五子连珠文档

1 杨滕超 2151298

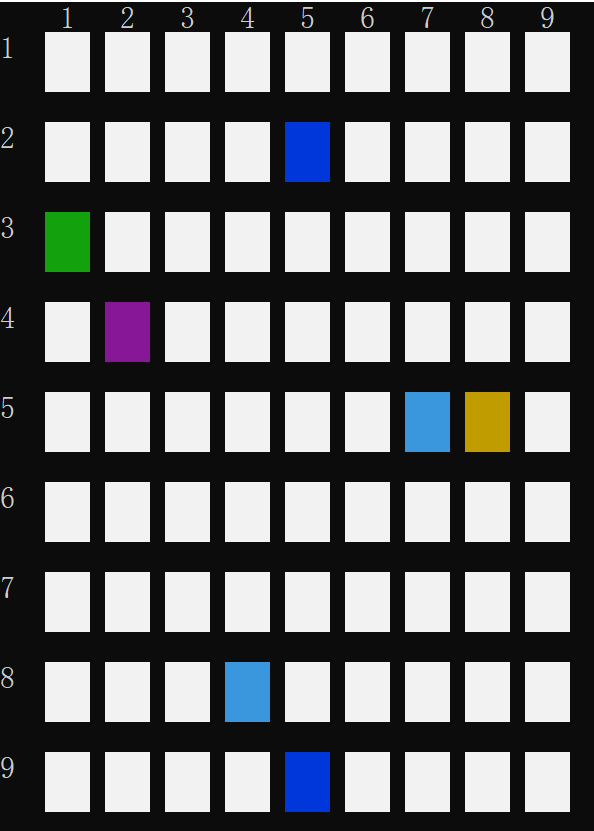
2 整体设计思路

首先利用9 \* 9的整型数组模拟游戏底层结构，用不同的数组代表珠子的不同颜色，最后根据数字的不同，在cmd窗口打印出不同的颜色。其次，对于命令行的输入和鼠标的控制，都有实现，利用高程课老师教导的一些函数对鼠标操作的读取，更好地提升游戏的趣味。再者，由于才疏学浅，苦思冥想之后只能用打印有颜色的方格代替珠子，算是美中不足。最后，关于移动的最短路径的函数，学习dfs、bfs、启发式等搜索后，决定采用bfs算法，完成该项功能。此外，五子连珠这个游戏封装在一个类中，可以非常方便地调用，还有背景音乐、分数统计、预告即将出现的珠子颜色等小功能，也是添加在这个类中。

3 程序功能以及使用

3.1 游戏界面展示

通过底层二维数组的不同的值，其中分为-1为没有珠子，0 - 5共计分别对应六种颜色的珠子，然后根据值的不同，打印出不同的颜色。设计打印宽度为3，高度为2，近似为一个正方形，以弥补圆形珠子的遗憾。此外，通过随机函数随机生成将要出现的珠子的坐标以及颜色，值得注意的一点是，当到最后剩下少于要生成的数量的空格时，需要根据剩余空格数量进行生成。

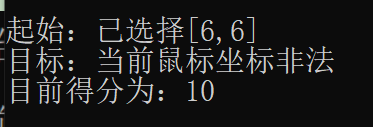


3.2 珠子移动以及移动动画

最初命令行移动的版本，使用输入起始位置和终止位置的坐标进行移动，共计四个数字。但加入了鼠标控制之后，通过将鼠标的位置通过一定的函数转换为游戏界面中方块的坐标，以及进行选择和移动。共同需要注意的是，两者均需要判断是否出界、以及能否到达目标位置。由于后者完全可以交给找最短路径这个功能，于是将在后面提到。

鼠标控制与命令行的不同点在于，分为两次点击，若是点击选中的珠子，我们通过高亮与原来的颜色交替进行选中的提示；若是点击空白处或者出界位置，将无任何反应。选择好珠子后，将进行选择去到的位置，若是点击到去不了的位置，其中包括被阻拦与已经被珠子占用，将清除第一次选择，以此重新选择珠子进行移动。

除此之外，鼠标控制有两个选择的文字提示于游戏界面的底部，可以方便玩家了解自己的选择珠子的位置或者是否位置非法。



珠子的移动特效：利用在较短时间内打印该珠子的颜色色块与补充背景色块实现。其中时间间隔利用Sleep函数，暂停程序来保证人眼能够观察到。

3.3 判断五子连珠

检查整个数组，从有珠子的地方进入函数，沿着八个方向，进行同颜色的搜索并且记录个数，相反的方向进行合并加合，某一方向共计超过五个便可以消去。

消去珠子包括游戏界面以及底层数组，游戏界面可以打印白色背景空格。此时需要注意的是并不是只要连在一起的珠子就消去，而是沿着某一方向才能消去。

具体实现消去珠子时候，我采用先消去最开始搜索的那个珠子，然后搜索的时候记录具体哪一个方向满足了消去的条件，由此进行该方向的消去递归。

3.4 找最短路径

利用广度优先搜索算法，通过STL中的队列以及以及pair<int, int> 记录坐标，其中类中有数据成员两个二维数组专门记录在搜索中可行的路径。记录的方法为反向记录，该位置记录的值为上一步的位置，最后能到达目标位置就返回共计步数，不能则返回零，以此判断能否到达目标位置。

最后通过上述两个二维数组得到经过的路径的各个坐标，以此通过移动的动画完成最短路径移动的展示。

3.5 预告新生成的三个珠子

首先在游戏循环开始就新生成三个颜色，在游戏界面展示，若是移动后满足五子连珠则不改变新生成的三个颜色，继续下一步的移动，否则随机生成位置，通过画色块的函数在游戏界面生成。

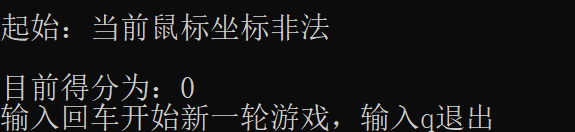


3.6 判断游戏是否结束

通过检查整个二维数组是否都被标记，判断是否该结束游戏。

3.7 游戏强制结束

若不想等界面都被塞满了珠子才结束游戏，可以选择按键盘q或Q结束这盘游戏，接着会在底部出现提示，回车则开启新一轮游戏，再输入q或Q则退出程序。



3.8 背景音乐

通过PlaySound函数以及相对路径的音乐文件，进行背景音乐的播放，将想听的背景音乐转成wav后缀，改名为background.wav并放入和发布版本同一个文件，即可播放音乐，若不想听音乐可以按键盘空格，即可暂停播放，再按以此，可重新播放。

3.9 得分记录

如图分数会在每次得分后实时更新于游戏底部，游戏结束后会输出最终得分。

4 各模块体系结构设计和描述

运行游戏public函数

void color\_ball::start\_game()

{

while (1)

{

const int time = 100;

bool checkfive = false;

init();

cct\_setcolor();

music(1);

score\_output();

show();

while (1)

{

int val[3];

//没有连成功，才会将要出来的球

if (checkfive == false)

{

//随机三个颜色

new\_color(val);

//显示将出现的三个球的颜色

show\_next\_three\_ball(val);

//随机出来的球连成五个

check();

}

//输出得分

score\_output();

//展示棋盘

Sleep(time);

//a.show();

//输入坐标并移动（找最短路径）/ 鼠标点击

mouse\_input();

if (game\_exit)

break;

//最短路径动画

//判断是否消失条件

//若可以消失则加分，不显示

//不可以消失则随机将要出现的位置，三个球出现

if (check())

{

checkfive = true;

}

else

{

checkfive = false;

new\_location(val);

}

Sleep(time);

//检查是否游戏结束

if (is\_over())

{

score\_output(true);

break;

}

}

//一局游戏结束之后

//恢复强制退出的，重新选择

game\_exit = false;

cct\_gotoxy(0, 31);

cout << "输入回车开始新一轮游戏，输入q退出";

char ch;

while (1)

{

ch = \_getch();

//回车十进制ascii码

if (ch == 13)

break;

if(ch == 'Q' || ch == 'q')

{

game\_exit = true;

break;

}

}

if (game\_exit)

break;

}

}

5 游戏使用说明

点击珠子所在位置，有闪动特效告知选中，同时也会有于底部的文字提示，再点击空白处可以移动珠子，形成五子连珠即可消去，否则每次移动会增加三个随机颜色随机位置的珠子。

若想重新选择移动的珠子，点击珠子不可移动到的位置即可。

若想重新开始游戏，按键盘q键，根据底部提示再按回车。

若想结束游戏，按两次q键。

关掉音乐，按键盘空格键。

再开始播放音乐，再按一次空格键。