**数据结构课程设计**

**课题名称： 走迷宫游戏**

**专业： 软件工程18级 班级： 2班**

**姓名： 李雨桦 学号：20 成绩：**

**指导教师： 张玲玲**

**完成日期： 2019 年 12 月 29 日**

任 务 书

|  |
| --- |
| 题目：走迷宫游戏 |
| 设计内容及要求：  1.课程设计任务内容  程序开始运行时显示一个迷宫地图，迷宫入口在左上方，迷宫的右下方有一个出口。游戏的任务是使用键盘上的方向键操纵在规定的时间内由入口走到出口。（走迷宫游戏）   1. 课程设计要求   (1)可用键盘操纵上下左右移动；  (2)迷宫的墙有明确标记，不能穿墙而过；  (3)正确检测结果，若在规定时间内走到出口，提示成功，否则提示失败；  (4)添加编辑迷宫功能，可修改当前迷宫，修改内容：墙变路、路变墙；  (5)提供机器自动找出出路的功能，并标记路线；  (6)提供定时器，提示走迷宫的剩余时间；  (7)利用栈实现迷宫的功能。 |
| 开发环境：Devc++编译器、Windows操作系统 |
| 实现目标：   1. 熟悉的运用c或c++语言程序编写代码。 2. 实现数据结构相关算法 3. 能够理清整个程序的运行过程并绘画流程图 4. 了解如何定义局部变量和整体变量； 5. 学会上机调试程序，发现问题，并解决 6. 学习用文档书写程序说明 |

摘 要

本设计是为了实现一个走迷宫游戏，利用数据结构中的栈实现寻找迷宫出路并将其明显打印在屏幕上；限制玩家不能穿越墙，只能在路径上移动，并且在规定时间内到达终点，才能获胜。所以玩家在游戏过程中要正确判断路径争取在最快时间内到达出口。在游戏菜单中玩家也可以根据自己需要在原有迷宫地图上重新编辑地图。

**关键词：**走迷宫 编程 设计

目 录

[1 引 言 4](#_Toc20079)

[2 课题分析 9](#_Toc24331)

[3 课题设计 1](#_Toc22017)0

[3.1设计思路 1](#_Toc5228)0

[3.2模块划分 1](#_Toc22069)1

[3.3重要数据结构和变量说明 1](#_Toc9455)3

[3.4关键算法描述 1](#_Toc6898)4

[3.5程序设计流程图 1](#_Toc30469)5

[4 程序运行结果 1](#_Toc14953)6

[5 软件使用说明 1](#_Toc29589)9

[6 结 论 2](#_Toc24436)1

[参考文献 2](#_Toc12639)3

[附录：源代码 2](#_Toc10233)4

1 引 言

本课设主要研究走迷宫游戏；

本题目要达到目标:熟练掌握入栈出栈实现的迷宫自动寻找路径的算法设计，以培养学生数据结构算法的实践性能力。掌握按键识别与读取功能，学习c语言解决实际能力的目标。

**1.1 意义及目的**

本课设的迷宫游戏主要为了增加用户的生活趣味性，同时使编写程序人员提高数据结构以及c语言算法能力。通过课程设计，巩固所学的理论知识，培养综合运用所学知识解决实际问题的能力。能根据实际问题的具体情况，结合数据.结构课程中的基本理论和基本算法，正确分析出数据的逻辑结构,.合理地选择相应的存储结构，并能设计出解决问题的有效算法。

**1.2 研究内容**

1.数据结构的算法与分析，本设计主要是栈的迷宫问题实际应用算法分析。

2.迷宫问题c语言解决实际问题算法掌握。

**1.3 国内外先进算法动态**

自动寻找路径有一种递归算法一种非递归算法，本实验采用非递归算法，可以更好的更了解递归思想，以及算法。

int MazePath(SqStack &S,PosType start,PosType end) //自动寻找路径非递归算法

{

PosType curposs;

SElemtp e;

InitStack(S);

curposs=start;

do

{

if(mz[curposs.x][curposs.y]==1)

{

mz[curposs.x][curposs.y]=-1;

e.seat=curposs;

e.di=0;

Push(S,e);

curstep++;

if(curposs.x==end.x&&curposs.y==end.y)

return 1;

NextPos(curposs,e.di);

}

else

{

if(!StackEmpty(S))

{

Pop(S,e);

curstep--;

while(e.di==3&&!StackEmpty(S))

{

mz[e.seat.x][e.seat.y]=-1;

Pop(S,e);

curstep--;

}

if(e.di<3)

{

e.di++;

Push(S,e);

curstep++;

curposs=e.seat;

NextPos(curposs,e.di);

}

}

}

}while(!StackEmpty(S));

return 0;

}

void simple\_getpath(maze \*M, pos cur, Stack \*s) //自动寻找路径递归算法

{

assert(M);

//2.判定当前点坐标是否可以走。（坐标合法且不为0）

if (can\_stay(cur, M) == 1)

{

//3.如果合法则将当前点标记成走过的并入栈

//（维护一个栈可以记录走过的路径，栈的长度就是路径的长度）

assign\_pos(cur, &M);

StackPush(s, cur);

}

else

{

return;

}

if (check\_exit(cur, M) == 1)

{

print\_stack(s);

return;

}

//4.判断当前点是否是出口，是出口就return（该迷宫不存在别的出口），

//如果不是出口，以顺时针的方向（上，右，下，左）探测临界点是否可以走（不为0且不为已经走过的点），

//并以递归形式重复步骤2-4.

//up

pos up = cur;

up.row -= 1;

simple\_getpath(M, up, s);

//right

pos right = cur;

right.col += 1;

simple\_getpath(M, right, s);

//down

pos down = cur;

down.row += 1;

simple\_getpath(M, down, s)

//left

pos left = cur;

left.col -= 1;

simple\_getpath(M, left, s);

//5.当一个点的4个方向都已经探测过，就返回上一层栈帧。

StackPop(s);

return;

}

2 课题分析

编写一个《走迷宫问题》的C程序，包括以下功能：

初始状态：有一个10\*10的地图显示给用户，通过输入1,2,3,4分别实现开始走迷宫游戏，自动寻找路径，编辑迷宫，小提示等功能。

(1)开始走迷宫游戏:

通过键盘上的上下左右方向键来控制“IMG_256”（代表用户）移动，设置倒计时提醒用户剩余游戏时间，在规定时间内到达终点即为获胜，否则视为失败。退回主界面。

(2)自动寻找路径:

进入界面后，程序中的“IMG_256”自动排列显示一条从入口到出口的路径，即为入口到出口的路径。退回主界面。

(3)编辑迷宫:

通过提示输入，将墙变为路，即将迷宫数组需要改变的位置置1，反之将路变成墙即赋值为0。退回主界面。

(4)小提示:

通过c语言设计提示界面，提示入口出口，以及需要注意的点。退回主界面。

1. 背景音乐:

进入控制台就播放背景音乐，全程电脑蜂鸣器音乐。

# 3 课题设计

## 3.1设计思路

在迷宫游戏中，通过迷宫二维数组代表迷宫地图，“1”代表可行的路（用★显示），“0”代表墙（用■表示），通过switch()检测输入数字，进入开始游戏，自动寻找路径、编辑迷宫、提示功能界面实现功能。

(1)开始走迷宫游戏:

通过键盘上的上下左右方向键来控制“IMG_256”（代表用户）移动，设置倒计时提醒用户剩余游戏时间，在规定时间内到达终点即为获胜，否则视为失败。退回主界面。

(2)自动寻找路径:

设置一个数组，用于改变方向，以改变当前位置。刚开始用户地址在入口处，从数组的第一个方向开始探索，如果移动位置为“1”，代表可行路径。此时入栈；否则出栈，如果数组四个方向都探索完成，发现还是没有路可走，将地图此位置数据变为“1”，出栈，直到可以行走为止。到达终点即停止。

在探索路径过程中走过的路径同样置1。

栈中此时已经存取了从入口到出口的路径，从base遍历栈并显示在地图上，即可明显标示出从入口到出口的路径。

(3)编辑迷宫:

通过提示输入，将墙变为路，即将迷宫数组需要改变的位置置1，反之将路变成墙即赋值为0。

(4)小提示:

通过c语言设计提示界面，提示入口出口，以及需要注意的点。

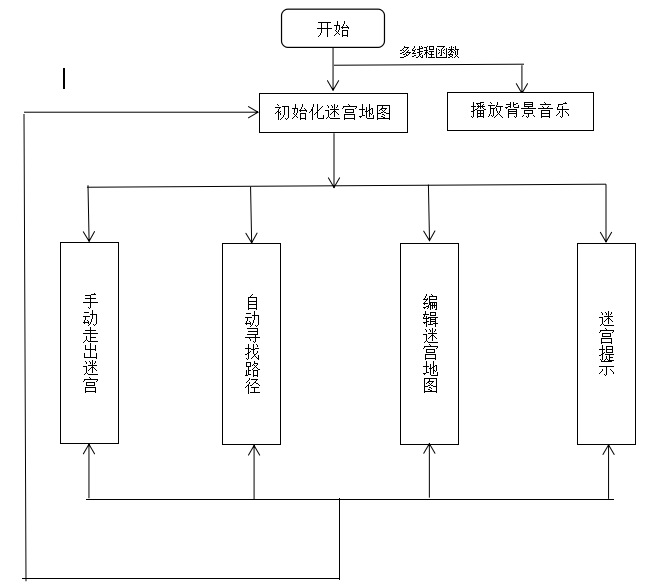
1. 背景音乐:

通过Beep函数设计童年背景音乐数组，然后循环播放，通过windows创建多线程函数（\_endthread();\_beginthread(yingyue,0,NULL);）实现背景音乐。

## 3.2模块划分

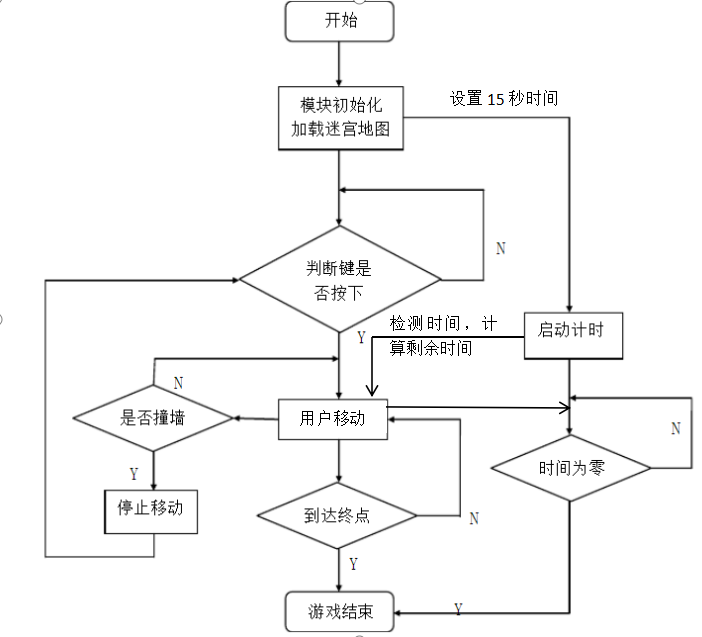
本程序共4个模块，分别如下：

1. 主程序模块 main( )



主程序模块图（即主界面）

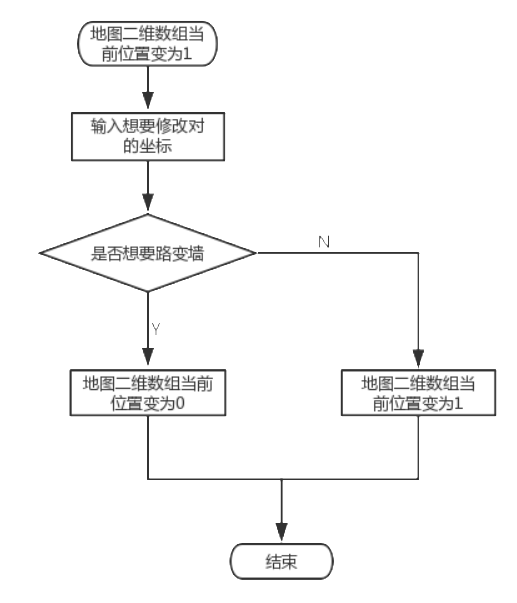
2.程序基本流程图：



开始游戏模块图（即手动到达迷宫出路）

## qt_temp

自动寻找路径模块



编辑地图模块图

## **3.3**重要数据结构和变量说明

1.玩家在地图上的位置坐标（全局变量）

**typedef struct Pos**

**{**

**int x;**

**int y;**

**}PosType;**

2.栈元素类型（全局变量）

**typedef struct SElemtp**

**{**

**PosType seat;**

**int di;**

**}SElemtp;**

3.栈结构体类型（全局变量）

**typedef struct**

**{**

**SElemtp \*base;**

**SElemtp \*top;**

**int stacksize;**

**}SqStack;**

4.游戏迷宫地图数组（全局变量）

**typedef int MazeType[N][N];**

**MazeType mz;**

5.游戏入口出口（全局变量）

**PosType begin,end;**

## 3.4关键算法描述

本人主要负责走迷宫游戏的自动寻找路径算法模块，以下即是本人所负责关键算法的描述。

设置一个数组，用于改变方向，以改变当前位置。刚开始用户地址在入口处，从数组的第一个方向开始探索，如果移动位置为“1”，代表可行路径。此时入栈；否则出栈，如果数组四个方向都探索完成，发现还是没有路可走，将地图此位置数据变为“1”，出栈，直到可以行走为止。到达终点即停止。

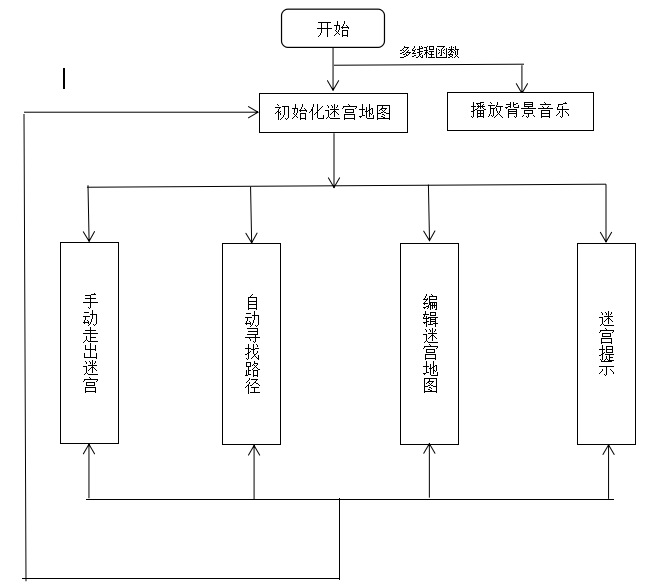
在探索路径过程中走过的路径同样置1。

栈中此时已经存取了从入口到出口的路径，从base遍历栈并显示在地图上，即可明显标示出从入口到出口的路径。

## 3.5程序设计流程图

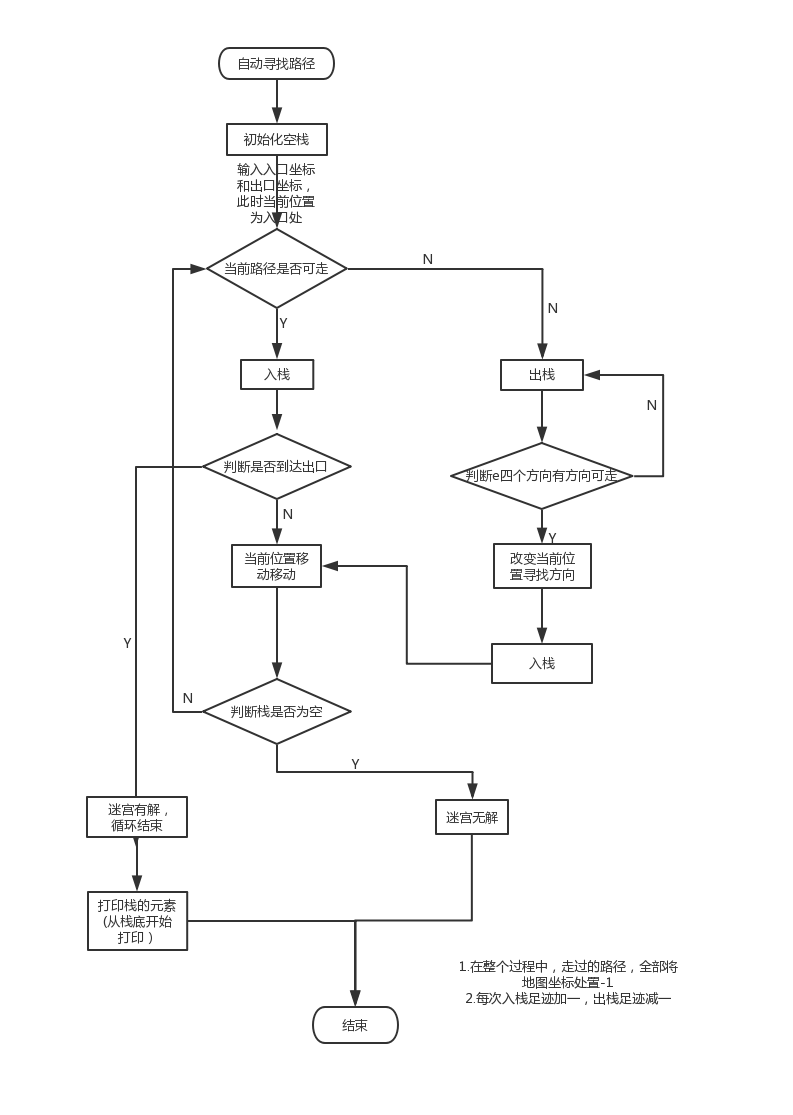
通过Beep函数设计童年背景音乐数组，然后循环播放，通过windows创建多线程函数（\_endthread();\_beginthread(yingyue,0,NULL);）实现背景音乐。

通过switch()函数实现迷宫各个功能的转换，通过输入1,2，3,4,进入不同的功能界面，以及不断的初始化迷宫让迷宫地图保持原样。



设置一个数组，用于改变方向，以改变当前位置。刚开始用户地址在入口处，从数组的第一个方向开始探索，如果移动位置为“1”，代表可行路径。此时入栈；否则出栈，如果数组四个方向都探索完成，发现还是没有路可走，将地图此位置数据变为“1”，出栈，直到可以行走为止。到达终点即停止。

自动寻找路径流程图



4 程序运行结果

图4.1为初始界面,当我们选择运行程序时，便会出现如下图的界面。根据上面文字的内容，我们可以了解游戏的注意事项，继续等待即可进行游戏。如下图所示：



图4.1 初始界面图

图4.2为进入游戏后的首界面，下图是主界面，在方框中有提示进入哪个选项，即输入不同的数字进入程序，并将地图显示在主界面上便于用户修改，如下图所示：

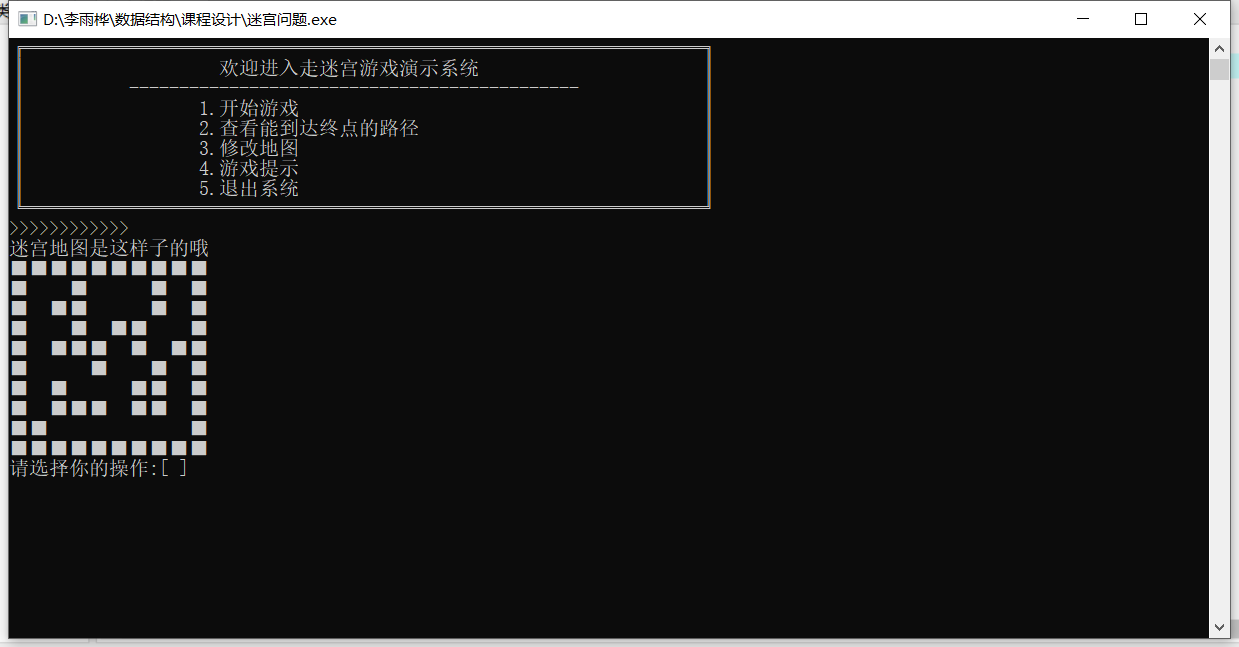


图4.2 游戏首界面图

图4.3为开始游戏运行界面（手动走出迷宫），通过键盘上的上下左右方向键移动“IMG_256”以到达最后终点，当然要在规定时间内，如下图所示：

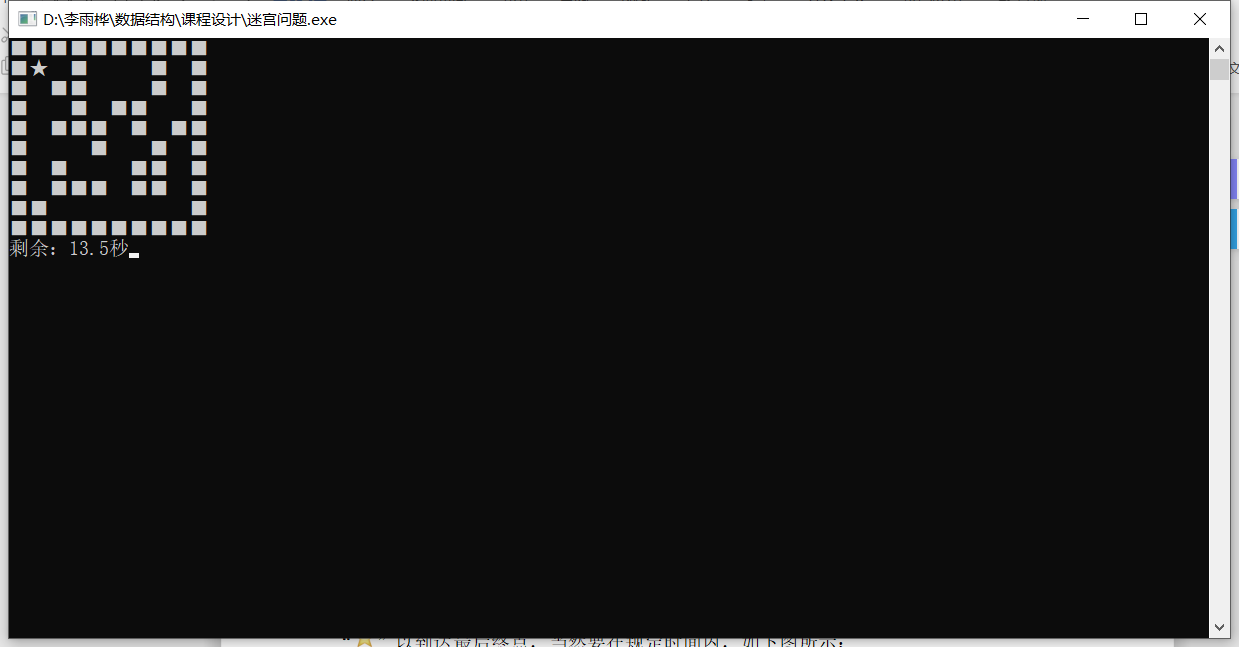


图4.3为开始游戏运行（手动走出迷宫）界面

图4.4为游戏结束后，得到游戏时间的结果图。游戏结束后会自动跳回主界面，参与其他功能或继续游戏，如下图所示：

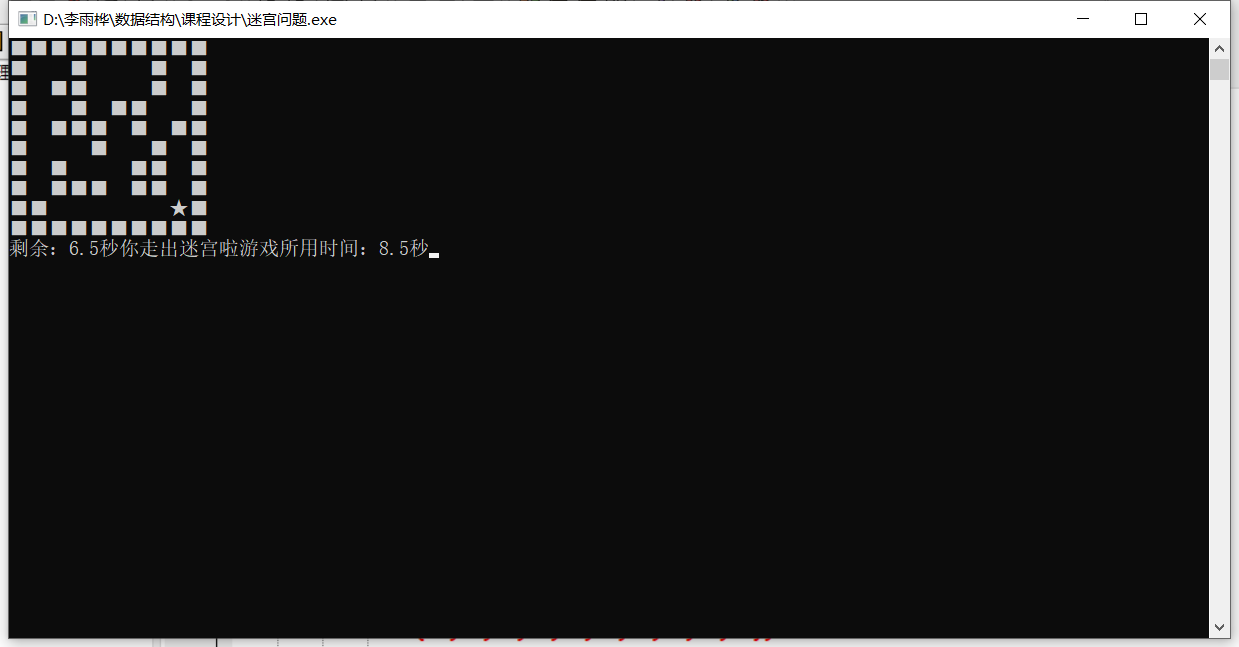


图4.4 游戏结果图

图4.5为自动寻找路径显示出的路径图，输入“2”后进入界面，自动行走，给用户显示出一条明显的路径。如下图所示：

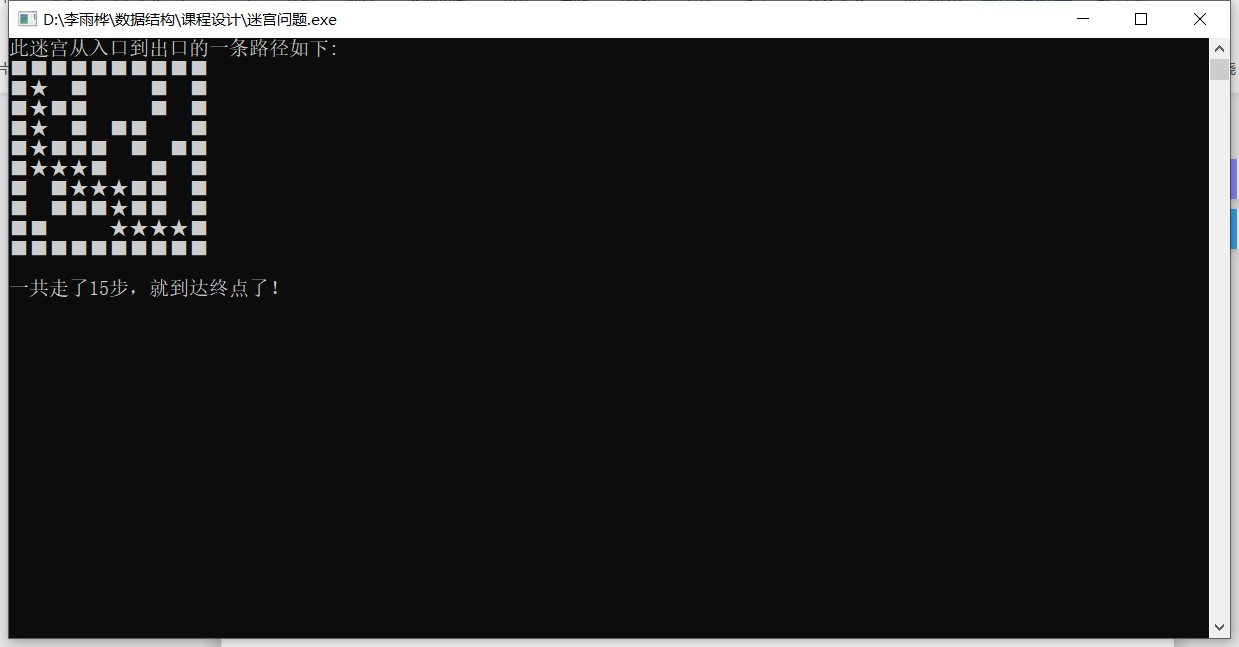


图4.5 自动寻找路径显示出的路径图

图4.6为编辑迷宫的图，如果想要坐标路变墙则输入0，否则输入1，，地图会自动修改，后面的游戏都会修改成这样。如下图所示：

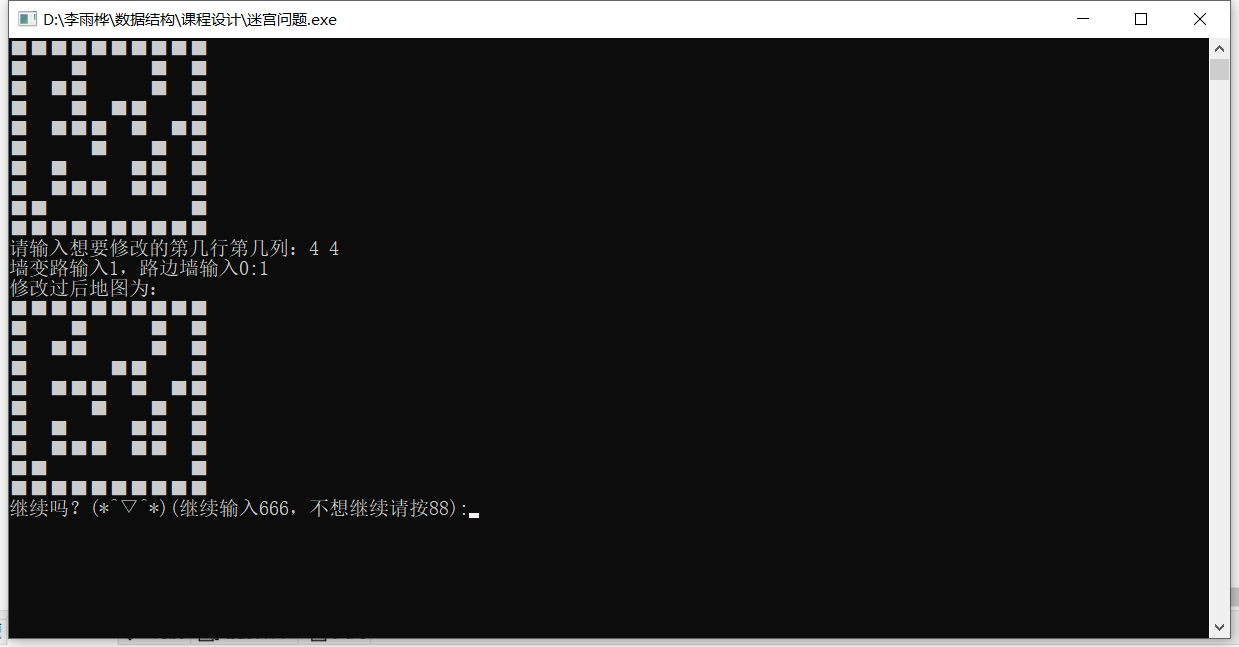


图4.7是游戏提示窗口，会显示迷宫中的一些游戏提示，

# 5 软件使用说明

迷宫游戏是一个娱乐性的游戏，用于锻炼人的脑力和反应能力以及快速的观察能力以在最快速度到达目标，取得胜利，同时可以和电脑路径进行比对，查看是否自己还有优化可能性，提高脑力。（通过上下左右方向键让玩家行走）

图5.1为程序进入游戏后的初始界面

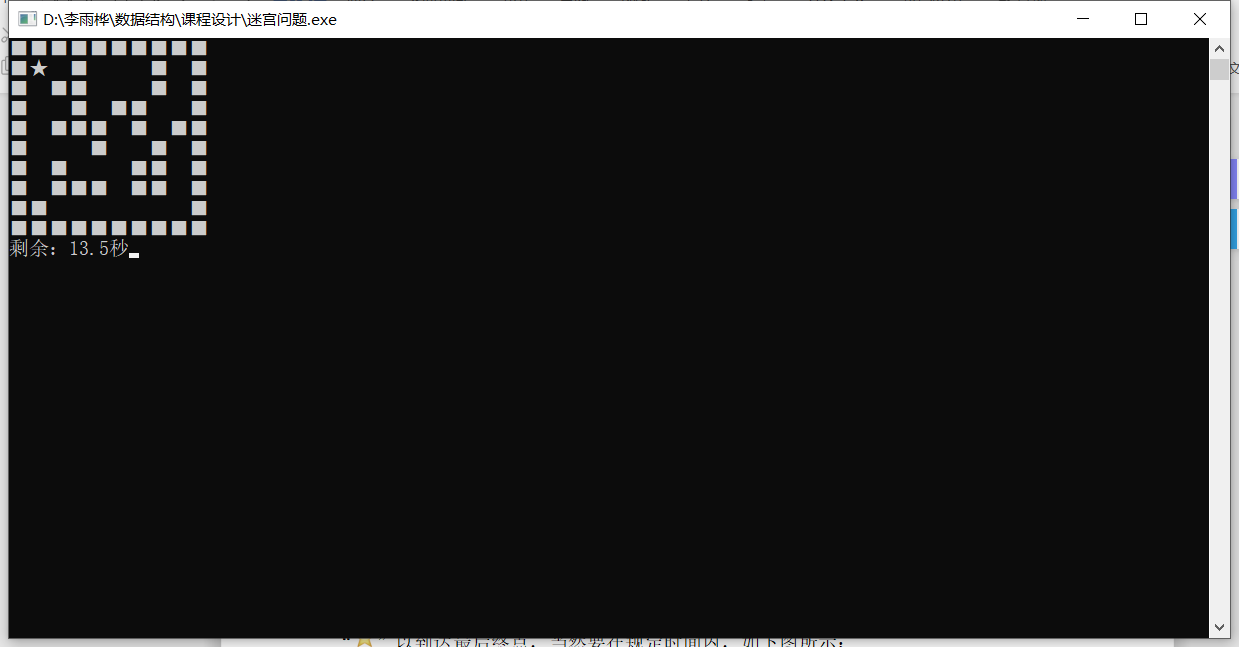


图5.1 游戏初始界面图

下图5.2是自动寻找路径中的截图，得到的是最优路径，作为玩家参考图。

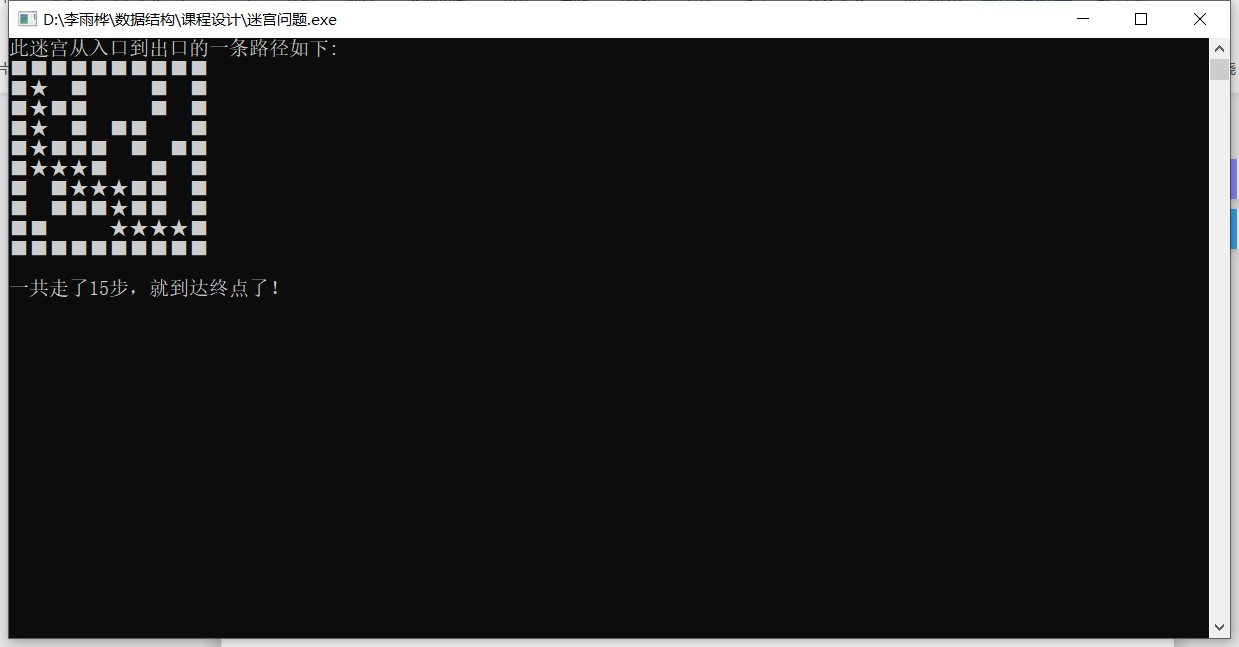
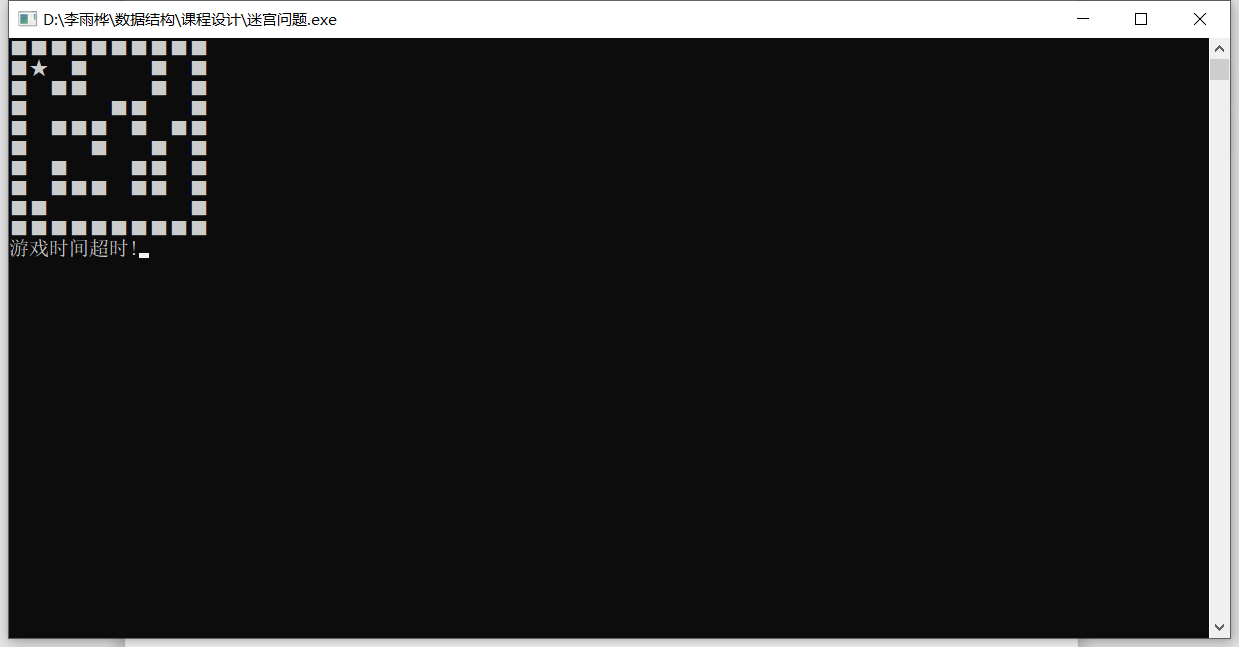


图5.2 自动寻找路径图

下图5.3是游戏结束后超时和在规定时间内到达终点的结果图，最后游戏在规定时间内到达终点会显示用去多长时间游戏。



游戏超时结果图

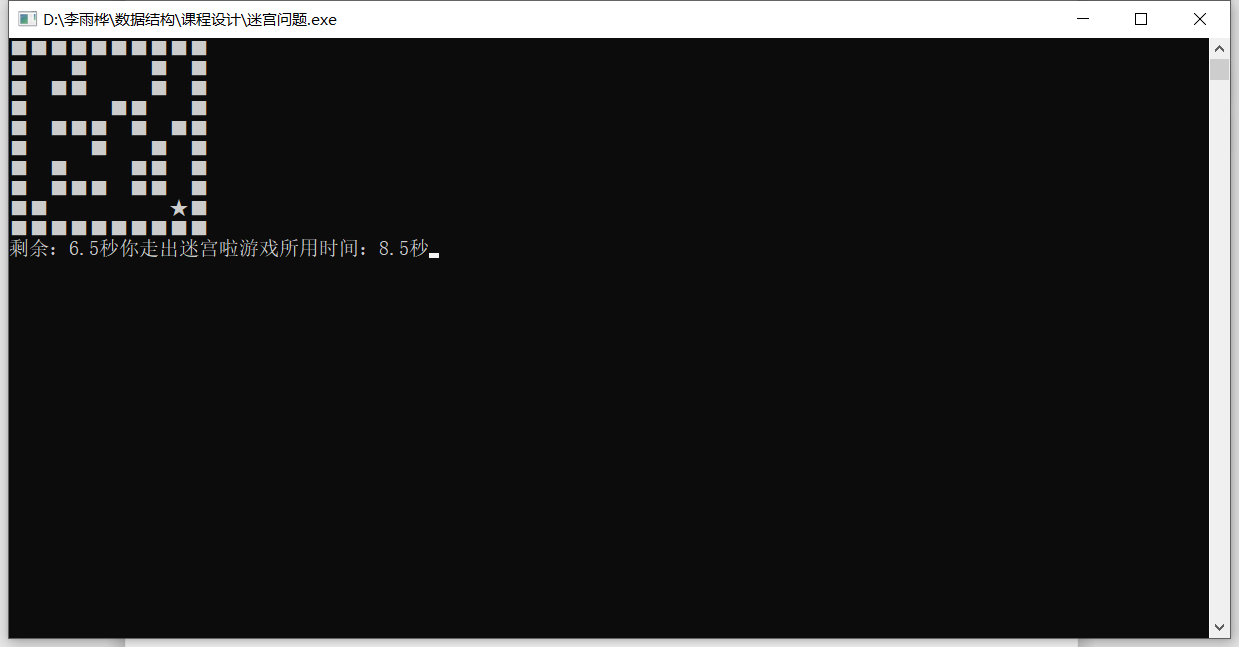


图5.3游戏在规定时间内到达终点结果图

结 论

数据结构是实践性很强的课程。课程设计是加强学生实践能力的一个强有力手段。课程设计要求学生在完成程序设计的同时能够写出比较规范的设计报告。严格实施课程设计这一-环节，对于学生基本程序设计素养的培养和软件工作者工作作风的训练，将起到显著的促进作用。

通过这次课程设计使我懂得了理论与实际相结合是很重要的，只有理论知识是远远不够的，比如栈的知识以前只知道如何建栈出栈，但是没有做过实际的问题，这次能做出实际的迷宫游戏我感到十分的骄傲。

同时对于c语言的实际应用又有了新的理解，比如如何检测键盘按键，如何将检测到的按键加以应用以达到自己想要做的东西，在迷宫手动设计开始游戏时有了诸多的思考，真正感受到自己在设计一个迷宫，去思考功能与实现。

我做的走迷宫游戏的设计中，对于自动走迷宫路径，本身自动寻找过程中没有出现很多问题，更多的问题出现在将手动走迷宫和自动走迷宫合起来的时候，一个printf()就让一个功能无法使用了，当时弄了很多天，想到底为什么？最后通过增大栈的初始分配空间或者清空栈解决了这个问题。说起来简单，但是当你换了一个个代码还是无法运行时的焦灼，现在想来还是一种历练。

通过上段出现的问题，我还了解了很多堆栈的知识，才明白原来数据结构中的栈是模拟栈的运行，同样也应模拟它的销毁。同时在运用栈的知识时，才发现原来模拟栈的知识是，程序员控制的是内存堆的空间，这样一个可控空间给程序员很大的便利。突然明白数据结构让我们学会的不仅仅是一种数据的存储，还有一些数据的释放，让我们更加明白电脑底层数据处理。我想在以后的日子里，我会更加注意存储空间的使用，释放，以及设计。

对于背景音乐，但是使用了很多方式去解决，有想过中断，但是查到的资料几乎都是单片机中断，而很少有windows的中断使用；还想过直接使用MAV的背景因为函数，但是因为编译器的限制，最终失败。最后选择自己写beep函数，但是因为不能做成背景音乐，然后找到一个windows的创建多线程函数解决了这个问题。也算了解了windows的很多多线程知识。

还有开始游戏的c语言的时间函数和按键检测函数，也算学习到的点，在设计地图的时候，移动而后面不留下痕迹也是思考了很久也遇到了很多问题，最后一一解决也是学到了很多。

最后，在这次课程设计中我还有很多思考和问题尚且没有明白，比如堆栈，printf会占用栈分配的空间吗？因为在设计过程中栈分配的空间明明是够的，单独自动寻找路径就能实现，而是合代码的时候，删除掉printf()，才能使用，否则无法运行，后来把栈的空间设置大才解决这个问题。查了很多资料还是不得其解。但是我相信通过以后的学习，一定可以更加了解电脑底层存储的问题。

# 参考文献

1. 谭浩强.《C程序设计（第三版）》，清华大学出版社，2005.7
2. 严蔚敏，吴伟民.《数据结构（C语言版）》，清华大学出版社，2007.6
3. 高一凡.《数据结构》算法实现及解析 (第二版),西安电子科技大学出版社,2004.04

# **附录：源代码**

#include<stdio.h>

#include<stdlib.h>

#include<malloc.h>

#include<iostream>

#include<windows.h>

#include <process.h>

#include<conio.h>

#include<time.h>

using namespace std;

#define STACK\_INIT\_SIZE 10000

#define N 10

typedef struct Pos

{

int x;

int y;

}PosType;

typedef struct SElemtp

{

PosType seat;

int di;

}SElemtp;

typedef struct

{

SElemtp \*base;

SElemtp \*top;

int stacksize;

}SqStack;

typedef int MazeType[N][N];

MazeType mz;

PosType begin,end;

int curstep=0;

int mm[N][N]={

{ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},

{ 0, 1, 1, 0, 1, 1, 1, 0, 1, 0},

{ 0, 1, 0, 0, 1, 1, 1, 0, 1, 0},

{ 0, 1, 1, 0, 1, 0, 0, 1, 1, 0},

{ 0, 1, 0, 0, 0, 1, 0, 1, 0, 0},

{ 0, 1, 1, 1, 0, 1, 1, 0, 1, 0},

{ 0, 1, 0, 1, 1, 1, 0, 0, 1, 0},

{ 0, 1, 0, 0, 0, 1, 0, 0, 1, 0},

{ 0, 0, 1, 1, 1, 1, 1, 1, 1, 0},

{ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

};

typedef struct

{

int freq; //赫兹

int duration;//持续时间

char text[32];//文本内容

}STU;

STU t[] = {{1318, 250, ""}, {1318, 500, ""}, {1318, 500, ""}, {1318, 250, ""}, {1175, 250, ""}, {1175, 500, ""},

{1046, 250, ""}, {1046, 500, ""}, {1046, 250, ""}, {1175, 250, ""}, {1046, 250, ""},

{880, 250, ""}, {784, 250, ""}, {784, 250, ""}, {784, 500, ""}, {784, 250, ""},

{880, 250, ""}, {784, 250, ""}, {1175, 250, ""}, {1318, 250, ""}, {1046, 500, ""},

{1046, 500, "\n"}, {659, 250, "池"}, {784, 250, "塘"}, {784, 500, "边"}, {784, 250, ""},

{659, 250, "的"}, {880, 250, "榕"}, {880, 250, "树"}, {988, 250, "上"}, {880, 250, ""},

{0, 250, ""}, {880, 250, "只"}, {880, 250, "了"}, {784, 250, "在"}, {1046, 500, "声"},

{1046, 250, "声"}, {1046, 250, "的"}, {1046, 250, "叫"}, {880, 250, ""}, {1046, 250, "着"},

{880, 250, "夏"}, {784, 500, "天"}, {784, 500, "，"}, {659, 250, "操"}, {784, 250, "场"},

{784, 500, "边"}, {784, 250, ""}, {659, 250, "的"}, {880, 250, "秋"}, {880, 250, "千"},

{988, 250, "上"}, {880, 250, "，"}, {0, 250, ""}, {880, 250, "只"}, {880, 250, "有"},

{784, 250, ""}, {1046, 500, "蝴"}, {1046, 375, "蝶"}, {1046, 125, ""}, {1046, 250, "停"},

{880, 250, "在"}, {880, 250, "上"}, {1046, 250, ""}, {1175, 500, "面"}, {1175, 500, "。"},

{1568, 250, "黑"}, {1568, 500, "板"}, {1568, 125, "上"}, {1568, 125, ""}, {1568, 250, ""},

{1568, 250, "老"}, {1318, 250, "师"}, {1175, 250, "的"}, {1046, 250, "粉"}, {1046, 500, "笔"},

{880, 250, "还"}, {880, 250, ""}, {880, 250, ""}, {1046, 250, "在"}, {880, 250, "拼"},

{1046, 250, "命"}, {1175, 250, "叽"}, {1175, 250, "叽"}, {1175, 250, "喳"}, {1046, 250, "喳"},

{1175, 250, "写"}, {1046, 250, "个"}, {1318, 250, "不"}, {1175, 250, ""}, {1175, 500, "停"},

{1175, 500, "，"}, {1318, 250, "等"}, {1318, 500, "待"}, {1318, 250, "着"}, {1318, 250, "下"},

{1175, 250, ""}, {1175, 500, "课"}, {1046, 250, "等"}, {1046, 500, "待"}, {1046, 250, "着"},

{1175, 250, "放"}, {1046, 250, ""}, {880, 250, "学"}, {784, 250, "，"}, {784, 250, "等"},

{784, 250, "待"}, {0, 250, ""}, {784, 250, "有"}, {880, 250, "趣"}, {784, 250, "的"},

{1175, 250, "童"}, {1318, 250, ""}, {1046, 500, "年"}, {1046, 500, "。"}, {1046, 500, "\n"},

{659, 250, "总"}, {784, 250, "是"}, {784, 500, "要"}, {784, 250, "等"}, {659, 250, "到"},

{880, 250, "睡"}, {880, 250, "觉"}, {988, 250, "前"}, {880, 250, ""}, {0, 250, ""},

{880, 250, "才"}, {880, 250, "知"}, {784, 250, "道"}, {1046, 500, "功"}, {1046, 250, "课"},

{1046, 250, "只"}, {1046, 250, "做"}, {880, 250, "了"}, {1046, 250, "一"}, {880, 250, "点"},

{784, 500, "点"}, {784, 500, "，"}, {659, 250, "总"}, {784, 250, "是"}, {784, 500, "要"},

{784, 250, "等"}, {659, 250, "到"}, {880, 250, "考"}, {880, 250, "试"}, {988, 250, "以"},

{880, 250, "后"}, {0, 250, ""}, {880, 250, "才"}, {880, 250, "知"}, {784, 250, "道"},

{1046, 500, "该"}, {1046, 375, "念"}, {1046, 125, "的"}, {1046, 250, "书"}, {880, 250, "都"},

{880, 250, "没"}, {1046, 250, "有"}, {1175, 500, "念"}, {1175, 500, "。"}, {1568, 250, "一"},

{1568, 500, "寸"}, {1568, 125, "光"}, {1568, 125, ""}, {1568, 250, ""}, {1568, 250, "阴"},

{1318, 250, "一"}, {1175, 250, "寸"}, {1046, 250, "金"}, {1046, 500, ""}, {880, 250, "老"},

{880, 250, "师"}, {880, 250, "说"}, {1046, 250, "过"}, {880, 250, "寸"}, {1046, 250, "金"},

{1175, 250, "难"}, {1175, 250, ""}, {1175, 250, "买"}, {1046, 250, ""}, {1175, 250, "寸"},

{1046, 250, ""}, {1318, 250, "光"}, {1175, 250, ""}, {1175, 500, "阴"}, {1175, 500, "，"},

{1318, 250, "一"}, {1318, 500, "天"}, {1318, 250, "又"}, {1318, 250, "一"}, {1175, 250, ""},

{1175, 500, "天"}, {1046, 250, "一"}, {1046, 500, "年"}, {1046, 250, "又"}, {1175, 250, "一"},

{1046, 250, ""}, {880, 250, "年"}, {784, 250, "，"}, {784, 250, "迷"}, {784, 250, "迷"},

{0, 250, ""}, {784, 250, "糊"}, {880, 250, "糊"}, {784, 250, "的"}, {1175, 250, "童"},

{1318, 250, ""}, {1046, 500, "年"}, {1046, 500, "。"}, {1046, 500, "\n"}, {1318, 500, "噢"},

{1318, 250, "一"}, {1318, 500, "天"}, {1318, 250, "又"}, {1318, 250, "一"}, {1175, 250, ""},

{1175, 500, "天"}, {1046, 250, "一"}, {1046, 500, "年"}, {1046, 250, "又"}, {1175, 250, "一"},

{1046, 250, ""}, {880, 250, "年"}, {784, 250, "，"}, {784, 250, "盼"}, {784, 250, "望"},

{0, 250, ""}, {784, 250, "长"}, {880, 250, "大"}, {784, 250, "的"}, {1175, 250, "童"},

{1318, 250, ""}, {1046, 500, "年"}, {1046, 500, "。"}, {0, 500, ""}, };

void yingyue(void \*)

{

int i;

//结构体数组的长度sizeof(t)/sizeof(STU)得到，总结构体数组大小 / 单个结构体大小

for (i=0; i<sizeof(t)/sizeof(STU); i++)

{

// printf("%s",t[i].text);

Beep(t[i].freq,t[i].duration);

}

\_endthread();

}

void Mazeinit()

{

for(int i=0;i<N;i++)

{

for(int j=0;j<N;j++)

{

mz[i][j]=mm[i][j];

}

}

begin.x=1;

begin.y=1;

end.x=8;

end.y=8;

}

void MazePrint()

{

for(int i=0;i<N;i++)

{

for(int j=0;j<N;j++)

{

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

cout<<"■";

else if(mz[i][j]==5)

cout<<"★";

else

cout<<" ";

}

printf("\n");

}

}

void NextPos(PosType &c,int di)

{

PosType direc[5]={{0,-1},{0,1},{-1,0},{1,0},{0,0}};

c.x+=direc[di].x;

c.y+=direc[di].y;

}

void InitStack(SqStack &S)

{

S.base=(SElemtp \*)malloc(STACK\_INIT\_SIZE\*sizeof(SElemtp));

S.top=S.base;

S.stacksize=STACK\_INIT\_SIZE;

}

void Push(SqStack &S,SElemtp e)//入栈

{

S.base=(SElemtp \*)realloc(S.base,(S.stacksize+2)\*sizeof(SElemtp));

S.stacksize++;

\*S.top=e;

S.top++;

}

bool Pop(SqStack &S,SElemtp &e)//出栈

{

if(S.top==S.base)

return false;

e=\*--(S.top);

S.stacksize--;

return true;

}

int StackEmpty(SqStack S)

{

if(S.top==S.base)

return 1;

else

return 0;

}

void ShowShow(SqStack S)

{

while(S.top>S.base)

{

Sleep(200);

mz[S.base->seat.x][S.base->seat.y]=5;

++S.base;

system("cls");

printf("此迷宫从入口到出口的一条路径如下:\n");

MazePrint();

}

printf("\n");

}

void DestroyStack(SqStack &S)// 销毁栈S，S不再存在

{

free(S.base);

S.base=NULL;

S.top=NULL;

S.stacksize=0;

}

int MazePath(SqStack &S,PosType start,PosType end)

{

PosType curposs;

SElemtp e;

InitStack(S);

curposs=start;

do

{

if(mz[curposs.x][curposs.y]==1)

{

mz[curposs.x][curposs.y]=-1;

e.seat=curposs;

e.di=0;

Push(S,e);

curstep++;

if(curposs.x==end.x&&curposs.y==end.y)

return 1;

NextPos(curposs,e.di);

}

else

{

if(!StackEmpty(S))

{

Pop(S,e);

curstep--;

while(e.di==3&&!StackEmpty(S))

{

mz[e.seat.x][e.seat.y]=-1;

Pop(S,e);

curstep--;

}

if(e.di<3)

{

e.di++;

Push(S,e);

curstep++;

curposs=e.seat;

NextPos(curposs,e.di);

}

}

}

}while(!StackEmpty(S));

return 0;

}

void Print()// 输出迷宫的解(m数组)

{

int i,j;

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

{

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

printf("%3d",mz[i][j]);

printf("\n");

}

}

int main()

{

int choose;

SqStack S;

\_beginthread(yingyue,0,NULL);

while(1)

{

Mazeinit();

system("cls");

printf("╔════════════════════════════════════════════════════════════════════╗\n");

printf("║ 欢迎进入走迷宫游戏演示系统 ║\n");

printf("║ --------------------------------------------- ║\n");

printf("║ 1.开始游戏 ║\n");

printf("║ 2.查看能到达终点的路径 ║\n");

printf("║ 3.修改地图 ║\n");

printf("║ 4.游戏提示 ║\n");

printf("║ 5.退出系统 ║\n");

printf("╚════════════════════════════════════════════════════════════════════╝\n");

printf(">>>>>>>>>>>>\n");

printf("迷宫地图是这样子的哦\n");

MazePrint();

printf("请选择你的操作:[ ]\b\b");

scanf("%d",&choose);

switch(choose)

{

// case 1:

// {

// curpos=begin;

// Show();

// }

// break;

case 2:curstep=0;

if(MazePath(S,begin,end))

{

printf(">>>>>>>>>运动路径马上来了，不要走哦！<<<<<<<<<<<");

ShowShow(S);

printf("一共走了%d步，就到达终点了！",curstep);

DestroyStack(S);

}

else

printf("此迷宫没有从入口到出口的路径\n");

break;

// case 3:

// {

// system("cls");

// Editor();

// }

// break;

// case 4:

// tip()

// color();

// break;

// case 5:exit(0);break;

}

Sleep(1000);

}

}

|  |  |
| --- | --- |
| 题目 | 走迷宫游戏 |
| 自我评价：在这次课程设计中，我感到收获良多，很多具体的我已经在结论中写到，同时对于代码问题的理解也理解的比较深入，对于电脑的底层存储，也有了更深的了解。但是最重要的是解决问题的能力得到了很大的提升。以及如何将自己学到的知识用于实践也有了新的进展。  虽然最后我分到的代码比较少，但是个人对于整个课程贡献一点不少，几乎每个功能都有所参与设计程序，最后的分工是根据功能划分，当然我还有很多不足，还有很多功能不完善，希望后续改进，使迷宫功能更强，让游戏玩家游戏体验感更好。  最后对于这次的课程设计，我觉得我已经达到课程设计的目标，有需要改进的后续也会继续学习改进。 | |
| 教师评语：  教师签名：  2019 年 12月31日 | |