

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**«Донской государственный технический университет»**

**(ДГТУ)**

Кафедра «Программное обеспечение вычислительной техники и

автоматизированных систем»

**ЛАБОРАТОРНАЯ РАБОТА № 2**

по дисциплине «Конструирование программного обеспечения»

тема: «Разработка программы Action»

Выполнила:

ст. гр. ВПР 42 Т.И. Мысливчик

Проверил:

профессор А. П. Коробейников

Ростов-на-Дону

2016

**Лист задания**

Имеется поле N х N, где N – параметр. N <= 20, параметр Т – такт времени. На поле расставляются два вида объектов: активные объекты (поисковая группа) и пассивные объекты, которых захватывают. Каждый активный элемент имеет свое количество игровых очков от -2 до 2 (изначально 0). В текущий такт времени происходят следующие события:

- перебираются все активные элементы;

- текущий активный элемент генерирует случайную пару координат – это клетка, в которой в следующий такт времени приземлится данный активный элемент (точка конца прыжка);

- если в 8-ми соседних клетках вокруг точки конца прыжка и в самой точке конца прыжка нет пассивных элементов (промах), то текущему активному элементу присваивается -1 очко.

- если в 8-ми соседних клетках вокруг точки конца прыжка и в самой точке конца прыжка находятся пассивные элементы, то данный активный элемент получает +1 очко, при этом к полученному очку алгебраически суммируются предыдущие, все пассивные в данных 9-ти клетках исчезают (в следующем такте времени их уже нет), а текущий активный элемент переходит в точку конца прыжка (появляется в ней) в следующий такт времени;

- если точка конца прыжка совпадает с местом расположения другого активного элемента, то прыжок повторяется;

- если у активного элемента -2 игровых очка, то в следующем такте времени он исчезает и не участвует в дальнейшем рассмотрении;

- если у активного элемента +2 игровых очка, то он генерирует в случайное место на поле (порождает) еще одного нового активного элемента, при этом его игровые очки обнуляются;

- если у любой пустой клетки есть 3 активных соседних элемента, то в ней рождается новый активный элемент.

Такт заканчивается после того, как все активные элементы сделали свой ход.

**Текст модуля на языке Pascal**

**program** laba3;

**uses** GraphABC;

**const**

step = 30; //ширина клетки игрового поля

width = 800; //ширина игрового поля

height = 600; //высота игрового поля

**var**

n: integer;//размер игрового поля

active\_input: integer;//количество активов уже внесенных на поле

passive\_input: integer;//количество пассивов уже внесенных на поле

active: integer;//количество активных объектов

passive: integer;//количество пассивных объектов

takt\_input:integer;

takt: integer; //количество тактов

game\_field: **array** [1..20, 1..20] **of** integer;//игровое поле

prev\_field: **array** [1..20, 1..20] **of** integer;//копия поля, для прорисовки предыдущего шага

//копирование игрового поля для прорисовки предыдущего шага

**procedure** CopyOfArray();

**begin**

**for var** i := 1 **to** n **do**

**for var** j := 1 **to** n **do**

prev\_field[i, j] := game\_field[i, j];

**end**;

//прорисовка игрового поля

**procedure** Field();

**var**

x, y: integer;

x1, y1: integer;

**begin**

x := 10;

y := 10;

x1 := 30 + step \* n;

y1 := 10;

SetBrushColor(clBlack);

SetPenColor(clBlack);

**for var** i := 1 **to** n + 1 **do**

**begin**

Line(x, y, x + step \* (n), y);

y += step;

Line(x1, y1, x1 + step \* (n), y1);

y1 += step;

**end**;

y := 10;

y1 := 10;

**for var** i := 1 **to** n + 1 **do**

**begin**

Line(x, y, x, y + step \* (n));

x += step;

Line(x1, y1, x1, y1 + step \* (n));

x1 += step;

**end**;

SetFontSize(12);

SetBrushColor(Color.White);

TextOut(10, 70 + step \* n, 'Мафия');

TextOut(150, 70 + step \* n, active.ToString);

SetFontSize(12);

TextOut(10, 100 + step \* n, 'Мирные жители');

TextOut(150, 100 + step \* n, passive.ToString);

SetFontSize(12);

TextOut(10, 130 + step \* n, 'Осталось тактов');

TextOut(150, 130 + step \* n, takt.ToString);

SetFontSize(12);

TextOut(100+ step \* n, 20 + step \* n, 'Такт');

**if**(takt\_input-takt-1)>=0 **then**

TextOut(150+ step \* n, 20 + step \* n, (takt\_input-takt-1).ToString);

SetFontSize(12);

TextOut(100, 20 + step \* n, 'Такт');

TextOut(150, 20 + step \* n, (takt\_input-takt).ToString);

**end**;

//прорисовка игровых элементов на поле

**procedure** Draw(flag: boolean);

**begin**

ClearWindow();

Field();

LockDrawing;

**if** (flag) **then**

**begin**

**for var** i := 1 **to** n **do**

**for var** j := 1 **to** n **do**

**begin**

**if** (prev\_field[i, j] = 7) **then**

**begin**

SetPenColor(clWhite);

SetBrushColor(clWhite);

rectangle(step \* i - 18, step \* j - 18, step \* i + 9, step \* j + 9);

SetPenColor(clBlack);

SetBrushColor(clBlack);

**continue**;

**end**;

**if** (prev\_field[i, j] = 5) **then**

**begin**

SetPenColor(clLime);

SetBrushColor(clLime);

rectangle(step \* i + 2 + step \* n, step \* j - 18, step \* i + 29 + step \* n, step \* j + 9);

SetPenColor(clBlack);

SetBrushColor(clBlack);

**continue**;

**end**;

SetPenColor(clCyan);

SetBrushColor(clCyan);

rectangle(step \* i + 2 + step \* n, step \* j - 18, step \* i + 29 + step \* n, step \* j + 9);

SetPenColor(clBlack);

SetBrushColor(clCyan);

SetFontSize(8);

**if** ((prev\_field[i, j] = -2) **or** (prev\_field[i, j] = -1) **or** (prev\_field[i, j] = 0) **or** (prev\_field[i, j] = 1) **or** (prev\_field[i, j] = 2)) **then**

TextOut(30 + step \* n + round(step / 2 + (i - 1) \* step), 10 + round(step / 2 + (j - 1) \* step), prev\_field[i, j].ToString());

**end**;

**end**;

**for var** i := 1 **to** n **do**

**for var** j := 1 **to** n **do**

**begin**

**if** (game\_field[i, j] = 7) **then**

**begin**

SetPenColor(clWhite);

SetBrushColor(clWhite);

rectangle(step \* i - 18, step \* j - 18, step \* i + 9, step \* j + 9);

SetPenColor(clBlack);

SetBrushColor(clBlack);

**continue**;

**end**;

**if** (game\_field[i, j] = 5) **then**

**begin**

SetPenColor(clLime);

SetBrushColor(clLime);

rectangle(step \* i - 18, step \* j - 18, step \* i + 9, step \* j + 9);

SetPenColor(clBlack);

SetBrushColor(clBlack);

**continue**;

**end**;

SetPenColor(clCyan);

SetBrushColor(clCyan);

rectangle(step \* i - 18, step \* j - 18, step \* i + 9, step \* j + 9);

SetPenColor(clBlack);

SetBrushColor(clCyan);

SetFontSize(8);

**if** ((game\_field[i, j] = -2) **or** (game\_field[i, j] = -1) **or** (game\_field[i, j] = 0) **or** (game\_field[i, j] = 1) **or** (game\_field[i, j] = 2)) **then**

TextOut(10 + round(step / 2 + (i - 1) \* step), 10 + round(step / 2 + (j - 1) \* step), game\_field[i, j].ToString());

**end**;

SetPenColor(clBlack);

SetBrushColor(clGray);

Rectangle(width - 298, height - 8, width - 448, height - 58);

SetPenColor(clBlack);

SetBrushColor(clWhite);

Rectangle(width - 300, height - 10, width - 450, height - 60);

SetFontSize(20);

TextOut(width - 400, height - 50, 'ШАГ');

Redraw;

**end**;

//создание нового активного элемента

**procedure** CreateNewActive();

**var**

x, y: integer;

**begin**

**while**(true) **do**

**begin**

x := random(n) + 1;

y := random(n) + 1;

**if** ((game\_field[x, y] = 5) **or** (game\_field[x, y] = 7)) **then**

**begin**

game\_field[x, y] := 0;

inc(active);

**break**;

**end**;

**end**;

**end**;

//добавление нового активного на поле

**procedure** CreateNewActive(i, j: integer);

**begin**

game\_field[i, j] := -10;

inc(active);

**end**;

//проверка выхода за границу поле:

**function** Check(i, j, key: integer): boolean;

**var**

res: boolean;

**begin**

res := false;

**case** key **of**

1:

**if** (i < n) **then**

res := true;

2:

**if** (i > 1) **then**

res := true;

3:

**if** (j < n) **then**

res := true;

4:

**if** (j > 1) **then**

res := true;

**end**;

result := res;

**end**;

//проверка свободно ли место для прыжка:

**procedure** CheckFree();

**var**

count: integer;

**begin**

count := 0;

**for var** i := 1 **to** n **do**

**begin**

**for var** j := 1 **to** n **do**

**begin**

count := 0;

**if** (game\_field[i, j] <> 7) **then continue**;

**if** (Check(i, j, 1)) **then**

**if** ((game\_field[i + 1, j] = -1) **or** (game\_field[i + 1, j] = 0) **or** (game\_field[i + 1, j] = 1)) **then**

inc(count);

**if** (Check(i, j, 2)) **then**

**if** ((game\_field[i - 1, j] = -1) **or** (game\_field[i - 1, j] = 0) **or** (game\_field[i - 1, j] = 1)) **then**

inc(count);

**if** (Check(i, j, 3)) **then**

**if** ((game\_field[i, j + 1] = -1) **or** (game\_field[i, j + 1] = 0) **or** (game\_field[i, j + 1] = -1)) **then**

**begin**

inc(count);

**if** (count = 3) **then**

**begin**

CreateNewActive(i, j);

**continue**;

**end**;

**end**;

**if** (Check(i, j, 4)) **then**

**if** ((game\_field[i, j - 1] = -1) **or** (game\_field[i, j - 1] = 0) **or** (game\_field[i, j - 1] = 1)) **then**

**begin**

inc(count);

**if** (count = 3) **then**

**begin**

CreateNewActive(i, j);

**continue**;

**end**;

**end**;

**if** ((Check(i, j, 3)) **and** (Check(i, j, 1))) **then**

**if** ((game\_field[i + 1, j + 1] = -1) **or** (game\_field[i + 1, j + 1] = 0) **or** (game\_field[i + 1, j + 1] = 1)) **then**

**begin**

inc(count);

**if** (count = 3) **then**

**begin**

CreateNewActive(i, j);

**continue**;

**end**;

**end**;

**if** ((Check(i, j, 4)) **and** (Check(i, j, 1))) **then**

**if** ((game\_field[i + 1, j - 1] = -1) **or** (game\_field[i + 1, j - 1] = 0) **or** (game\_field[i + 1, j - 1] = 1)) **then**

**begin**

inc(count);

**if** (count = 3) **then**

**begin**

CreateNewActive(i, j);

**continue**;

**end**;

**end**;

**if** ((Check(i, j, 3)) **and** (Check(i, j, 2))) **then**

**if** ((game\_field[i - 1, j + 1] = -1) **or** (game\_field[i - 1, j + 1] = 0) **or** (game\_field[i - 1, j + 1] = 1)) **then**

**begin**

inc(count);

**if** (count = 3) **then**

**begin**

CreateNewActive(i, j);

**continue**;

**end**;

**end**;

**if** ((Check(i, j, 4)) **and** (Check(i, j, 2))) **then**

**if** ((game\_field[i - 1, j - 1] = -1) **or** (game\_field[i - 1, j - 1] = 0) **or** (game\_field[i - 1, j - 1] = 0)) **then**

**begin**

inc(count);

**if** (count = 3) **then**

**begin**

CreateNewActive(i, j);

**continue**;

**end**;

**end**;

**end**;

**end**;

**end**;

//проверка свободно ли место для нового активного:

**function** ChengeStepActive(i, j: integer): boolean;

**var**

res: boolean;

**begin**

**if** (game\_field[i, j] = 5) **then**

**begin**

game\_field[i, j] := 7;

res := true;

dec(passive);

**end**;

**if** (Check(i, j, 1)) **then**

**if** (game\_field[i + 1, j] = 5) **then**

**begin**

game\_field[i + 1, j] := 7;

res := true;

dec(passive);

**end**;

**if** (Check(i, j, 2)) **then**

**if** (game\_field[i - 1, j] = 5) **then**

**begin**

game\_field[i - 1, j] := 7;

res := true;

dec(passive);

**end**;

**if** (Check(i, j, 3)) **then**

**if** (game\_field[i, j + 1] = 5) **then**

**begin**

game\_field[i, j + 1] := 7;

res := true;

dec(passive);

**end**;

**if** (Check(i, j, 4)) **then**

**if** (game\_field[i, j - 1] = 5) **then**

**begin**

game\_field[i, j - 1] := 7;

res := true;

dec(passive);

**end**;

**if** ((Check(i, j, 3)) **and** (Check(i, j, 1))) **then**

**if** (game\_field[i + 1, j + 1] = 5) **then**

**begin**

game\_field[i + 1, j + 1] := 7;

res := true;

dec(passive);

**end**;

**if** ((Check(i, j, 4)) **and** (Check(i, j, 1))) **then**

**if** (game\_field[i + 1, j - 1] = 5) **then**

**begin**

game\_field[i + 1, j - 1] := 7;

res := true;

dec(passive);

**end**;

**if** ((Check(i, j, 2)) **and** (Check(i, j, 3))) **then**

**if** (game\_field[i - 1, j + 1] = 5) **then**

**begin**

game\_field[i - 1, j + 1] := 7;

res := true;

dec(passive);

**end**;

**if** ((Check(i, j, 4)) **and** (Check(i, j, 2))) **then**

**if** (game\_field[i - 1, j - 1] = 5) **then**

**begin**

game\_field[i - 1, j - 1] := 7;

res := true;

dec(passive);

**end**;

result := res;

**end**;

//такт

**procedure** p\_takt();

**var**

x, y: integer;

**begin**

CheckFree;

**for var** i := 1 **to** n **do**

**for var** j := 1 **to** n **do**

**if** (game\_field[i, j] = -10) **then** game\_field[i, j] += 10;

Draw(false);

CopyOfArray();

LockDrawing;

Draw(false);

Redraw;

**for var** i := 1 **to** n **do**

**for var** j := 1 **to** n **do**

**begin**

**if** ((game\_field[i, j] = -2) **or** (game\_field[i, j] = -1) **or** (game\_field[i, j] = 0) **or** (game\_field[i, j] = 1) **or** (game\_field[i, j] = 2)) **then**

**begin**

**while** (true) **do**

**begin**

x := random(n) + 1;

y := random(n) + 1;

**if** ((game\_field[x, y] = 5) **or** (game\_field[x, y] = 7)) **then**

**break**;

**end**;

**if** (ChengeStepActive(x, y)) **then**

game\_field[x, y] := game\_field[i, j] + 11

**else**

game\_field[x, y] := game\_field[i, j] - 11;

game\_field[i, j] := 7;

LockDrawing;

SetPenColor(Color.Green);

Redraw();

**end**;

**end**;

**for var** i := 1 **to** n **do**

**for var** j := 1 **to** n **do**

**begin**

**if** (game\_field[i, j] > 7) **then**

game\_field[i, j] -= 10;

**if** (game\_field[i, j] < -5) **then**

game\_field[i, j] += 10;

**end**;

Draw(true);

**for var** i := 1 **to** n **do**

**for var** j := 1 **to** n **do**

**begin**

**if** (game\_field[i, j] = -2) **then**

**begin**

LockDrawing;

SetPenColor(clGray);

SetBrushColor(clGray);

rectangle(step \* i - 18, step \* j - 18, step \* i + 9, step \* j + 9);

SetFontSize(8);

TextOut(10 + round(step / 2 + (i - 1) \* step), 10 + round(step / 2 + (j - 1) \* step), game\_field[i, j].ToString());

game\_field[i, j] := 7;

Redraw();

dec(active);

**end**;

**if** (game\_field[i, j] = 2) **then**

**begin**

LockDrawing;

SetPenColor(clLightCyan);

SetBrushColor(clLightCyan);

rectangle(step \* i - 18, step \* j - 18, step \* i + 9, step \* j + 9);

SetFontSize(8);

TextOut(10 + round(step / 2 + (i - 1) \* step), 10 + round(step / 2 + (j - 1) \* step), game\_field[i, j].ToString());

CreateNewActive();

game\_field[i, j] := 0;

Redraw;

**end**;

**end**;

**end**;

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

**var**

i, j: integer;

**begin**

**if** (x <width - 300) **and** (y < height - 10) **and** (x > width - 450) **and** (y > height - 60) **and** (takt<>0) **then**

**begin**

takt-=1;

p\_takt();

**end**;

**if** (x <width - 300) **and** (y < height - 10) **and** (x > width - 450) **and** (y > height - 60) **and**(takt=0) **then**

**begin**

SetPenColor(clWhite);

SetBrushColor(clWhite);

Rectangle(width - 298, height - 8, width - 448, height - 58);

SetPenColor(clWhite);

SetBrushColor(clWhite);

Rectangle(width - 300, height - 10, width - 450, height - 60);

SetFontSize(20);

TextOut(width - 400, height - 50, 'КОНЕЦ ИГРЫ');

Redraw;

**end**;

**if** (x < 10) **or** (y < 10) **or** (x > n \* step + 10) **or** (y > n \* step + 10) **then exit**;

LockDrawing;

i := (x - 10) **div** step + 1;

j := (y - 10) **div** step + 1;

**if** (active\_input < active) **then**

**begin**

game\_field[i, j] := 0;

inc(active\_input);

Draw(false);

Redraw();

**exit**;

**end**;

**if** (passive\_input < passive) **then**

**begin**

game\_field[i, j] := 5;

inc(passive\_input);

Draw(false);

Redraw();

**exit**;

**end**;

Redraw;

**end**;

//основная программа

**begin**

SetWindowCaption('Action Мысливчик Т.И.');

SetWindowPos(400, 50);

SetWindowWidth(width);

SetWindowHeight(height);

active\_input := 0;

passive\_input := 0;

write('Введите размер поля ');

readln(n);

writeln(n);

**for var** i := 1 **to** n **do**

**for var** j := 1 **to** n **do** game\_field[i, j] := 7;

CopyOfArray();

Write('Количество активных объектов ');

ReadLN(active);

writeln(active);

Write('Количество пассивных объектов ');

ReadLN(passive);

writeln(passive);

write('Количество тактов ');

readln(takt\_input);

writeln(takt\_input);

takt:=takt\_input;

ClearWindow();

Draw(false);

OnMouseDown := MouseDown;

**end**.

# 

# **Контрольный пример**

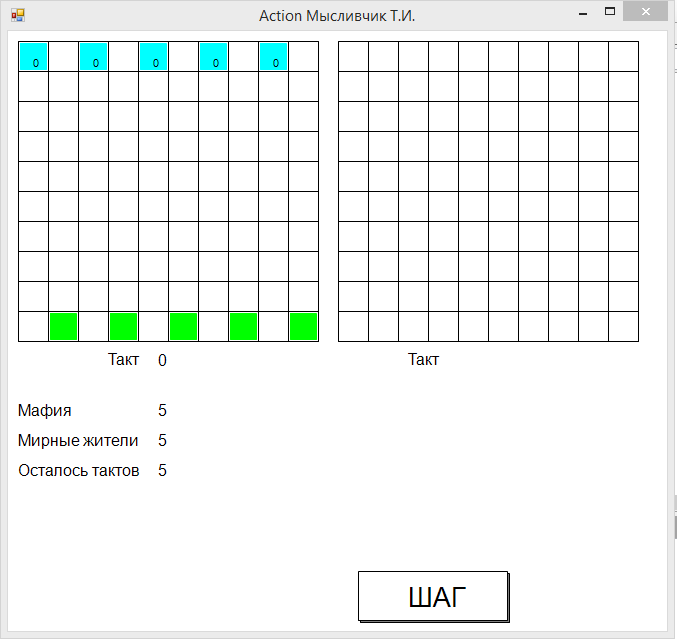


рис 1. Начальная расстановка

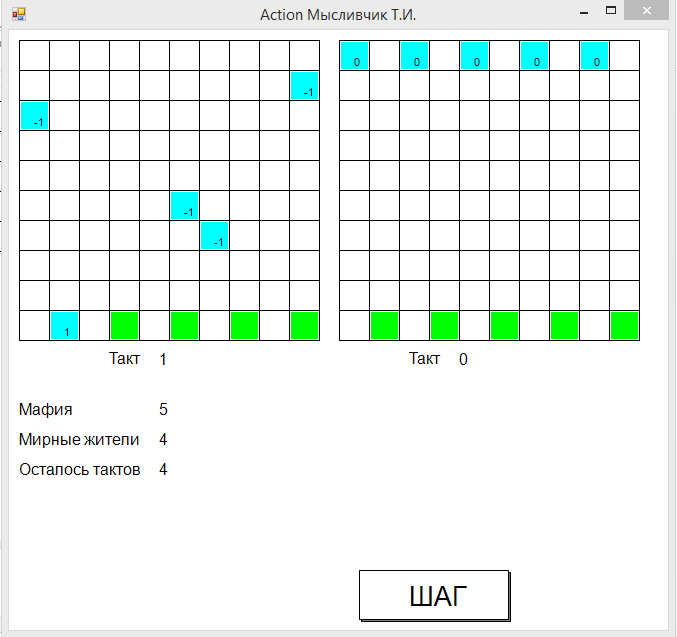


рис 2. Положение после первого такта

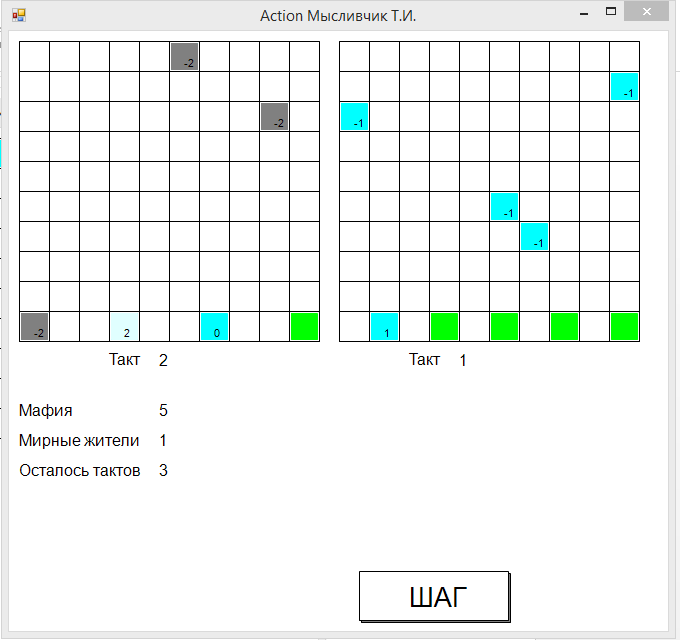


рис 3. Положение после второго такта

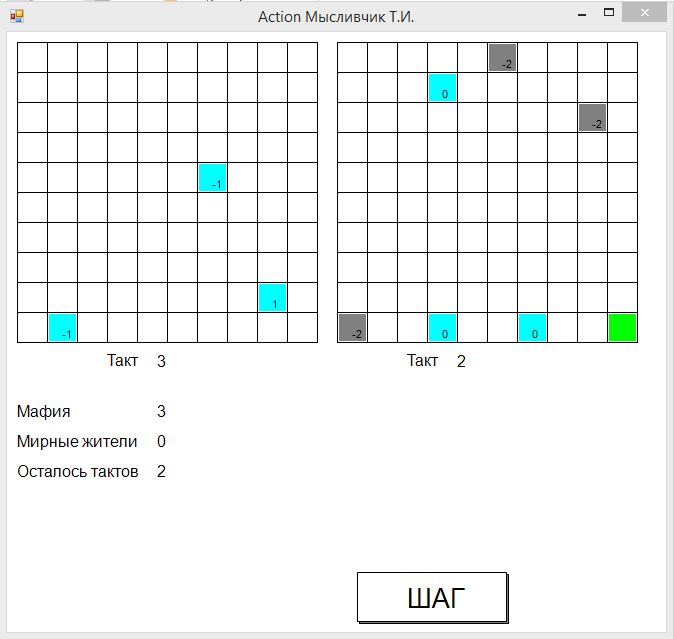


рис 4. Положение после третьего такта

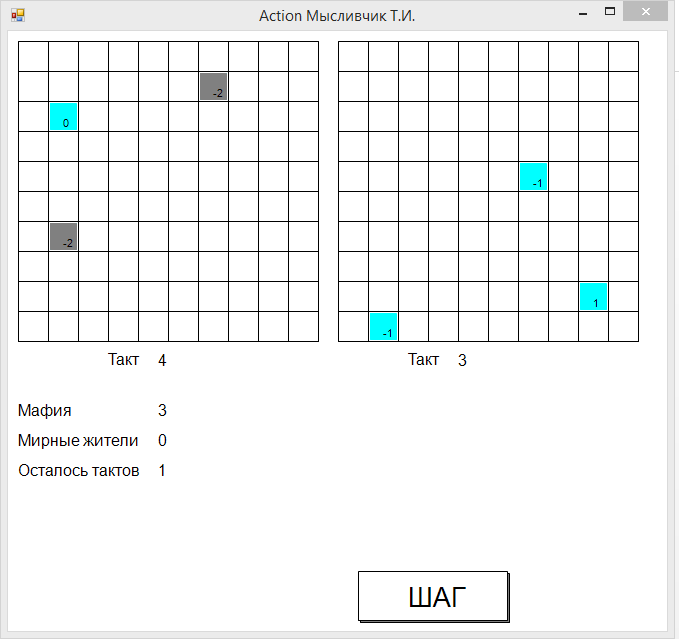


рис 5. Положение после четвертого такта

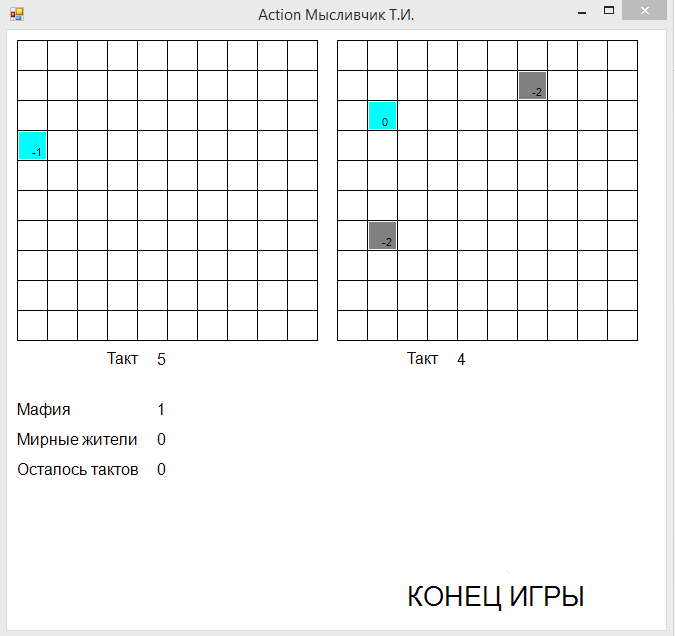


рис 6. Положение после пятого такта