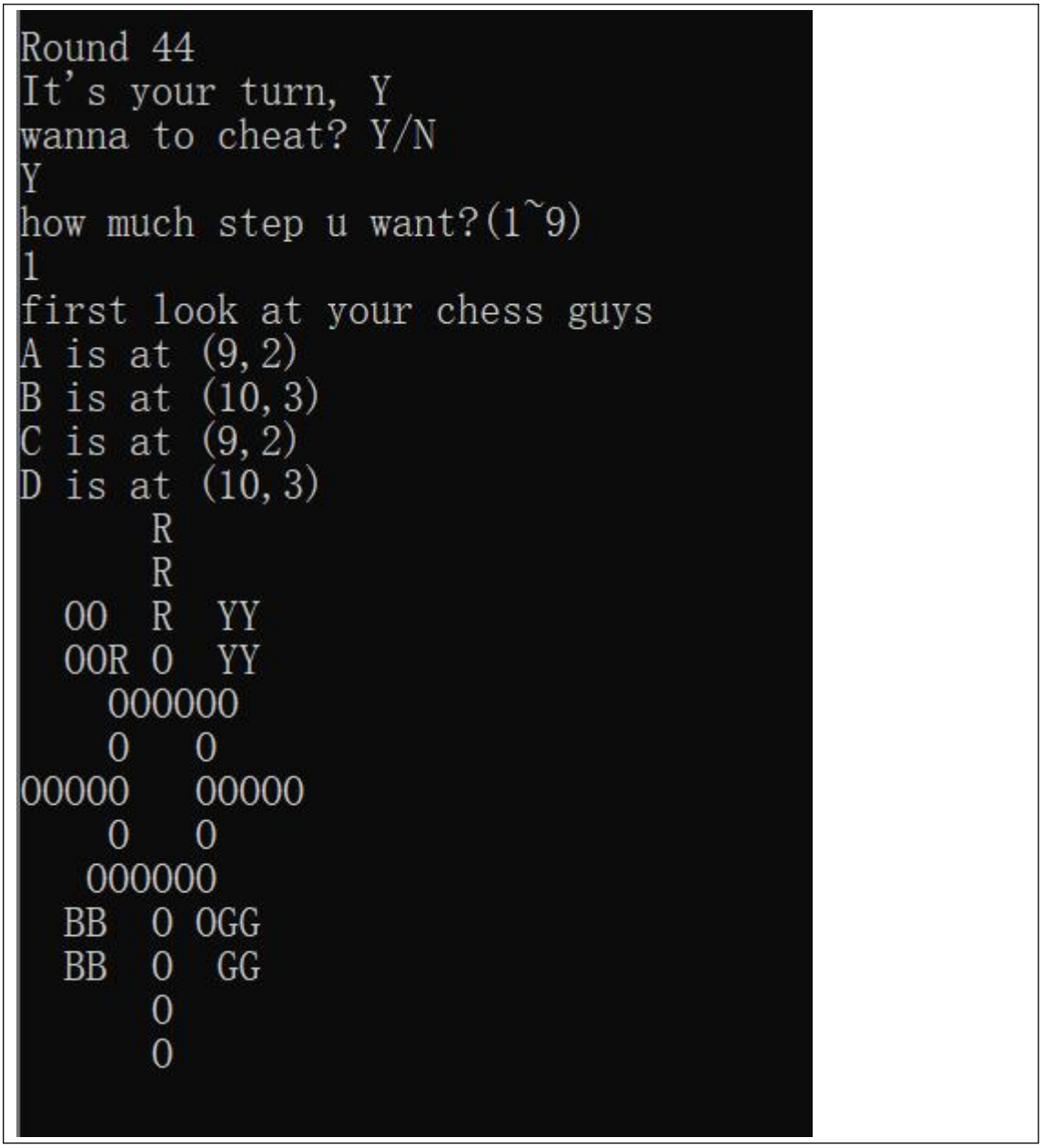

Round 7
It's your turn, G
wanna to cheat? Y/N
Y
how much step u want?(1~9)
1
first look at your chess guys
A is at (8,9)
B is at (9,10)
C is at (10,9)
D is at (10,10)
look at your movable chess and make a choice:
A is at (8,9)
A
Now A is at (8,8)
0
0
OR 0 YY
RRR 0 YY
OOOOOO
0 0
OOOOO OOOOO
0 0
OBOOOG
OB 0 OOG
BB 0 GG
0
0

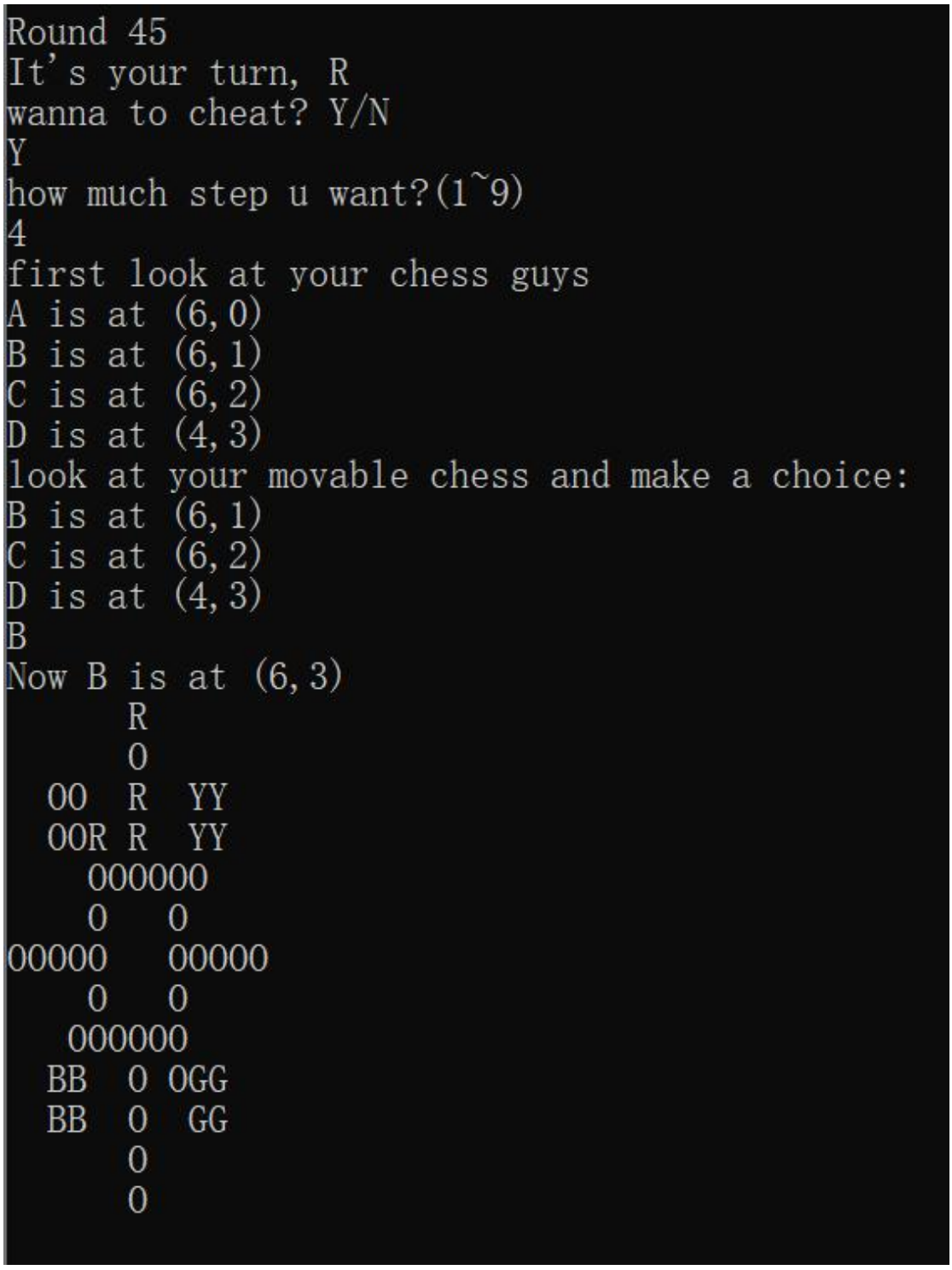
5. 让两个蓝色棋子重叠


Round 10
It's your turn, B
wanna to cheat? Y/N
Y
how much step u want?(1~9)
6
first look at your chess guys
A is at (4,8)
B is at (2,10)
C is at (3,9)
D is at (3,10)
wanna a new chess to start? Y/N
Y
OR O YY
RRO O YY
ROOOOO
OOOOO OOOOO
BBOOOG
OB O OOG
OB O GG


Round 14
It's your turn, B
wanna to cheat? Y/N
Y
how much step u want?(1~9)
1
first look at your chess guys
A is at (4,8)
B is at (3,8)
C is at (3,9)
D is at (3,10)
look at your movable chess and make a choice:
A is at (4,8)
B is at (3,8)
B
Now B is at (4,8)
O
O
OR O YY
RRO O YY
OOOOOO
R O
OOOOO OOOOO
O O
OBOOOG
OB O GOG
OB O OG
O
O

6. 弹回的红色棋子


Round 44
It's your turn, Y
wanna to cheat? Y/N
Y
how much step u want?(1~9)
1
first look at your chess guys
A is at (9,2)
B is at (10,3)
C is at (9,2)
D is at (10,3)
R
R
OO R YY
OOR O YY
OOOOOO
O O
OOOOO OOOOO
O O
OOOOOO
BB O OGG
BB O GG
O
O



最后结束时用一种奇妙的方法（getchar()）暂停终端的显示

```
if(RED->approched==4)

{ printf("Game over, RED is the winner");}

else if(BLUE->approched==4)
```

```
{ printf("Game over, BLUE is the winner");}

else if(GREEN->approched==4)

{ printf("Game over, GREEN is the winner");}

else if(YELLOW->approched==4)

{ printf("Game over, YELLOW is the winner");}

getchar();

return 0;
```

7. 最后展示一下 4 个电脑玩家发癫式游戏

电脑下棋（如果超链接点不开就麻烦 gg 手动看一下了（雾））

以上为核心代码的实现，因为还未加入 GUI 所以坐标未标明，终端不刷新，阅读很痛苦

Phase 2：GUI 的实现，Qt 的自学与运用

环境 Qt creator (6 还是 8 来的，忘了)，MinGW64，qmake

1. 图形界面的设计（略）

2. 重启功能的实现

```
void MainWindow::on_reboot_button_clicked()
{
    // 启动新的进程
    QProcess::startDetached(QApplication::applicationFilePath());
    // 终止当前进程
    QApplication::exit();
}
```

3. 诸如此类为了防止 gg 们错误输入搞破坏的按钮能否按判断

```
void MainWindow::on_getdieres_button_clicked()
{
    srand(time(0));  // 使用当前时间作为种子初始化随机数生成器
                dicepoint = rand() % 6+1;  // 生成 1~6 之间的随机数
                ui->Point_text->clear();
                ui->Point_text->setText(QString::number(dicepoint));
                if(dicepoint==6)
                {
                    if(Player->unstart>0)
                    {
                        ui->newchess->setEnabled(true);
                        ui->message->setText("你可以出发一个棋子\n");
                    }
                }
                else if((Player->unstart+Player->approched)==4)
                {
                    Player=Player->next;
                    ui->message->setText(QString(Player->ID));
                    ui->message->setText("你的回合结束，轮到下一名玩家\n");
                }
                else
                {
                    ui->move_A->setEnabled(true);//再让按钮可以点击
                    ui->move_B->setEnabled(true);
                    ui->move_C->setEnabled(true);
                    ui->move_D->setEnabled(true);
                    ui->message->setText("请选择移动棋子\n");
                }
}
```

4. 核心代码部分，与前述 C 版本差不多，略

5. 打包为可执行文件）

<1>进行 release 版本构建

<2>利用 Qt 自带的命令行工具和 Enigma Virtual Box 进行打包，实现较为简单，略

Last but not least, 助教 gg 和老师新年快乐，身体健康~