消消乐

实验报告

1551265 计 1 张伯阳

1. 消消乐

1.1.操作方法及要求

【游戏规则:】

1.游戏区域为 5*5 - 9*9 可选，共有 9 种不同（颜色/形状）的彩球随机出现，初始占满全部空间.

2.可消除及消除后的得分规则如下：

a)任意行同色球满 3 个及以上可消除，得分为消除个数

b)任意列同色球满 3 个及以上可消除，得分为消除个数

c)同时满足行列要求要求的则同时可消除，得分为消除个数

d)初始生成后未经移动即可消除的不得分游戏共9个功能

3.消除后，上方的彩球按垂直方向下落，上方空余位置随机补充彩球至满，如果下落及补充后满足消除要求，则自动消除并积分（初始生成后未经移动的消除后的下落补充导致再次消除的不得分）

4、初始状态生成并完成初始消除后，要标识出可移动位置的彩球

5、鼠标对标识出的彩球进行操作（分别单击源位置及目标位置）

6、无任何可消除彩球则游戏结束

【显示要求:】

1、可移动位置的彩球、被选中的彩球要有不同的显示效果

2、彩球下落时，要有动画效果沿着通路进行移动

3、消除时要有相应的动画效果

菜单项 1：输入行列后，在规定范围内随机生成所有位置上的彩球，然后打印整个内部数组，随后对数组搜索，找出行/列方向连续有三个以上相同值的并打印出来

- 为方便观察，打印时有球的位置用不同颜色输出

菜单项 2：在 1 的基础上，继续进行消除初始项、下落、空闲位置填入新随机值的操作

- 填入新随机值后需要再次查找，如果有可消除项，仍要重复操作直至无初始可消除项为止

菜单项 3：在 2 的基础上（已无初始可消除项），找出所有可被互换的彩球位置

- 球的位置用不同颜色标出

菜单项 4：在 cmd 伪图形界面上画出框架（无分隔线）及初始状态生成后的所有彩球

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 5 的基础上显示初始可消除项

- 彩球的颜色、背景色等不需要和 demo 一样

菜单项 7：在 6 的基础上完成消除初始项、下落、空闲位置填入新随机值的操作

- 与菜单项 2 的过程一致，差别只在于内部数组还是伪图形界面的展示
- 填入新随机值后需要再次查找，如果有可消除项，仍要重复操作直至无初始可消除项为止

菜单项 8：在 cmd 伪图形界面上实现完整版的游戏

- 要求用鼠标操作，基本操作为：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*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

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1.2.3.4.5个功能不需要移动的图形化操作,优先设计.定义出储存彩球的数组a[11][11],消除数组del[11][11],提示移动数组mov[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.**2功能需要判断小球是否连续,使用judge1函数遍历每一个小球,气上下左右四个方向是否有三个及三个以上的连续,del数组的该点处置1.3功能需要判断小球是否能够通过移动消去,使用judge2函数遍历每个球上下左右移动后是否会发生消去,可移动彩球的两个mov点置1.将可移动球的标记重复打印在原棋盘处进行覆盖.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.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.只是要充分考虑传参的问题.这次消消乐游戏使用的函数和之前彩球连线的有大量相同,在此不再赘述

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

2.3.6.7.8功能需要调用judge1,judge2函数进行遍历查找 7.8函数还要图形界面的移动的打印 8函数需要进行鼠标的读取

4. 调试过程碰到的问题

碰到的问题主要有下:

函数参数的传递

二维数组的第一个参数是行数(r值)第二个参数是列数(l值),而我又有一些函数在传二维数组时第一个参数写成了l,第二个写成了r,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.

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 */

#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<fstream>
#include <profileapi.h>
#include"cmd_console_tools.h"

using namespace std;
#define length  100
#define hight 40
#define pertime 50

struct ball {
    int a[11][11];
    bool del[11][11];
    bool mov[11][11];
};

const char  str_[3] = "◎", str00[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] = "—";
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 read1_1(int *choice);
void read1_2(int *row, int *line);
void showin(HANDLE hout, int X, int Y, int A, const int bg_color, const int fg_color);
void showdb(HANDLE hout, int X, int Y, double A, const int bg_color, const int fg_color);
int judge1(int a[][11], bool del[][11], int row, int line);
int judge2(int a[][11], bool mov[][11], int row, int line);
int judge3(int a[][11], bool mov[][11], int x1, int y1, int x2, int y2);
int judge4(int a[][11], bool del[][11], bool mov[][11], int choice, int row, int line, int
score0);
void printarr(int a[][11], bool del[][11], bool mov[][11], int flag, int row, int line);
void random_gene(int a[][11], const int row, const int line);
void choice1(int a[][11], bool del[][11], bool mov[][11], int row, int line);
void choice2(int a[][11], bool del[][11], bool mov[][11], int row, int line);
void choice3(int a[][11], bool del[][11], bool mov[][11], int row, int line);
void drop(int a[][11], bool del[][11], int row, int line);
void gen(int a[][11], int row, int line);
int choice6(int a[][11], bool del[][11], int row, int line);
int choice7(int a[][11], bool del[][11], bool mov[][11], int choice, int row, int line, int
score0);
void choice8(int a[][11], bool del[][11], bool mov[][11], int choice, int row, int line);
int mouse(int a[][11], bool del[][11], bool mov[][11], int choice, int row, int line, int score);
void choice45(int a[][11], const int choice, int row, int line);
```

```
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);
void print45_3(int a[][11], const int choice, int row, int line);
void tapeenter(int row, int choice);
void swap(int *a, int *b);
int input_dat(int a[][11], bool mov[][11], int *line, int *row,char name[]);
int input_ans(int a[][11], bool mov[][11], int line, int row,char name[]);
void choice9(int a[][11], bool del[][11], bool mov[][11]);

#include"90-b1.h"
#include"cmd_console_tools.h"

void printarr(int  a[][11], bool del[][11], bool mov[][11], int flag, 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 l = 1; l <= line; l++)
	{
		cout << endl << char('A' + l - 1) << " |";
		for (int r = 1; r <= row; r++)
		{
			setcolor(hout,  0, 7);
			cout << "   ";
			if (flag && (del[r][l] || mov[r][l]))
				setcolor(hout,  6, 0);
			cout << a[r][l];
		}
		setcolor(hout,  0, 7);
	}
	cout << endl << endl;
}

void random_gene(int  a[][11], const int row, const int line)
{
	srand((unsigned)time(NULL));
	for (int i = 1; i <= line; i++)
		for (int j = 1; j <= row; j++)
			a[j][i] = rand() % 9 + 1;
}

void choice1(int  a[][11], bool del[][11], bool mov[][11], int row, int line)
{
	printarr(a, del, mov, 0, row, line);
	if (judge1(a, del, row, line))
	{
		cout << endl << "输入回车继续" << endl;
		while (_getch() != '\r');
		printarr(a, del, mov, 1, row, line);
	}
}

void choice2(int  a[][11], bool del[][11], bool mov[][11], int row, int line)
{
	printarr(a, del, mov, 0, row, line);
	while (judge1(a, del, row, line))
	{
		cout << endl << "输入回车继续" << endl;
		while (_getch() != '\r');
```

```
            printarr(a, del, mov, 1, row, line);
            cout << endl << "输入回车继续" << endl;
            while (_getch() != '\r');
            drop(a, del, row, line);
            printarr(a, del, mov, 1, row, line);
            cout << endl << "输入回车继续" << endl;
            while (_getch() != '\r');
            gen(a, row, line);
            printarr(a, del, mov, 1, row, line);
        }
}

void choice3(int a[][11], bool del[][11], bool mov[][11], int row, int line)
{
        choice2(a, del, mov, row, line);
        if (judge2(a, mov, row, line))
        {
            cout << endl << "输入回车继续" << endl;
            while (_getch() != '\r');
            printarr(a, del, mov, 1, row, line);
        }
}

void drop(int a[][11], bool del[][11], int row, int line)
{
        for (int i = row; i >= 1; i--)
            for (int j = line; j >= 1; j--)
                for (int k = 0; del[i][j] && k <= 9; k++)
                {
                    for (int jj = j; jj >= 1; jj--)
                    {
                        a[i][jj] = a[i][jj - 1];
                        del[i][jj] = del[i][jj - 1];
                    }
                }
}

void gen(int a[][11], int row, int line)
{
        for (int i = row; i >= 1; i--)
            for (int j = line; j >= 1; j--)
                if (!a[i][j])
                    a[i][j] = rand() % 9 + 1;
}

#include"90-b1.h"
#include"cmd_console_tools.h"

void choice45(int a[][11], const int choice, int row, int line)
{
        bool del[11][11], mov[11][11];
        memset(del, 0, sizeof(del));
        memset(mov, 0, sizeof(mov));
        if (choice == 4 || choice == 5)
        {
            cout << "初始数组:" << endl;
            printarr(a, del, mov, 0, row, line);
            cout << "按回车键显示图形...";
            while (_getch() != '\r');
        }
        if (choice == 4 || choice == 5)
            setfontsize(hout, L"新宋体", 24);
        if (choice == 4)
```

```
            setconsoleborder(hout, 2 * line + 8, row + 8, row + 8);
        else if (choice == 5)
            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);
}

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 * i + 1, j + 1, str0, 15, 15);
                else
                    showstr(hout, 2 * i + 1, j + 1, str0, a[i][j] / 7 + 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 * i + 1, j + 1, str0, 15, 15);
                        else
                            showstr(hout, 2 * i + 1, j + 1, str0, a[(i + 1) / 2][(j + 1) / 2] / 7 + 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 choice6(int a[][11], bool del[][11], int row, int line)
{
    int num;
    setconsoleborder(hout, 4 * line + 8, 2 * row + 8, 2 * row + 8);
    setfontsize(hout, L"新宋体", 24);
    choice45(a, 6, row, line);
    if (num = judge1(a, del, row, line))
        for (int r2 = 1; r2 <= row; r2++)
            for (int l2 = 1; l2 <= line; l2++)
                if (del[r2][l2])
                    showstr(hout, 4 * r2 - 1, 2 * l2, str00, a[r2][l2] / 7 + a[r2][l2] - 1, 0);//?
    gotoxy(hout, 1, 2 * row + 4);
    return num;
}
```

装 订 线

```
int choice7(int a[][11], bool del[][11], bool mov[][11], int choice, int row, int line, int score0)
{
	int score = 0, over = 1;
	setconsoleborder(hout, 4 * line + 8, 2 * row + 8, 2 * row + 8);
	setfontsize(hout, L"新宋体", 24);
	while (score = choice6(a, del, row, line))//hint del
	{
		tapeenter(row, choice);
		for (int r2 = 1; r2 <= row; r2++)
			for (int l2 = 1; l2 <= line; l2++)
				if (del[r2][l2])
				{
					showstr(hout, 4 * r2 - 1, 2 * l2, str0, a[r2][l2] / 7 + a[r2][l2] - 1, 0);
					Sleep(pertime * 2);
					showstr(hout, 4 * r2 - 1, 2 * l2, str0, 15, 15);
					Sleep(pertime * 2);
				}//delete
		for (int r2 = row; r2 >= 1; r2--)
			for (int l2 = line; l2 >= 1; l2--)
			{
				if (del[r2][l2])
				{
					int i = 1;
					if (l2 == 1)
						continue;
					while (del[r2][l2 - i] && a[r2][l2 - i])
						i++;
					if (!a[r2][l2 - i])
						continue;
					for (int j = i; j >= 1; j--)
					{
						showstr(hout, (r2 - 1) * 4 + 3, 2 * (l2 - j), str0, 15, 15);
						showstr(hout, (r2 - 1) * 4 + 3, 2 * (l2 - j) + 1, str0, a[r2][l2 - j] / 7 + a[r2][l2 - j] - 1,
15);
						Sleep(pertime);
						showstr(hout, (r2 - 1) * 4 + 3, 2 * (l2 - j) + 1, str12, 15, 0);
						showstr(hout, (r2 - 1) * 4 + 3, 2 * (l2 - j) + 2, str0, a[r2][l2 - j] / 7 + a[r2][l2 - j] - 1,
15);
						Sleep(pertime);
						swap(a[r2][l2 - j + 1], a[r2][l2 - j]);
					}//drop down
					bool t = del[r2][l2 - i];
					del[r2][l2 - i] = del[r2][l2];
					del[r2][l2] = t;//swap del
				}
			}
		for (int r2 = row; r2 >= 1; r2--)
			for (int l2 = line; l2 >= 1; l2--)
				if (del[r2][l2])
					a[r2][l2] = 0;//
		tapeenter(row, choice);
		gen(a, row, line);
		for (int r2 = row; r2 >= 1; r2--)
			for (int l2 = line; l2 >= 1; l2--)
				if (del[r2][l2])
				{
					showstr(hout, 4 * r2 - 1, 2 * l2, str0, a[r2][l2] / 7 + a[r2][l2] - 1, 0);
					Sleep(pertime * 2);
				}//generate
		for (int i = 0; i <= line + 1; i++)
			for (int j = 0; j <= row + 1; j++)
				del[j][i] = 0;
		tapeenter(row, choice);
		score0 += score;
```

```
        }
        if (choice != 10)
            judge2(a, mov, row, line);
        for (int r2 = row; r2 >= 1; r2--)
            for (int l2 = line; l2 >= 1; l2--)
                if (mov[r2][l2])
                {
                    showstr(hout, 4 * r2 - 1, 2 * l2, str_, a[r2][l2] / 7 + a[r2][l2] - 1, 0);
                    Sleep(pertime);
                    over = 0;//judge game over
                }//hint mov
        if (over)
            return -1;
        gotoxy(hout, 1, 2 * row + 4);
        return score0;
}

void choice8(int a[][11], bool del[][11], bool mov[][11], int choice, int row, int line)
{
        int score0 = 0;
        if (!judge4(a, del, mov, choice, row, line, score0))
            while (1)
            {
                gotoxy(hout, 0, 2 * row + 4);
                setcolor(hout, 0, 7);
                cout << "Score:" << score0 << endl;
                int score = mouse(a, del, mov, choice, row, line, 0);
                if (score == -1 || score == -2)
                    break;
                else
                    score0 += score;
                if (judge4(a, del, mov, choice, row, line, score0))
                    break;
            }
        setcursor(hout, CURSOR_VISIBLE_NORMAL);   //open the cursor
        setcolor(hout, 0, 7);
        gotoxy(hout, 1, 2 * row + 4);
        cout << "游戏结束 最终得分:" << score0 << endl << "输入回车键继续";
        while (_getch() != '\r');
}

int mouse(int a[][11], bool del[][11], bool mov[][11], int choice, int row, int line, int score)
{
        enable_mouse(hin);
        int X = 0, Y = 0, action, x = 0, y = 0, x1 = 0, x2 = 0, y1 = 0, y2 = 0;
        setcursor(hout, CURSOR_INVISIBLE); //close the cursor
        while (1)
        {
            action = read_mouse(hin, X, Y);
            if (action == MOUSE_RIGHT_BUTTON_CLICK)
                return -2;
            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)
                {
                    if (!x1)
                        x1 = x, y1 = y;
                    else if (!x2)
                    {
                        x2 = x, y2 = y;
                        int b = judge3(a, mov, x1, y1, x2, y2);
                        if (!b)
                        {
```

```
                                gotoxy(hout, 0, 2 * row + 5);
                                cout << "Can't switch\n";
                        }
                        else
                        {
                                swap(&a[x1][y1], &a[x2][y2]);
                                int score = choice7(a, del, mov, choice, row, line, 0);
                                if (!score)
                                {
                                        gotoxy(hout, 0, 2 * row + 5);
                                        setcolor(hout, 0, 7);
                                        cout << "Can't switch\n";
                                        swap(&a[x1][y1], &a[x2][y2]);
                                        return 0;
                                }
                                else
                                {
                                        Sleep(pertime * 4);
                                        showstr(hout, 4 * x1 + 1, 2 * y1 + 1, str0, a[x1][y1] / 7 + a[x1][y1] - 1, 0);
                                        showstr(hout, 4 * x2 + 1, 2 * y2 + 1, str0, a[x2][y2] / 7 + a[x2][y2] - 1, 0);
                                        return score / 2;
                                }
                        }
                        x1 = 0, x2 = 0, y1 = 0, y2 = 0;
                }
        }
        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";
        }
    }
  }
}

void choice9(int a[][11], bool del[][11],bool mov[][11])
{
    int line, row;
    char name[30] = "test\\",num[8], name_[5] = ".dat";
    cout << "输入学号:" << endl;
    cin >> num;
    strcat(name, num);
    strcat(name, name_);
    input_dat(a, mov, &line, &row, name);
    choice7(a, del, mov, 9, row, line, 0);
    tapeenter(row, 7);
    input_ans(a, mov, line, row, name);
    for (int r = row; r >= 1; r--)
        for (int l = line; l >= 1; l--)
            if (a[r][l] > 10)
                mov[r][l] = 1, a[r][l] -= 90;
            else
                mov[r][l] = 0;
    choice7(a, del, mov, 10, row, line, 0);
}

#include"90-b1.h"
#include"cmd_console_tools.h"

int main()
{
    int choice, row, line;
    ball b;
```

```
	setcolor(hout, 0, 7);
	srand((unsigned)time(NULL));
	while (1)
	{
		memset(b.del, 0, sizeof(b.del));
		memset(b.mov, 0, sizeof(b.mov));
		memset(b.a, 0, sizeof(b.a));
		setconsoleborder(hout, length, hight, hight);
		read1_1(&choice);
		if (!choice)
			break;
		if (choice != 9)
		{
			read1_2(&row, &line);
			random_gene(b.a, row, line);
		}
		system("cls");
		if (choice == 1)
			choice1(b.a, b.del, b.mov, row, line);
		if (choice == 2)
			choice2(b.a, b.del, b.mov, row, line);
		if (choice == 3)
			choice3(b.a, b.del, b.mov, row, line);
		if (choice == 4 || choice == 5)
			choice45(b.a, choice, row, line);
		if (choice == 6)
			choice6(b.a, b.del, row, line);
		if (choice == 7)
			choice7(b.a, b.del, b.mov, choice, row, line, 0);
		if (choice == 8)
			choice8(b.a, b.del, b.mov, choice, row, line);
		if (choice == 9)
			choice9(b.a, b.del,b.mov);
		setcolor(hout, 0, 7);
		if (choice != 8)
		{
			cout << "本小题结束，请输入回车键继续..." << endl;
			while (_getch() != '\r');
		}
		setfontsize(hout, L"新宋体", 18);
		system("cls");
	}
}

void read1_1(int *choice)
{
	while (1)
	{
		cout << "-----------------------------------" << endl;
		cout << "1.内部数组，生成初始状态，寻找是否有初始可消除项\n2.内部数组，消除初始可消除项后非0项下落并用0填充\n3.内部数组，消除初始可消除项后查找消除提示\n4.n*n的框架(无分隔线)，显示初始状态\n5.n*n的框架(有分隔线)，显示初始状态\n6.n*n的框架(有分隔线)，显示初始状态及初始可消除项\n7.n*n的框架(有分隔线)，消除初始可消除项后显示消除提示\n8.cmd图形界面完整版\n9.从文件中读取数据以验证查找消除提示的算法的正确性\n0.退出\n";
		cout << "-----------------------------------" << endl;
		cin >> *choice;
		if (!cin || *choice < 0 || *choice>9)
		{
			cin.clear();
			cin.sync();
			cin.ignore(1024, '\n');
		}
		else
```

装 订 线

```
                break;
        }
}

void read1_2(int *row, int *line)
{
        while (1)
        {
                cout << "请输入行数(5-9)：";
                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(5-9)：";
                cin >> *line;
                if (!cin || *line>9 || *line<7)
                {
                        cin.clear();
                        cin.sync();
                        cin.ignore(1024, '\n');
                }
                else
                        break;
        }
}

#include"90-b1.h"
#include"cmd_console_tools.h"

int judge1(int a[][11], bool del[][11], int row, int line)
{
        for (int i = 0; i <= line + 1; i++)
                for (int j = 0; j <= row + 1; j++)
                        del[j][i] = 0;
        for (int i = 0; i <= line + 1; i++)
                del[0][i] = 1, del[row + 1][i] = 1;
        for (int i = 0; i <= row + 1; i++)
                del[i][0] = 1, del[i][line + 1] = 1;
        int ir = 0, il = 0, i1 = 0, i2 = 0, score0 = 0;
        for (int r2 = 1; r2 <= row; r2++)
                for (int l2 = 1; l2 <= line; l2++)
                {
                        ir = 0, il = 0;
                        int 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++);
                        if (ir >= 2)
                        {
                                for (int j = 1; a[r2 + j][l2] == a[r2][l2]; j++)
                                        del[r2 + j][l2] = 1;
                                for (int j = 1; a[r2 - j][l2] == a[r2][l2]; j++)
                                        del[r2 - j][l2] = 1;
                                score += ir;
                        }
```

装 订 线

```
                if (il >= 2)
                {
                    for (int j = 1; a[r2][l2 + j] == a[r2][l2]; j++)
                        del[r2][l2 + j] = 1;
                    for (int j = 1; a[r2][l2 - j] == a[r2][l2]; j++)
                        del[r2][l2 - j] = 1;
                    score += il;
                }
                if (score)
                    del[r2][l2] = 1;
                score0 += score;
            }
        return score0;
}//judge del

int judge2(int a[][11], bool mov[][11], int row, int line)
{
    for (int i = 0; i <= line + 1; i++)
        for (int j = 0; j <= row + 1; j++)
            mov[j][i] = 0;
    int ir[5] = { 0 }, il[5] = { 0 }, iu[5] = { 0 }, id[5] = { 0 }, flag = 0;
    for (int r2 = 1; r2 <= row; r2++)
        for (int l2 = 1; l2 <= line; l2++)
        {
            for (int i = 0; i < 5; i++)
                ir[i] = 0, il[i] = 0, iu[i] = 0, id[i] = 0;
            int score = 0;
            for (int j = 1; a[r2 + 1][l2] && a[r2 + 1][l2] != a[r2][l2] && a[r2 + 1 + j][l2] == a[r2][l2]; j++, ir[1]++);
            for (int j = 1; a[r2 + 1][l2] && a[r2 + 1][l2] != a[r2][l2] && a[r2 + 1][l2 + j] == a[r2][l2]; j++, ir[3]++);
            for (int j = 1; a[r2 + 1][l2] && a[r2 + 1][l2] != a[r2][l2] && a[r2 + 1][l2 - j] == a[r2][l2]; j++, ir[4]++);
            for (int j = 1; a[r2 - 1][l2] && a[r2 - 1][l2] != a[r2][l2] && a[r2 - 1 - j][l2] == a[r2][l2]; j++, il[2]++);
            for (int j = 1; a[r2 - 1][l2] && a[r2 - 1][l2] != a[r2][l2] && a[r2 - 1][l2 - j] == a[r2][l2]; j++, il[3]++);
            for (int j = 1; a[r2 - 1][l2] && a[r2 - 1][l2] != a[r2][l2] && a[r2 - 1][l2 + j] == a[r2][l2]; j++, il[4]++);
            for (int j = 1; a[r2][l2 + 1] && a[r2][l2 + 1] != a[r2][l2] && a[r2 + j][l2 + 1] == a[r2][l2]; j++, id[1]++);
            for (int j = 1; a[r2][l2 + 1] && a[r2][l2 + 1] != a[r2][l2] && a[r2 - j][l2 + 1] == a[r2][l2]; j++, id[2]++);
            for (int j = 1; a[r2][l2 + 1] && a[r2][l2 + 1] != a[r2][l2] && a[r2][l2 + 1 + j] == a[r2][l2]; j++, id[3]++);
            for (int j = 1; a[r2][l2 - 1] && a[r2][l2 - 1] != a[r2][l2] && a[r2 + j][l2 - 1] == a[r2][l2]; j++, iu[1]++);
            for (int j = 1; a[r2][l2 - 1] && a[r2][l2 - 1] != a[r2][l2] && a[r2 - j][l2 - 1] == a[r2][l2]; j++, iu[2]++);
            for (int j = 1; a[r2][l2 - 1] && a[r2][l2 - 1] != a[r2][l2] && a[r2][l2 - 1 - j] == a[r2][l2]; j++, iu[4]++);
            if (ir[1] >= 2 || ir[3] >= 2 || ir[4] >= 2)
            {
                mov[r2][l2] = 1;
                mov[r2 + 1][l2] = 1;
                flag = 1;
            }
            if (il[2] >= 2 || il[3] >= 2 || il[4] >= 2)
            {
                mov[r2][l2] = 1;
                mov[r2 - 1][l2] = 1;
                flag = 1;
            }
            if (iu[1] >= 2 || iu[2] >= 2 || iu[4] >= 2)
            {
                mov[r2][l2] = 1;
                mov[r2][l2 - 1] = 1;
                flag = 1;
            }
            if (id[1] >= 2 || id[2] >= 2 || id[3] >= 2)
            {
                mov[r2][l2] = 1;
                mov[r2][l2 + 1] = 1;
                flag = 1;
            }
        }
```

装 订 线

```
    return flag;
}//judge mov

int judge3(int a[][11], bool mov[][11], int x1, int y1, int x2, int y2)
{
    return (x1 == x2&&y1 == y2 + 1 || x1 == x2&&y1 == y2 - 1 || x1 == x2 + 1 && y1 == y2 || x1 == x2 - 1 && y1 == y2) && mov[x1][y1] && mov[x2][y2];
}//judge switch

int judge4(int a[][11], bool del[][11], bool mov[][11], int choice, int row, int line, int score0)
{
    if (choice7(a, del, mov, choice, row, line, 0) == -1)//not add score in the beginning
    {
        gotoxy(hout, 0, 2 * row + 4);
        cout << "游戏结束  最终得分:" << score0 << endl;
        return 1;
    }
    return 0;
}

void swap(int *a, int *b)
{
    int t = *a;
    *a = *b;
    *b = t;
}

void tapeenter(int row, int choice)
{
    if (choice == 7)
    {
        gotoxy(hout, 1, 2 * row + 4);
        setcolor(hout, 0, 7);
        cout << "输入回车键继续";
        while (_getch() != '\r');
        gotoxy(hout, 1, 2 * row + 4);
        cout << endl;
    }
}

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);
}

int input_dat(int a[][11], bool mov[][11], int *line, int *row,char name[])
{
    ifstream fin;
    fin.open(name, ios::in);
    if (!fin.is_open())
    {
        cout << "无法打开" << endl;
        return -1;
    }
    char l[4], r[4], str[4];
```

装 订 线

```
	fin >> l >> r;
	*line = l[0] - '0';
	*row = r[0] - '0';
	for (int i = 1; i <= *row; i++)
		for (int j = 1; j <= *line; j++)
		{
			fin >> str;
			a[j][i] = str[0] - '0';
		}
	fin.close();
	return 0;
}

int input_ans(int a[][11], bool mov[][11],int line, int row,char name[])
{
	ifstream fin;
	fin.open(name, ios::in);
	if (!fin.is_open())
	{
		cout << "无法打开" << endl;
		return -1;
	}
	char str[4];
	for (int i = 1; i <= row; i++)
		for (int j = 1; j <= line; j++)
		{
			fin >> str;
			if (!str[1])
				a[j][i] = str[0] - '0';
			else
				a[j][i] = (str[0] - '0') * 10 + str[1] - '0';
		}
	fin.close();
	return 0;
}
```

装 订 线