МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

Кафедра «Информационные системы»

Дисциплина «Основы алгоритмизации и программирования»

Лабораторная работа №10

Структуры. Работа с файлами. Текстовое меню

Выполнил:

студент гр. ПИбд-12

Дозорова Алена

Ульяновск, 2021 г

Задание:

1. Собрат код из презентации
2. Добавить элемент в заданный столбец
3. Сделать меню выбора действий
4. Удалить «землю» из заданной горки
5. Срыть самую высокую горку
6. Подсыпать земли в низину
7. Сохранить состояние в файле
8. Доработать игру

Код:

// Laba10.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "stdio.h"

#include <stdlib.h>

#define WIN32\_LEAN\_AND\_MEAN

// индексы входа и выхода

struct Position {

int i, j;

};

// Уровень игры

struct Level {

int map[10][10]; // карта уровня

// 0 – воздух

// 1 – земля

int n; // количество строк

int m; // количество столбцов

struct Position entry; // вход

struct Position exit; // выход

};

void printLevel(struct Level\* level) {

for (int i = 0; i < level->n; i++) {

for (int j = 0; j < level->m; j++) {

printf("%5d ", level->map[i][j]);

}

printf("\n");

}

printf("\n");

printf("Entry i = %d; j = %d\n", level->entry.i, level->entry.j);

printf("Exit i = %d; j = %d\n\n", level->exit.i, level->exit.j);

}

char filename[] = "1.txt";

int loadLevel(struct Level\* level) {

FILE\* fin = fopen(filename, "rt");

if (fin == NULL) {

printf("File %s is not opened", filename);

return 0;

}

fscanf(fin, "%d", &level->n);

fscanf(fin, "%d", &level->m);

for (int i = 0; i < level->n; i++) {

for (int j = 0; j < level->m; j++) {

fscanf(fin, "%d", &level->map[i][j]);

}

}

fscanf(fin, "%d", &level->entry.i);

fscanf(fin, "%d", &level->entry.j);

fscanf(fin, "%d", &level->exit.i);

fscanf(fin, "%d", &level->exit.j);

fclose(fin);

return 1;

}

char filename2[] = "D:\\моя папка\\институт\\ОАиП\\лабы\\Labs\\Laba10\\2.txt";

int saveLevel(struct Level\* level) {

FILE\* fin = fopen(filename, "wt");

if (fin == NULL) {

printf("File %s is not opened", filename);

return 0;

}

fprintf(fin, "%d ", level->n);

fprintf(fin, "%d\n", level->m);

for (int i = 0; i < level->n; i++) {

for (int j = 0; j < level->m; j++) {

fprintf(fin, "%d ", level->map[i][j]);

}

fprintf(fin, "\n");

}

fprintf(fin, "%d ", level->entry.i);

fprintf(fin, "%d\n", level->entry.j);

fprintf(fin, "%d ", level->exit.i);

fprintf(fin, "%d\n", level->exit.j);

fclose(fin);

return 1;

}

void addInColumn(struct Level\* level, int indexCol)

{

for (int i = level->n - 1; i >= 0; i--)

{

if (level->map[i][indexCol] == 0)

{

level->map[i][indexCol] = 1;

break;

}

}

}

void deleteChoose(struct Level\* level, int indexCol)

{

for (int i = 0; i < level->n; i++)

{

if (level->map[i][indexCol] == 1)

{

level->map[i][indexCol] = 0;

break;

}

}

}

void deleteMaxColumn(struct Level\* level)

{

int kol\_z = 0;

int maxcol = 0;

int indexmax = 0;

for (int j = 0; j < level->m; j++)

{

for (int i = 0; i < level->n; i++)

{

if (level->map[i][j] == 1)

{

kol\_z += 1;

if (maxcol < kol\_z)

{

maxcol = kol\_z;

indexmax = j;

}

}

}

kol\_z = 0;

}

for (int i = 0; i < level->n; i++)

{

if (level->map[i][indexmax] == 1)

{

level->map[i][indexmax] = 0;

}

}

}

void addMinColumn(struct Level\* level)

{

int kol\_z = 0;

int mincol = 1000;

int indexmin = 0;

for (int j = 0; j < level->m; j++)

{

for (int i = 0; i < level->n; i++)

{

if (level->map[i][j] == 1)

{

kol\_z += 1;

}

}

if (mincol > kol\_z)

{

mincol = kol\_z;

indexmin = j;

}

kol\_z = 0;

}

for (int i = level->n-1; i > 0; i--)

{

if (level->map[i][indexmin] == 0)

{

level->map[i][indexmin] = 1;

break;

}

}

}

int main()

{

struct Level g;

int k = 0;

int j;

do

{

switch (k) {

case 0:

loadLevel(&g);

break;

case 1:

printf("\n Input j >");

scanf("%d", &j);

addInColumn(&g, j);

break;

case 2:

printf("\n Input j =");

scanf("%d", &j);

deleteChoose(&g, j);

break;

case 3:

deleteMaxColumn(&g);

break;

case 4:

addMinColumn(&g);

break;

case 5:

saveLevel(&g);

break;

}

printLevel(&g);

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

printf("Please select action:\n");

printf("0: Reload file\n");

printf("1: addInColumn\n");

printf("2: deleteChoose\n");

printf("3: deleteMaxColumn\n");

printf("4: addMinColumn\n");

printf("5: saveLevel\n");

printf("-1: Exit\n");

scanf("%d", &k);

} while (k != -1);

loadLevel(&g);

printLevel(&g);

addInColumn(&g, 1);

printLevel(&g);

addInColumn(&g, 1);

printLevel(&g);

{

int x;

scanf("%d", &x);

}

saveLevel(&g);

}

Скрины:







