

**МИНОБР НАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет**

**«СТАНКИН»**

**(ФГБОУ ВО МГТУ «СТАНКИН»)**

Кафедра управления и информатики в технических системах

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

по дисциплине «Управление и информатика в технических системах»

на тему: «Игра: жизнь по своим правилам»

Подготовил: студент группы ИДБ-18-11

Исии Майя

Проверил:

Звонарев Антон Алексеевич

Москва – 2019

**Оглавление**

[**1.** **Введение** 4](#_Toc9330313)

[**2.** **Игра «Жизнь» по своим правилам** 5](#_Toc9330314)

[**Литература** 12](#_Toc9330315)

# **Введение**

*Клеточный автомат* — дискретная модель, изучаемая в математике, теории вычислимости, физике, теоретической биологии и микромеханике. Включает регулярную решётку ячеек, каждая из которых может находиться в одном из конечного множества состояний, таких как 1 и 0. Для каждой ячейки определено множество ячеек, называемых окрестностью. Для работы клеточного автомата требуется задание начального состояния всех ячеек и правил перехода ячеек из одного состояния в другое. На каждой итерации, используя правила перехода и состояния соседних ячеек, определяется новое состояние каждой ячейки. Обычно правила перехода одинаковы для всех ячеек и применяются сразу ко всей решётке.

*Игра «Жизнь» (англ. Conway's Game of Life)* — клеточный автомат, придуманный английским математиком Джоном Конвеем в 1970 году.

Место действия этой игры — «вселенная» — это размеченная на клетки поверхность или плоскость — безграничная, ограниченная, или замкнутая (в пределе — бесконечная плоскость).

Каждая клетка на этой поверхности может находиться в двух состояниях: быть «живой» (заполненной) или быть «мёртвой» (пустой). Клетка имеет восемь соседей, окружающих её.

Распределение живых клеток в начале игры называется первым поколением. Каждое следующее поколение рассчитывается на основе предыдущего по таким правилам:

в пустой (мёртвой) клетке, рядом с которой ровно три живые клетки, зарождается жизнь;

если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить; в противном случае, если соседей меньше двух или больше трёх, клетка умирает («от одиночества» или «от перенаселённости»)

Игра прекращается, если на поле не останется ни одной «живой» клетки

конфигурация на очередном шаге в точности (без сдвигов и поворотов) повторит себя же на одном из более ранних шагов (складывается периодическая конфигурация)

при очередном шаге ни одна из клеток не меняет своего состояния (складывается стабильная конфигурация; предыдущее правило, вырожденное до одного шага назад)

Эти простые правила приводят к огромному разнообразию форм, которые могут возникнуть в игре.

Игрок не принимает прямого участия в игре, а лишь расставляет или генерирует начальную конфигурацию «живых» клеток, которые затем взаимодействуют согласно правилам уже без его участия (он является наблюдателем).Игра «Жизнь» (англ. Conway's Game of Life) — клеточный автомат, придуманный английским математиком Джоном Конвеем в 1970 году.

# **Игра «Жизнь» по своим правилам**

Правила:

Правила рождения и смерти клеток аналогичны с правилами классической «Жизни». Задаются координаты живых клеток и так называемые черные дыры, попадая в которую клетка умирает, зарождая в то же время вокруг себя еще четыре живые клетки: сверху, справа, снизу и слева. Так игра продолжается: клетки начинают жить и умирать уже взаимодействуя с окружающими их клетками.

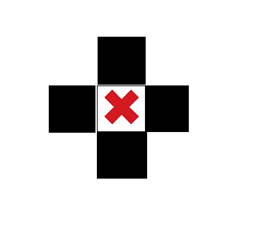
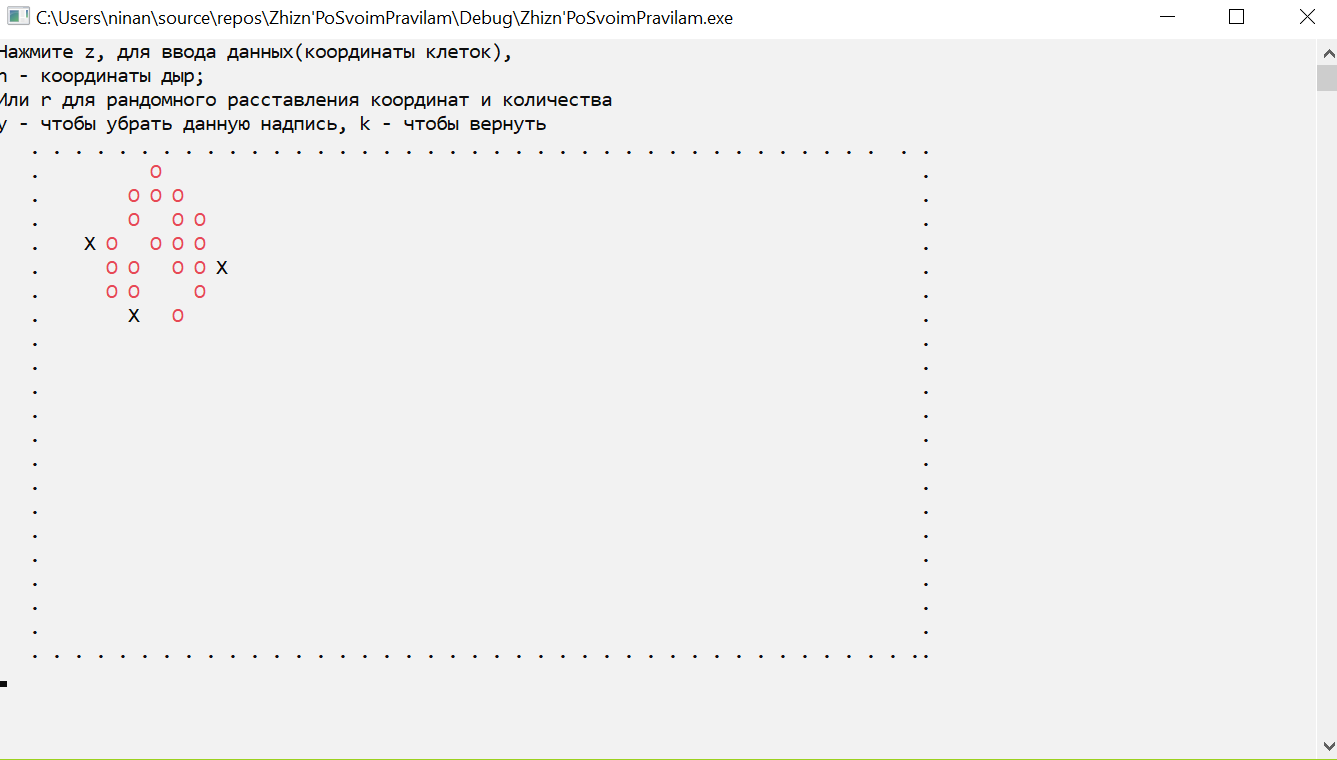
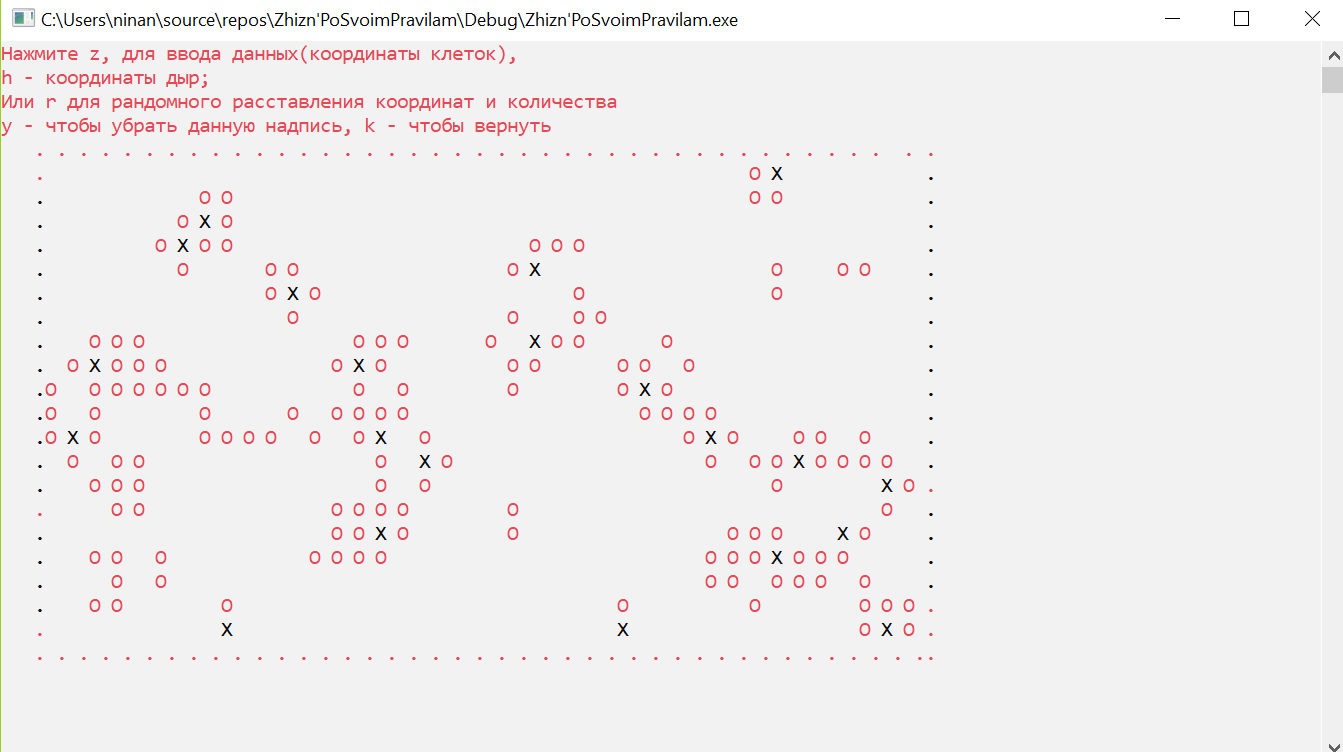


Рис.4 Схема зарождения клеток при попадании в «черную дыру», отмеченную крестиком.

 Рис.5 Вид сетки при введении координат живых клеток и черных дыр пользователем.

Рис. 6 Вид сетки при рандомном расставлении клеток при задании их количества в 100 штук.

Проект состоит из 3 файлов: Source.cpp, который определяет как будет выглядеть консоль с игрой (цвет экранна — белый (system("color F0")); Header.h, задает классы, используемые в файле Univerce.cpp (всего их два: class Map и class Universe).

В Univerce.cpp описано то, как будет выглядеть сетка, живые и мертвые клетки, их расположение относительно друг друга, а также относительно черных дыр, определяется размер, для определения состояния ячеек вводится объект класса Map, где устанавливаются правила и принципы клеток.

Текст программы

Universe.cpp

#include "Header.h"

Map::Map() {}

Map::Map(int s1, int s2)

{

size1 = s1;

size2 = s2;

arr = new char\*[size1];

for (int i = 0; i < size1; i++)

arr[i] = new char[size2];

zeroer();

}

void Map::zeroer()

{

for (int i = 0; i < size1; i++)

for (int j = 0; j < size2; j++)

arr[i][j] = ' ';

}

void Map::assign(Map object)

{

for (int i = 0; i < size1; i++)

for (int j = 0; j < size2; j++)

arr[i][j] = object.arr[i][j];

}

void Map::newshow()

{

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

cout << "\n .";

for (int k = 0; k < size2 - 1; k++)

{

if (k == size2 - 2)

cout << " .";

cout << " .";

}

cout << "\n";

for (int i = 0; i < size1; i++)

{

cout << " .";

for (int j = 0; j < size2; j++)

{

if (arr[i][j] == 'O')

{

SetConsoleTextAttribute(hConsole, (WORD)((15 << 4) | 12)); //4 - red, 12 - lightred,

cout << arr[i][j] << " ";

}

else

{

SetConsoleTextAttribute(hConsole, (WORD)((15 << 4) | 0));

cout << arr[i][j] << " ";

}

}

cout << ".\n";

}

cout << " .";

for (int k = 0; k < size2; k++)

cout << " .";

cout << ".\n";

}

void Map::getsize(int& s1, int& s2)

{

s1 = size1;

s2 = size2;

}

void Map::find(char sym, int& number1, int& number2, int& status)

{

status = 0;

for (int i = 0; i < size1; i++)

{

for (int j = 0; j < size2; j++)

{

if (arr[i][j] == sym)

{

number1 = i;

number2 = j;

status++;

}}}}

Universe::Universe() { symbol = 'O'; }

void Universe::add(Map object, int status)

{

int x = 0, y = 0;

if (status == 1 || status == 2)

{

cout << " Пожалуйста, введите координаты: \n x: ";

cin >> y;

y = y - 1;

cout << " y: ";

cin >> x;

x = x - 1;

}

if (status == 1)

object.arr[x][y] = symbol;

if (status == 2)

{

object.arr[x][y] = 'X';

if (check(object, x - 1, y, 1) == true)

if (object.arr[x - 1][y] == ' ')

object.arr[x - 1][y] = 'O';

if (check(object, x + 1, y, 1) == true)

if (object.arr[x + 1][y] == ' ')

object.arr[x + 1][y] = 'O';

if (check(object, x, y - 1, 1) == true)

if (object.arr[x][y - 1] == ' ')

object.arr[x][y - 1] = 'O';

if (check(object, x, y + 1, 1) == true)

if (object.arr[x][y + 1] == ' ')

object.arr[x][y + 1] = 'O';

}

if (status == 3)

{

int n = 0;

cout << "Введите количество рандомных клеток: ";

cin >> n;

int a = 20, b = 40;

srand(time(NULL));

int x = 0, y = 0;

for (int i = 0; i < 25; i++)

{

x = rand() % a;

y = rand() % b;

if (object.arr[x][y] == ' ')

{

object.arr[x][y] = 'X';

if (check(object, x - 1, y, 1) == true)

if (object.arr[x - 1][y] == ' ')

object.arr[x - 1][y] = 'O';

if (check(object, x + 1, y, 1) == true)

if (object.arr[x + 1][y] == ' ')

object.arr[x + 1][y] = 'O';

if (check(object, x, y - 1, 1) == true)

if (object.arr[x][y - 1] == ' ')

object.arr[x][y - 1] = 'O';

if (check(object, x, y + 1, 1) == true)

if (object.arr[x][y + 1] == ' ')

object.arr[x][y + 1] = 'O';

}}

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

{

x = rand() % a;

y = rand() % b;

if (object.arr[x][y] == ' ')

object.arr[x][y] = 'O';

}}}

void Universe::buttons(Map object, char sym)

{

if (sym == 'z')

add(object, 1);

if (sym == 'h')

add(object, 2);

if (sym == 'r')

add(object, 3);

}

void Universe::NewBirthAndDeath(Map object)

{

int s1 = 0, s2 = 0;

object.getsize(s1, s2);

Map ChangeObject(s1, s2);

ChangeObject.zeroer();

ChangeObject.assign(object);

int counter = 0;

for (int i = 0; i < s1; i++)

for (int j = 0; j < s2; j++)

if (ChangeObject.arr[i][j] == ' ' || ChangeObject.arr[i][j] == 'O')

{

if (check(ChangeObject, i, j + 1, 1) == true)

if (ChangeObject.arr[i][j + 1] == symbol)

++counter;

if (check(ChangeObject, i, j - 1, 1) == true)

if (ChangeObject.arr[i][j - 1] == symbol)

++counter;

if (check(ChangeObject, i - 1, j, 1) == true)

if (ChangeObject.arr[i - 1][j] == symbol)

++counter;

if (check(ChangeObject, i + 1, j, 1) == true)

if (ChangeObject.arr[i + 1][j] == symbol)

++counter;

if (check(ChangeObject, i - 1, j - 1, 1) == true)

if (ChangeObject.arr[i - 1][j - 1] == symbol)

++counter;

if (check(ChangeObject, i + 1, j + 1, 1) == true)

if (ChangeObject.arr[i + 1][j + 1] == symbol)

++counter;

if (check(ChangeObject, i - 1, j + 1, 1) == true)

if (ChangeObject.arr[i - 1][j + 1] == symbol)

++counter;

if (check(ChangeObject, i + 1, j - 1, 1) == true)

if (ChangeObject.arr[i + 1][j - 1] == symbol)

++counter;

if (counter == 3 && object.arr[i][j] == ' ')

object.arr[i][j] = symbol;

if ((counter < 2 || counter > 3) && object.arr[i][j] == symbol)

object.arr[i][j] = ' ';

counter = 0;

}}

bool Universe::check(Map object, int x, int y, int status)

{

int s1 = 0, s2 = 0;

object.getsize(s1, s2);

if ((x < 0 || y < 0 || x >= s1 || y >= s2) && status == 1)

return false;

else return true;

}

void Universe::test()

{

setlocale(LC\_ALL, "Rus");

int a = 20, b = 40;

Map A(a, b);

char button = ' ';

int letter = 0;

while (1)

{

if (letter == 0)

cout << "Нажмите z для ввода данных (координаты клеток),\n h - координаты дыр;\nИли r для рандомного расставления координат и количества\n y - чтобы убрать данную надпись, k - чтобы вернуть";

A.newshow();

NewBirthAndDeath(A);

button = (char)\_getch();

buttons(A, button);

if (button == 'y')

letter = 1;

if (button == 'k')

letter = 0;

system("cls");

}}

# **Литература**

1. Электронный ресурс: <https://gcup.ru/forum/62-40908-1>
2. Электронный ресурс: <https://ravesli.com/urok-113-klassy-obekty-i-metody-klassov/>