*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана***  ***(национальный исследовательский университет)»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

**Отчет**

**по домашней работе**

**Дисциплина:** Прикладная теория цифровых автоматов

**Название домашней работы:**

Игра с конём

Вариант 20

Студент гр. ИУ6-42  **\_\_\_\_\_\_15.05.2018 Бурлаков А.С.**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_ Губарь А.М**

(Подпись, дата) (И.О. Фамилия)

Москва, 2018

**ЗАДАНИЕ**

Переместить шахматного коня с поля a1 на поле а8 так, чтобы он побывал на каждой клетке шахматной доски по одному разу.

**ОСНОВНАЯ ЧАСТЬ**

**1 Спецификация автомата**

**1.1 Состояния автомата**

S0 – Конем можно ходить, игра не закончена

S1 – Игра закончена

**1.2 Входные сигналы**

«a» – сделан ход куда угодно кроме a8

«b» - сделан ход в а8, обойдя все клетки доски

«с» - нет ходов

«d» - начать заново

**1.3 Выходные сигналы**

**«**Ход»- сделан ход

«Победа» - Игрок победил

«Поражение» - Игрок проиграл

«Заново» - Игра начинается заново

**2 Полученный конечный автомат**

Составим таблицу, описывающую конечный автомат на основе полученной спецификации и текста задания

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Состояние** |  | | | |  | | | |
| **a** | **b** | **c** | **d** | **a** | **b** | **c** | **d** |
| **S0** | S0 | S1 | S1 | - | Ход | Победа | Поражение | - |
| **S1** | - | - | - | S0 | - | - | - | Заново |

По полученной таблице составим схему конечного автомата.

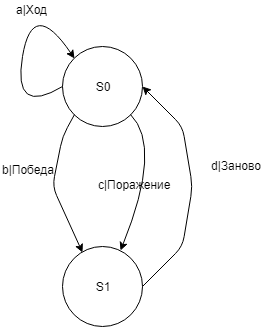


Рисунок 1 – Схема конечного автомата

Переход из S0 в S1 производится при проверке компьютером игрового поля.

**3 Реализация конечного автомата**

**3.1 Текст программы реализации конечного автомата**

**uses**

GraphAbc;

**const**

dx: **array**[0..7] **of** integer = (1, 2, 2, 1, -1, -2, -2, -1);

dy: **array**[0..7] **of** integer = (2, 1, -1, -2, -2, -1, 1, 2);

**var**

Kx, Ky: integer;

count: integer = 0;

mas: **array**[0..7, 0..7] **of** integer;

win1: boolean = false;

lose1: boolean = false;

**procedure** DrawX(x, y: integer);

**begin**

Pen.Color := clRed;

Pen.Width := 3;

Line(x \* 50, y \* 50 + 50, x \* 50 + 50, y \* 50);

Line(x \* 50, y \* 50, x \* 50 + 50, y \* 50 + 50);

Pen.Color := clBlack;

Pen.Width := 1;

**end**;

**procedure** Lighting();

**begin**

Pen.Color := rgb(0, 255, 0);

Pen.Width := 4;

**for** i: byte := 0 **to** 7 **do**

**begin**

**if** (Kx + dx[i] < 8) **and** (Kx + dx[i] >= 0) **and**

(Ky + dy[i] < 8) **and** (Ky + dy[i] >= 0) **then**

**if** mas[Kx + dx[i], Ky + dy[i]] = 0 **then**

DrawEllipse((Kx + dx[i]) \* 50 + 5, (Ky + dy[i]) \* 50 + 5,

(Kx + dx[i] + 1) \* 50 - 5, (Ky + dy[i] + 1) \* 50 - 5);

**end**;

Pen.Color := clBlack;

Pen.Width := 1;

**end**;

**procedure** Redraw();

**begin**

ClearWindow();

Pen.Color := clBlack;

**for** i: byte := 0 **to** 7 **do**

**for** j: byte := 0 **to** 7 **do**

**begin**

**if** ((i + j) **mod** 2 = 1) **then**

Brush.Color := clGray

**else**

Brush.Color := clWhite;

FillRect(i \* 50, j \* 50, (i + 1) \* 50, (j + 1) \* 50);

**end**;

Font.Size := 16;

**for** i: byte := 0 **to** 7 **do**

**for** j: byte := 0 **to** 7 **do**

**begin**

**if** mas[i, j] > 0 **then begin**

**if** ((i + j) **mod** 2 = 1) **then**

Brush.Color := clGray

**else**

Brush.Color := clWhite;

**if** mas[i, j] < 10 **then**

textout(i \* 50 + 19, j \* 50 + 14, mas[i, j])

**else**

textout(i \* 50 + 14, j \* 50 + 14, mas[i, j]);

**end**;

**end**;

Font.Size := 12;

Pen.Color := clBlack;

Brush.Color := clWhite;

textout(450, 50, 'Количество: ');

textout(450, 100, count);

**for** i: byte := 0 **to** 7 **do**

textout(i \* 50 + 19, 405, chr(ord('a') + i));

**for** i: byte := 0 **to** 7 **do**

textout(405, i \* 50 + 14, 8 - i);

**end**;

**procedure** CheckWin();

**begin**

**if** (Kx = 0) **and** (Ky = 7) **and** (count = 63) **then begin**

Brush.Color := clwhite;

textout(100, 200, 'Ты выиграл, нажми, чтобы сыграть еще раз');

win1 := true;

**end**;

**end**;

**procedure** CheckLose();

**var**

c: byte;

**begin**

**if not** win1 **then begin**

c := 0;

**for** i: byte := 0 **to** 7 **do**

**if** (Kx + dx[i] < 8) **and** (Kx + dx[i] >= 0) **and**

(Ky + dy[i] < 8) **and** (Ky + dy[i] >= 0) **then**

**if** mas[Kx + dx[i], Ky + dy[i]] = 0 **then**

inc(c);

**if** c = 0 **then begin**

Brush.Color := clwhite;

textout(100, 200, 'Ты проиграл, нажми, чтобы сыграть еще раз');

Lose1 := true;

**end**;

**end**;

**end**;

**procedure** NewGame();

**begin**

Pen.Color := clBlack;

count := 0;

**for** i: byte := 0 **to** 7 **do**

**for** j: byte := 0 **to** 7 **do**

**begin**

mas[i, j] := 0;

**end**;

Kx := 0;

Ky := 7;

Lose1 := false;

Win1 := false;

Redraw();

DrawX(Kx, Ky);

Lighting();

**end**;

**procedure** MouseDown(x, y, mb: integer);

**var**

h: integer;

**begin**

**if** lose1 **or** win1 **then**

NewGame()

**else begin**

h := -1;

**for** i: byte := 0 **to** 7 **do**

**if** (x **div** 50 = Kx + dx[i]) **and** (y **div** 50 = Ky + dy[i]) **and**

(x < 400) **and** (y < 400) **then**

h := i;

**if** (h >= 0) **and** (mas[x **div** 50, y **div** 50] = 0) **then begin**

mas[Kx, Ky] := count + 1;

inc(count);

Redraw();

Kx := x **div** 50;

Ky := y **div** 50;

DrawX(Kx, Ky);

CheckWin();

CheckLose();

Lighting();

**end**;

**end**;

**end**;

**begin**

OnMouseDown := MouseDown;

SetWindowSize(600, 430);

NewGame();

**end**.

**3.2 Интерфейс программы**

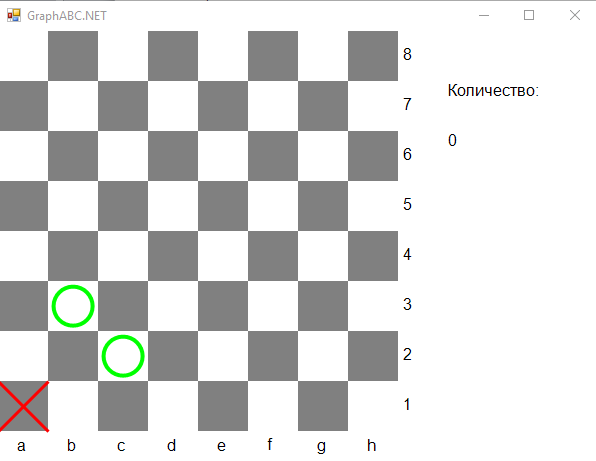


Рисунок 2 – Начало игры

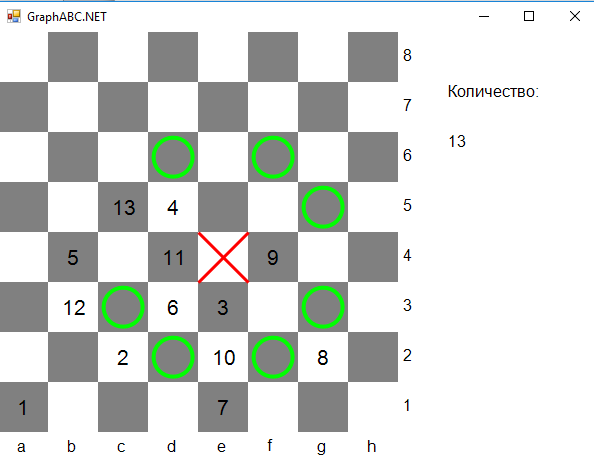


Рисунок 3 – Середина игры

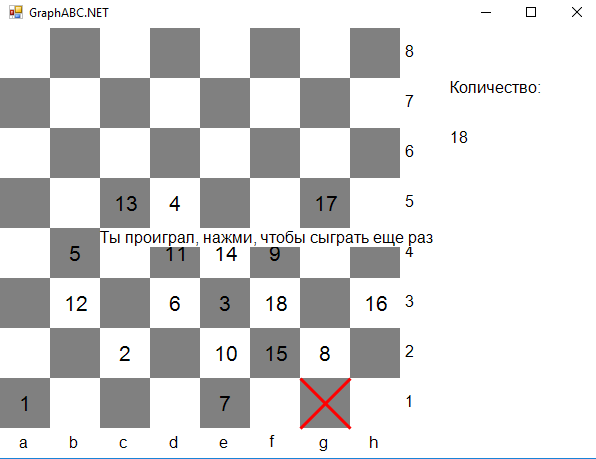


Рисунок 4 – Конец игры

**ЗАКЛЮЧЕНИЕ**

В ходе данного домашнего задания спроектирован и получен конечный автомат, реализующий игру с конём. Создана реализация данного автомата с графическим интерфейсом в среде разработки программного обеспечения PascalABC на языке Pascal.