2152667 李奥

1. 整体设计思路

一局游戏所需要的时间和棋盘的大小密切相关，玩家可以根据自己的需要选择棋盘的大小，由于需要“五子相连”，设置棋盘的行列数为7/8/9，用户可以自行选择行数和列数，一共会有7\*7,7\*8,7\*9,8\*7,8\*8,8\*9,9\*7,9\*8,9\*9这9种棋盘可以供用户选择，为了优化体验，采用鼠标点击的方式选择行列值。

在游戏进行过程中，玩家要能够实时地看到棋盘上珠子的信息，所以在游戏界面需要有一个信息框，用以给出棋盘上6种颜色的小球的数量，为了使其更美观，信息更具有参考性，同时给出小球数量的占比。玩家在界面上移动鼠标时，如果鼠标移动到棋盘上时，显示出鼠标的坐标，即显示出鼠标的位置是棋盘的第几行、第几列。

玩家在意的往往是得分，显示分数是必然的，设置一个分数栏用于实时地显示分数。在游戏过程中统计并且显示6种颜色小球的消去数量，玩家可以根据消去小球的数量自己计算出得分。对于玩家来说，下一次生成的三个球是什么颜色影响着整体局势的判断，但是又不能给出太多的信息，所以在当前轮次显示出下一轮将加入到棋盘的三个球的颜色，而不会给出这第三个球的具体位置，通过添加一个“球框”来显示下一轮加入棋盘的三个球。

为了使得用户在游戏过程中体验更佳，背景音乐必不可少，同时也不能只有一首歌作为背景音乐，这样的话过于枯燥。所以游戏至少会有两种背景音乐，每一局游戏的背景音乐到底是什么是随机产生的，同时将背景音乐的名字显示在界面的一角，方便用户知晓。

在游戏进行过程中，通过鼠标点击来移动珠子，点击左键选中珠子或者目的地进行移动，点击右键退出本局。通过循环来重复进行操作，循环的跳出条件有两个，一个是点击右键手动退出，另外一个是棋盘满，无空闲位置，这时会自动退出循环。

用户可以进行在启动游戏之后进行多局游戏，当某一局游戏结束之后，玩家可以选择再来一局，也可以点击退出游戏。

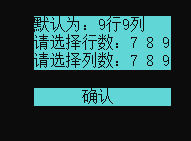
玩家会比较关心自己的历史得分，程序可以记录最高的三次得分，并且在游戏开始和游戏结束的时候显示、更新得分排行榜，主要通过文件的读写实现这一功能。

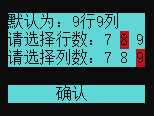
1. 功能描述

**2.1 选择棋盘**

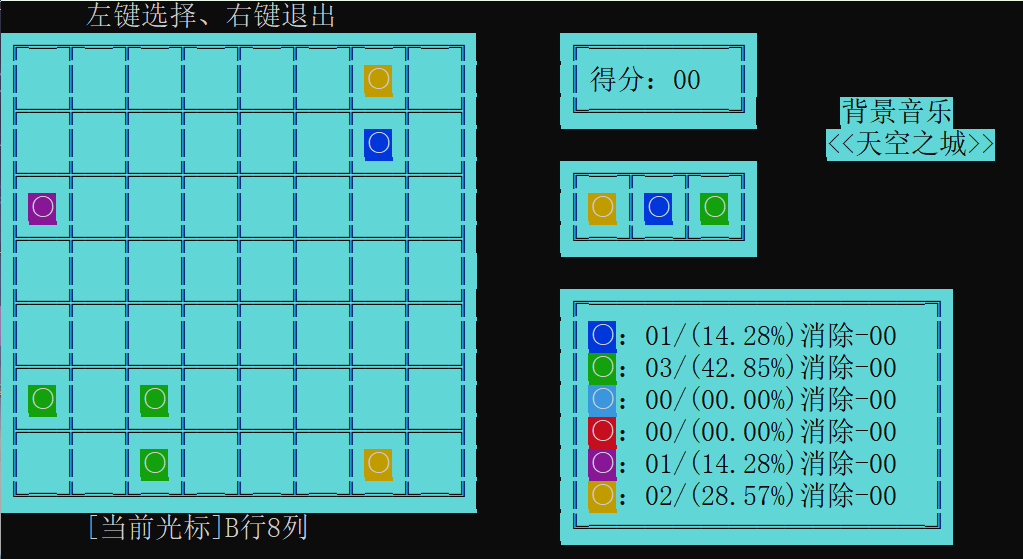
进入游戏之后，会看到选择行列值的框，如下图，：

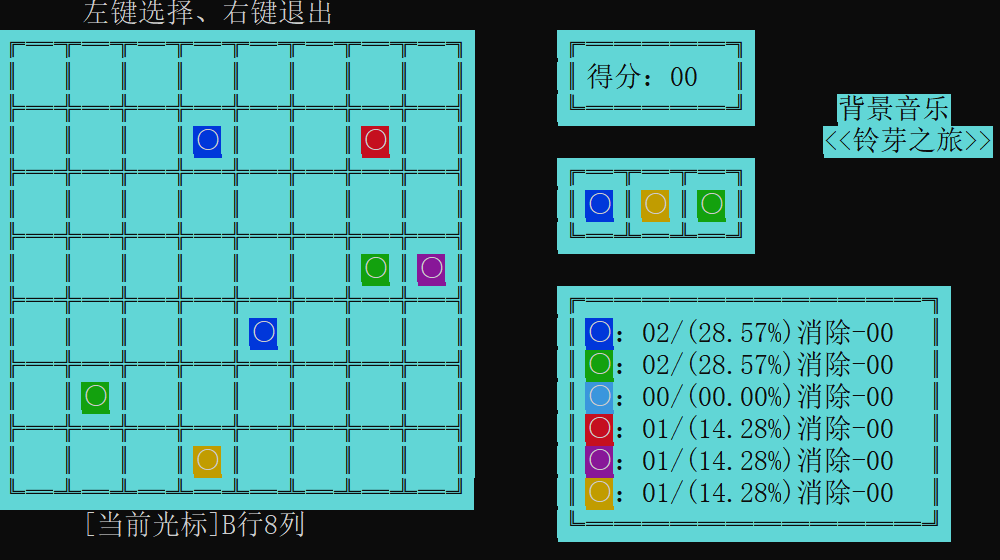
默认棋盘是9行9列，当直接点击“确认”按钮时，棋盘就是9\*9。

也可以自定义行列值。如下图所示，使用鼠标点击行数和列数的数字便可以选择，选择的行列数会变成红色背景以表示区分，这时点击便可以得到8\*9的棋盘。



**2.2 背景音乐**

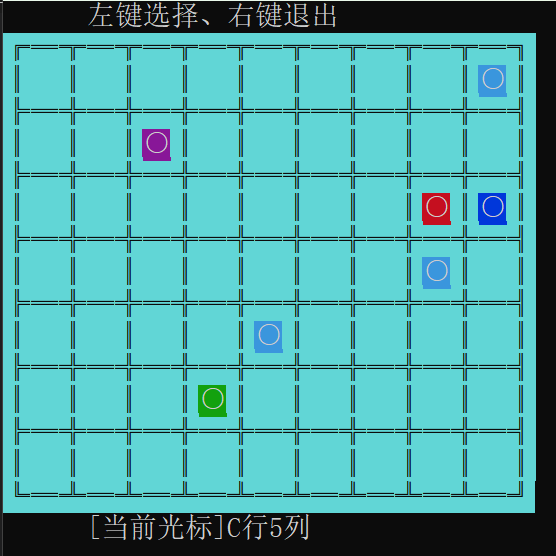
当开始某一局的游戏时，会有背景音乐增加游戏体验。程序添加了两首背景音乐，分别为《铃芽之旅》和《天空之城》。具体是哪一首音乐随机选择。本局游戏的背景音乐将会显示在界面的右上角，用户可以查看。如下面两张图所示：



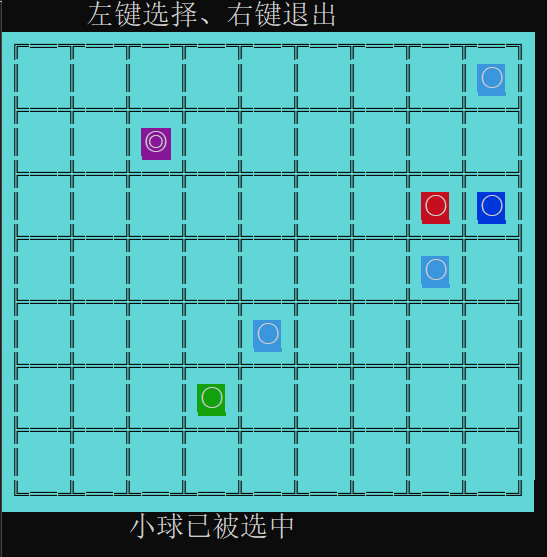
**2.3 移动珠子**

棋盘上会有6种不同颜色的珠子，移动珠子是游戏的核心操作。

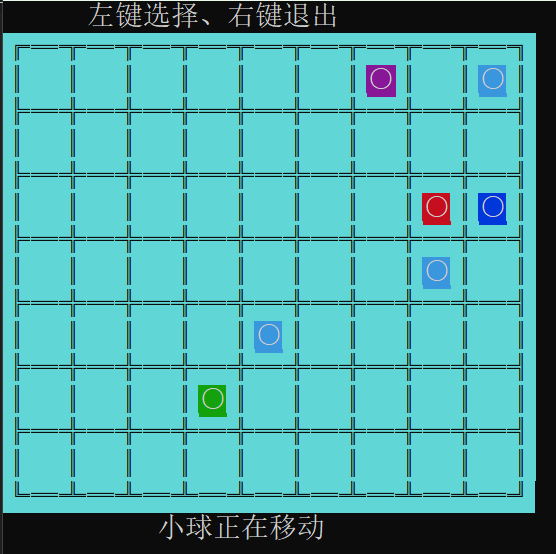
下图为一局游戏的棋盘：

在棋盘中，最上方的“左键选择、右键退出”是游戏的规则提示，当移动鼠标的光标时，会在棋盘下方显示光标所在的棋盘的坐标，上图的光标位置在“C行5列”,由于图片来源是电脑截图，鼠标光标没有进入截图中。

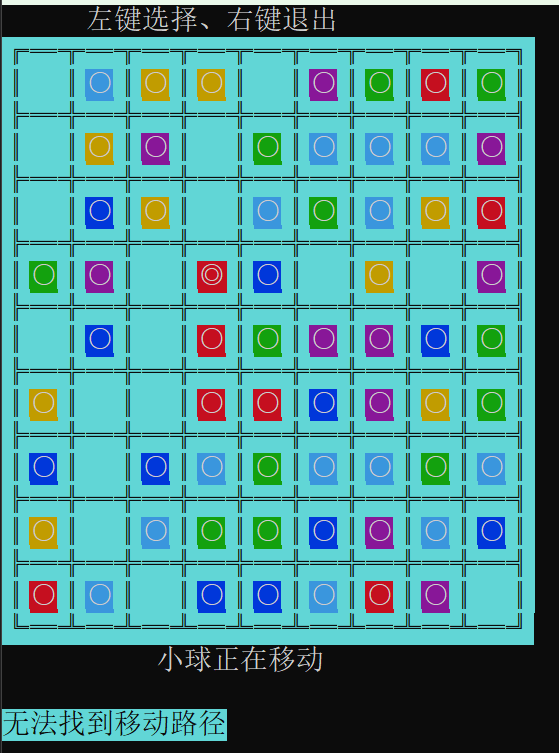
点击某一个小球，即可选中这个小球，同时小球的形状也会发生变化，让用户可以更清楚地看到选中的小球是哪一个，同时棋盘下方给出提示：“小球已被选中”。用户也可以继续选中其他的小球，相应的“选中效果”也会发生变化。如下图所示，第二行第三列的小球已经被选中，小球由“单圈”变成了“双圈”，并显示“小球已被选中”。



选中小球之后，点击棋盘的其他位置，如果点击的是其他小球，则选中其他小球，如果点击的是空位置，则将把选中的小球移动到此空位置，同时棋盘下方给出提示：“小球正在移动”。小球会按照一定的路径动态地移动到目的地。

之后程序会判断是否出现“五连珠”（单独判断四种方向，是否有5个及以上的同颜色珠子相连），出现“五连珠”则消去这些连在一起的珠子，并且根据消去珠子的数量获得得分，然后可以继续选择珠子进行下一轮的移动；如果没有出现“五连珠”，则会将三个新的珠子加入到棋盘中，然后才可以进入下一轮的移动。

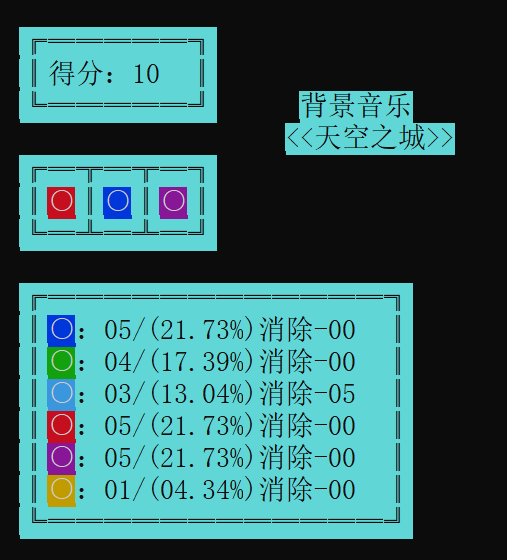
如果选中的珠子无法到达目的地，即没有移动路径，则会提示“无法找到移动路径”，如下图所示，意味着本轮的移动没能成功，本轮结束。棋盘将添加3个小球，进入下一轮的移动。



**2.4 查看信息**

界面的右侧是各种信息栏。

如下图所示：



得分框会显示这局游戏到现在为止的总得分。

下面的球框中有3个珠子，这三个珠子是下一轮即将加入棋盘的珠子，如果本次移动没有出现“五连珠”，这三个珠子将随机加入到棋盘的空闲位置。

最下面的一个框是统计数据，6种颜色的珠子、棋盘上各种颜色珠子的数量、相应的占比和消除的数量。

如上图所示，当前的棋盘中有5个蓝色珠子，4个绿色珠子，3个浅蓝色珠子，5个红色珠子，5个紫色珠子，1个黄色珠子。已经消除了5个红色珠子，得分为10分，即将加入棋盘的三个珠子的颜色分别是红色、蓝色、紫色。

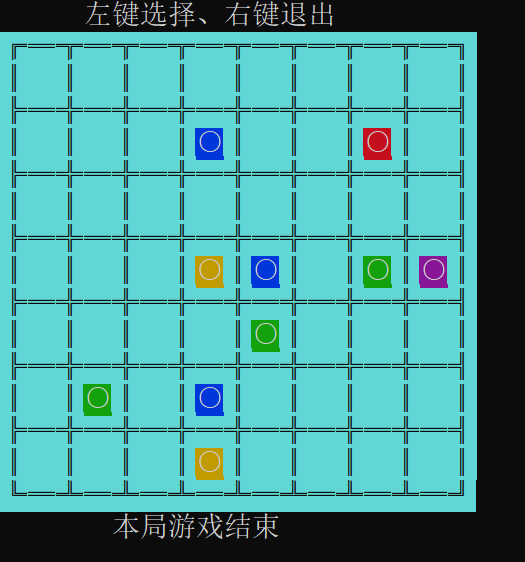
**2.5 退出本局**

退出本局有两种方式，一种是主动退出，另一种是棋盘已满，被动退出。

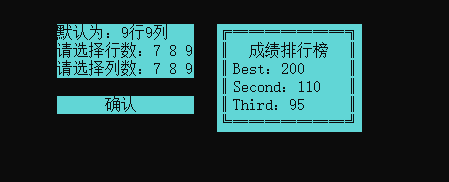
如下图，移动之后并没有消去小球，此时棋盘上只剩3个空闲位置，将有3个小球加入棋盘，加入之后会出现棋盘全满、无法移动的现象，所以在加入3个小球之前就可以判断游戏已经结束了，这时会在界面下方显示“游戏结束”。这是被动结束的情况。



在游戏当中，点击鼠标右键（前提条件是没有小球选中，小球不在移动状态中），即可直接退出本局。如下图所示，当前状态下棋盘并没有满，棋盘下方会显示“本局游戏结束”。等待2秒钟之后，退出本局，回到游戏的初始界面。



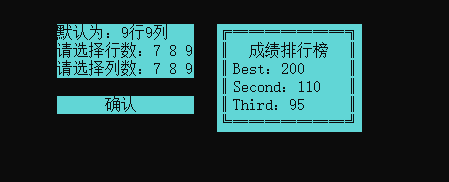
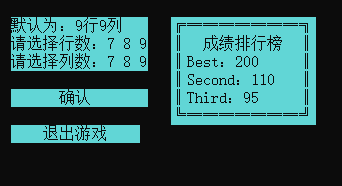
**2.6 查看排行榜**

当进入游戏时，或者依据游戏结束后，会在初始界面看到成绩排行榜。如下图：  
 游戏会记录最好的三次成绩，并且按照“Best”、“Second”、“Third”的顺序显示出来。记录的成绩是本地成绩，不涉及联网其他玩家的成绩。

**2.7 退出游戏**

退出游戏指的是关闭游戏，而不是指的是退出某一局游戏。

游戏的初始选择界面有两种，如下图所示，当第一次进入游戏时会显示左边的界面，之后完成某局游戏后会返回右边的界面。当玩家想要退出游戏时，一般的流程是：点击鼠标右键退出本局游戏，在初始界面点击“退出游戏”，即可退出游戏。



1. 模块设计

**3.1 结构体**

创建结构体direction，用于在寻找珠子移动路径时给出坐标变化。

struct direction//记录去某个方向时，横纵坐标的变化值

{

int a;//横坐标变化量

int b;//纵坐标变化量

};

使用时，使用结构体数组move，记录上下左右移动时的坐标变化。

direction move[4] = { {-1,0},{0,1},{0,-1},{1,0} };//四个方向的坐标变化值

**3.2游戏类**

创建游戏类，名字为five\_game。

类的私有成员有map[9][9]，元素值为0说明当前位置为空，值为1~6说明当前位置有珠子，珠子的颜色用数字一一对应，对应规则是：

|  |  |
| --- | --- |
| 数字 | 颜色 |
| 1 | 蓝色 |
| 2 | 绿色 |
| 3 | 浅蓝色 |
| 4 | 红色 |
| 5 | 紫色 |
| 6 | 黄色 |

私有成员数组record[3]用于暂时记录个人最好的三次成绩。私有成员还有整形数hang和lie，分别是棋盘的函数和列数，一个整形数Music\_num记录本局游戏的音乐，有0和1两种音乐，值为0时背景音乐是《铃芽之旅》，值为1时背景音乐为《天空之城》。整形数total\_score记录本局的得分。

public下的的各种函数以及类的完整内容如下：  
class five\_game

{

private:

int map[9][9];//最大是9\*9的棋盘

int record[3];//记录个人历史最好的十次成绩

int hang;//实际行数，在7/8/9中选择

int lie;//实际列数，在7/8/9中选择

int Music\_num;//本局游戏的音乐，暂时有0和1两种音乐

int total\_score;//依据游戏的成绩

public:

five\_game();//构造函数

void show\_music();//显示当前的背景音乐

void best\_record(int a);//记录本次游戏的成绩，更新排行榜

bool restore();//用于开始下一次游戏时将某些数据恢复到初始值

void Strat();//游戏开始函数

int fun();//游戏过程控制函数

void Music();//音乐播放函数

void show\_score();//打印出成绩排行榜

bool choose(int flag);//选择棋盘的规格以及是否退出游戏

void Color(int next\_3[3]);//给出下一轮即将产生的3个小球的颜色

void Score();//显示出得分栏

void xinxi(int now[10], int die[10]);//打印出棋盘中各种颜色的小球的信息，包括现有数量、占比，已经消去的数量

void fenjie(float i, int& a1, int& a2, int& a3, int& a4);//将float类型的小数（小球数量占比）分解成四位整数

int mouse();//鼠标控制函数

int judge(int remove[]);//判断四个方向能否删除，并返回得分，并且通过参数数组返回消去的小球的具体颜色和具体数量

void find\_num(int now[7]);//统计6种小球的数量

int Isfull();//判断棋盘是不是满的，游戏的终止条件之一

void writeArrayToFile(const std::string& filename);//将游戏得分写入文件

void readArrayFromFile(const std::string& filename);//从文件中读取排行榜数据

void shape2();//打印出五子连珠的棋盘

void add( int next\_3[3]);//在棋盘上添加3个小球

void donghua(int startx, int starty, int printff[]);//动态模拟小球移动过程

void move\_qiu( int startx, int starty, int endx, int endy);//小球移动的控制函数

int movepath1( int mark[9][9], int printff[50], int num, int startx, int starty, int s, int t, int endx, int endy, direction move[]);//回溯法寻找路径

};

**3.3 主函数**

通过主函数来生成类的对象并控制游戏的开始与结束，主函数如下：

int main()

{

five\_game A;//生成类的对象

while (1)

{

A.Strat();

int temp=A.restore();

if (temp == 0)

break;

}

return 0;

}  
**3.4 寻找路径算法**

对于hang\*lie 的棋盘，给出起点和终点，如何找到一条移动路径，可以通过回溯法解决。

进入寻找路径函数movepath1后，首先判断当前位置是否是终点，是的话直接返回1，不是终点则对四个方向进行试探，依次进入四个方向进行试探，试探某个方向时先判断这个方向对应的第一个位置是否可走，即判断是不是没有珠子的空位置，是否已经走过，有没有越过棋盘的边界。条件都满足之后，标记位置为“走过”，然后进入这个方向的movepath1函数，如果该函数返回值为1，则将该位置计入路径中，返回1，如果返回值为0，则进入下一个方向进行试探。如果四个方向都试探结束，说明从当前位置无法找到路径，返回0。如果最后的返回值为0，说明无路径，返回值为0，路径保存在printff数组中。

代码如下：

movepath1(int mark[9][9], int printff[MAX], int num, int startx, int starty, int s, int t, int endx, int endy, direction move[])

{

int i, m, p;

if (startx == endx && starty == endy)//已经到达终点

{

return 1;

}

else//未到达终点

{

for (i = 0; i < 4; i++)//对四个方向依次进行试探

{

m = startx + move[i].a;

p = starty + move[i].b;//坐标变化

if (map[m][p] == 0 && mark[m][p] == 0 && 0 <= m && m < hang && p < lie && 0 <= p)//表明可以走

{

mark[startx][starty] = 1;

int temp = movepath1( mark, printff, num + 2, m, p, s, t, endx, endy, move);

if (temp != 0)

{

printff[num++] = m;

printff[num++] = p;//记录路径

return 1;

}

else

{

continue;

}

}

}

if (i == 4)//四个方向都已经试探完成

{

return 0;

}

}

return 0;

}

**3.5 动态移动模块**

根据起点、终点、保存有移动路径的printff数组，便可以对珠子进行动态的移动了。

printff中记录的是相对坐标，并且是连续记录，即(printff[0],printff[1])是一个位置，(printff[2],printff[3])是一个位置。

首先进行位置变换，将相对位置转变成控制台界面的绝对位置。根据前后两个位置的纵坐标变化就可以得知是上下移动还是左右移动，如果纵坐标位置相同，说明横坐标一定不同，这时是上下移动，纵坐标位置不同，则是左右移动。

移动过程中珠子会先到达两个位置之间的边框，然后进入下一个位置，中间有时间暂停，以达到动态的效果。

主要代码如下：

While(1){  
 x1 = printff[i + 1] \* 4 + 2;

y1 = printff[i] \* 2 + 2;

x = printff[i - 1] \* 4 + 2;

y = printff[i - 2] \* 2 + 2;

if ((x + x1) / 2 != x)//左右移动

{

cct\_showstr(x, y, " ", COLOR\_HCYAN, COLOR\_HCYAN, 1);

cct\_showstr((x + x1) / 2, y, "○", map[startx][starty], COLOR\_WHITE, 1);

Sleep(Sleep\_time);

cct\_showstr((x + x1) / 2, y, "║", COLOR\_HCYAN, COLOR\_BLACK, 1);

cct\_showstr(x1, y, "○", map[startx][starty], COLOR\_WHITE, 1);

Sleep(Sleep\_time);

}

else//上下移动

{

cct\_showstr(x, y, " ", COLOR\_HCYAN, COLOR\_HCYAN, 1);

cct\_showstr(x, (y + y1) / 2, "○", map[startx][starty], COLOR\_WHITE, 1);

Sleep(Sleep\_time);

cct\_showstr(x, (y + y1) / 2, "═", COLOR\_HCYAN, COLOR\_BLACK, 1);

cct\_showstr(x, y1, "○", map[startx][starty], COLOR\_WHITE, 1);

Sleep(Sleep\_time);

}

i = i + 2;

}

**3.6 鼠标控制模块**

游戏的整个流程都是通过鼠标来进行操作的，通过鼠标选择行列值、开始游戏、移动珠子、结束本局、结束游戏等。

以移动珠子为例，描述鼠标操作的实现流程。通过循环一直读取鼠标的相关信息，MX, MY记录鼠标的光标的绝对位置，将绝对位置进行转换，如果光标的相对位置在棋盘上，将会实时显示光标的行列，如“[当前光标]C行3列”。 MAction记录鼠标的操作， MAction== MOUSE\_RIGHT\_BUTTON\_CLICK时，说明鼠标点击了右键，意思是“退出本局游戏”，则跳出循环，结束本局。如果MAction == MOUSE\_LEFT\_BUTTON\_CLICK，说明点击了左键，这时MX, MY将记录点击的位置，这个位置是界面的绝对位置，然后对绝对位置进行判断，如果是在棋盘之外、棋盘的框架或者是棋盘的空位置上，不会有反应，如果点击的位置有小球，则会选中小球，选中小球之后点击右键将不会退出游戏，可以再次点击左键更换选中的小球，然后点击空位置表示“将选中的小球移动到空位置”，这时选中的小球的位置以及空位置将作为起点和终点，珠子将进行动态的移动。

**3.7 音乐模块**

添加头文件

#pragma comment(lib, "winmm.lib")

通过Music\_num来先选择具体的音乐，使用PlaySound函数循环播放音乐。

代码：

if (Music\_num == 0)

{

PlaySound(TEXT("铃芽之旅.wav"), NULL, SND\_FILENAME | SND\_ASYNC | SND\_LOOP);

}

else

{

PlaySound(TEXT("天空之城.wav"), NULL, SND\_FILENAME | SND\_ASYNC | SND\_LOOP);

}

**3.8 成绩排行榜模块**

成绩保存在array\_data.txt文本文档中，游戏开始时，会读取本地文件，然后将组好的三次成绩读取到record数组中。当某一局游戏结束时，会将本局游戏的得分与record数组的元素进行比较，如果比数组中成绩要好，则更新record数组的内容，然后写回文本文件中，排行榜在每一局后更新。

1. 游戏使用说明

游戏的设计环境是VS2022，Release，x86。

游戏基本流程是：程序运行后，鼠标点击选择行列值，点击“确认”，进入游戏，左键点击小球选中小球，左键点击空位置，如果有路径的话，小球便会移动到空位置。如果达到“五连珠”，会消去相连的珠子并且得分，如果没有达到“五连珠”,不会消去珠子，反而会在棋盘上添加三个珠子。之后循环进行下去，想要退出本局游戏，直接点击右键即可，也可以等到棋盘变满，也会退出本局游戏。退出游戏后可以再次选择行列值，再次游戏，也可以点击“退出游戏”，程序结束。

注意事项：

游戏默认是9\*9的棋盘，如果不选择行列值，直接点击“确认”，将会得到9\*9的棋盘。

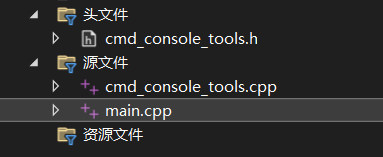
珠子在移动过程中、在选中状态时，点击鼠标右键是无法退出游戏的。

array\_data.txt文件中的三个整数类型的得分按照从大到小的顺序填写，中间用空格分开，最后换行。

**4.1 在VS中编译使用**

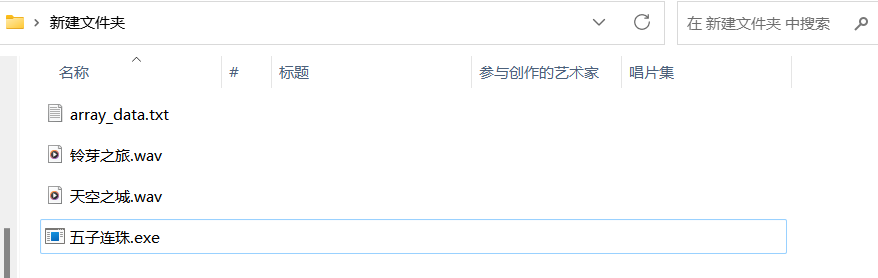
一共有3个文件，一个头文件，两个cpp文件。

具体放置位置如下图：

将array\_data.txt，头文件，cpp文件，两个音乐文件放在同一个文件夹下，就是将txt文本和wav文件放在与cpp同一个文件夹下。

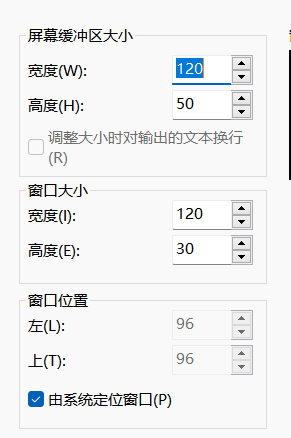
**4.2 可执行文件使用**

将exe文件、两个wav文件、一个txt文件放在同一个文件夹下，双击exe文件开始游戏，放置如下图

1. 测试参数设置

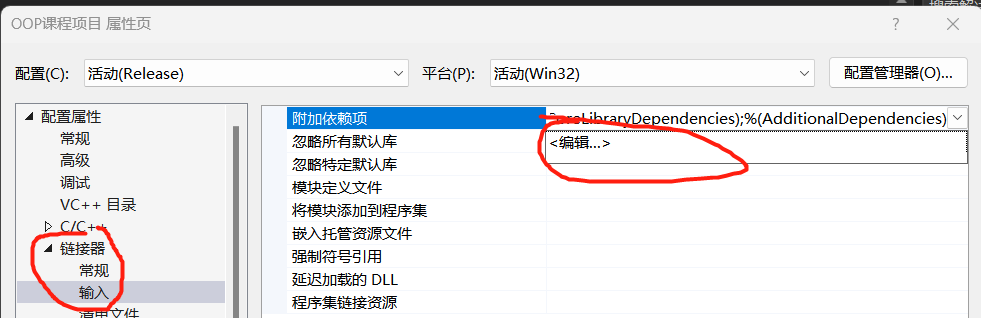
**5.1 控制台设置**

如下图：使用旧版控制台，取消快速编辑模式，取消插入模式，设置好之后需要重新启动才有效。为了显示效果，缓冲区以及窗口设置建议如下。

**5.2 VS编译设置**

在VS 中运行时，还需要对链接器进行操作才能播放背景音乐。

流程为：右键项目，选择“属性”，“链接器”，“输入”，“编辑”



添加内容“Winmm.lib”，“确认”，“确认”。

