徽标

描述已自动生成

**完成时间：2024.6.16**

信16 2352495 张竹和

**高级语言程序设计**

**彩球游戏的设计与实现**

**实验报告**

1. **题目及基本要求描述**
   1. **整体及菜单页面要求**

**用伪图形界面方式完成彩球游戏。设置可消除球为实心，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。**

* 1. **菜单选项要求**

**菜单1要求：数组形式，根据行列输入生成初始状态并找出初始可消除项；**

**菜单2要求：数组形式，完成生成初始状态、找出初始可消除项、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，完成后还需查找是否有初始可消除项，如有需要反复进行；**

**菜单3要求：数组形式，先完成菜单2中的各个步骤，然后查找消除提示项（即相邻的可互换项）；**

**菜单4要求：伪图形界面，画出初始状态，彩球之间无分隔线；**

**菜单5要求：伪图形界面，画出初始状态，彩球之间有分隔线；**

**菜单6要求：伪图形界面，画出初始状态，找出初始可消除项，彩球之间无分隔线；**

**菜单7要求：伪图形界面，画出初始状态，找出初始可消除项，消除后显示消除提示，彩球之间有分隔线；**

**菜单8要求：伪图形界面，先完成菜单7中的各个步骤，支持鼠标，能正确判断出鼠标的行列位置，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；**

**菜单9要求：伪图形界面完整版，在菜单8的基础上完成，需要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. **整体设计思路**
   1. **总体思路**

**将各个功能分在各个自定义的处理具体需求的函数中；另外，如有需要再通过自定义函数进行集成并实现与用户的交互；通过核心函数将这些处理具体需求的函数集成在一起并与菜单选择的函数通过参数联系起来。**

* 1. **具体论述**

**程序的核心函数主要包括magic\_ball\_menu (菜单函数，管理用户的输入)、menu\_main (对各个输入进行处理以及调用函数选择的函数)。除去个别需要单独处理的菜单，主要围绕这两个函数进行补充；**

**程序的具体处理函数包括包括magic\_ball\_array\_initialize (初始化彩球数组的函数)、find\_y\_n\_draw\_first (查找可消除项并标记的函数)、find\_y\_n\_draw (查找可消除项并标记的函数，这个函数由于部分题的特殊要求进行了单独处理)、update\_array (有重复项可消除的数组下落函数)、update\_array\_new (有重复项可消除的数组消除后更新的函数)、search\_tips (查找消除提示项，即相邻的可互换项的函数)、draw\_table (根据数组的输入情况画表的函数)、draw\_windows (设置cmd窗口大小和字号等的函数)、draw\_inception (画初始状态，根据输入选择是否画分隔线的函数)、drop\_down (掉落效果的函数)、input\_end\_continue (输入end继续的函数)、read\_mouse (读鼠标的函数)、read\_mouse\_9 (读鼠标的函数，多一个形参数组用于记录选择的状态)；**

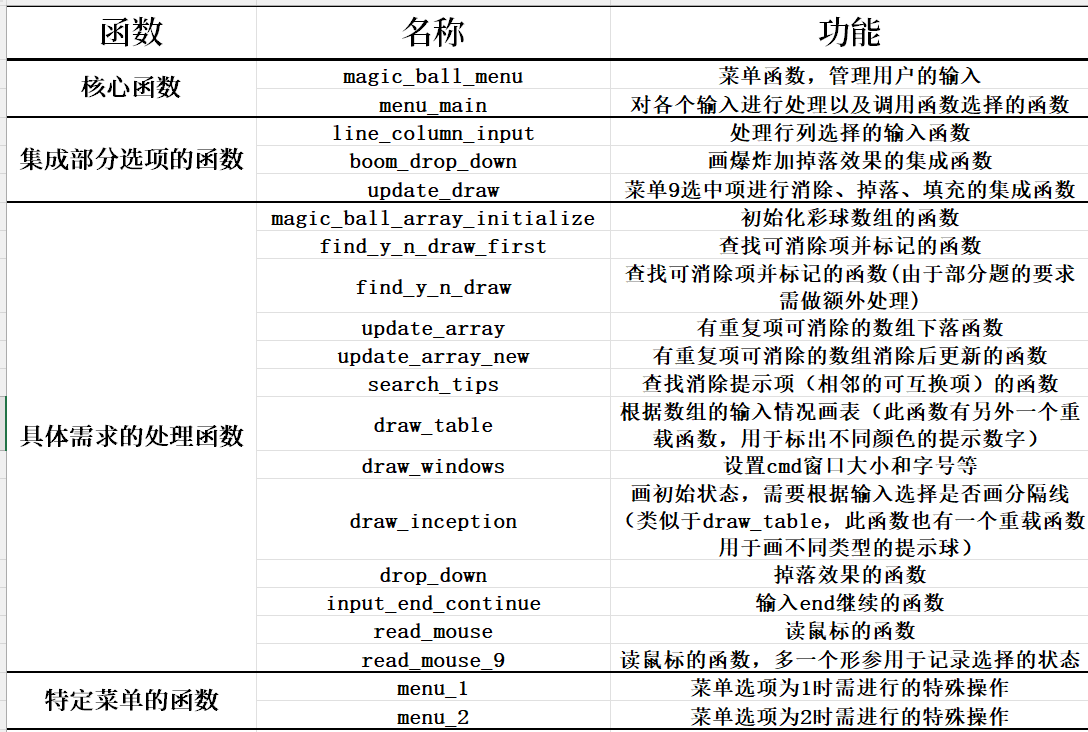
**集成部分选项的函数包括line\_column\_input (处理行列选择的输入函数)、boom\_drop\_down (画爆炸加掉落效果的集成函数)、update\_draw (选项9中进行消除、掉落、填充的集成函数)；**

**另外，由于部分菜单在循环以及循环内数组处理方面的特殊性，单独写了menu\_1 (菜单选项为1时的操作)、menu\_1(菜单选择为2时需进行的特殊操作)；**

**通过以上函数的相互调用等完成整体程序的实现。**

1. **主要功能的实现**
   1. **自定义函数**

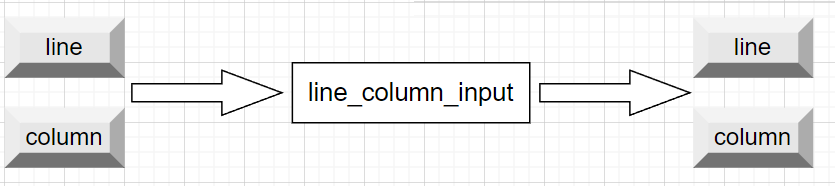
**为了更清楚的展现，首先将程序中出现过的所有自定义函数(除main)制表如下：**

****

* 1. **主要集成函数**

**集成部分函数的示意图与解释如下：**

**line\_column\_input函数根据输入改变line与column的值并传回；**

****

**boom\_drop\_down函数先根据数组的值在指定位置将原本的图案替换成爆炸图案，然后通过循环，针对每个需要下落的位置调用drop\_down函数实现伪图形界面上的下落效果；**

**图片包含 图示

描述已自动生成update\_draw函数主要包括菜单9中在鼠标选择并交换位置后的控制爆炸、消除、掉落、填充以及判断和可能进行的循环的函数**

* 1. **主要结构**

**程序开始，首先运行magic\_ball\_menu，等待用户输入菜单选择。**

**然后进入menu\_main。先执行所有菜单项均需进行的部分，这包括先进入line\_column\_input函数进行line与column的输入，然后进入magic\_ball\_array\_initialize函数根据行列的输入完成数组的随机数初始化；再根据菜单输入选择下一步，这之中尽量做了合并，单独的支路菜单项有1、2、3、8、9，合并处理的菜单项有4、5和6、7。**

**如果菜单项为1，先调用draw\_table绘制数组，然后直接调用menu\_1函数，该函数内包含调用find\_y\_n\_draw\_first函数进行初始可消除项的寻找，然后调用draw\_table进行绘制的一个循环。**

**如果菜单项为2，同样先调用draw\_table绘制数组，然后调用menu\_1函数寻找初始可消除项，如果有可消除项则调用menu\_2进行消除、下落和填充，menu\_2先后调用update\_array、draw\_table、update\_array\_new、draw\_table函数进行数组的掉落、填充，并在每一步结束进行绘制。**

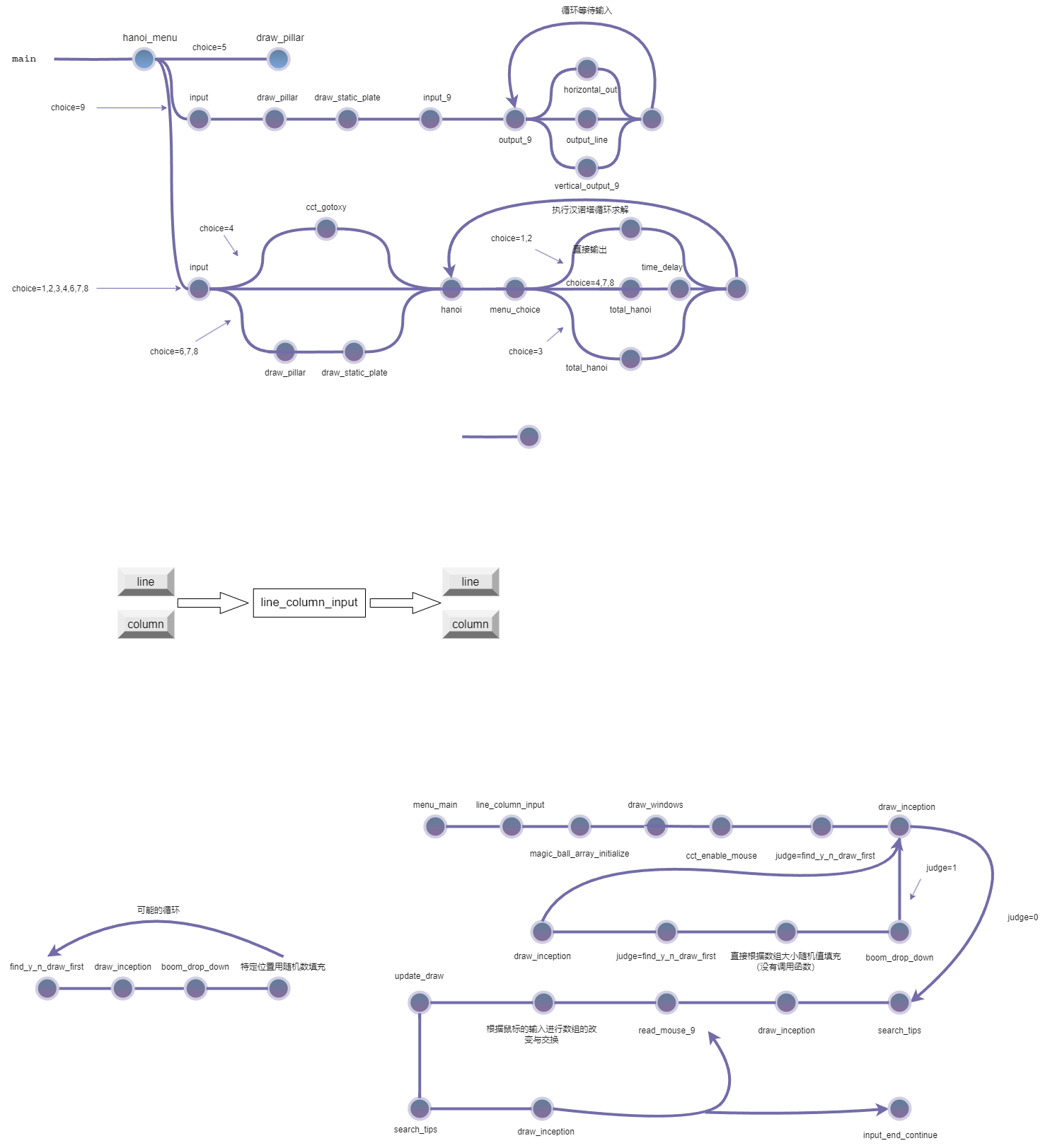
**如果菜单项为3，在进行完菜单项2的步骤后，运行search\_tips函数寻找可消除项的提示，并调用draw\_table进行绘制。**

**如果菜单项为4或5，先调用draw\_table绘制数组，然后调用draw\_windows设置窗口的一些参数，再根据菜单调用参数不同的draw\_inception函数绘制初始状态，参数决定了是否有分隔线。**

**如果菜单项为6或7，先执行4/5中的步骤（除绘制数组），然后调用find\_y\_n\_draw\_first判断可消除项。如果参数是6，直接draw\_inception绘制含可消除项提示的界面；如果参数是7，绘制完提示界面后，调用boom\_drop\_down进行掉落效果绘制，然后通过遍历数组的值在特定位置用随机值进行新值填充，再重新判断、绘制并循环。**

**如果菜单项为8，先进行菜单项为7的一系列步骤，然后调用search\_tips进行可消除提示项的寻找，调用draw\_inception绘制，再调用read\_mouse进行鼠标位置的读取与输出。**

**如果菜单项为9，先进行菜单项为8的一些步骤，在8的read\_mouse处改为进入读鼠标的一个循环，调用read\_mouse\_9读取鼠标，包括鼠标位置的输出，单击左键选择/取消选择的提示的输出等，然后将选择位置进行交换，并调用update\_draw进行爆炸、掉落、填充效果的伪图形界面显示，然后重新寻找可消除项与可消除提示并循环。**

**主要结构的流程图（选项9）如下：**

1. **调试过程碰到的问题**

**问题1：选项5画边框时坐标错误，导致伪图形界面出现混乱；**

**解决方案：先用F5开启调试，F9设置断点，在图形处用F11逐步运行，观察伪图形界面的输出，找出导致错误的语句，并从逻辑角度改进。**

**问题2：选项7中下落效果的处理，在逻辑及坐标、数组指针的值上存在问题导致期望的效果无法出现；**

**解决方案：同样用F5调试，F9设置断点，在进行到下落函数时，用F11逐步运行来体会函数的逻辑，并观察每步的参数是否符合预期，进行修改。**

1. **心得体会**
   1. **完成本次作业的心得体会与经验教训**
      1. **核心选择函数部分的合并**

**合并的部分没有做到最优，仍然有很多冗余代码，如菜单项1、2、3完全可以合并，并使用菜单项的参数选择后续进行操作与否，但本题考虑到每题输出的提示等有差异而没有合并，选择了更省事的分开写的方法。**

* + 1. **自定义函数的拆分**

**一个函数只包含自己的部分内容而不包含其他的内容，尽管这样可能导致传递参数在某一道题中有一些麻烦，如menu\_1中的**

**//画**

**if (judge == 1) {**

**cout << endl;**

**draw\_table(line, column, magic\_ball\_array, pp);**

**}**

**else if (judge == 0) {**

**cout << "初始已无可消除项" << endl;**

**}**

**//结束的函数**

**input\_end\_continue();**

**这一部分本来是放到find\_y\_n\_draw\_first函数中的，但为了后续题目考虑，每个函数都应该只执行自己的那部分，原则上拆的越细越好，但这样传递参数可能有点麻烦，所以需要折中一下。**

**同时由于各个小题之间有关系，这里的函数拆分也需谨慎。**

* + 1. **熟练使用VS中的debug工具**

**debug时灵活使用F5、F9、F11以及其他VS的debug工具可以大量节省时间，在对于循环的debug过程中，可以在程序调试中设置断点，这样可以直接从bug出现的那一次循环开始，操作可以为：在相应位置添加需要读取输入的代码(若无)，然后在程序等待输入的时候就可以在需要的位置设置断点，并使用F11进行逐步查验。另外，在F11逐步处理的时候可以在cmd窗口上看到实时的输出，对伪图形界面图形绘制中的坐标检查有很大帮助。**

* 1. **通过综合题1/2中有关函数的分解与使用，总结在完成过程中是否考虑了前后小题的关联关系，是否能尽可能做到后面小题有效利用前面小题已完成的代码，如何才能更好地重用？**

**本次完成过程中尽可能考虑了与后面题目的关联关系，也尽可能做到了对前面小题代码的有效利用，最后结果体现在选项9较快速地完成。**

**在对代码的利用方面，本次完成作业的总思路是：重复超过两处的代码则集成为函数，对于函数在后期改进时主要采用三种方式。第一种是在原函数中添加参数，进行个别步骤或者补充步骤的选择性执行；第二种是使用重载函数，在本次作业中主要使用的是添加形参的重载方法，主要用在部分函数在用于特定题目时需要传入额外数组的情况；第三种就是直接复制创建新函数，并进行修改，这种主要用在原函数修改不太方便，但重建函数修改比较快的情况，但个人感觉不太好，容易导致代码臃肿，所以尽量的少用了。**

1. **附件：源程序（注：源程序部分主要是菜单9的实现程序）**

**考虑到源程序的页数问题将标题放在本页，并且删掉了函数启示处的一个换行。**

void menu\_main(int choice\_menu)

{ if (choice\_menu != 0) {

int line = 0, column = 0;

line\_column\_input(&line, &column);

int magic\_ball\_array[11][11] = { 0 };

int pp[10][10] = { 0 };

magic\_ball\_array\_initialize(magic\_ball\_array, line, column);

int score = 0;

else if (choice\_menu == 9) {

int re = 0;

int choose[10][10] = { 0 };

draw\_windows(line);

cct\_enable\_mouse();

int judge = 0;

judge = find\_y\_n\_draw\_first(magic\_ball\_array, line, column, pp);

draw\_inception(line, column, magic\_ball\_array, 1, pp);

cct\_gotoxy(14, 0);

cout << "(当前分数:0)";

if (judge == 1) {

while (judge == 1) {

boom\_drop\_down(line, column, magic\_ball\_array, pp, &score);

for (int i = 1; i <= line; i++) {

for (int j = 1; j <= column; j++) {

if (magic\_ball\_array[i][j] == 0)

magic\_ball\_array[i][j] = rand() % 9 + 1;

}

}

for (int i = 0; i <= line; i++) {

for (int j = 0; j <= column; j++) {

pp[i][j] = 0;

}

}

judge = find\_y\_n\_draw\_first(magic\_ball\_array, line, column, pp);

cct\_gotoxy(0, 1);

draw\_inception(line, column, magic\_ball\_array, 1, pp);

}

}

re = search\_tips(line, column, magic\_ball\_array, pp);

cct\_gotoxy(0, 1);

draw\_inception(line, column, magic\_ball\_array, 1, pp);

score = 0;

while (1) {

int tt = read\_mouse\_9(line, column, magic\_ball\_array, pp, choose);

if (tt == 1) {

cout << endl;

break;

}

for (int i = 1; i <= line; i++) {

for (int j = 1; j <= column; j++) {

if (choose[i][j] == 1) {

if (choose[i + 1][j] == 1) {

int t = magic\_ball\_array[i][j];

magic\_ball\_array[i][j] = magic\_ball\_array[i + 1][j];

magic\_ball\_array[i + 1][j] = t;

update\_draw(line, column, magic\_ball\_array, pp, &score);

choose[i][j] = 0;

choose[i + 1][j] = 0;

}

else if (choose[i][j + 1] == 1) {

int t = magic\_ball\_array[i][j];

magic\_ball\_array[i][j] = magic\_ball\_array[i][j + 1];

magic\_ball\_array[i][j + 1] = t;

update\_draw(line, column, magic\_ball\_array, pp, &score);

choose[i][j] = 0;

choose[i][j + 1] = 0;

}

else

continue;

}

int x, y;

cct\_getxy(x, y);

cct\_gotoxy(14, 0);

cout << "(当前分数:" << score << ") 右键退出";

cct\_gotoxy(x, y);

re = search\_tips(line, column, magic\_ball\_array, pp);

}

}

if (re == 1) {

cct\_gotoxy(0, 1);

draw\_inception(line, column, magic\_ball\_array, 1, pp);

}

else if (re == 0)

break;

}

}

input\_end\_continue();

}

}

int read\_mouse\_9(const int line, const int column, int magic\_ball\_array[][11], int pp[][10], int choose[][10])

{ int MX, MY, MAction, keycode1, keycode2;

cout << "[当前光标]";

int x, y;

cct\_getxy(x, y);

while (1) {

cct\_read\_keyboard\_and\_mouse(MX, MY, MAction, keycode1, keycode2);

char H\_line[] = { ' ','A','B','C','D','E','F','G','H','I' };

int L\_col[] = { 0,1,2,3,4,5,6,7,8,9 };

if ((MY != 0) && (MY % 2 == 0)) {

if ((MX + 2) % 4 == 0) {

MX = (MX + 2) / 4;

MY = MY / 2;

if ((MX <= 9) && (MY <= 9)) {

cct\_gotoxy(x, y);

cout << " ";

cct\_gotoxy(x, y);

cout << H\_line[MY] << "行" << L\_col[MX] << "列";

}

else {

cct\_gotoxy(x, y);

cout << "位置非法";

}

}

else {

cct\_gotoxy(x, y);

cout << "位置非法";

}

}

else {

cct\_gotoxy(x, y);

cout << "位置非法";

}

if ((MAction == MOUSE\_LEFT\_BUTTON\_CLICK) && ((MX <= 9) && (MY <= 9))) {

if (pp[MY][MX] == 2) {

cct\_gotoxy(0, y);

cout << " ";

cct\_gotoxy(0, y);

if (choose[MY][MX] == 0) {

cout << "当前选择" << H\_line[MY] << "行" << L\_col[MX] << "列";

Sleep(500);

choose[MY][MX] = 1;

}

else if (choose[MY][MX] == 1) {

cout << "取消选择";

Sleep(500);

choose[MY][MX] = 0;

}

break;

}

else {

cct\_gotoxy(0, y);

cout << " ";

cct\_gotoxy(0, y);

cout << "不能选择" << H\_line[MY] << "行" << L\_col[MX] << "列";

Sleep(500);

cct\_gotoxy(0, y);

cout << " ";

cct\_gotoxy(0, y);

cout << "[当前光标]";

continue;

}

}

if (MAction == MOUSE\_RIGHT\_BUTTON\_CLICK)

return 1;

}

return 0;

}

void update\_draw(const int line, const int column, int magic\_ball\_array[][11], int pp[][10], int\* score)

{ int judge = 0;

judge = find\_y\_n\_draw\_first(magic\_ball\_array, line, column, pp);

cct\_gotoxy(0, 1);

draw\_inception(line, column, magic\_ball\_array, 1, pp);

if (judge == 1) {

while (judge == 1) {

boom\_drop\_down(line, column, magic\_ball\_array, pp, score);

for (int i = 1; i <= line; i++) {

for (int j = 1; j <= column; j++) {

if (magic\_ball\_array[i][j] == 0)

magic\_ball\_array[i][j] = rand() % 9 + 1;

}

}

for (int i = 0; i <= line; i++) {

for (int j = 0; j <= column; j++) {

pp[i][j] = 0;

}

}

judge = find\_y\_n\_draw\_first(magic\_ball\_array, line, column, pp);

cct\_gotoxy(0, 1);

draw\_inception(line, column, magic\_ball\_array, 1, pp);

}

}

}

int search\_tips(const int line, const int column, int p[][11], int pp[][10])

{ int judge = 0;

int re = 0;

for (int i = 1; i <= line; i++) {

for (int j = 1; j <= column; j++) {

const int temporary = p[i][j];

//上

p[i][j] = p[i - 1][j];

p[i - 1][j] = temporary;

judge = find\_y\_n\_draw(p, line, column, pp);

if (judge == 1) {

pp[i][j] = 2;

pp[i - 1][j] = 2;

re = 1;

}

p[i - 1][j] = p[i][j];

p[i][j] = temporary;

//下

p[i][j] = p[i + 1][j];

p[i + 1][j] = temporary;

judge = find\_y\_n\_draw(p, line, column, pp);

if (judge == 1) {

pp[i][j] = 2;

pp[i + 1][j] = 2;

re = 1;

}

p[i + 1][j] = p[i][j];

p[i][j] = temporary;

//左

p[i][j] = p[i][j - 1];

p[i][j - 1] = temporary;

judge = find\_y\_n\_draw(p, line, column, pp);

if (judge == 1) {

pp[i][j] = 2;

pp[i][j - 1] = 2;

re = 1;

}

p[i][j - 1] = p[i][j];

p[i][j] = temporary;

//右

p[i][j] = p[i][j + 1];

p[i][j + 1] = temporary;

judge = find\_y\_n\_draw(p, line, column, pp);

if (judge == 1) {

pp[i][j] = 2;

pp[i][j + 1] = 2;

re = 1;

}

p[i][j + 1] = p[i][j];

p[i][j] = temporary;

}

}

for (int i = 1; i <= line; i++) {

for (int j = 1; j <= column; j++) {

if (pp[i][j] == 1) {

pp[i][j] = 0;

}

}

}

return re;

}

void draw\_inception(const int line, const int column, const int p[][11], int choose\_frame, const int pp[][10])

{ cct\_setcolor(COLOR\_HWHITE, COLOR\_BLACK);

if (choose\_frame == 0) {

for (int i = 0; i <= line + 1; i++) {

if (i == 0)

cout << "┏";

else if (i == line + 1)

cout << "┗";

else

cout << "┃";

for (int j = 1; j <= column + 1; j++) {

if (i == 0) {

if (j == column + 1)

cout << "┓" << endl;

else

cout << "━";

}

else if (i == line + 1) {

if (j == column + 1)

cout << "┛" << endl;

else

cout << "━";

}

else {

if (j == column + 1)

cout << "┃" << endl;

else {

cct\_setcolor(p[i][j], COLOR\_BLACK);

if (pp[i][j] == 0)

cout << "○";

else if (pp[i][j] == 1)

cout << "●";

cct\_setcolor(COLOR\_HWHITE, COLOR\_BLACK);

}

}

}

}

}

else if (choose\_frame == 1) {

int m = 1, n = 1;

for (int i = 0; i <= 2 \* line; i++) {

if (i == 0)

cout << "┏";

else if (i == 2 \* line)

cout << "┗";

else if ((i % 2) == 0)

cout << "┣";

else

cout << "┃";

for (int j = 1; j <= 2 \* column; j++) {

if (i == 0) {

if (j == 2 \* column)

cout << "┓" << endl;

else if ((j % 2) == 0)

cout << "┳";

else

cout << "━";

}

else if (i == 2 \* line) {

if (j == 2 \* column)

cout << "┛" << endl;

else if ((j % 2) == 0)

cout << "┻";

else

cout << "━";

}

else if ((i % 2) == 0) {

if (j == 2 \* column)

cout << "┫" << endl;

else if ((j % 2) == 0)

cout << "╋";

else

cout << "━";

}

else {

if ((j % 2) == 1) {

cct\_setcolor(p[m][n], COLOR\_BLACK);

if (pp[m][n] == 0)

cout << "○";

else if (pp[m][n] == 1)

cout << "●";

else if (pp[m][n] == 2)

cout << "◎";

cct\_setcolor(COLOR\_HWHITE, COLOR\_BLACK);

if (n < column)

n++;

else if (n == column) {

m++;

n = 1;

}

}

if (j == 2 \* column)

cout << "┃" << endl;

else if ((j % 2) == 0)

cout << "┃";

}

}

}

}

cct\_setcolor();

}

void boom\_drop\_down(const int line, const int column, int p[][11], int pp[][10], int\* score)

{ int x = 0, y = 0;

cct\_getxy(x, y);

for (int i = 1; i <= line; i++) {

for (int j = 1; j <= column; j++) {

if (pp[i][j] == 1) {

cct\_gotoxy(2 + 4 \* (j - 1), 2 \* i);

cct\_setcolor(p[i][j], COLOR\_BLACK);

cout << "¤";

(\*score)++;

Sleep(500);

cct\_gotoxy(2 + 4 \* (j - 1), 2 \* i);

cct\_setcolor(COLOR\_HWHITE, COLOR\_BLACK);

cout << " ";

Sleep(200);

}

}

}

int ppp[10][10];

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

ppp[i][j] = pp[i][j];

}

}

for (int j = 1; j <= column; j++) {

if (ppp[1][j] == 1)

p[1][j] = 0;

}

cct\_setcursor(CURSOR\_INVISIBLE);

for (int j = 1; j <= column; j++) {

for (int i = 1; i <= line; i++) {

if (ppp[i][j] == 1) {

if (i > 1) {

drop\_down(line, column, p, ppp, i, j);

i = i - 2;

}

}

}

}

cct\_gotoxy(x, y);

cct\_setcolor();

}

void drop\_down(const int line, const int column, int p[][11], int ppp[][10], int i, int j)

{ int m = 4 \* j - 2;

for (int n = 2 \* i - 2; n <= 2 \* i - 1; n++) {

cct\_gotoxy(m, n);

cct\_setcolor(COLOR\_HWHITE, COLOR\_BLACK);

if (n % 2 == 1)

cout << "━";

else

cout << " ";

cct\_gotoxy(m, n + 1);

if (p[i - 1][j] != 0) {

cct\_setcolor(p[i - 1][j], COLOR\_BLACK);

cout << "○";

}

Sleep(50);

}

ppp[i - 1][j] = 1;

ppp[i][j] = 0;

p[i][j] = p[i - 1][j];

p[i - 1][j] = 0;

}