МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра информационных систем

Отчет по дисциплине Основы алгоритмизации и программирование

Домашняя работа №1

Выполнил студент гр. ИСЭбд-11

Гапон Николай Иванович

Преподаватель

Власенко Олег Федосович

ССЫЛКА НА GIT <https://github.com/NikGapon/1year/tree/main/lab9>

ДОМАШНЯЯ РАБОТА

Доделать задачи 1-6

Задача 7

Переделать кода так так, чтобы игрок мог поедать монстров если он собрал не менее 5 золота

Задача 8

Придумать и реализовать новое поведение робота

Задача 9

Загрузка игры из файла

void moveToLeft() {

int i, j;

i = 0;

while (i < N) {

j = 1;

while (j < M) {

if (a[i][j] == 3) {

if (a[i][j - 1] == 0) {

a[i][j - 1] = 3;

a[i][j] = 0;

steps++;

}

else if (a[i][j - 1] == 1) {

a[i][j - 1] = 3;

a[i][j] = 0;

steps++;

gold++;

}

}

j++;

}

i++;

}

}

void moveToRight() {

int i = 0;

while (i < N) {

int j = M - 2;

while (j >= 0) {

if (a[i][j] == 3) {

if (a[i][j + 1] == 0) {

a[i][j + 1] = 3;

a[i][j] = 0;

steps++;

}

else if (a[i][j + 1] == 1) {

a[i][j + 1] = 3;

a[i][j] = 0;

steps++;

gold++;

}

}

j--;

}

i++;

}

}

void moveUp() {

int i = 1;

while (i < N) {

int j = 0;

while (j < M) {

if (a[i][j] == 3) {

if (a[i - 1][j] == 0) {

a[i - 1][j] = 3;

a[i][j] = 0;

steps++;

}

else if (a[i - 1][j] == 1) {

a[i - 1][j] = 3;

a[i][j] = 0;

steps++;

gold++;

}

}

j++;

}

i++;

}

}

void UPST() {

int i = 0;

while (i < N) {

int j = 0;

while (j < M) {

if (a[i][j] == 3) {

a[i - 1][j] = 2;

steps++;

}

j++;

}

i++;

}

}

void UPST\_0() {

int i = 0;

while (i < N) {

int j = 0;

while (j < M) {

if (a[i][j] == 3) {

a[i - 1][j] = 0;

steps++;

}

j++;

}

i++;

}

}

void MoveMonsters() {

int i = 0;

while (i < N) {

int j = 0;

while (j < M) {

if (a[i][j] == 4) {

int di = rand() % 3 - 1; // от -1 до +1

int dj = 0;

if (di == 0) {

dj = rand() % 3 - 1; // от -1 до +1

}

if (di != 0 || dj != 0) {

if (i + di >= 0 && i + di < N

&& j + dj >= 0 && j + dj < M)

{

a[i][j] = 0;

if (a[i + di][j + dj] == 3) {

win = -1;

}

a[i + di][j + dj] = -4;

}

}

}

j++;

}

i++;

}

i = 0;

while (i < N) {

int j = 0;

while (j < M) {

if (a[i][j] == -4) {

a[i][j] = 4;

}

j++;

}

i++;

}

}

void genergold() {

int i, j;

int k;

for (k = 0; k < 10 / 2; k++) {

i = rand() % N;

j = rand() % M;

if (a[i][j] == 0) {

a[i][j] = 1;

}

}

}

void generateLevel() {

// Коды ячеек

// 0 - свободно

// 1 - золото

// 2 - стена

// 3 - игрок

// 4 - МОНСТР!

// всё заново!

steps = 0;

gold = 0;

// все ячейки делаем свободными

int i, j;

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

for (j = 0; j < M; j++) {

a[i][j] = 0;

}

}

// в часть ячеек помещаем много ЗОЛОТА

int k;

for (k = 0; k < (N \* M) / 2; k++) {

i = rand() % N;

j = rand() % M;

a[i][j] = 1;

}

for (k = 0; k < (N \* M) / 5; k++) {

i = rand() % N;

j = rand() % M;

a[i][j] = 2;

}

// в одну из ячеек помещаем игрока

i = rand() % N;

j = rand() % M;

a[i][j] = 3;

}

void moveDown() {

int i = N;

while (i >= 0) {

int j = 0;

while (j < M) {

if (a[i][j] == 3) {

if (a[i + 1][j] == 0) {

a[i + 1][j] = 3;

a[i][j] = 0;

steps++;

}

else if (a[i + 1][j] == 1) {

a[i + 1][j] = 3;

a[i][j] = 0;

steps++;

gold++;

}

}

j++;

}

i--;

}

}

void deadmonster() {

int i = 0;

while (i < N) {

int j = 0;

while (j < M) {

if (a[i][j] == 4) {

if (gold > 3) {

a[i][j] = 0;

gold = gold - 3;

}

}

j++;

}

i++;

}

}

int sizeX = 36;

int sizeY = 30;

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)

{

switch (message)

{

case WM\_CREATE:

SetTimer(hWnd, 1, 500, 0); // Запускаем таймер

// srand(time(NULL)); // Перезапуск генератора случайных чисел

break;

case WM\_TIMER:

MoveMonsters(); // перемещаем монстров

InvalidateRect(hWnd, NULL, TRUE);

break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

default:

return DefWindowProc(hWnd, message, wParam, lParam);

case WM\_PAINT:

{

PAINTSTRUCT ps;

HDC hdc = BeginPaint(hWnd, &ps);

DrawField(hdc);

EndPaint(hWnd, &ps);

}

break;

case WM\_KEYDOWN:

switch (wParam)

{

case VK\_DOWN:

moveDown();

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_LEFT:

moveToLeft();

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_UP:

moveUp();

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_RIGHT:

moveToRight();

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_TAB:

UPST();

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_CAPITAL:

UPST\_0();

InvalidateRect(hWnd, NULL, TRUE);

break;

case 0x4e: // N - New

generateLevel();

InvalidateRect(hWnd, NULL, TRUE);

break;

case VK\_F1:

genergold();

InvalidateRect(hWnd, NULL, TRUE);

case VK\_F2:

deadmonster();

InvalidateRect(hWnd, NULL, TRUE);

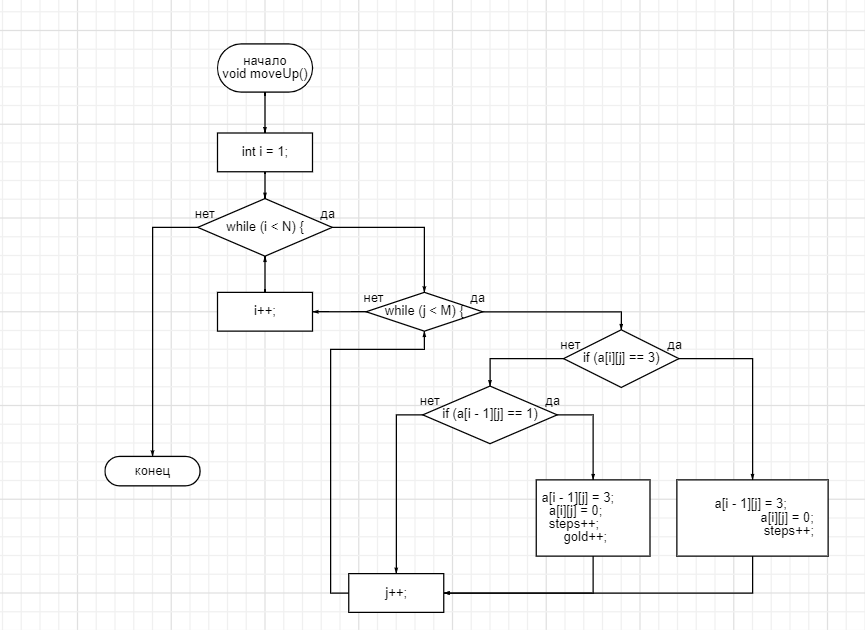
}

break;

return 0;

}

}



ВЫВОД

В данной лабораторной работе мы научились ещё в большей мере работать с двухмерными массивами

И таймером