

**项目名称 ： 超级方块**

**成员 ： 黄勇博，彭飞，王浩然**

**指导老师 ： 陈松，范波**

**日期：**

**2022年4月26日**

**项目名称 ： 超级方块**

**成员 ： 黄勇博，彭飞，王浩然**

**指导老师 ： 陈松，范波**

**程序设计报告**

目录

1.摘要.........................................................................................................................................................1

2.系统方案................................................................................................................................................2

2.1比较与选择..........................................................................................................................2

2.2需求分析...............................................................................................................................3

2.2.1输入，输出.....................................................................................................................3

2.2.2性能要求..........................................................................................................................3

2.2.3运行要求..........................................................................................................................3

2.3系统总体设计思路.............................................................................................................4

2.3.1设计思想..........................................................................................................................4

2.3.2实现思路..........................................................................................................................5

2.3.3功能划分..........................................................................................................................5

3.理论分析与计算...................................................................................................................................6

3.1遇到的难点和解决方法....................................................................................................6

4.游戏流程.................................................................................................................................................8

5.主要算法.................................................................................................................................................9

5.1BFS寻路算法.........................................................................................................................9

5.2随机算法...............................................................................................................................11

5.3Astar寻路..............................................................................................................................13

5.3游戏胜利时更新排行榜....................................................................................................15

5.4撤回算法...............................................................................................................................16

5.5消除算法...............................................................................................................................17

6.测试.........................................................................................................................................................20

6.1测试方案...............................................................................................................................20

6.2测试条件...............................................................................................................................20

6.3测试结果...............................................................................................................................20

6.3测试结果分析......................................................................................................................23

**1.摘要**

随着TD-LTE移动通信、智能手机等软硬件技术的发 展，越来越多的益智型小程序诞生，大大的促进了人们的娱乐生活。C语言作为一种面向过程的程序开发语言，拥有数据、指针、结构体等多种数据类型，运行速度非常快，可移植性强，编制游戏 程序具有先天性优势，具有重要的作用和意义。

现在我们所要介绍的这一款游戏——超级方块，就是满足以上要求而设计出来的，希望能给玩家带来娱乐。超级方块这一游戏简单易行，操作方便，娱乐性极强，吸引了不少人。这一款游戏紧紧地抓住了人们的心理，虽然简单，却其乐无穷，在人们不断追求更多的欲望下，该游戏给人们带来了逐级的快感，以及成功后的满足感，对于一直处于高压下的现代人是很好的放松工具。

**关键词**：函数，文件，算法，数据结构

1. **系统方案**

**2.1比较与选择**

1. **开发环境选择**

目前主流的开发环境有DevC++，IDEA，Visual Studio code，Visual Studio 2022。

DevC++界面简洁，适合写简单的算法题，当是功能比较少，不适合用来项目。

IDEA，Visual Studio code，操作复杂，暂时不考虑。

Visual Studio（简称 VS），它是 Windows 下的标准 IDE，VS系列为IDE（集成开发环境），为了适应最新的 Windows 操作系统，微软每隔一段时间（一般是一两年）就会对 VS 进行升级，为目前各大公司Windows下开发的主流选择工具。

Visual Studio 2022它是64位编译器，功能齐全，它可以使开发人员能够更轻松地创建、编辑、构建和调试跨平台应用程序。

总体来说，Visual Studio 2022非常适合这次程序设计。

1. **图形库选择**

目前主流的图形库有BGI图形库，[GRX](http://grx.gnu.de/index.html) 图形库，[dislin](http://www.mps.mpg.de/dislin/) 库，[allegro图形库](http://alleg.sourceforge.net/) ，[opengl](http://www.opengl.org/) 和[sdl](http://www.libsdl.org/) 库。

BGI图形库：即turbo c所带的图形库，又叫easyx图形库，安装上手简单，功能强大适合C语言开发。其他图形库较为繁琐，且安装上手复杂。

总的来说，easyx图形库是非常适合这次程序设计。

1. **寻路算法选择**

现有寻路算法：BFS，DFS，A\*，下面讲讲这三种算法的区别。

DFS在于从一个初始状态出发，一直转移状态直到搜到目标或搜到状态无法进行转移为止，其中的剪枝便是通过应题目具体要求和仔细分析而去掉某些没必要或不存在的状态以节约时间和空间，而BFS在于一层一层地扩展状态,这样首先找到的目标便一定是用时或步数最少的。DFS几乎适用于所有情况，因为这本身是一种遍历所有状态的搜索，只不过在寻找最小步或最小时间的情况下比较慢，无法快速找到目标，但在程序世界里DFS本身也有不适用的情况，如果状态没有下限即转移的深度没有下限那么便无法深搜，同样，BFS也可能存在一层状态都扩展不完的情况。而A\*则多了评估函数（或者说剪枝），每次预估如果这条路如果继续下去也无法到达终点，则放弃这条路（剪枝的一种，即最优性剪枝）。IDA\*原理实现起来简单，非常方便，但难的地方是评估函数的编写，足够好的评估函数可以避免走更多的不归路，如果评估函数太差或者没有评估函数，那会退化到迭代加深搜索那种浪费程度。

所以，我们采用BFS和A\*进行寻路，评估函数中的距离采用曼哈顿值。当图为稀疏图时采用BFS广度搜索算法，当图为稠密图时采用A\*算法，这样极大的提高了运行效率。

**2.2需求分析**

**2.2.1输入、输出**

输入：鼠标操作选择、级别

输出：时间、分数

**2.2.****2性能要求**

**（1）时间特性**

（a）鼠标点击响应时间不超过1s;

（b）进入游戏界面时间不超过2s;

（c）目标消除响应时间不超过1s;

**（2）显示特性**

（a）控件分布合理;

（b）游戏界面简单、漂亮;

（c）消除动画流畅;

（d）菜单深度不超过2层;

**2.2.3运行要求**

**（1）界面**

各个界面主题风格一致、界面布局简单功能按钮分布合理。

**（2）基本操作**

进入游戏后倒计时开始，通过手指滑动来改变方块的位置，如果五个活五个以上相同的方块连在一起则消除这些方块，会给出相应的分数作为奖励，分数达到500分后游戏胜利。点击帮助按钮显示游戏的操作信息，点击排行榜按钮显示游戏的排行榜信息，点击退出按钮退出游戏界面，回到主菜单。

**（3）显示要求**

进入游戏后所有方块随机生成，刷新地图显示最新所用的时间，方块消除后两秒内不生成方块，游戏失败可退出游戏，点击退出则重新开始游戏。

2.2.3输入输出

**2.3系统总体设计思路**

**2.3.1设计思想**

1. 超级方块是一款简单的方块消除类游戏，界面是一个9\*9的棋盘。
2. 用户可以点击方块进行移动，使之5个及以上相同的方块连成一线达到消除的目的，每消除一个方块 可得到10积分。此外，如果成功消除方块，则不会生成新的方块。反之，随机生成3个方块
3. 游戏目标是获得500积分及以上为胜利。当棋盘被方块填满时，游戏失败。
4. 游戏界面包含排行榜，帮助，实时分数显示，存档功能，撤回功能，退出功能，难度选择。
5. 菜单应包括开始游戏和继续游戏。
6. 游戏界面设计使游戏的画面简单美观，具备比较强的视觉效果。



流程图2-1

**2.3.2实现思路**

1. 可以使用二维数组来保存9\*9棋盘状态，不同的数代表不同的方块。
2. 用户选择方块，再点击空地，若有可移动的路径，则通过寻路算法选择最短路径。移动后，若不符合 消除条件，则通过随机算法随机生成方块。若符合消除条件，则获得相应的积分。
3. 寻路算法可采用BFS盲目式搜索或者A\*启发式搜索。
4. 游戏界面采用EasyX图形库实现。
5. 随机算法可调用随机函数以时间为种子，进行播种。
6. 判断游戏胜利的函数，记录游戏分数和游戏时间的变量。
7. 当游戏结束时，能够询问用户是否重新开始。

**2.3.3****功能划分**

|  |  |  |
| --- | --- | --- |
| **模块名称** | **功能要求** | **功能说明** |
| 游戏开始模块 | 启动游戏，并进入设定好的游戏界面 | 正常启动游戏 |
| 显示游戏开始界面 |
| 显示游戏界面上的按键 |
| 游戏界面设计 | 界面美观 | 完成对整个游戏界面的设计与实现 |
| 游戏功能键模块 | 开始按钮，帮助按钮，关于按钮，退出按钮，鼠标点击识别 | 开始游戏 |
| 显示帮助信息 |
| 显示版本等信息 |
| 退出游戏 |
| 判断鼠标信息 |
| 游戏算法设计模块 | 算法的编写符合各个功能的需要 | 寻路算法 |
| 游戏计分、计时功能算法 |
| 鼠标点击控件的事件触发算法 |

表2-1

**3.理论分析与计算**

**3.1遇到的难点及解决办法**

1. **运行过程中遇到闪屏问题**

**解：**查阅资料：双缓

BeginBatchDraw();

EndBatchDraw();

所在头文件：EasyX.h

**原理:**

**1.将要输出的数据写在缓冲区一（写的过程中显示的是缓冲区二的内容）**

**2.显示缓冲区一的内容**

**3.将要输出的数据写在缓冲区二（写的过程中显示的是缓冲区一的内容）**

**4.显示缓冲区二的内容 ，回到第1步**

流程图3-1

1. **地图赋值为0，用for循环比较繁琐**

**解：**查阅资料：memset();

所在头文件：string.h

函数原型：void \*memset(void \*s,int c,size\_t n);

1. **C语言同时开辟两个数组，两个数组的地址会连续造成访问数组时越界**

**解：**传统的直接用数组名加下标的访问方式，可能会在下标大于数组大小但不会报错，而是直接访问另一 个数组，进而导致影响另一个数组的问题。我们可以对数组进行包装，防止越界，来解决此问题。

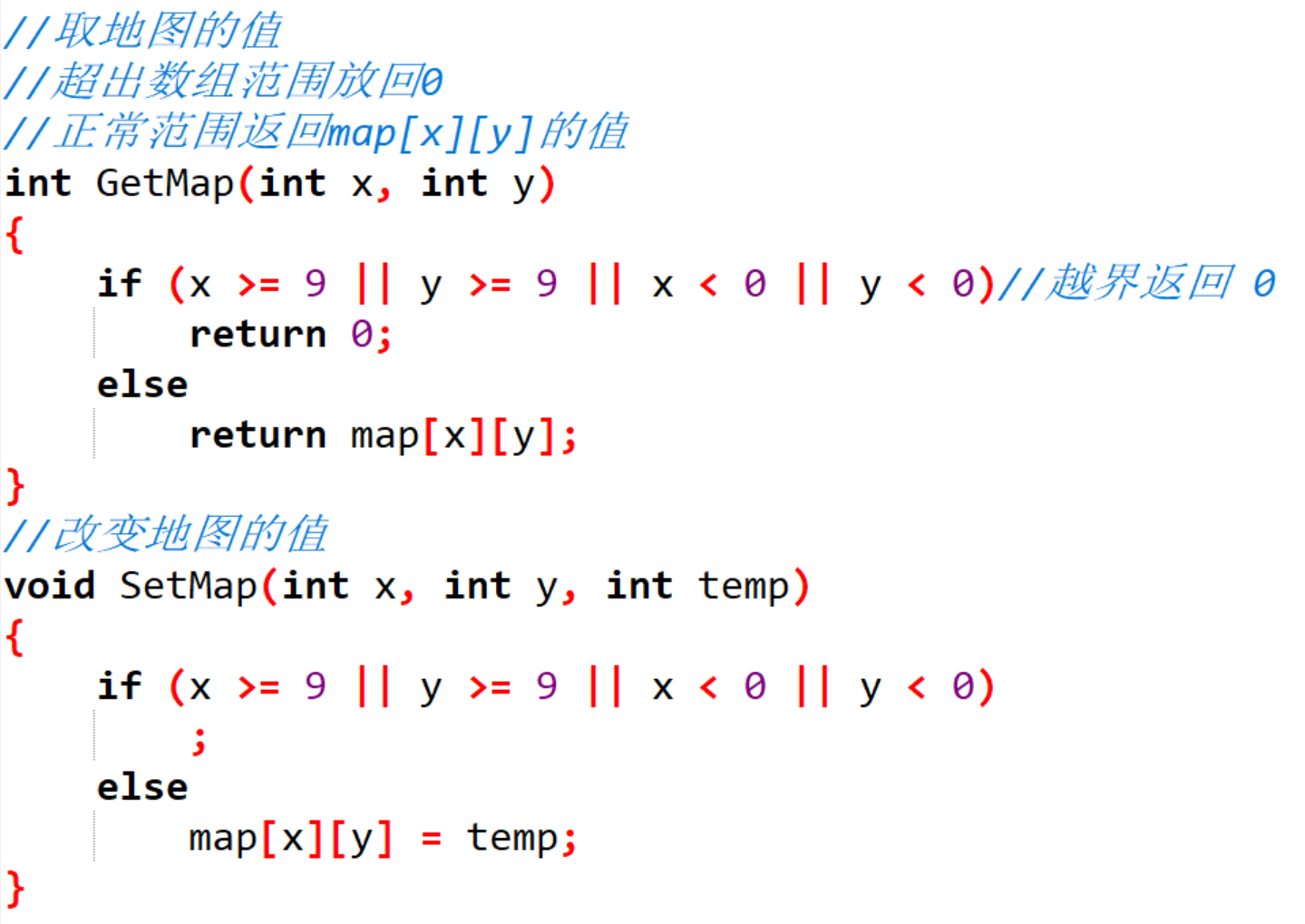


图3-1

1. **关于 GetMouseMsg()函数没有读取到鼠标信息而等待，暂停程序运行的问题**

**解：**查阅资料：MouseHit()检测鼠标是否存在鼠标信息，当存在鼠标信息才 GetMouseMsg()，防 止 GetMouseMsg()等待鼠标信息

**4.游戏流程**



流程图4-1

**5.主要算法**

**5.1 BFS寻路算法**



流程图5-1

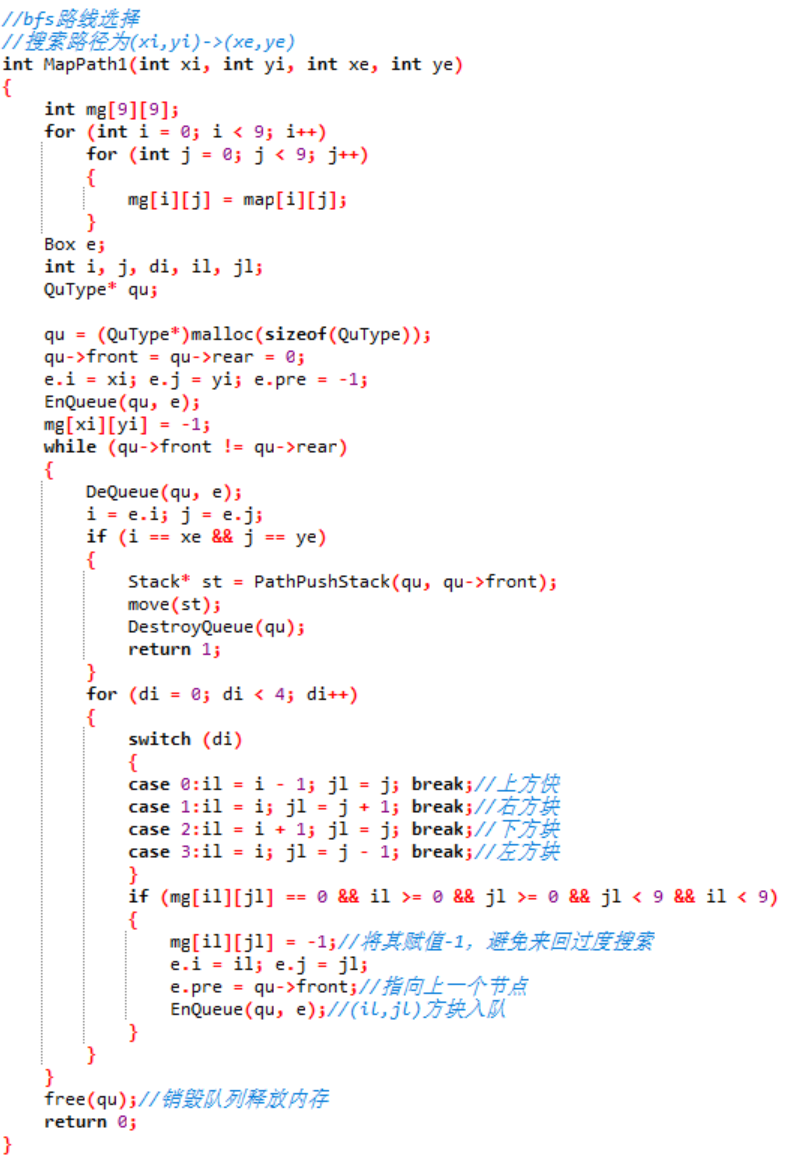
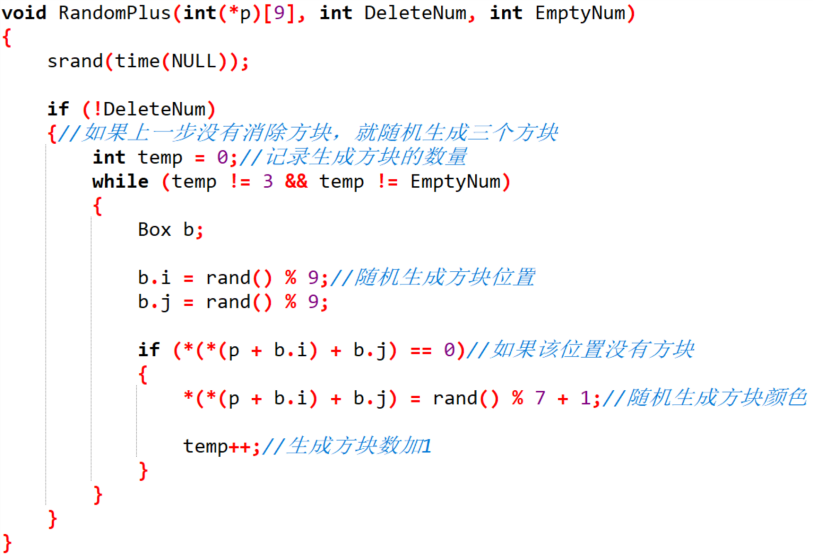


图5-1

**5.2 随机算法**



流程图5-2

图5-2

**5.3 Astar寻路（启发式搜索）**



流程图5-3

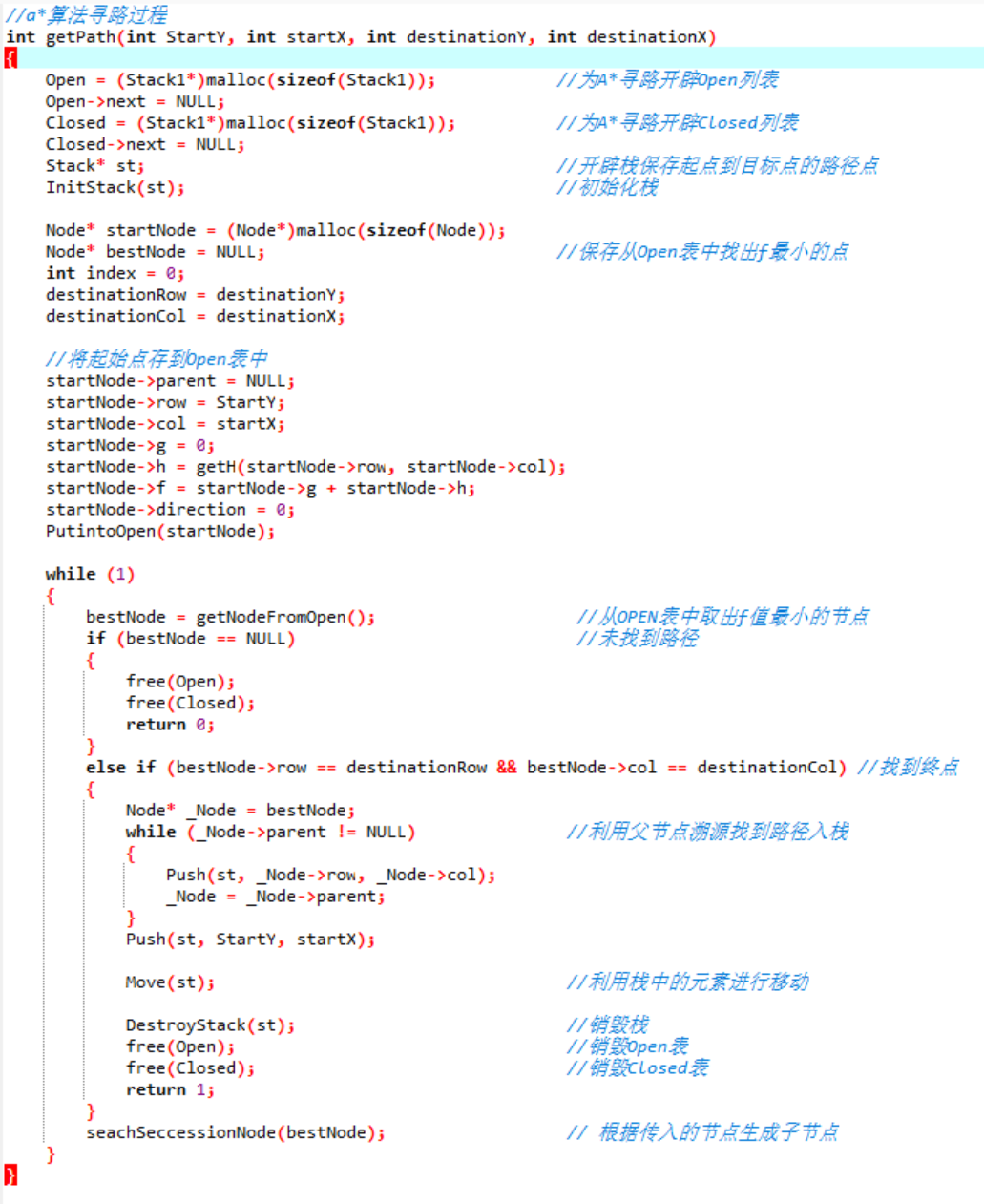


图5-3

**5.4 游戏胜利时更新排行榜**



流程图5-4

**5.5 撤回算法**

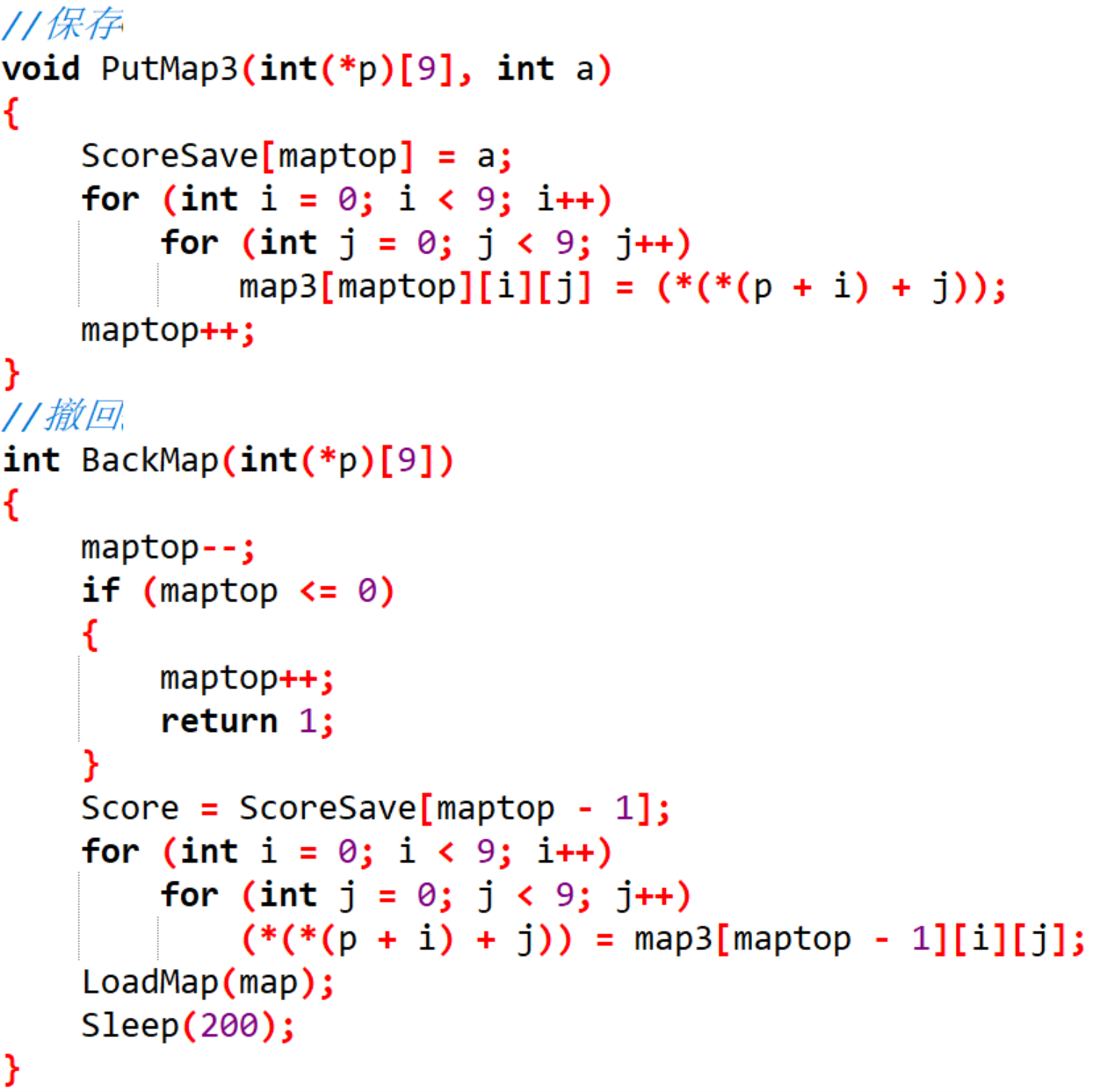


图5-4

**5.6消除算法**



流程图5-5 流程图5-6

int JudgeWin()

{

int i = 0;

int j = 0;

Stack\* st;

int k = 0;

InitStack(st);

for (i = 0; i < 9; i++)

{

for (j = 0; j < 9; j++)

{

//判断竖直方向是否可以消除

if (GetMap(i, j) && GetMap(i, j) == GetMap(i + 1, j) && GetMap(i + 1, j) == GetMap(i + 2, j) && GetMap(i + 2, j) == GetMap(i + 3, j) && GetMap(i + 3, j) == GetMap(i + 4, j))

{

int I = i; int J = j; //（I,J)为待消除方块组的起始方块

while (GetMap(I, J) == GetMap(I + 1, J))

//以（I，J)方块为起点的竖线上，将相通颜色的方块入栈，等待消除

{

Push(st, I, J);//将待消除方块入栈

I++;//推到下一个方块

}

Push(st, I, J);

}

//正对角线方向是否可以消除

if (GetMap(i, j) && GetMap(i, j) == GetMap(i + 1, j + 1) && GetMap(i + 1, j + 1) == GetMap(i + 2, j + 2) && GetMap(i + 2, j + 2) == GetMap(i + 3, j + 3) && GetMap(i + 3, j + 3) == GetMap(i + 4, j + 4))

{

int I = i; int J = j;//（I,J)为待消除方块组的起始方块

while (GetMap(I, J) == GetMap(I + 1, J + 1))

//以（I，J)方块为起点的正对角线方向上，将相同颜色的方块入栈，等待消除

{

Push(st, I, J);

I++;

J++;//推到下一个方块

}

Push(st, I, J);

}

//水平方向是否可以消除

if (GetMap(i, j) && GetMap(i, j) == GetMap(i, j + 1) && GetMap(i, j + 1) == GetMap(i, j + 2) && GetMap(i, j + 2) == GetMap(i, j + 3) && GetMap(i, j + 3) == GetMap(i, j + 4))

{

int I = i; int J = j; //（I,J)为待消除方块组的起始方块

while (GetMap(I, J) == GetMap(I, J + 1))

//以（I，J)方块为起点的水平方向上，将相同颜色的方块入栈，等待消除

{

Push(st, I, J);

J++;//推到下一个方块

}

Push(st, I, J);

}

//反对角线方向是否可以消除

if (GetMap(i, j) && GetMap(i, j) == GetMap(i + 1, j - 1) && GetMap(i + 1, j - 1) == GetMap(i + 2, j - 2) && GetMap(i + 2, j - 2) == GetMap(i + 3, j - 3) && GetMap(i + 3, j - 3) == GetMap(i + 4, j - 4))

{

int I = i; int J = j;//（I,J)为待消除方块组的起始方块

while (GetMap(I, J) == GetMap(I + 1, J - 1))

//以（I，J)方块为起点的水平方向上，将相同颜色的方块入栈，等待消除

{

Push(st, I, J);

I++;//推到下一个方块

J--;

}

Push(st, I, J);

}

}

}

//删除，k记录删除方块的数量

while (!IsEmpty(st))

{

Pop(st, i, j);

if (GetMap(i, j))

{

SetMap(i, j, 0);

k++;

}

}

DestroyStack(st);

return k;

}

**6.测试**

**6.1测试方案**

1. 菜单系统功能测试

有排行榜，帮助，开始游戏，继续游戏，保存游戏等功能，并且开始游戏时可以选择难度。

1. 游戏操作测试

能正常操作方块移动，撤回。

**6.2测试条件**

**平台：Visual Studio 2022**

**系统：window11，window10**

**网络环境：无**

**6.3测试结果**

**（1）菜单系统**



图6-1

**（2）排行榜**



图6-2

**（3）帮助**

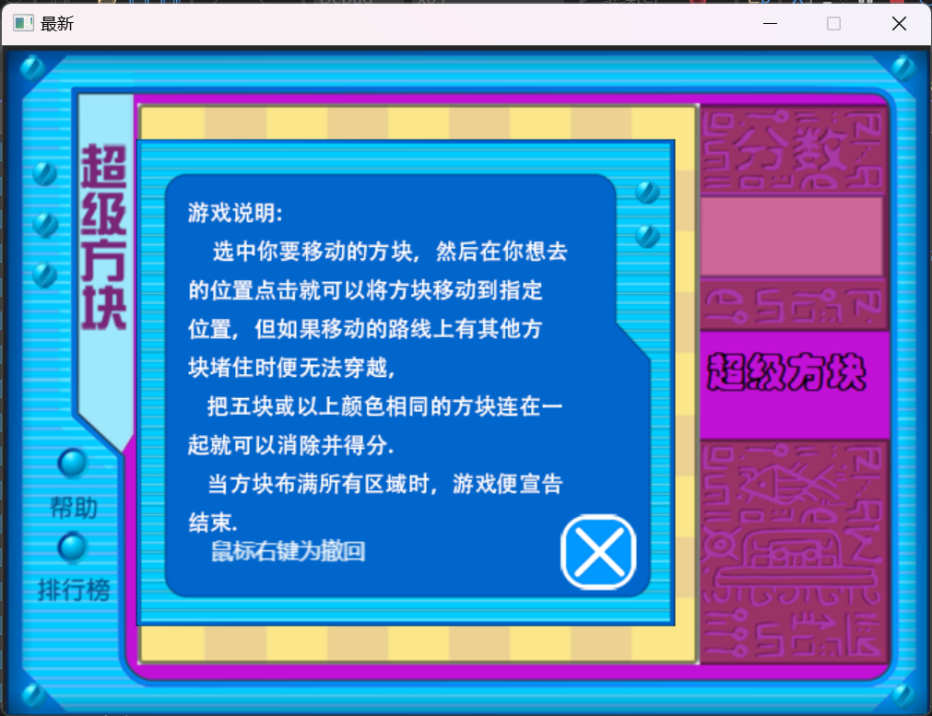


图6-3

1. **开始游戏&&选择难度**

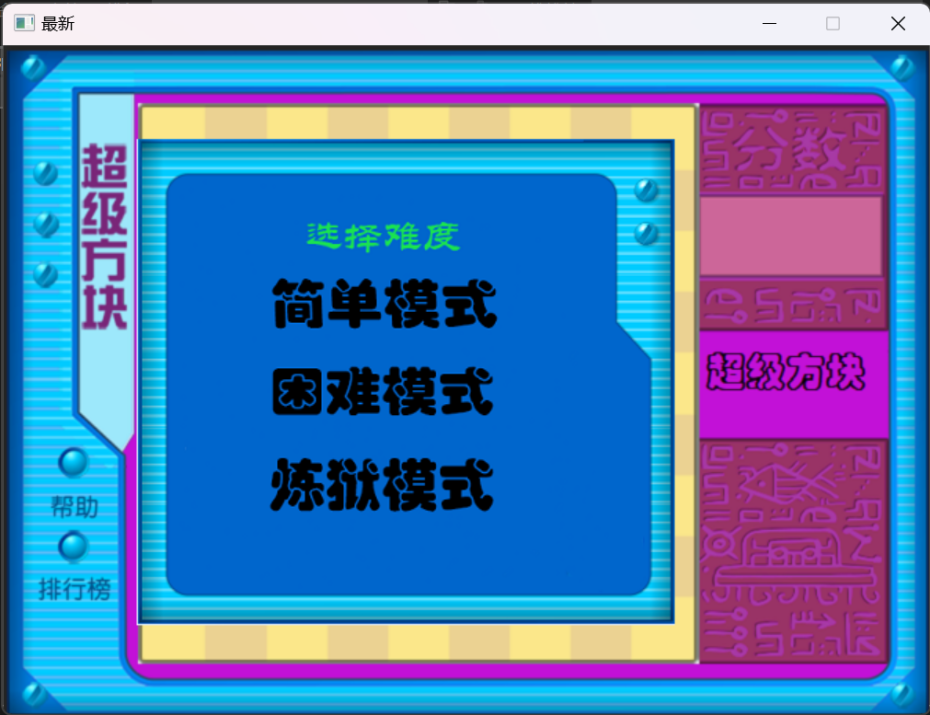


图6-4

1. **保存游戏**



图6-5

**6.4测试结果分析**

**通过全方位的测试，游戏没有发现bug，功能符合需求，界面美观，且有背景音乐和点击音效。运行时无卡顿，无闪屏，无闪退的情况。**