装

订

线

高级语言程序设计

彩球游戏实验报告

作 者 姓 名：陈应波
学 号：2352219
班 级：信 11

同济大学

Tongji University

二〇二四年六月

1. 题目及基本要求描述

1.1. 题目综述

子题目 1:

生成初始状态并找出初始可消除项键盘输入行列(要处理输入错误，下同)

显示初始数组，行号从 A 开始，列号从 1 开始，值 1-9 分别表示 9 种颜色的球按回车键查找初始可消除项，即初始生成后某行/某列有三个以上连续相同值，用不同颜色标记出来，如果没有，则提示找不到

子题目 2:

消除初始可消除项、非 0 项下落、用 0 填充、再在 0 位填充新值在子题目 1 完成的基础上进行，0 代表空位:

非零项下落、填充 0、新值填充 0 位置分步进行

完成后，需要再次查找是否有初始可消除项，如有，则反复进行

子题目 3:

初始可消除项消除完成后，查找消除提示项(即相邻的可互换项)在子题目 2 完成的基础上进行，

子题目 4:

在伪图形界面下画出初始状态彩球之间无分隔线

子题目 5:

在伪图形界面下画出初始状态彩球之间有分隔线

希望和子题目 4 共用画框架的参数，是否有分隔线通过参数决定(后续同)

子题目 6:

在伪图形界面下画出初始状态并找出初始可消除项在子题目 4 完成的基础上进行彩球之间无分隔线

子题目 7:

在伪图形界面下画出初始状态并找出初始可消除项，消除后显示消除提示在子题目 5 完成的基础上进行

彩球之间有分隔线

子题目 8:

在伪图形界面下支持鼠标，能正确判断出鼠标的行列位置在子题目 7 完成的基础上进行

彩球之间有分隔线

鼠标只能选择可消除项(选到不可消除项要给出提示，

支持鼠标右键结束

子题目 9:

伪图形界面完整版

在子题目 8 完成的基础上进行

能用鼠标完成一次完整的游戏能用鼠标选择可消除项，再按一次则取消选择

能交换可消除项并进行消除、下落、填充

能计算分数

能判断游戏是否结束

1.2. 要求与限制

1、用伪图形界面方式完成彩球游戏(MagicBal1)

2、提供 90-b2-demo.exe 供参考

a)需设置为旧版控制台，去除快速编辑和插入模式

b) Windows 版的游戏区域是 5*5~9*9 的正方形，自制版是 59 之间任意行列

c)自制版的可消除球为实心，可互换球为空心双圈，普通球为空心单圈，颜色通过背景色区分

3、附件提供了一个图形界面版的 MagicBal1 游戏供参考，如果参考游戏的规则和本作业的具体要求不同，以作业要求为准

5、项目命名及提交要求:整个项目由 8 个文件组成(需提交的为 6 个)

装

2. 整体设计思路

订

1. 数据结构定义

游戏面板数据结构： 使用二维数组来表示游戏面板，每个位置存储一个整数，代表球的颜色或空位（0 表示空位）。

线

2. 初始状态生成与初始可消除项查找

生成初始状态：

接收用户输入的行数和列数。

初始化二维数组，随机生成每个位置的值（1 到 9 的随机数，表示不同颜色的球）。

查找初始可消除项：

遍历每行和每列，找出连续三个或以上相同值的部分。

标记这些位置为初始可消除项，并记录其位置和颜色。

如果没有初始可消除项，则重新生成游戏面板，直到找到满足条件的面板为止。

3. 消除和下落过程实现

消除过程：

根据找到的初始可消除项，执行消除操作，将这些位置置为 0。

计算分数并更新。

下落过程：

从底部往上遍历每列，对于每个位置，如果是 0，则向上找到最近的非 0 值并将其移至当前位置。

填充过程：

从顶部向下遍历每列，对于每个位置，如果是 0，则填充一个新的随机数（1 到 9 的随机数，

表示新的球）。

4. 查找消除提示项

查找消除提示项：

在每次消除和下落之后，重新遍历整个游戏面板，找出所有相邻可互换的项，即可以进行交换的位置。

标记这些位置，以便在界面上显示给玩家。

5. 绘制游戏界面

伪图形界面绘制：

使用控制台或简易图形库来实现游戏界面的绘制。

可根据需求选择是否显示彩球之间的分隔线。

6. 鼠标操作支持

鼠标交互支持：

在图形界面的基础上，实现鼠标点击事件的响应。

点击位置的转换，根据具体实现环境确定行列位置。

确认点击是否在可消除项上，如果是则执行消除操作，否则给出提示或取消选择。

7. 游戏逻辑控制

游戏循环：

主循环中实现游戏的整体控制逻辑。

包括初始化游戏、处理用户输入、执行消除和下落、更新界面显示、判断游戏结束等功能。

根据需要实现分数计算和显示。

8. 结束游戏

结束游戏：

支持鼠标右键结束游戏。

2.1. 抽象化彩球游戏及其移动的思路

需要进行的操作有包括查找可消除项、消除、计算得分、下落消除0和在0位置产生新数据，提示可交换消除项等。通过设置二维数组ball[9][9]来存储游戏功能的图，值即为其显示值；设置rs[20][4]来存储查找相邻可交换的值的图。通过这两个数组的操作，加以随机数产生函数，即可完成全部功能。

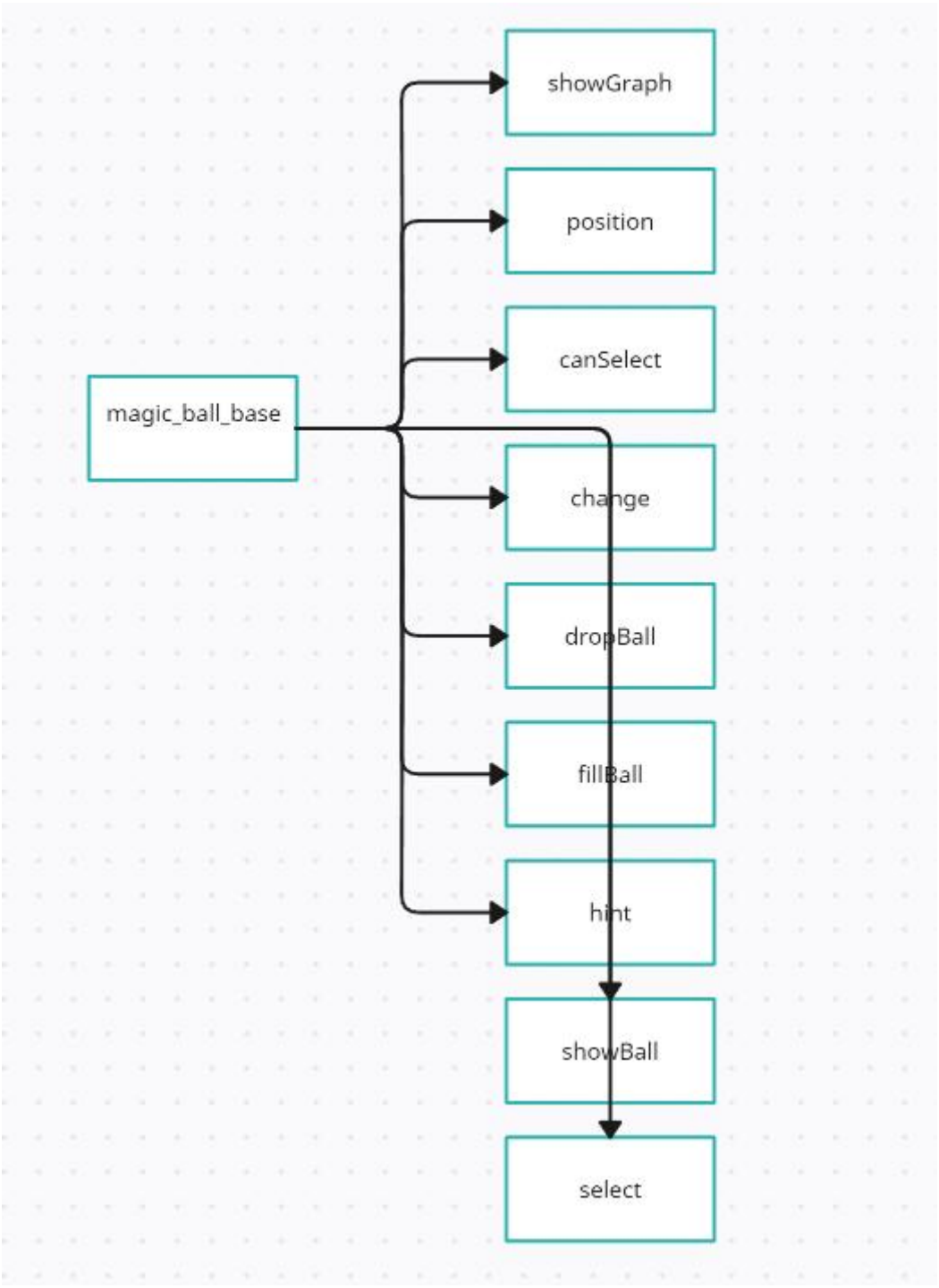
2.2. 程序整体实现思路

Main函数用一个循环和switch()函数来做到各小题函数的调用。而与上次不同的是，我将函数按功能分为了几类：逻辑功能函数、数组函数、进行命令行输出的函数和进行cmd伪图形化输出的函数。并分别将其放在不同的cpp文件中。在各个小题中我使用这些函数并使用一些基础的

循环、条件语句来完成所给的要求。各个部分将在下一部分进行具体介绍，而那些基础的循环、条件语句则略去。

整体的设计思路是处理输入的行列，通过改变数组的函数来改变数组。之后再通过进行命令行输出的函数和进行cmd伪图形化输出的函数来将这种变化可视化，从而彩球游戏的任务。

3 设计图



4. 调试过程碰到的问题

4.1. 越界问题

这个问题老生常谈，但是在写的函数非常多的时候很难察觉到，虽然知道数组下标加减之前要判断是否越界，但是写的代码一多真的很容易忘记。

举几个例子

●判断相邻的函数。在达到边界条件时应该判断+1/-1 之后是否越界，不然同样会越界。

●递归函数的返回条件。Row 其实是行数+1，column 是列数+1，所以如果判断是到 row/column 的话，事实上已经越界了，但是发现 bug 之后才想到这一点，耗费了很多时间。

4.2. gotoxy函数输出问题

我最开始的各类print函数中使用了cct_gotoxy()函数来分别输出框架和内部的数组。但是后来我发现gotoxy()函数只能到达当前窗口的坐标，而不能产生滚轮而自动向下。这就导致当我输出到最下面的时候再使用gotoxy函数就会出现预期之外的结果。最后我将函数改写成了单纯使用cout来一行行输出，解决了这个问题。

5. 心得体会

1)反复出现的常量应该用常变量或者宏定义的方式写在头文件中，便于维护。

2)变量命名要准确易于理解，最好使用下划线命名法或驼峰命名法。

本次作业我尝试了这两种命名法都应用一遍。Cosole 中使用驼峰命名法，其他 cpp 中使用下划线命名法，让我对这两种命名方法的理解更进了一步。

3)在源程序中要善于写注释，并且对不同的函数和不同的定义做分类以及说明，减少维护线程序时浪费的时间。在同一函数中也要注意用空行来区分不同代码块，便于区分。

4)要善于构建临时测试，并学会打断点，边写代码边做测试，这样可以大大节省后续调试的时间，同时也不会因为程序过于庞大而找不到错误在哪。

5)充分思考后再开始写代码，增加代码精巧程度，减少无效代码和垃圾代码。

在这次大作业中我利用改变参数来使函数完成不同的功能，合并了一些函数和更改了以前的公共函数。在我看来，这样做的最大好处就是能够使函数的数量减少，看起来也更加简洁明了。

相对复杂函数的主要特点就是大。代码量大，函数多，很容易出现遗忘的问题。

这个时候平时小作业所嗤之以鼻的写注释、规范命名、模块函数分类就显得非常重要。

●不写注释，过几天就忘了函数/参数每个是干嘛的了，非常不利于维护。

●不规范命名，比如全部命名为 abcd1234，甚至不用过几天，只要这个函数写得稍微长一点，轻则忘了参数是干嘛的，重则相互混淆，造成很难排除的 bug。函数更是如此，命名如果没有意义，后续调用都不知道是什么功能，整个程序很容易崩掉。

●模块函数分类。这次的作业，应该是函数写的最多的一次作业，有很多很多功能需要实现，

如果不将其分类，找起来真的是眼花缭乱，很容易看困了，或者上下翻半天才知道是干嘛的，非常浪费时间不说，心态还容易受影响。

所以我觉得应该：

1.做好函数分类：

将函数按照功能分为类，放在不同的cpp文件中。便于寻找与理解。（详见5.1.2）

2.养成好的函数与变量命名规则：

易于理解函数的含义，减少错误引用。（详见 5.1.1）

6. 附件：源程序

```
void fun8() {
                int row = 0, col = 0, score = 0;
                scanRowAndCol(&row,
&col);
                int arr[9][9];
                initArr(arr, row, col);
                showArr(arr, row, col, 0);
                showGraph(arr, row, col, 1);
                Sleep(100);
                showBall(arr, row, col, 1, 1);
                while (!isFinish(arr, row, col))
{
                 Sleep(100);
                 dropBall(arr, row, col, 1,
score);
                 Sleep(100);
                 fillBall(arr, row, col, 1);
                }
                Sleep(100);
                score = 0;
                if (hint(arr, row, col, 1)) {
                 cct_showstr(14, 0, "(当前分
数：    右键退出)");
                 cct_gotoxy(25, 0);//打印分
数的地方
                 cout << score;
                 int v1 = 0, v2 = 0;
                 if (select(arr, row, col, &v1,
&v2)) {
                      Sleep(500);
                 }
                }
                else {
                 cct_showstr(14, 0, "(已无可
消除项，游戏结束！)");
                }
                cct_showch(0, 2 + 2 * row, ' ',
0, 7, 40);
                theEnd();
}
void fun9() {
                int row = 0, col = 0, score = 0;
                scanRowAndCol(&row,
&col);
                int arr[9][9];
                initArr(arr, row, col);
                showArr(arr, row, col, 0);
                showGraph(arr, row, col, 1);
                Sleep(100);
                showBall(arr, row, col, 1, 1);
                while (!isFinish(arr, row, col))
{
                 Sleep(100);
                 dropBall(arr, row, col, 1,
score);
                 Sleep(100);
                 fillBall(arr, row, col, 1);
                }
                Sleep(100);
                score = 0;
                while (hint(arr, row, col, 1)) {
                 cct_showstr(14, 0, "(当前分
数：    右键退出)");
                 cct_gotoxy(25, 0);//打印分
数的地方
                 cout << score;
```

```
                int v1 = 0, v2 = 0;
                if (select(arr, row, col, &v1, &v2)) {
                    int v3 = 0, v4 = 0;
                    select(arr, row, col, &v3, &v4);
                    if (v1 == v3 && v2 == v4)
                        cct_showstr(2 + v2 * 4, 2 + v1 * 2, "◎", arr[v1][v2], 7);
                    else {
                        int x, y;
                        cct_getxy(x, y);
                        change(arr, row, col, v1, v2, v3, v4, score);
                        cct_gotoxy(25, 0);//打印分数的地方
                        cout << score;
                        cct_gotoxy(x,y);
                    }
                }
                else {
                    cct_showch(0, 2 + 2 * row, ' ', 0, 7, 40);
                    theEnd();
                    return;
                }
            }
            cct_showstr(14, 0, "(已无可消除项，游戏结束！)");
            cct_showch(0, 2 + 2 * row, ' ', 0, 7, 40);
            theEnd();
}
void showGraph(int arr[][9], int row, int col, int showBorder) {
            cct_setconsoleborder(40, 6 + row + 8 * showBorder, -1, -1);
            cct_setfontsize("新宋体", 68 - row * 4);
            cct_gotoxy(0, 1);
            cct_setcolor(7, 0);
            for (int i = 0; i < row + 2; i++)
            {
             if (i == 0)
                cout << "╔";
            else if (i == row + 1)
                cout << "╚";
            else if (showBorder && i < row)
                cout << "╠";
            else if (!showBorder || i != row)
                cout << "║";
            if (showBorder && i != row)
                for (int j = 0; j < col - 1; j++)
                {
                    cout << "═";
                    if (i == 0)
                        cout << "╦";
                    else if (i == row + 1)
                        cout << "╩";
                    else
                        cout << "╬";
                }
            cout << "═";
            }
            if(!showBorder)
                for (int j = 0; j < col; j++) {
                    if (i == 0 || i == row + 1)
                        cout << "═";
                    else
                        cout << "    ";
                }
            if (i == 0)
                cout << "╗";
            else if (i == row + 1)
                cout << "╝";
            else if (showBorder && i < row)
                cout << "╣";
            else if (!showBorder || i != row)
                cout << "║";
            if(!showBorder || i != row)
```

```
                cout << endl;
            if (showBorder && i < row)
{
                for (int j = 0; j < col;
j++)
                    cout << "┃  ";
                cout << "┃" << endl;
            }
            }
            cct_setcolor(0, 7);
            cct_gotoxy(0, 0);
            cout << "屏幕：" << (6 + row
+ 8 * showBorder) << "行" << 40 << "列";
            cct_gotoxy(0,row + (row - 1)
* showBorder + 3);
}
void showBall(int arr[][9], int row, int col, int
hasBorder, int showDifferent) {
            int x = 2, y = 2;
            for (int i = 0; i < row; i++)
            {
             for (int j = 0; j < col; j++)
             {
                if (showDifferent &&
removable(arr,i,j,row,col)) {
                    cct_showstr(x + j *
2 * (1 + hasBorder), y + i * (hasBorder + 1), "●",
arr[i][j], 0);
                }
                else
                    cct_showstr(x + j *
2 *(1 + hasBorder), y + i * (hasBorder + 1), "○",
arr[i][j], 0);
             }
            }
            cct_setcolor(0, 7);
            cct_gotoxy(0, 3 + row + 8 *
hasBorder);
}
void dropBall(int arr[][9], int row, int col, int
hasBorder, int& score) {
            int tmp[9][9];
            for (int i = 0; i < 9; i++)
             for (int j = 0; j < 9; j++)
                tmp[i][j] = arr[i][j];
            int x = 2, y = 2;
            for (int i = 0; i < row; i++)
             for (int j = 0; j < col; j++)
                if (removable(arr, i, j,
row, col)) {
                    for (int k = 0; k < 8;
k++) {
                        Sleep(50);
                        cct_showstr(x
+ j * 2 * (1 + hasBorder), y + i * (hasBorder + 1),
"○", arr[i][j], 0);
                        Sleep(50);
                        cct_showstr(x
+ j * 2 * (1 + hasBorder), y + i * (hasBorder + 1),
"¤", arr[i][j], 0);
                    }
                    tmp[i][j] = 0;
                    score++;
                    cct_showstr(x + j *
2 * (1 + hasBorder), y + i * (hasBorder + 1), "  ",
7, 7);
                }
            for (int i = col - 1; i >= 0; i--)
            {
             for(int j = 0; j<row;j++){
                if (tmp[j][i] == 0) {
                    for (int k = j; k > 0;
k--) {
                        tmp[k][i] =
tmp[k - 1][i];
                        Sleep(100);
                        if (tmp[k - 1][i]
== 0)

                cct_showstr(x + i * 2 * (1 +
hasBorder), y + k * (hasBorder + 1), "  ", 7, 7);
                        else
                        cct_showstr(x
+ i * 2 * (1 + hasBorder), y + k * (hasBorder + 1),
"○", tmp[k][i], 0);
                        cct_showstr(x
+ i * 2 * (1 + hasBorder), y + (k - 1) *
(hasBorder + 1), "  ", 7, 7);
                    }
                    tmp[0][i] = 0;
            }
            for (int i = 0; i < 9; i++)
```

装 订 线

```
                for (int j = 0; j < 9; j++)
                    arr[i][j] = tmp[i][j];
            cct_setcolor(0, 7);
            cct_gotoxy(0, 3 + row + 8 * hasBorder);
}
void fillBall(int arr[][9], int row, int col, int hasBorder) {
            int x = 2, y = 2;
            for (int i = 0; i < row; i++)
                for (int j = 0; j < col; j++)
                    if (arr[i][j] == 0) {
                        arr[i][j] = randomInt(1, 9);
                        cct_showstr(x + j * 2 * (1 + hasBorder), y + i * (hasBorder + 1), "○", arr[i][j], 0);
                        Sleep(100);
                    }
            cct_setcolor(0, 7);
            cct_gotoxy(0, 3 + row + 8 * hasBorder);
}
int hint(int arr[][9], int row, int col, int hasBorder)
{
            int rs[20][4];
            for (int i = 0; i < 20; i++)
                for (int j = 0; j < 4; j++)
                    rs[i][j] = -1;
            results(arr,rs,row,col);
            int x = 2, y = 2;
            for (int i = 0; i < 20; i++)
            {
                if (rs[i][0] == -1)
                    break;
                cct_showstr(x + rs[i][1] * 2 * (1 + hasBorder), y + rs[i][0] * (hasBorder + 1), "◎", arr[rs[i][0]][rs[i][1]], 0);
                cct_showstr(x + rs[i][3] * 2 * (1 + hasBorder), y + rs[i][2] * (hasBorder + 1), "◎", arr[rs[i][2]][rs[i][3]], 0);
            }
            cct_setcolor(0, 7);
            cct_gotoxy(0, 3 + row + 8 * hasBorder);
            return rs[0][0] != -1;
}
int positionValid(int arr[][9], int row, int col, int mX, int mY, int *v1, int *v2) {
            int flag = 0;
            for (int i = 0; i < row; i++)
            {
                for (int j = 0; j < col; j++)
                    if ((mX == 2 + j * 4 || mX == 2 + j * 4 + 1) && mY == 2 + i * 2) {
                        *v1 = i;
                        *v2 = j;
                        flag = 1;
                        break;
            return flag;
int canSelect(int arr[][9], int row, int col, int mX, int mY) {
            int flag = 0;
            int rs[20][4];
            for (int i = 0; i < 20; i++)
                for (int j = 0; j < 4; j++)
                    rs[i][j] = -1;
            results(arr,rs,row,col);
            for (int i = 0; i < 20; i++)
            {
                if (rs[i][0] == -1)
                    break;
                if (mX == 2 + rs[i][1] * 4 && mY == 2 + rs[i][0] * 2) {
                    flag = 1;
                    break;
                }
                if (mX == 2 + rs[i][3] * 4 && mY == 2 + rs[i][2] * 2) {
                    flag = 1;
                    break;
            return flag;
}
int select(int arr[][9], int row, int col, int *v1, int *v2) {
            int rs[20][4];
            for (int i = 0; i < 20; i++)
                for (int j = 0; j < 4; j++)
                    rs[i][j] = -1;
```

```
                results(arr, rs, row, col);
                cct_gotoxy(0, 2 + 2 * row);
                cct_enable_mouse();
                int mX, mY, mAction, kValue1, kValue2;
                while(true){

                cct_read_keyboard_and_mouse(mX, mY, mAction, kValue1, kValue2);
                 if (positionValid(arr, row, col, mX, mY, v1, v2)) {
                        cct_showch(0, 2 + 2 * row, ' ', 0, 7, 40);
                        cct_showstr(0, 2 + 2 * row, "[当前光标] ", 0, 7, -1);
                        cout << char(*v1 + 'A') << "行" << char(*v2 + '1') << "列";
                        if (mAction == MOUSE_LEFT_BUTTON_CLICK) {
                            if (canSelect(arr, row, col, mX, mY)) {
                                cct_showstr(2 + *v2 * 4, 2 + *v1 * 2, "◎", arr[*v1][*v2], 7);
                                cct_showch(0, 2 + 2 * row, ' ', 0, 7, 40);
                                cct_showstr(0, 2 + 2 * row, "当前选择", 0, 7, -1);
                                cout << char(*v1 + 'A') << "行" << char(*v2 + '1') << "列";
                                return 1;
                            }
                            else {
                                cct_showch(0, 2 + 2 * row, ' ', 0, 7, 40);
                                cct_showstr(0, 2 + 2 * row, "不能选择", 0, 7, -1);
                                cout << char(*v1 + 'A') << "行" << char(*v2 + '1') << "列";
                            }
                        }
                        else if (mAction == MOUSE_RIGHT_BUTTON_CLICK) {
                            return 0;
                    }
                    else {
                        cct_showch(0, 2 + 2 * row, ' ', 0, 7, 40);
                        cct_showstr(0, 2 + 2 * row, "[当前光标] 位置非法", 0, 7, -1);
void change(int arr[][9], int row, int col, int v1, int v2, int v3, int v4, int& score) {
                int rs[20][4];
                for (int i = 0; i < 20; i++)
                 for (int j = 0; j < 4; j++)
                        rs[i][j] = -1;
                results(arr, rs, row, col);
                for (int i = 0; i < 20; i++)
                {
                 if (rs[i][0] == -1) {
                        cct_showstr(2 + v2 * 4, 2 + v1 * 2, "◎", arr[v1][v2], 0);
                        cct_showstr(2 + v4 * 4, 2 + v3 * 2, "◎", arr[v3][v4], 7);
                        break;
                 }
                 if ((v1 == rs[i][0] && v2 == rs[i][1] && v3 == rs[i][2] && v4 == rs[i][3]) || (v1 == rs[i][2] && v2 == rs[i][3] && v3 == rs[i][0] && v4 == rs[i][1])) {
                        int t = arr[v1][v2];
                        arr[v1][v2] = arr[v3][v4];
                        arr[v3][v4] = t;
                        cct_showstr(2 + v2 * 4, 2 + v1 * 2, "◎", arr[v1][v2], 0);
                        cct_showstr(2 + v4 * 4, 2 + v3 * 2, "◎", arr[v3][v4], 0);
                        while (!isFinish(arr, row, col)) {

                showBall(arr,row,col,1,1);
                            Sleep(100);
                            dropBall(arr, row, col, 1, score);
                            Sleep(100);
                            fillBall(arr, row, col, 1);
                        break;
}
```