**ЗАДАНИЕ**

на курсовое проектирование по МДК 02.02 Управление проектами

профессионального модуля ПМ02 Участие в разработке информационных систем

студента Пестова Константина Сергеевича

специальности 09.02.04 «Информационные системы (по отраслям)»

курса 3 группы ИС-17

Федерального казенного профессионального образовательного учреждения "Новокузнецкий государственный гуманитарно-технический колледж-интернат" Министерства труда и социальной защиты Российской Федерации

Тема задания: "Разработка программного продукта "Игра МАРИО" с применением СУБД"

**ИСХОДНЫЕ ДАННЫЕ**

Разрабатываемая «Игра МАРИО» должна быть реализована с использованием базы данных (рекомендовано MS SQL Server). БД должна находиться в третьей нормальной форме.

В игре должно быть предусмотрено регистрация и авторизация пользователей. Хранения результатов прохождения игры каждым игроком. Возможность сохранения результатов игры для дальнейшего прохождения. Суть игры состоит в управлении персонажем, который проходит препятствия и собирает бонусы.

Курсовой проект выполняется в следующем объеме: 25-30 страниц

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Введение

1. Техническое задание

2 Общая часть

2.1 Выбор и обоснование средств реализации БД

2.2 Выбор и обоснование средств разработки программного продукта

* 1. Анализ прототипов существующих программных продуктов

3 Специальная часть

3.1 Разработка алгоритма и проектирование структуры БД

3.2 Разработка программного продукта

3.3 Руководство пользователя

Заключение

Список использованных источников

Приложения

**Рекомендуемая литература**

1. Советов, Б. Я. [Базы данных : учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 3-е изд., перераб. и доп. — М. : Издательство Юрайт, 2019. — 420 с. — (Серия : Профессиональное образование). — ISBN 978-5-534-09324-7.](https://biblio-online.ru/book/bazy-dannyh-438438?utm_campaign=rpd&utm_source=web&utm_content=7a051c534c738558cbf6ad919829933f)
2. Илюшечкин, В. М. [Основы использования и проектирования баз данных : учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М. : Издательство Юрайт, 2019. — 213 с. — (Серия : Профессиональное образование). — ISBN 978-5-534-01283-5.](https://biblio-online.ru/book/osnovy-ispolzovaniya-i-proektirovaniya-baz-dannyh-437670?utm_campaign=rpd&utm_source=web&utm_content=7a051c534c738558cbf6ad919829933f)
3. Кудрина, Е. В. [Основы алгоритмизации и программирования на языке c# : учеб. пособие для СПО / Е. В. Кудрина, М. В. Огнева. — М. : Издательство Юрайт, 2019. — 322 с. — (Серия : Профессиональное образование). — ISBN 978-5-534-10772-2.](https://biblio-online.ru/book/osnovy-algoritmizacii-i-programmirovaniya-na-yazyke-c-431505?utm_campaign=rpd&utm_source=web&utm_content=15ea88963f0356741eae8a500596396d)
4. Черпаков, И. В. [Основы программирования : учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2019. — 219 с. — (Серия : Профессиональное образование). — ISBN 978-5-9916-9984-6.](https://biblio-online.ru/book/osnovy-programmirovaniya-436557?utm_campaign=rpd&utm_source=web&utm_content=15ea88963f0356741eae8a500596396d)
5. Кувшинов, Д. Р. [Основы программирования : учеб. пособие для СПО / Д. Р. Кувшинов. — М. : Издательство Юрайт, 2019. — 105 с. — (Серия : Профессиональное образование). — ISBN 978-5-534-07560-1.](https://biblio-online.ru/book/osnovy-programmirovaniya-441571?utm_campaign=rpd&utm_source=web&utm_content=15ea88963f0356741eae8a500596396d)

**Календарный план выполнения курсового проекта**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Содержание работ | Примерный объем работы в % | Месяцы и дни выполнения | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
| 1 | Постановка целей и задач курсового проектирования | 5 | 5 |  |  |  |  |  |  |  |  |  |
| 2 | Сбор сведений об изучаемой области | 10 |  | 15 |  |  |  |  |  |  |  |  |
| 3 | Сбор сведений об изучаемой области | 10 |  |  | 25 |  |  |  |  |  |  |  |
| 4 | Проектирование базы данных | 10 |  |  |  | 35 |  |  |  |  |  |  |
| 5 | Созданиебазы данных | 10 |  |  |  |  | 45 |  |  |  |  |  |
| 6 | Проектирование пользовательского интерфейса | 10 |  |  |  |  |  | 55 |  |  |  |  |
| 7 | Создание пользовательского интерфейса | 20 |  |  |  |  |  |  | 75 |  |  |  |
| 8 | Проведение исследования | 10 |  |  |  |  |  |  |  | 85 |  |  |
| 9 | Анализ полученной информации. | 10 |  |  |  |  |  |  |  |  | 95 |  |
| 10 | Выводы. Защита КП | 5 |  |  |  |  |  |  |  |  |  | 100 |

Дата выдачи «16» сентября 2019 года

Срок окончания «20» декабря 2019 года

Преподаватель-руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Грисман С.С./

Утверждено на заседании МК Информационные системы, протокол № \_\_

Председатель МК Информационные системы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/Грисман С.С./

# **ВВЕДЕНИЕ**

В последнее время компьютерные технологии начали свое стремительное развитие, появление мощнейших графических ускорителей и центральных процессоров способствовало не менее бурному развитию индустрии компьютерных игр. Выдающиеся разработки этой отрасли – это сложнейшие программы, как правило, с очень высокими требованиями к аппаратной части компьютера. Но, и существует великое множество небольших компьютерных игр, которые также популярны. Для возможности отдохнуть в перерыве от выполнения какой-либо работы оператору компьютера не всегда требуется новейшая компьютерная игра, а зачастую использовать её не позволяет маломощное оборудование офисного компьютера. Именно этой цели – отдыху от монотонной работы служит разрабатываемая в рамках данного курсового проекта игра «МАРИО».

Актуальность темы обуславливается необходимостью создания продукта, который будет помогать развивать реакцию, внимание и приносить удовольствие от игры, с использованием информационных технологий, которые прочно вошли в нашу повседневную жизнь.

Выдвигается гипотеза, что подобные игры помогают развивать концентрацию, внимание, реакцию принятия решений и помогает отдохнуть от повседневности.

Целью данного курсового проекта, является разработка программного продукта "Игра МАРИО" с применением СУБД.

Объектом в данной работе является игра «МАРИО».

Предметом в данной работе является автоматизация процессов игры.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* Разработать техническое задание
* Проанализировать прототипы;
* Произвести анализ и проектирование структуры базы данных;
* Выбрать и обосновать программные средства реализации;
* Разработать программный продукт;
* Провести тестирование и отладить программный продукт.
* Разработать руководство пользователя

1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ПРОГРАММНОГО ПРОДУКТА

Разработка приложения для автоматизации игрового процесса

Прототипом данной системы можно считать игру «МАРИО»

Требования к функциональным характеристикам, следующие:

1. Необходимо предусмотреть два режима работы – режим администратора и режим пользователя.

2. В режиме пользователя должна быть возможность зарегистрироваться, заполнить свою анкету (Nickname, password). Реализовывать возможность сохранения прогресса игры.

3. В режиме пользователя можно полноценно играть, а также посмотреть историю своих достижений.

4. При повторном входе в игру, должна быть возможность продолжать с места сохранения.

5. В режиме администратора должна быть возможность просматривать, удалять старые сохранения.

6. В режиме администратора должна быть реализована возможность регистрации пользователя в игре.

7. В игре должно присутствовать: препятствия, враги, бонусы, оружие, награда, количество жизней, счетчик монет, таймер времени дающееся на прохождение уровня.

Требования к информационному обеспечению, следующие:

К приложению должна быть предоставлена документация, содержащая техническое описание продукта и руководство пользователя.

Требования к лингвистическому обеспечению, следующие:

Приложение должно иметь интерфейс и документацию на русском языке.

Требования к программному обеспечению, следующие:

Разработка программного продукта предусматривается под OS Windows не ниже версии 7.

Требования к техническому обеспечению, следующие:

Для работы приложения требуется MS SQL Server, Visual Studio С#. Также, необходимо не менее 2Гб оперативной памяти и 800Мб свободного места на жестком диске.

Анализ и выполнение задачи

Общий анализ

Для создания проекта использовалась программа Microsoft Visual C++ 2013. Проект создавался методом проб и ошибок, с постоянными изменениями и улучшениями, как в плане производительности, так и в плане удобства для пользователя. Исходя из того, что нет предела совершенству, можно сказать, что программа не окончена и требует дополнений.

Программа состоит из файла исходного кода и двух текстовых документов.

Программа имеет возможность считывать карту из текстового документа и самостоятельно дополнять ее по мере прохождения. Конец игры зависит только от желания игрока (пользователя) продолжать игру и в любой момент имеет выход.

В игре существуют следующие объекты:

Таблица 1. Игровые объекты

|  |  |
| --- | --- |
| Тип объекта | Условное обозначение в консоли |
| Персонаж | 7 (цвет - серый) |
| Малое дерево | 2 (цвет - зелёный)  2  6 (цвет - коричневый) |
| Среднее дерево | 10 (цвет – светло-зелёный)  10 10 10  6 (цвет - коричневый) |
| Бонусный блок | 4 (цвет - красный) |
| Земля | 1 (цвет - синий) |
| Воздух | 0 (цвет - черный) |

Алгоритм решения

Объявляю два двумерных массива А(массив, отвечающий за фон будущей карты) и В(массив, отвечающий за блоки, на которые можно воздействовать). А так же перечисление, отвечающее за цвет объектов.

Точка входа программы находится в файле Mario\_1.cpp.

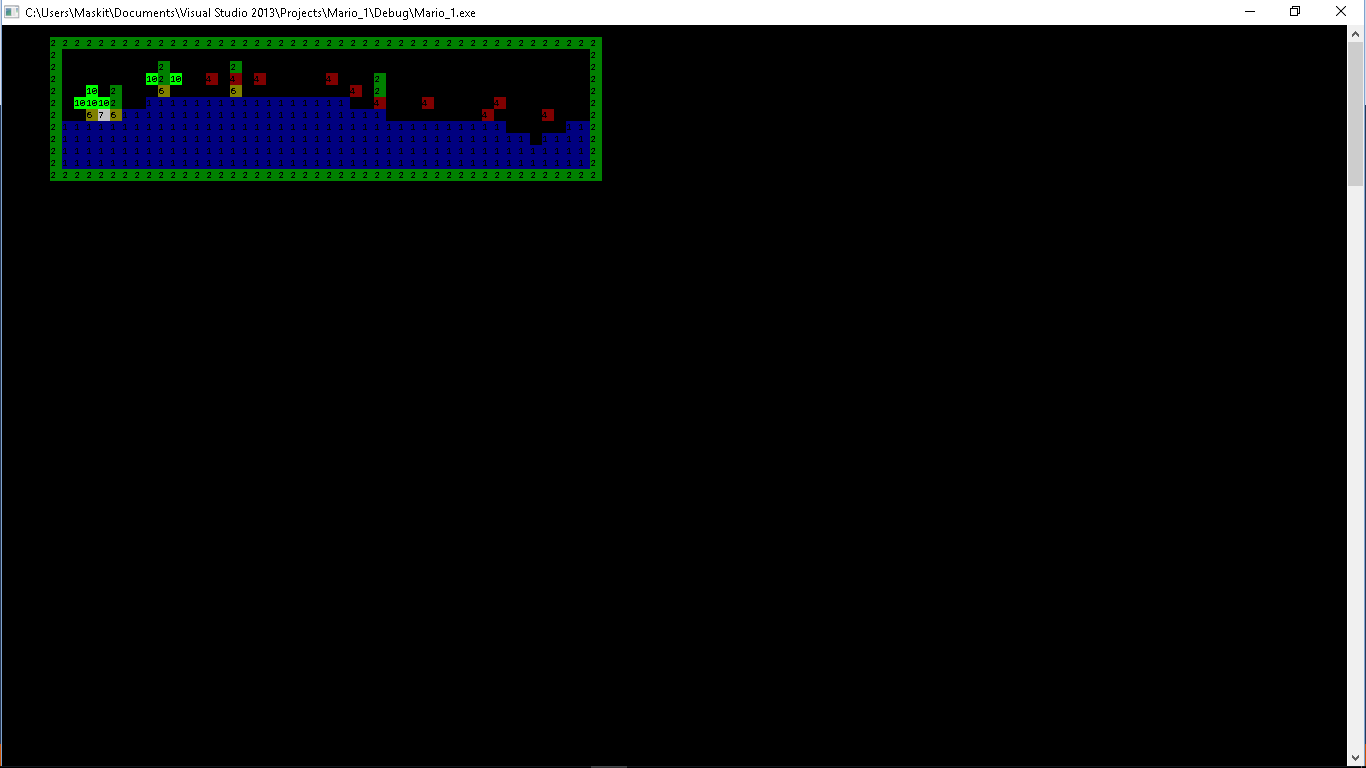
В первую очередь, программа осуществляет вызов текстового документа, с которого считывает значения и присваивает их каждому элементу массива А. Массив В заполняется нулевыми значениями.

С помощью функций «void LTree(int j)», «void MTree(int j)» создаются деревья для красоты программы с вероятностью 1:10 и 13:100 соответственно и вводятся в массив А, при чем они создаются на «поверхности», то есть над первым значением «1» в этом массиве в столбце j. Учитывается вероятность того, что верхушки деревьев могут выходить за верх карты, поэтому они обрезаются сверху.

Функция «void Coins(int j)» создает бонусные блоки с вероятностью 6:100, которые «лежат» на «земле», или 1:10, которые «зависают» над «землей» на высоте одного блока в массиве В в столбце j.

На этом первичная генерация карты окончена, можно переходить к «основным действиям».

Создаю персонажа, который появляется в этом мире на втором столбце, стоящий на земле.

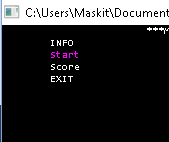


Далее идет цикл типа do…while, который определяет движение персонажа по нажатию игроком клавиш. При каждом действии происходит отрисовка карты с помощью функции «void mass(int N, int M)».

«W» - прыжок. Персонаж подпрыгивает на один блок вверх и ждет следующей команды. Есть возможность прыгнуть вверх, вперед и назад. Учитывается, что при прыжке персонаж может подобрать бонусный блок. В таком случае, прибавляется одно очко.

«D» - движение вперед. Происходит движение вперед, если после этого действия персонаж находится не на земле, то он падает. Учитывается, что при движении персонаж может подобрать бонусный блок. В таком случае, прибавляется одно очко. Если персонаж подходит близко к краю карты (за 15 блоков до конца), то к ней прибавляется новый столбец с помощью функции «int main1(int N, int M)».

«A» - движение назад. Происходит движение назад, если после этого действия персонаж находится не на земле, то он падает. Учитывается, что при движении персонаж может подобрать бонусный блок. В таком случае, прибавляется одно очко. Если перед этим происходило создание нового столбца, то карта сдвигается назад.



В программе существует возможность обратиться к Меню игры. На данный момент там можно сохранить свой результат в файл на ПК, посмотреть информацию об игре, или выйти из нее. В будущем планируется создать рейтинговую таблицу, в которой будет происходить сравнение предыдущих результатов и выводить первые десять мест и имена чемпионов по количеству очков.

«Q» - вызов меню. Управление выбранным пунктом меню производится клавишами «W» и «S». Для выбора пункта необходимо нажать клавишу «D». Сохранение производится автоматически при выборе пункта меню «Score».

«P» - конец цикла. Происходит выход из основной части программы. После чего происходит освобождение памяти массивов А и В и завершение программы.

Отрисовка

Как уже было сказано, цвет был добавлен для удобства игрока и зрелищности программы. Изменение цвета фона и шрифта происходит с помощью функции «void SetColor(int text, int background)»

Вывод

В ходе выполнения курсовой работы были освоены навыки работы с файлами, двумерными и одномерными динамическими массивами, функциями, строками и циклами. Отработаны собственные алгоритмы и методы выполнения работы. Создана рабочая программа игрового формата, имеющая много возможностей улучшения и оптимизации.

Приложение 1

язык программирование файл массив

Код программы

#include "stdafx.h"

#include <conio.h>

#include <iostream>

#include <cstdlib>

#include <fstream>

#include <string>

#include <Windows.h>

using namespace std;

int \*\*A; // Массив "заднего" фона

int \*\*B; // Массив "переднего" фона

enum ConsoleColor

{

Black = 0,

Blue = 1,

Green = 2,

Cyan = 3,

Red = 4,

Magenta = 5,

Brown = 6,

LightGray = 7,

DarkGray = 8,

LightBlue = 9,

LightGreen = 10,

LightCyan = 11,

LightRed = 12,

LightMagenta = 13,

Yellow = 14,

White = 15

};

void SetColor(int text, int background)

{

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hStdOut, (WORD)((background << 4) | text));

}

void menu(int c, int v, int score)

{

cout << "\t\t\t\*\*\*MENU\*\*\*\n";

if (c == 1){ SetColor(13, 0); }

cout << "\tINFO\n"; SetColor(15, 0);

if (v == 1 && c == 1){ cout << "\tЭто информацияV1.1\n"; }// информация о программе

if (c == 2){ SetColor(13, 0); }

cout << "\tStart\n"; SetColor(15, 0); // переходит к разделу игры

if (v == 1 && c == 2){}

if (c == 3){ SetColor(13, 0); } // выдает 10 лучших результатов

cout << "\tScore\n"; SetColor(15, 0);

if (v == 1 && c == 3){

cout << "\tSaved\n";

ofstream in("C://Users//Maskit//Desktop//php3.txt");

in << "Ваш счет: " << score;

in.close();

} //Закрываем файл

if (c == 4){ SetColor(13, 0); } // выход из программы

cout << "\tEXIT"; SetColor(15, 0);

if (v == 1 && c == 4){exit(0);}

}

void LTree(int j){

int i, r;

i = 1;

r = 1 + rand() % 100;

while (A[i][j] != 1)

{

++i;

}

if (r < 10){

if (i > 1){ A[i - 1][j] = 6; }

if (i > 2){ A[i - 2][j] = 10; A[i - 2][j - 1] = 10; A[i - 2][j + 1] = 10; }

if (i > 3){ A[i - 3][j] = 10; }

}

}

void MTree(int j){

int i, r;

i = 1;

r = 1 + rand() % 100;

while (A[i][j] != 1)

{

++i;

}

if (r < 13){

if (i > 1){ A[i - 1][j] = 6; }

if (i > 2){ A[i - 2][j] = 2; }

if (i > 3){ A[i - 3][j] = 2; }

}

}

void Coins(int j){

int i, r;

i = 1;

r = 1 + rand() % 100;

while (A[i][j] != 1)

{

++i;

}

if (r < 6){if (i > 1){ B[i - 1][j] = 4; }}

if (r >= 6 && r<14){ if (i > 2){ B[i - 2][j] = 4; } }

}

void mass(int N, int M){

int i, j;

system("cls");

for (i = 0; i < M-1; ++i){

cout << "\n\t";

for (j = N - 46; j < N; ++j){

if(i<1 || j<N-45 || i>M-2 || j>N-2){ SetColor(0,2); cout<<"2 "; SetColor(15,0);}

else{

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

SetColor(0, A[i][j]);

cout << A[i][j];

if (A[i][j] < 10)cout << " ";

}

else

{

SetColor(0, B[i][j]); cout << B[i][j]; if (B[i][j] < 10) cout << " ";

}

SetColor(15, 0);

}

}

}

}

int main1(int N, int M)// создание нового столбца и заполнение

{

int i, e, r;

e = 0;

r = (0 + rand() % 3);

while (A[e][N - 1] != 1)

{

++e;

}

if (e < M - 2 && r == 0){ ++e; }

if (e > 2 && r == 1){ --e; }

int \*\*temp = new int\*[M];

for (int i = 0; i < M; i++){

temp[i] = new int[N];//Выделяю память для временного массива

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

temp[i][j] = A[i][j]; //Записываю туда данные

}

}

delete[](\*A); //вот так освобождаю массив, принятый функцией

for (int i = 0; i < M; i++){

A[i] = new int[N + 1]; //Выделяю для массива, принятой функцией новую память

}

for (int i = 0; i < M; i++){

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

A[i][j] = temp[i][j]; //Записываю туда данные

}

}

int \*\*temp2 = new int\*[M];

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

temp2[i] = new int[N]; //Выделяю новую память для временного массива

for (int i = 0; i < M; i++){

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

temp2[i][j] = B[i][j]; //Записываю туда данные

}

}

delete[](\*B); //вот так освобождаю массив, принятый функцией

for (int i = 0; i < M; i++){

B[i] = new int[N + 1]; //Выделяю для массива, принятой функцией новую память

}

for (int i = 0; i < M; i++){

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

B[i][j] = temp2[i][j]; //Записываю туда данные

}

}

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

{

if (e <= i){ A[i][N] = 1; }

else{ A[i][N] = 0; }

B[i][N] = 0;

}

delete[]temp; //освобождаю память от временного массива

delete[]temp2; //освобождаю память от временного массива

MTree(N); LTree(N - 1); Coins(N);

++N;

return N;

}

void main()

{

int i, j;

string a, b;

b = \*"";

setlocale(LC\_ALL, "Russian");

int N = 0, M = 0; //Число колонок=N, Число строк= M

/\*Чтение из файла текста\*/

ifstream in("C://Users//Maskit//Desktop//php.txt");// получение файла с картой

int t;

t = 0;

while (!in.eof())

{

getline(in, a); // Считываем строчку в переменную а

b = b + ' ' + a;

if (N < a.size()){ N = a.size(); }

for (i = 0; a[i] != \*""; i++){ if (i > a.size()){ break; } }

++M;

}

in.close(); //Закрываем файл

/\*Создание динамического двумерного массива\*/

//A = new int\*[M];

A = (int\*\*)malloc(sizeof(int\*) \* M);

B = (int\*\*)malloc(sizeof(int\*) \* M);

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

{

// A[i] = new int[N];

A[i] = (int\*)malloc(sizeof(int) \* N);

B[i] = (int\*)malloc(sizeof(int) \* N);

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

{

a = b.substr(t + 2, 1);

++t;

A[i][j] = atoi(a.c\_str());

B[i][j] = 0;

if (a == " "){

while (j < N)

{

A[i][j] = 0;

B[i][j] = 0;

j++;

}

}

if (t == b.size() - 2){ break; }

}

if (t == b.size() - 2){ break; }// зарисовка предметов

}

cout << "\n\n";

for (j = 1; j < N - 1; ++j){ LTree(j); MTree(j); Coins(j); }

i = M - 2;

j = 2;

while (A[i][j] == 1)

{

i = i - 1;

}

B[i][j] = 7;// создание персонажа

char vvod;

int K = N, c = 0, score = 0, v = 0;

do{

vvod = \_getche();

if (vvod == 'w' && i > 0 && A[i - 1][j] != 1){ if (B[i - 1][j] == 4){ ++score; }B[i][j] = 0; --i; B[i][j] = 7; mass(N, M); vvod = \_getche(); }

if (vvod == 'a' && j > 0 && A[i][j - 1] != 1){ if (B[i][j - 1] == 4){ ++score; }B[i][j] = 0; --j; B[i][j] = 7; if (N > 45){ --N; } mass(N, M); }

if (vvod == 'd' && j < N - 2 && A[i][j + 1] != 1){ if (j >= K - 15){/\*\*/ if (N < K){ ++N; } else { K = main1(K, M); ++N;/\*\*/ } } if (B[i][j + 1] == 4){ ++score; }B[i][j] = 0; ++j; B[i][j] = 7; mass(N, M); }

while (A[i + 1][j] != 1){ if (B[i + 1][j] == 4){ score++; }if (B[i + 1][j] == 4){ ++score; } B[i][j] = 0; ++i; B[i][j] = 7; mass(N, M); }

if (vvod == 'q'){// Меню

c = 1;

do

{

system("cls");

menu(c, v, score);

vvod = \_getche();

if (vvod == 'w'){ if (c > 1) { --c; } }

if (vvod == 'a'){ v = 0; }

if (vvod == 's'){ if (c < 5){ ++c; } }

if (vvod == 'd'){ v = 1; }

//if (c == 1){}

} while (vvod != 'q');

}

} while (vvod != 'p');

/\*Очистка памяти от созданного двумерного массива\*/

for (int i = 0; i < N; i++) { delete[]A[i]; delete[]B[i]; }

delete[]A; delete[]B;

A = NULL; B = NULL;

system("pause");

}

Приложение 2

Блок-схемы

void LTree(int j)

Да

Нет

A[i - 2][j] = 10; A[i - 2][j - 1] = 10;

A[i - 2][j + 1] = 10;

Да

i > 3

A[i - 3][j] = 10;

Нет

Да

i > 2

Нет

Нет

Да

++i;

A[i - 1][j] = 6;

i > 1

r < 10

Нет

Да

A[i][j] != 1;

int i = 1;

int r = 1 + rand() % 100;

void MTree(int j)

Да

A[i - 2][j] = 2;

Нет

Да

i > 3

A[i - 3][j] = 2;

Нет

Да

i > 2

Нет

Нет

Да

++i;

A[i - 1][j] = 6;

i > 1

r < 13

Нет

Да

A[i][j] != 1;

int i = 1;

int r = 1 + rand() % 100;

void Coins(int j)

Да

B[i - 2][j] = 4;

Нет

Да

Нет

i > 3

Да

r >= 6 && r<14

Нет

Нет

Да

++i;

B[i - 1][j] = 4;

i > 1

r < 6

Нет

Да

A[i][j] != 1;

int i = 1;

int r = 1 + rand() % 100;

void mass(int N, int M)

Нет

Нет

SetColor(15, 0);

A[i][j] < 10

cout << " ";

Да

SetColor(0, B[i][j]);

cout << B[i][j];

Нет

Нет

Да

cout << " ";

A[i][j] < 10

Да

SetColor(0, A[i][j]);

cout << A[i][j];

B[i][j] == 0

Нет

SetColor(0,2); cout<<"2 "; SetColor(15,0);

Да

i<1||j<N-45||i>M-2||j>N-2

j=N-46; j<N; ++j

Да

Да

cout << "\n\t";

Нет

i = 0;i<M-1; ++i

system("cls");

Нет

Нет

Да

temp[i][j] = A[i][j];

int j=0;j<N;++j

Нет

temp[i] = new int[N];

int i=0;i<M;++i

Да

int \*\*temp = new int\*[M];

--e;

Да

e > 2 && r == 1

Нет

Нет

Да

++e;

e < M-2 && r==0

Да

++e;

Нет

Да

A[e][N - 1] != 1

e = 0;

r = (0 + rand() % 3);

int main1(int N, int M)

Да

delete[](\*B);

Да

int i=0;i<M;++i

Нет

temp2[i][j] = B[i][j];

B[i] = new int[N + 1];

int j=0;j<N;++j

temp2[i] = new int[N];

temp2[i][j] = B[i][j];

Нет

Нет

int i=0;i<M;++i

Да

int \*\*temp2 = new int\*[M];

A[i] = new int[N + 1];

A[i][j] = temp[i][j];

Нет

int j=0;j<N;++j

Да

Нет

int i=0;i<M;++i

Да

delete[](\*A);

Coins(N);

LTree(N - 1);

MTree(N);

delete[]temp;

delete[]temp2; ++N;

B[i][N] = 0;

A[i][N] = 1;

A[i][N] = 0;

Нет

Да

e <= i

Да

Нет

i=0; i<M; ++i

Да

Нет

B[i][j] = temp2[i][j];

int j=0;j<N;++j

void main()

break;

t==b.size()-2

Нет

Да

A[i][j] = 0;

B[i][j] = 0; j++;

j < N

Да

Нет

Нет

a == " "

a = b.substr(t + 2, 1); ++t;

A[i][j] = atoi(a.c\_str());

B[i][j] = 0;

Нет

Нет

Да

j=0; j<=N; j++

A[i] = (int\*)malloc(sizeof(int) \* N);

B[i] = (int\*)malloc(sizeof(int) \* N);

Да

i=0; i<=M; i++

A = (int\*\*)malloc(sizeof(int\*) \* M);

B = (int\*\*)malloc(sizeof(int\*) \* M);

b = \*""; setlocale(LC\_ALL, "Russian");

ifstream in("C://Users//Maskit// Desktop//php.txt");

Нет

Да

Нет

Да

vvod=='w' && i>0 && A[i-1][j]!= 1

vvod != 'q'

Нет

Да

B[i][j] = 7; char vvod;

int K = N, c=0, score=0, v=0;

i = i - 1;

A[i][j] == 1

Нет

Да

i = M - 2; j = 2;

Coins(j);

MTree(j);

LTree(j);

j=1;j<N-1;++j

Нет

Да

t==b.size()-2

Нет

Да

break;

Да

Нет

vvod =='d' && j<N-2 && A[i][j + 1] != 1

mass(N, M);

--N;

N > 45

Нет

Да

Да

Нет

B[i][j-1] == 4

++score;

B[i][j] = 0; --i; B[i][j] = 7;

Да

Нет

vvod == 'a' && j>0 && A[i][j - 1] != 1

vvod = \_getche();

mass(N, M);

B[i][j] = 0; --i; B[i][j] = 7;

++score;

B[i-1][j] == 4

Нет

Да

Да

Нет

vvod =='d' && j<N-2 && A[i][j + 1] != 1

mass(N, M);

--N;

N > 45

Нет

Да

Да

Нет

B[i][j-1] == 4

++score;

B[i][j] = 0; --i; B[i][j] = 7;

Да

Нет

vvod == 'a' && j>0 && A[i][j - 1] != 1

vvod = \_getche();

mass(N, M);

B[i][j] = 0; --i; B[i][j] = 7;

++score;

B[i-1][j] == 4

Нет

Да

delete[]A; delete[]B;

A = NULL; B = NULL;

mass(N, M);

B[i][j] = 0; ++i; B[i][j] = 7;

++score;

Да

Нет

B[i+1][j] == 4

Да

Нет

A[i+1][j] != 1

mass(N, M);

B[i][j] = 0; ++j; B[i][j] = 7;

++score;

B[i][j+1]== 4

Нет

Да

++N;

K = main1(K, M); ++N;

Нет

Да

N < K

j >= K - 15

Нет

Да

Размещено на Allbest.ru