



# 彩球游戏

实验报告

1551265 计 1 张伯阳



## 1. 彩球游戏

### 1.1. 操作方法及要求

游戏区域为  $9 \times 9$ ，共有 7 种颜色的彩球随机出现，其中初始状态 5 个。

选择一个彩球并指定一个位置可以判断是否能够将彩球移动过去，移动完成后判断是否有连续的5个彩球，若有则发生消去并获得相应的得分，若没有则随机生成三个彩球。若棋盘被彩球占满，则游戏结束。

游戏共9个功能

1. 内部数组，随机生成初始5个球
  2. 内部数组，随机生成60%的球，寻找移动路径
  3. 内部数组，完整版
  4. 画出 $n \times n$ 的框架（无分隔线），随机显示5个球
  5. 画出 $n \times n$ 的框架（有分隔线），随机显示5个球
  6.  $9 \times 9$ 的框架，50个球，完成一次移动
  7. cmd图形界面完整版
  8. 鼠标键盘同时读入
  9. 鼠标双击加左右键演示
- 按0为退出

菜单项 1:

输入行列后，在规定范围内随机生成五个球的位置，然后打印整个内部数组

- 为方便观察，打印时有球的位置用不同颜色输出

菜单项 2:

输入行列后，在规定范围内随机生成 60%的球的位置，然后输入要移动球的起始坐标及目的坐标，找出将球移动过去的路径（不要求最短，能找到即可）

- 起始位置必须有球，目的位置必须为空
- 生成过程中，如果该位置已经有球，要重新生成

菜单项 3: 结合菜单项 1 和 2，完成一个完整的实现过程（纯内部数组表现形式）

- 球的位置用不同颜色标出
- 连续 5 个则消除，并可以得分（规则可以自定义，demo 的规则是消除数量为  $n$ ，则得分为  $(n-1) \times (n-2)$ ，和之前的游戏并不相同，双五连等情况，交叉点要重复计数）
- 本次移动若得分，则不产生新球，否则会产生三个新球
- 没有任何空位则游戏结束
- 游戏得分不需要记录在文件中，每次开始均从 0 开始即可

菜单项 4:

在 cmd 伪图形界面上画出框架（无分隔线）及初始的五个球

- demo 程序为了看清楚，加了延时，实现时可以不加

- 彩球的颜色、背景色等不需要和 demo 一样
- demo 程序加了设置字体的操作, 该部分内容会下发样例程序 (不需要自己查资料)

菜单项 5:

在 cmd 伪图形界面上画出框架 (有分隔线) 及初始的五个球

- 要求同菜单项 4

菜单项 6:

在 cmd 伪图形界面上显示 60% 的球, 用鼠标选择要移动的球及目的位置, 完成一个移动

- 鼠标操作为: 左键选择, 右键退出本小题
- 鼠标移动过程中, 要实时显示当前移动到  $n \times n$  矩阵的哪个位置 (行: A-I, 列: 1-9), 放在边框线上不算
- 移动过程需要完整的移动轨迹显示, 动画效果必须跨越分隔线
- 如果寻找最佳移动路线不做要求, 只要能找到即可 (另: demo 中的移动路线很奇葩, 但保证了每个位置只移动一次, 你作业中找到的路线不能比这个再差了)

菜单项 7:

在 cmd 伪图形界面上实现完整版的 game

- 要求用菜单项 6
- 右侧显示得分、预告彩球以及统计信息

注: 共用函数中, 均允许调用其它函数, 基本原则就是高效完成程序, 减少冗余代码

尽量保证每个函数 (包括 main) 不要超过 50 行

## 2. 整体设计思路

按照实验的功能顺序分步骤设计, 较分立的功能设单独的函数进行操作, 本着尽量共用函数的准则进行接下来的函数设计. 第1. 4. 5个功能不需要移动和寻路操作, 优先设计. 定义出储存彩球的数组  $a[11][11]$ ,  $a$  数组开设11行11列的目的是为了将边缘行列初始化为-1, 便于判断边界并且可以有效地防止越界. 浪费空间不多却效果显著. 第4. 5个功能需要打印棋盘, 用for循环加数学计算实现循环打印, 使用预先给的在指定位置打印字符串的函数可以简化代码. 下来的功能便需要进行寻路及判断路径的操作, 这也是本题的关键之处. 寻路操作经过考虑了深搜广搜等算法最后使用了递归进行深搜, 并用path数组记录是否走过, 防止走重复的路径, 用arr数组记录走过的路径顺序, 防止进行移动时无法找到合适的移动顺序. 图形界面移动时, 先在分隔线上打印彩球在原彩球位置打印空白, 给延时后在后一个空白处打印彩球, 在原分隔线上还原分隔线, 再给一个延时. 通过循环这样的步骤按照arr给出的路径移动. 之后的得分栏, 彩球预告栏, 彩球分布栏用同4. 5功能的方法打印, 只是要充分考虑传参的问题. 功能8我使用的方法是在检测到除了鼠标的操作进行判断, 若是F3, F4, F5这三个键则进行相应的操作, 否则正常执行鼠标的操作. 功能9屏蔽双击的单击操作直接加上相应的合适的延时即可. 防止两个键同时按下后显示单个按键的操作方法即加上一个flag, 若两键同时按下事件触发, flag置1, 当flag为1时, 不进行判断action的操作及输出, 直到检测到移动或无操作的事件发生, 即两键都松开, flag置0即可完成指定功能.

## 3. 主要功能的实现

1. 4. 5功能直接进行相应的随机生成和打印即可

2. 3. 6. 7功能需要调用move函数进行递归深搜 6. 7函数还要注意鼠标的使用和图形界面的移动的打印 7功能需要打印得分栏, 彩球预告栏, 彩球分布栏.

8. 功能进行鼠标键盘同时输入并在检测到相应的输入后打印黑色将原栏覆盖或重新调用choice37即可 不过实际调试过程发现运行并不稳定, 由于时间紧迫暂未解决.

9功能进行改良鼠标输入, 屏蔽双击的单击操作直接加上相应的合适的延时即可. 防止两个键同时按下后显示单个按键的操作方法即加上一个flag, 若两键同时按下事件触发, flag置1, 当flag为1时, 不进行判断action的操作及输出, 直到检测到移动或无操作的事件发生, 即两键都松开, flag置0即可完成指定功能.

## 4. 调试过程碰到的问题

碰到的问题主要有下:

### 1. 寻路算法的实现与路径的保存

在彩球移动时需要使用寻路算法, 判断是否能够移动到达并规划出路径, 开始时只使用了一个path数组并没有很好地考虑递归函数的实现过程, 发现路径显示为所有的可到达之处, 并且在之后的图形化显示路径的过程中发现无法显示路径, 加入了arr数组保存路径及行走顺序, 经过修改得到正确的dfs函数.

### 2. 函数参数的传递

二维数组的第一个参数是行数(Y值) 第二个参数是列数(X值), 而我的大多函数参数为第一个X第二个Y, 在写代码时就出现了一个函数传参传反了的bug. 导致彩球移动X, Y相反, 界面非常混乱, 并且这个bug非常隐蔽, 我检查了2个小时才查出来. 修正之后程序正常运行.

### 3. 图形界面打印混乱

5功能界面打印时, 因为最开始没有注意到汉字字符与英文字符的差别, 直接采用showch函数进行打印, 直接编译就报错了. 改成showstr函数后完美运行. 还有7功能进行打印小球移动过程中出现的颜色不正确, 位置不正确, 点击出现双圈之前的还原单圈, 之类的一些bug还时有发生.

### 4. 8功能键盘鼠标同时输入不稳定

这个问题就比较恶心了, 网上几乎没有任何可参考的信息, 最开始同时允许两种输入方式时直接两种方式都无法输入, 我在分析和尝试了几种方法后只好选择现在这种相对比较稳定一些的方法, 按F3-F5可能需要多按几次, 鼠标操作也变得不灵敏.

## 5. 心得体会

从这次作业我得到的主要教训有下:

### 1. 写之前多考虑一些之后的可能情况

在写后几个图形界面函数时很明显的遇到了很多次的问题就是之前在写这个函数时只想到了实现眼下的功能所需要的参数进行传入, 后续功能或者完善程序时发现还需要别的参数, 于是又要修改, 这些函数很多还是中间函数, 修改传参就要从前面一个一个函数依次修改, 浪费了一些时间.

### 2. 多加注释 多使用宏定义和常变量

加注释的作用很明显: 让阅读的人更容易看懂, 包括自己之后debug时也会变得容易.

为了达到这个目的, 定义函数名参数名也应该避免使用fun, a, b, c这样的名字, 在一个大程序中这样的参数多了会让阅读变得非常困难. 使用宏定义常变量的目的, 一是为了让阅读变得方便, 一个OK, ERROR肯定要比1, 0看起来更加易懂, -1, -2等特殊返回值也变得更直观. 二是为了让修改方便, 一个使用很多次的常量如果要修改只需要在.h里面修改一次即可实现, 使用宏定义常变量也可以让计算式变得易写易懂不易出错. 因为这方面的练习较少, 我在这次作业中的使用还很不好, 图形界面的计算还是在用常数实现, 以后需要多加注意.

### 3. 保证一个程序的统一

暂且不说养成平时的习惯, 我这次在写这一个程序中也没有做到前后统一. 有些地方使用x, y顺序, 有些地方使用y, x顺序使传参的过程产生了一个很隐蔽的bug, 用了很久的时间去修改.

这次的大作业的完成, 我用了两天时间, 但是纯写代码只用了不到一天时间, 剩下的时间几乎都是在debug, 说明我的代码在完成的过程中没有考虑到可能出现的bug, 和如果出现bug的处理方法, 毕竟bug几乎是难以避免的, 但是如何减少bug, 有意把可能出现bug的地方暴露出来和在出现了bug后将debug时间减少到最短 比写代码还要重要. 这是以后要注意的, 并不是写代码写得快一些就可以完事.

装

订

线

## 6. 附件：源程序

/\* 1551265 计1 张伯阳 \*/

```
90-b2.h
#pragma once
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <iomanip>
#include <ctime>
#include <cmath>
#include <cstdio>
#include <stdlib.h>
#include <cstring>
#include <conio.h>
#include <Windows.h>
#include "cmd_console_tools.h"
#include <profileapi.h>
#define PRINT_X_0 3
#define PRINT_Y_0 3
#define OK 1
#define ERROR 0
#define initial 5
#define proportion 0.6
#define pertime 20
#define length 100
#define hight 40
#define F3 61
#define F4 62
#define F5 63
#define times 300
const char str_[3] = "◎", str0[3] = "○", str1[3] = "—", str2[3] = "┐", str3[3] = "└", str4[3] = "┌", str5[3] = "└", str6[3] = "┐", str7[3] = "┐", str8[3] = "└", str9[3] = "└", str10[3] = "┐", str11[3] = "┐", str12[3] = "—";
using namespace std;
const HANDLE hout = GetStdHandle(STD_OUTPUT_HANDLE); //取标准输出设备对应的句柄
const HANDLE hin = GetStdHandle(STD_INPUT_HANDLE); //取标准输入设备对应的句柄
void readl_1(int *choice);
void readl_2(int *row, int *line);
void clr(int a[][11], const int row, const int line);
void random_gene(int a[][11], const int row, const int line, const int num);
void printarr(int a[][11], int row, int line);
void choice1(int a[][11], int row, int line);
void choice2(int a[][11], const int choice, const int row, const int line);
int move(int a[][11], int path[][11], int arr[][11], int r1, int l1, int r2, int l2);
void choice45(int a[][11], const int choice, int row, int line);
int read2(int a[][11], int b[], const int choice, const int row, const int line);
int dfs(int a[][11], int path[][11], int arr[][11], int r1, int l1, int r2, int l2, const char c);
bool fun(int a[][11], int path[][11], int arr[][11], int row, int line, int *r1, int *l1, int *r2, int *l2);
void choice37(int a[][11], const int choice, int row, int line);
void random(int b[]);
void print2(int a[][11], int arr[][11], int row, int line);
int judge1(int a[][11], int row, int line, int r2, int l2);
void print45_1(int a[][11], const int choice, int row, int line);
void print45_2(int a[][11], const int choice, int row, int line);
```

装

订

线

```
void print45_3(int a[][11], const int choice, int row, int line);
void print7(int a[][11], int b[], int dead[], int last[], int score, int row, int line);
void print7_1(int a[][11], int score, int line);
void print7_2(int a[][11], int b[], int line);
void print7_3(int a[][11], int dead[], int last[], int row, int line);
int judge2(int a[][11], int row, int line, int r2, int l2);
void clrpa(int path[][11], int arr[][11]);
int mouse(int a[][11], int choice, int b[], int dead[], int last[], int choose[], int choose_[], int
&x, int &y, int row, int line);
int judge_left(int a[][11], int choose[], int choose_[], int score[], int x, int y, int row, int line,
int &is_move);
int judge_move(int a[][11], int choose[], int score[], int x, int y, int row, int line, int &is_move);
void movepic(int a[][11], int arr[][11], int n, int choose[], int x2, int y2, int row, int line);
void swap(int *a, int *b);
void choice6(int a[][11], int row, int line);
void mouse_test();
int read_mouse_and_key(const HANDLE hin, int &X, int &Y, const int enable_read_mouse_moved);
```

90-b2-main.cpp

```
#include "90-b2.h"
#include "cmd_console_tools.h"
int main()
{
    int a[11][11], choice, row, line;
    setcolor(hout, 0, 7);
    srand((unsigned)time(NULL));
    while (1)
    {
        setconsoleborder(hout, length, hight, hight);
        readl_1(&choice);
        if (!choice)
            break;
        if (choice != 9)
        {
            readl_2(&row, &line);
            clr(a, row, line);
            random_gene(a, row, line, initial);
        }
        if (choice == 1)
            choicel(a, row, line);
        if (choice == 2)
            choice2(a, choice, row, line);
        if (choice == 3 || choice == 7 || choice == 8)
            choice37(a, choice, row, line);
        if (choice == 4 || choice == 5)
            choice45(a, choice, row, line);
        if (choice == 6)
            choice6(a, row, line);
        if (choice == 9)
            mouse_test();
        system("cls");
    }
}

void readl_1(int *choice)
{

```

装

订

线

```

while (1)
{
    cout << "-----" << endl;
    cout << "1. 内部数组，随机生成初始5个球\n2. 内部数组，随机生成60%的球，寻找移动路径\n3. 内部
数组，完整版\n4. 画出n*n的框架（无分隔线），随机显示5个球\n5. 画出n*n的框架（有分隔线），随机显示5个
球\n6. 9 * 9的框架，50个球，完成一次移动\n7. cmd图形界面完整版\n8. 鼠标键盘同时读入(要多按几次)\n9. 鼠
标双击加左右键演示\n0. 退出\n" << endl;
    cout << "-----" << endl;
    cin >> *choice;
    if (!cin || *choice < 0 || *choice>9)
    {
        cin.clear();
        cin.sync();
        cin.ignore(1024, '\n');
    }
    else
        break;
}

}

void readl_2(int *row, int *line)
{
    while (1)
    {
        cout << "请输入行数(7-9): ";
        cin >> *row;
        if (!cin || *row>9 || *row<7)
        {
            cin.clear();
            cin.sync();
            cin.ignore(1024, '\n');
        }
        else
            break;
    }
    while (1)
    {
        cout << "请输入列数(7-9): ";
        cin >> *line;
        if (!cin || *line>9 || *line<7)
        {
            cin.clear();
            cin.sync();
            cin.ignore(1024, '\n');
        }
        else
            break;
    }
}

}

90-b2-base.cpp
#include "90-b2.h"
#include "cmd_console_tools.h"
void printarr(int a[][11], int row, int line)
{
    cout << " ";

```



```

for (int i = 1; i <= line; i++)
    cout << " " << i;
cout << endl << "---+";
for (int i = 1; i <= line; i++)
    cout << "---";
for (int r = 1; r <= row; r++)
{
    cout << endl << char('A' + r - 1) << " |";
    for (int l = 1; l <= line; l++)
    {
        setcolor(hout, 0, 7);
        cout << " ";
        if (a[r][l])
            setcolor(hout, 6, a[r][l] / 7 + a[r][l] - 1);
        cout << a[r][l];
    }
    setcolor(hout, 0, 7);
}
cout << endl;
}

void choice1(int a[][11], int row, int line)
{
    printarr(a, row, line);
    cout << "本小题结束, 请输入回车键继续..." << endl;
    while (_getch() != '\r');
}

void choice2(int a[][11], const int choice, const int row, const int line)
{
    int b[3];
    random(b);
    random_gene(a, row, line, int(proportion*row*line - 5));
    printarr(a, row, line);
    read2(a, b, choice, row, line);
}

int read2(int a[][11], int b[], const int choice, const int row, const int line)
{
    srand((unsigned)time(NULL));
    int r1, r2, l1, l2, path[11][11], arr[11][11], score = 0, num = 0;
    cout << "下3个彩球的颜色分别是: " << b[0] << " " << b[1] << " " << b[2] << endl << "请以字母+
数字形式[例: c2]输入要移动球的矩阵坐标: ";
    if (fun(a, path, arr, row, line, &r1, &l1, &r2, &l2))
    {
        if (choice == 2)
            print2(a, arr, row, line);
        else
        {
            a[r2][l2] = a[r1][l1];
            a[r1][l1] = 0;
            score = judge2(a, row, line, r2, l2);
        }
    }
    if (choice == 2)
    {
        cout << endl << "本小题结束, 请输入回车键继续...";
        while (_getch() != '\r');
    }
}

```

```

        return OK;
    }
    return score;
}

void print2(int a[][11], int arr[][11], int row, int line)
{
    int x, y;
    system("cls");
    for (int i = 1; i < row; i++)
    {
        cout << endl;
        for (int j = 1; j <= line; j++)
        {
            gotoxy(hout, x, y);
            if (!arr[i][j])
            {
                cout << arr[i][j] << " ";
                gotoxy(hout, x, y + 12);
                cout << a[i][j] << " ";
                gotoxy(hout, x + 2, y);
            }
            else
            {
                cout << "* ";
                gotoxy(hout, x, y + 12);
                setcolor(hout, 6, a[i][j]);
                cout << a[i][j];
                setcolor(hout, 0, 7);
                cout << " ";
                gotoxy(hout, x + 2, y);
            }
        }
    }
}

bool fun(int a[][11], int path[][11], int arr[][11], int row, int line, int *r1, int *l1, int *r2,
int *l2)
{
    char c[10];
    while (1)
    {
        cin >> c;
        if (c[0]<'a' || c[0]>'a' + row - 1 || c[1]<'1' || c[1]>line + '0')
            cout << "输入错误, 请重新输入.";
        else if (!a[*r1 = int(c[0] - 'a' + 1)][*l1 = int(c[1] - '0')])
            cout << "起始位置为空, 请重新输入.";
        else
            break;
    }
    cout << "输入为" << char(c[0] + 'A' - 'a') << "行" << c[1] << "列" << endl << "请以字母+数字
形式[例: c2]输入要移动球的目的坐标: ";
    while (1)
    {
        cin >> c;
        if (c[0]<'a' || c[0]>'a' + row - 1 || c[1]<'1' || c[1]>line + '0')
            cout << "输入错误, 请重新输入.";
        else if (a[*r2 = int(c[0] - 'a' + 1)][*l2 = int(c[1] - '0')])

```

```

        cout << "目标位置非空，请重新输入。";
    else
        break;
}
clrpa(path, arr);
if (move(a, path, arr, *r1, *l1, *r2, *l2))
    return OK;
cout << "无法找到移动路径";
return ERROR;
}

90-b2-console.cpp
#include "90-b2.h"
#include "cmd_console_tools.h"
void showin(HANDLE hout, int X, int Y, int A, const int bg_color, const int fg_color)
{
    gotoxy(hout, X, Y);
    setcolor(hout, bg_color, fg_color);
    cout << A;
}

void showdb(HANDLE hout, int X, int Y, double A, const int bg_color, const int fg_color)
{
    gotoxy(hout, X, Y);
    setcolor(hout, bg_color, fg_color);
    printf("%.2f", A);
}

void choice45(int a[][11], const int choice, int row, int line)
{
    cout << "初始数组:" << endl;
    printarr(a, row, line);
    cout << "按回车键显示图形...";
    while (_getch() != '\r');
    setfontsize(hout, L"新宋体", 24);
    if (choice == 4)
        setconsoleborder(hout, 2 * line + 8, row + 8, row + 8);
    else
        setconsoleborder(hout, 4 * line + 8, 2 * row + 8, 2 * row + 8);
    print45_1(a, choice, row, line);
    if (choice == 4)
        gotoxy(hout, 1, row + 4);
    else
        gotoxy(hout, 1, 2 * row + 4);
    cout << "本小题结束，请输入回车键继续...";
    while (_getch() != '\r');
    setfontsize(hout, L"新宋体", 18);
}

void choice6(int a[][11], int row, int line)
{
    int score0 = 0, x, y, score = 0, choose[2] = { 0 }, choose_[2] = { 0 }, b[3], dead[8] = { 0 },
    last[8] = { 0 };
    setconsoleborder(hout, 4 * line + 6, 2 * row + 6, 2 * row + 6);
    setfontsize(hout, L"新宋体", 24);
    print45_1(a, 6, row, line);
    mouse(a, 6, b, dead, last, choose, choose_, x, y, row, line);
}

```

装

订

线

```

setcolor(hout, 0, 7);
setcursor(hout, CURSOR_VISIBLE_NORMAL); //打开光标
gotoxy(hout, 1, 2 * row + 4);
cout << "本小题结束，请输入回车键继续...";
while (_getch() != '\r');
setfontsize(hout, L"新宋体", 18);
}

void print45_1(int a[][11], const int choice, int row, int line)
{
    showstr(hout, 1, 1, str2, 15, 0);
    if (choice == 4)
    {
        for (int i = 1; i <= line; i++)
        {
            showstr(hout, i * 2 + 1, 1, str1, 15, 0);
            showstr(hout, i * 2 + 1, row + 2, str1, 15, 0);
        }
        showstr(hout, line * 2 + 3, row + 2, str5, 15, 0);
        showstr(hout, line * 2 + 3, 1, str6, 15, 0);
    }
    else
    {
        for (int i = 1; i <= 2 * line - 1; i++)
        {
            if (i % 2)
            {
                showstr(hout, i * 2 + 1, 1, str1, 15, 0);
                showstr(hout, i * 2 + 1, 2 * row + 1, str1, 15, 0);
            }
            else
            {
                showstr(hout, i * 2 + 1, 1, str3, 15, 0);
                showstr(hout, i * 2 + 1, 2 * row + 1, str4, 15, 0);
            }
        }
        showstr(hout, line * 4 + 1, 2 * row + 1, str5, 15, 0);
        showstr(hout, line * 4 + 1, 1, str6, 15, 0);
    }
    print45_2(a, choice, row, line);
    print45_3(a, choice, row, line);
}

void print45_2(int a[][11], const int choice, int row, int line)
{
    if (choice == 4)
    {
        for (int i = 1; i <= row; i++)
        {
            showstr(hout, 1, i + 1, str7, 15, 0);
            showstr(hout, line * 2 + 3, i + 1, str7, 15, 0);
        }
        showstr(hout, 1, row + 2, str10, 15, 0);
    }
    else
    {
        for (int i = 1; i <= 2 * row - 1; i++)

```

```

        {
            if (i % 2)
            {
                showstr(hout, 1, i + 1, str7, 15, 0);
                showstr(hout, line * 4 + 1, i + 1, str7, 15, 0);
            }
            else
            {
                showstr(hout, 1, i + 1, str8, 15, 0);
                showstr(hout, line * 4 + 1, i + 1, str9, 15, 0);
            }
        }
        showstr(hout, 1, 2 * row + 1, str10, 15, 0);
    }
}

void print45_3(int a[][11], const int choice, int row, int line)
{
    if (choice == 4)
        for (int i = 1; i <= row; i++)
            for (int j = 1; j <= line; j++)
                if (!a[i][j])
                    showstr(hout, 2 * j + 1, i + 1, str0, 15, 15);
                else
                    showstr(hout, 2 * j + 1, i + 1, str0, a[i][j] - 1, 15);
    else
        for (int i = 1; i < 2 * row; i++)
            for (int j = 1; j < 2 * line; j++)
            {
                gotoxy(hout, 2 * j + 1, i + 1);
                if (i % 2)
                {
                    if (j % 2)
                    {
                        if (!a[(i + 1) / 2][(j + 1) / 2])
                            showstr(hout, 2 * j + 1, i + 1, str0, 15, 15);
                        else
                            showstr(hout, 2 * j + 1, i + 1, str0, a[(i + 1) / 2][(j + 1) / 2]
- 1, 15);
                    }
                    setcolor(hout, 15, 0);
                }
                else
                    cout << " | ";
            }
            else if (j % 2)
                cout << " — ";
            else
                cout << " + ";
        }
    setcolor(hout, 0, 7);
}

int mouse(int a[][11], int choice, int b[], int dead[], int last[], int choose[], int choose_[], int
&x, int &y, int row, int line)
{
    enable_mouse(hin);
    int X = 0, Y = 0, is_move = 0, action, score[1] = { 0 }, flagf3 = 0, flagf5 = 0;
    setcursor(hout, CURSOR_INVISIBLE);    //关闭光标

```

```

while (1)
{
    if (choice == 7 || choice == 6)
        action = read_mouse(hin, X, Y);
    else
        action = read_mouse_and_key(hin, X, Y, 1);
    if (action == MOUSE_RIGHT_BUTTON_CLICK)
        return -2;
    if (action == F3)
    {
        flagf3 = !flagf3;
        if (flagf3)
            for (int i = 0; i < 3; i++)
                for (int j = 0; j <= 6; j++)
                    showstr(hout, line * 4 + 5 + 2 * j, 5 + i, str0, 0, 0);
        else
            print7_2(a, b, line);
    }
    if (action == F4)
        return -3;
    if (action == F5)
    {
        flagf5 = !flagf5;
        if (flagf5)
            for (int i = 0; i <= 9; i++)
                for (int j = 0; j <= 12; j++)
                    showstr(hout, line * 4 + 5 + 2 * j, 9 + i, str0, 0, 0);
        else
            print7_3(a, dead, last, row, line);
    }
    if (X < line * 4 + 1 && Y < 2 * row + 1 && (!(X % 4) || X % 4 == 3) && !(Y % 2))
    {
        x = (X + 1) / 4, y = Y / 2;
        if (action == MOUSE_LEFT_BUTTON_CLICK)
        {
            int b = judge_left(a, choose, choose_, score, x, y, row, line, is_move);
            choose_[0] = choose[0], choose_[1] = choose[1];
            if (!b)
            {
                gotoxy(hout, 0, 2 * row + 3);
                cout << "无法移动到该点";
            }
            else if (b == 1)
            {
                choose[0] = x, choose[1] = y;
                return score[0];
            }
            else if (b == -1)
                choose[0] = x, choose[1] = y;
        }
        else if (action == MOUSE_ONLY_MOVED)
        {
            setcolor(hout, 0, 15);
            gotoxy(hout, 0, 2 * row + 3);
            cout << "[当前光标]: " << char(y + 'A' - 1) << " 行   Y: " << x << " 列";
        }
    }
}

```

```

    }
}

int judge_left(int a[][11], int choose[], int choose_[], int score[], int x, int y, int row, int line,
int &is_move)
{
    if (choose[0])
        if (a[y][x])
        {
            choose[0] = x, choose[1] = y;
            showstr(hout, (choose[0] - 1) * 4 + 3, 2 * choose[1], str_, a[y][x] - 1, 15);
            if (!is_move)
                showstr(hout, (choose_[0] - 1) * 4 + 3, 2 * choose_[1], str0,
a[choose_[1]][choose_[0]] - 1, 15);
            return -1;
        }
    else
        return judge_move(a, choose, score, x, y, row, line, is_move);
    else if (a[y][x])
    {
        choose[0] = x, choose[1] = y;
        showstr(hout, (choose[0] - 1) * 4 + 3, 2 * choose[1], str_, a[y][x] - 1, 15);
        return -1;
    }
    return -2;
}

int judge_move(int a[][11], int choose[], int score[], int x, int y, int row, int line, int &is_move)
{
    int path[11][11], arr[11][11], n;
    clrpa(path, arr);
    n = move(a, path, arr, choose[1], choose[0], y, x);
    if (arr[choose[1]][choose[0]] = n)
    {
        is_move = 1;
        movepic(a, arr, n, choose, x, y, row, line);
        if (0)
        {
            gotoxy(hout, 0, 13);
            cout << "球的分布" << endl;
            for (int i = 1; i <= row; i++)
            {
                for (int j = 1; j <= line; j++)
                    cout << a[i][j] << ' ';
                cout << endl;
            }
            while (_getch() != '\r');
        }
        score[0] = judge2(a, row, line, y, x);
    }
    else
    {
        is_move = 0;
        return ERROR;
    }
    return OK;
}

```

```

void movepic(int a[][11], int arr[][11], int n, int choose[], int x2, int y2, int row, int line)
{
    for (int x = 0, y = 0, x_ = choose[0], y_ = choose[1], i = n; i >= 1; i--)
    {
        for (int j = 1; j <= row; j++)
            for (int k = 1; k <= line; k++)
                if (arr[j][k] == i)
                    y = j, x = k;

        if (!x)
            break;
        else if (x == x_ + 1)
        {
            showstr(hout, (x_ - 1) * 4 + 3, 2 * y, str0, 15, 15);
            showstr(hout, (x_ - 1) * 4 + 5, 2 * y, str_, a[y][x_] - 1, 15);
            Sleep(pertime);
            showstr(hout, (x_ - 1) * 4 + 5, 2 * y, str11, 15, 0);
            showstr(hout, (x_ - 1) * 4 + 7, 2 * y, str_, a[y][x_] - 1, 15);
            Sleep(pertime);
            swap(a[y] + x_ + 1, a[y] + x_);
        }
        else if (x == x_ - 1)
        {
            showstr(hout, (x_ - 1) * 4 + 3, 2 * y, str0, 15, 15);
            showstr(hout, (x_ - 1) * 4 + 1, 2 * y, str_, a[y][x_] - 1, 15);
            Sleep(pertime);
            showstr(hout, (x_ - 1) * 4 + 1, 2 * y, str11, 15, 0);
            showstr(hout, (x_ - 1) * 4 - 1, 2 * y, str_, a[y][x_] - 1, 15);
            Sleep(pertime);
            swap(a[y] + x_ - 1, a[y] + x_);
        }
        else if (y == y_ + 1)
        {
            showstr(hout, (x - 1) * 4 + 3, 2 * y_, str0, 15, 15);
            showstr(hout, (x - 1) * 4 + 3, 2 * y_ + 1, str_, a[y_][x] - 1, 15);
            Sleep(pertime);
            showstr(hout, (x - 1) * 4 + 3, 2 * y_ + 1, str12, 15, 0);
            showstr(hout, (x - 1) * 4 + 3, 2 * y_ + 2, str_, a[y_][x] - 1, 15);
            Sleep(pertime);
            swap(&a[y_ + 1][x], a[y_] + x);
        }
        else if (y == y_ - 1)
        {
            showstr(hout, (x - 1) * 4 + 3, 2 * y_, str0, 15, 15);
            showstr(hout, (x - 1) * 4 + 3, 2 * y_ - 1, str_, a[y_][x] - 1, 15);
            Sleep(pertime);
            showstr(hout, (x - 1) * 4 + 3, 2 * y_ - 1, str12, 15, 0);
            showstr(hout, (x - 1) * 4 + 3, 2 * y_ - 2, str_, a[y_][x] - 1, 15);
            Sleep(pertime);
            swap(a[y_ - 1] + x, a[y_] + x);
        }
        x_ = x, y_ = y;
    }
    choose[0] = x2, choose[1] = y2;
}

void print7(int a[][11], int b[], int dead[], int last[], int score, int row, int line)
{
    print7_1(a, score, line);
}

```



```

    print7_2(a, b, line);
    print7_3(a, dead, last, row, line);
}
/*打印得分栏*/
void print7_1(int a[][11], int score, int line)
{
    char ch[7] = "得分:";
    showstr(hout, line * 4 + 5, 1, str2, 15, 0);
    showstr(hout, line * 4 + 5 + 12, 1, str6, 15, 0);
    showstr(hout, line * 4 + 5, 3, str10, 15, 0);
    showstr(hout, line * 4 + 5 + 12, 3, str5, 15, 0);
    showstr(hout, line * 4 + 5 + 12, 2, str7, 15, 0);
    showstr(hout, line * 4 + 5, 2, str7, 15, 0);
    for (int i = 0; i < 5; i++)
    {
        showstr(hout, line * 4 + 7 + 2 * i, 1, str1, 15, 0);
        showstr(hout, line * 4 + 7 + 2 * i, 3, str1, 15, 0);
    }
    for (int i = 0; i < 5; i++)
        showstr(hout, line * 4 + 12 + i, 2, " ", 15, 15);
    showstr(hout, line * 4 + 7, 2, ch, 15, 0);
    showin(hout, line * 4 + 12, 2, score, 15, 0);
}
/*打印彩球预告栏*/
void print7_2(int a[][11], int b[], int line)
{
    showstr(hout, line * 4 + 5, 5, str2, 15, 0);
    showstr(hout, line * 4 + 5 + 12, 5, str6, 15, 0);
    showstr(hout, line * 4 + 5, 7, str10, 15, 0);
    showstr(hout, line * 4 + 5 + 12, 7, str5, 15, 0);
    showstr(hout, line * 4 + 5 + 12, 6, str7, 15, 0);
    showstr(hout, line * 4 + 5, 6, str7, 15, 0);
    showstr(hout, line * 4 + 5 + 4, 6, str11, 15, 0);
    showstr(hout, line * 4 + 5 + 8, 6, str11, 15, 0);
    for (int i = 0; i < 5; i++)
    {
        if (i % 2)
        {
            showstr(hout, line * 4 + 7 + 2 * i, 5, str3, 15, 0);
            showstr(hout, line * 4 + 7 + 2 * i, 7, str4, 15, 0);
        }
        else
        {
            showstr(hout, line * 4 + 7 + 2 * i, 5, str1, 15, 0);
            showstr(hout, line * 4 + 7 + 2 * i, 7, str1, 15, 0);
        }
    }
    for (int i = 0; i < 3; i++)
        showstr(hout, line * 4 + 7 + 4 * i, 6, str0, b[i] - 1, 15);
}
/*打印彩球比例分布栏*/
void print7_3(int a[][11], int dead[], int last[], int row, int line)
{
    char ch1[2] = ":"; ch2[3] = "/((", ch3[7] = ") del-";
    int col[8] = { 0 };
    double cent[8] = { 0 };

```

```

for (int i = 1; i <= row; i++)
    for (int j = 1; j <= line; j++)
        col[a[i][j]]++;
for (int i = 0; i < 8; i++)
    cent[i] = col[i] * 1.0 / row / line;
if (last[0] - col[0] < 0)
    if (col[0] != row*line - 5)
        for (int i = 1; i < 8; i++)
            dead[i] += (last[i] - col[i]);
for (int i = 0; i < 8; i++)
    last[i] = col[i];
showstr(hout, line * 4 + 5, 9, str2, 15, 0);
showstr(hout, line * 4 + 5 + 24, 9, str6, 15, 0);
showstr(hout, line * 4 + 5, 18, str10, 15, 0);
showstr(hout, line * 4 + 5 + 24, 18, str5, 15, 0);
for (int i = 1; i < 12; i++)
{
    showstr(hout, line * 4 + 5 + 2 * i, 9, str1, 15, 0);
    showstr(hout, line * 4 + 5 + 2 * i, 18, str1, 15, 0);
}
for (int i = 1; i < 9; i++)
{
    showstr(hout, line * 4 + 5 + 24, 9 + i, str7, 15, 0);
    showstr(hout, line * 4 + 5, 9 + i, str7, 15, 0);
}
for (int i = 0; i < 8; i++)
    for (int j = 0; j < 5; j++)
        showstr(hout, line * 4 + 23 + j, 10 + i, " ", 15, 15);
for (int i = 0; i < 8; i++)
{
    if (!i)
        showstr(hout, line * 4 + 7, 10 + i, str0, 15, 15);
    else
        showstr(hout, line * 4 + 7, 10 + i, str0, i - 1, 15);
    showstr(hout, line * 4 + 9, 10 + i, ch1, 15, 0);
    if (col[i] < 10)
    {
        showin(hout, line * 4 + 10, 10 + i, 0, 15, 0);
        showin(hout, line * 4 + 11, 10 + i, col[i], 15, 0);
    }
    else
    {
        showin(hout, line * 4 + 10, 10 + i, col[i], 15, 0);
        showstr(hout, line * 4 + 12, 10 + i, ch2, 15, 0);
        showdb(hout, line * 4 + 14, 10 + i, cent[i], 15, 0);
        showstr(hout, line * 4 + 18, 10 + i, ch3, 15, 0);
        showin(hout, line * 4 + 24, 10 + i, dead[i], 15, 0);
    }
}
}
void mouse_test()
{
    system("cls");
    int X = 0, Y = 0, flagrl = 0;
    int action;
    int loop = 1;

    enable_mouse(hin);

```

```

/* 打印初始光标位置[0,0] */
setcursor(hout, CURSOR_INVISIBLE); //关闭光标
cout << "可测试左键单/双击,右键单/双击,左右键同时单击五种,其中右键双击结束鼠标测试" << endl;
cout << "[当前鼠标位置] X:0 Y:0"; //打印初始鼠标位置

while (loop) {
    /* 读鼠标,返回值为下述操作中的某一种,当前鼠标位置在<X,Y>处 */
    action = read_mouse(hin, X, Y);

    /* 转到第1行进行打印 */
    gotoxy(hout, 0, 1);
    cout << "[当前光标位置] X:" << setw(2) << X << " Y:" << setw(2) << Y << " 操作:";
    if (action != MOUSE_LEFT_BUTTON_CLICK && action != MOUSE_RIGHT_BUTTON_CLICK /*&& action !=
MOUSE_LEFTRIGHT_BUTTON_CLICK*/)
        flagrl = 0;
    if (!flagrl)
        switch (action) {
            case MOUSE_LEFT_BUTTON_CLICK: //按下左键
                cout << "按下左键" << endl;
                Sleep(times);
                showch(hout, X, Y, '1'); //在鼠标指针位置显示1
                break;
            case MOUSE_LEFT_BUTTON_DOUBLE_CLICK: //双击左键
                cout << "双击左键" << endl;
                showch(hout, X, Y, '2'); //在鼠标指针位置显示2
                break;
            case MOUSE_RIGHT_BUTTON_CLICK: //按下右键
                cout << "按下右键" << endl;
                Sleep(times);
                showch(hout, X, Y, '3'); //在鼠标指针位置显示3
                break;
            case MOUSE_RIGHT_BUTTON_DOUBLE_CLICK: //双击右键
                cout << "双击右键" << endl;
                showch(hout, X, Y, '4'); //在鼠标指针位置显示4
                loop = 0;
                break;
            case MOUSE_LEFTRIGHT_BUTTON_CLICK: //同时按下左右键
                cout << "同时按下左右键" << endl;
                showch(hout, X, Y, '5'); //在鼠标指针位置显示5
                flagrl = 1;
                break;
            case MOUSE_ONLY_MOVED:
                cout << "移动" << endl;
                //showch(hout, X, Y, '*');不打印任何内容
                break;
            case MOUSE_NO_ACTION:
            default:
                cout << "其它操作" << endl;
                showch(hout, X, Y, '0'); //在鼠标指针位置显示0
                break;
        } //end of switch
    } //end of while(1)

    setcursor(hout, CURSOR_VISIBLE_NORMAL); //打开光标
    gotoxy(hout, 1, 30);
    cout << "本小题结束,请输入回车键继续...";
    while (_getch() != '\r');
}

```

```

}
int read_mouse_and_key(const HANDLE hin, int &X, int &Y, const int enable_read_mouse_moved)
{
    INPUT_RECORD    mouseRec;
    DWORD           res;
    COORD            crPos;

    while (1) {
        /* 从hin中读输入状态（包括鼠标、键盘等） */
        ReadConsoleInput(hin, &mouseRec, 1, &res);
        if (mouseRec.EventType != MOUSE_EVENT)
        {
            int a = _getch();
            if (!a)
            {
                a = _getch();
                if (a == F3)
                    return F3;
                if (a == F4)
                    return F4;
                if (a == F5)
                    return F5;
            }
            else
                continue;
        }
        ReadConsoleInput(hin, &mouseRec, 1, &res);
        /* 从返回中读鼠标指针当前的坐标 */
        crPos = mouseRec.Event.MouseEvent.dwMousePosition;
        X = crPos.X;
        Y = crPos.Y;

        if (enable_read_mouse_moved && mouseRec.Event.MouseEvent.dwEventFlags == MOUSE_MOVED)
            return MOUSE_ONLY_MOVED;

        if (mouseRec.Event.MouseEvent.dwButtonState == (FROM_LEFT_1ST_BUTTON_PRESSED |
        RIGHTMOST_BUTTON_PRESSED)) { //同时按下左右键
            return MOUSE_LEFTRIGHT_BUTTON_CLICK;
        }
        else if (mouseRec.Event.MouseEvent.dwButtonState == FROM_LEFT_1ST_BUTTON_PRESSED) { //按下左
        下左键
            if (mouseRec.Event.MouseEvent.dwEventFlags == DOUBLE_CLICK)
                return MOUSE_LEFT_BUTTON_DOUBLE_CLICK;
            else
                return MOUSE_LEFT_BUTTON_CLICK;
        }
        else if (mouseRec.Event.MouseEvent.dwButtonState == RIGHTMOST_BUTTON_PRESSED) { //按下右
        键
            if (mouseRec.Event.MouseEvent.dwEventFlags == DOUBLE_CLICK)
                return MOUSE_RIGHT_BUTTON_DOUBLE_CLICK;
            else
                return MOUSE_RIGHT_BUTTON_CLICK;
        }
        else //忽略其它按键操作（如果需要滚轮，则判断 FROM_LEFT_2ND_BUTTON_PRESSED）
            ;
    } //end of while(1)
    return -1;
}

```

```

}

90-b2-tools.cpp
#include "90-b2.h"
#include "cmd_console_tools.h"

void clr(int a[][11], const int row, const int line)
{
    for (int i = 1; i < 11; i++)
        for (int j = 1; j < 11; j++)
            a[i][j] = -1;
    for (int i = 1; i <= row; i++)
        for (int j = 1; j <= line; j++)
            a[i][j] = 0;
}

void random_gene(int a[][11], const int row, const int line, const int num)
{
    srand((unsigned)time(NULL));
    for (int i = 0; i < num; i++)
        while (1)
            if (!a[b = (rand()*rand()) % row + 1][c = rand() % line + 1])//确认为空位
            {
                a[b][c] = (rand() / 2) % 7 + 1;//避免取得rand相同
                break;
            }
}

int move(int a[][11], int path[][11], int arr[][11], int r1, int l1, int r2, int l2)
{
    path[r1][l1] = 1;
    arr[r1][l1] = 1;
    int b;
    if (b = dfs(a, path, arr, r1, l1, r2, l2, 'l'))
        return b;
    if (b = dfs(a, path, arr, r1, l1, r2, l2, 'r'))
        return b;
    if (b = dfs(a, path, arr, r1, l1, r2, l2, 'd'))
        return b;
    return dfs(a, path, arr, r1, l1, r2, l2, 'u');
}

int dfs(int a[][11], int path[][11], int arr[][11], int r1, int l1, int r2, int l2, const char c)
{
    int t;
    if (r1 == r2 && l1 == l2)
    {
        arr[r1][l1] = 1;
        return OK;
    }
    if (c != 'r' && !a[r1 - 1][l1] && !path[r1 - 1][l1])/*若左侧可以走并且没走过*/
    {
        path[r1][l1] = 1;
        if (t = dfs(a, path, arr, r1 - 1, l1, r2, l2, 'l'))
        {
            arr[r1][l1] = t + 1;
            return t + 1;
        }
    }
}

```

```

}
if (c != 'l' && !a[r1 + 1][l1] && !path[r1 + 1][l1])/*若右侧可以走并且没走过*/
{
    path[r1][l1] = 1;
    if (t = dfs(a, path, arr, r1 + 1, l1, r2, l2, 'r'))
    {
        arr[r1][l1] = t + 1;
        return t + 1;
    }
}
if (c != 'u' && !a[r1][l1 + 1] && !path[r1][l1 + 1])/*若下侧可以走并且没走过*/
{
    path[r1][l1] = 1;
    if (t = dfs(a, path, arr, r1, l1 + 1, r2, l2, 'd'))
    {
        arr[r1][l1] = t + 1;
        return t + 1;
    }
}
if (c != 'd' && !a[r1][l1 - 1] && !path[r1][l1 - 1])/*若上侧可以走并且没走过*/
{
    path[r1][l1] = 1;
    if (t = dfs(a, path, arr, r1, l1 - 1, r2, l2, 'u'))
    {
        arr[r1][l1] = t + 1;
        return t + 1;
    }
}
return ERROR;
}
int judge1(int a[][11], int row, int line, int r2, int l2)
{
    int ir = 0, il = 0, i1 = 0, i2 = 0, flag = 0, score = 0;
    for (int j = 1; a[r2 + j][l2] == a[r2][l2]; j++, ir++);
    for (int j = 1; a[r2 - j][l2] == a[r2][l2]; j++, ir++);
    for (int j = 1; a[r2][l2 + j] == a[r2][l2]; j++, il++);
    for (int j = 1; a[r2][l2 - j] == a[r2][l2]; j++, il++);
    for (int j = 1; a[r2 - j][l2 - j] == a[r2][l2]; j++, il++);
    for (int j = 1; a[r2 + j][l2 + j] == a[r2][l2]; j++, il++);
    for (int j = 1; a[r2 + j][l2 - j] == a[r2][l2]; j++, i2++);
    for (int j = 1; a[r2 - j][l2 + j] == a[r2][l2]; j++, i2++);
    if (ir >= 4)
    {
        for (int j = 1; a[r2 + j][l2] == a[r2][l2]; j++)
            a[r2 + j][l2] = 0;
        for (int j = 1; a[r2 - j][l2] == a[r2][l2]; j++)
            a[r2 - j][l2] = 0;
        flag = 1;
        score += ir*(ir - 1);
    }
    if (il >= 4)
    {
        for (int j = 1; a[r2][l2 + j] == a[r2][l2]; j++)
            a[r2][l2 + j] = 0;
        for (int j = 1; a[r2][l2 - j] == a[r2][l2]; j++)
            a[r2][l2 - j] = 0;
        flag = 1;
    }
}

```

```

        score += i1*(i1 - 1);
    }
    if (i1 >= 4)
    {
        for (int j = 1; a[r2 + j][l2 + j] == a[r2][l2]; j++)
            a[r2 + j][l2 + j] = 0;
        for (int j = 1; a[r2 - j][l2 - j] == a[r2][l2]; j++)
            a[r2 - j][l2 - j] = 0;
        flag = 1;
        score += i1*(i1 - 1);
    }
    if (i2 >= 4)
    {
        for (int j = 1; a[r2 - j][l2 + j] == a[r2][l2]; j++)
            a[r2 - j][l2 + j] = 0;
        for (int j = 1; a[r2 + j][l2 - j] == a[r2][l2]; j++)
            a[r2 + j][l2 - j] = 0;
        flag = 1;
        score += i2*(i2 - 1);
    }

    if (flag)
        a[r2][l2] = 0;
    return score;
}

int judge2(int a[][11], int row, int line, int r2, int l2)
{
    int num = 0, score = 0;
    score = judge1(a, row, line, r2, l2);
    if (!score)
    {
        for (int i = 1; i <= row; i++)
            for (int j = 1; j <= line; j++)
                if (!a[i][j])
                    num++;
        if (num <= 3)
        {
            for (int i = 1; i <= row; i++)
                for (int j = 1; j <= line; j++)
                    if (!a[i][j])
                        a[i][j] = 1;
            score = -1;
        }
    }
    return score;
}

void clrpa(int path[][11], int arr[][11])
{
    for (int i = 0; i < 11; i++)
        for (int j = 0; j < 11; j++)
            path[i][j] = 0, arr[i][j] = 0;
}

void choice37(int a[][11], const int choice, int row, int line)
{
    int score0 = 0, b[3], c, d, x, y, score = 0, score_ = 0, dead[8] = { 0 }, last[8] = { 0 }, choose[2]
    = { 0 }, choose_[2] = { 0 };

```

```

random(b);
system("cls");
choice == 7 || choice == 8 ? setconsoleborder(hout, 4 * row + 35, 2 * line + 6, 2 * line + 6),
setfontsize(hout, L"新宋体", 24) : 1;
while (1)
{
    if (choice == 3)
    {
        printarr(a, row, line);
        score = read2(a, b, choice, row, line);
    }
    else
    {
        print45_1(a, choice, row, line);
        print7(a, b, dead, last, score0, row, line);
        if (choice[0] && !score)
            showstr(hout, (choice[0] - 1) * 4 + 3, 2 * choice[1], str_, a[choice[1]][choice[0]]
- 1, 15);
        else if (score)
            choose[0] = 0, choose[1] = 0;
        score = 0;
        cout << (score = mouse(a, choice, b, dead, last, choose, choose_, x, y, row, line));
        setcolor(hout, 0, 7);
    }
    if (score == -1 || score == -2)
        break;
    if (score == -3)
    {
        clr(a, row, line);
        random_gene(a, row, line, initial);
        choice37(a, choice, row, line);
    }
    if (!score)
    {
        for (int i = 0; i < 3; i++)
            while (1)
                if (!a[c = (rand()*rand()) % row + 1][d = rand() % line + 1])
                {
                    a[c][d] = b[i];
                    break;
                }
            random(b);
    }
    score0 += score;
    if (choice == 3)
        cout << "本次得分:" << score << "\t总得分:" << score0 << endl;
}
if (choice == 7)
{
    print45_1(a, choice, row, line);
    print7(a, b, dead, last, score0, row, line);
}
setcursor(hout, CURSOR_VISIBLE_NORMAL); //打开光标
setcolor(hout, 0, 7);
gotoxy(hout, 1, 2 * row + 4);
if (score == -1)
    cout << "游戏结束 最终得分:" << score0 << endl;

```



```
cout << "本小题结束，请输入回车键继续...";
while (_getch() != '\r');
setfontsize(hout, L"新宋体", 18);
}
void random(int b[])
{
    for (int i = 0; i < 3; i++)
        b[i] = rand() % 7 + 1;
}
void swap(int *a, int *b)
{
    int t = *a;
    *a = *b;
    *b = t;
}
```

装

订

线