# Механика игры

Программа разработана и написана на языке C#.

Игра в пятнашки - популярная головоломка, придуманная в 1878 году, которая представляет собой набор одинаковых квадратных костяшек с нанесёнными числами.

Игроку доступно поле размером 3х3 (4x4; 5х5) состоящее из 9(16; 25) клеток. Все клетки кроме одной (свободное место под одну костяшку) заняты костяшками с номерами от 1 до 8 (от 1 до 15; от 1 до 24), которые перемешаны между собой. Цель игры состоит в том, чтобы посредством последовательных передвижений, допускаемых наличием свободного поля упорядочить костяшки по порядку, желательно сделав как можно меньше перемещений. Концом игры считается, когда все костяшки от 1 до 8 (15;25) стоят друг за другом.  
Игра поможет вам [развить память](https://cepia.ru/memory/) и логическое мышление. У вас будет развиваться возможность просчитывать ходы вперед без ошибок.

## **Алгоритм игры:**

***Стадия 1: сборка верхней строки.***

Вы соберёте строку слева на право.

Найдите первую костяшку строки, которую вы хотите поместить в левый верхний угол.

Если это не последняя цифра строки, достаточно просто правильно её разместить, просто держите в уме следующие заметки:

* Никогда не трогайте части, собранные ранее.
* Чтобы сдвинуть цифру в определённом направлении, двигайте другие части по кругу, пока пропуск не окажется перед вашей цифрой на стороне, в которую вы хотите его сдвинуть. Далее вы можете сдвинуть цифру.

Если последняя часть строки не на месте, переместите её на позицию прямо под её правильным местом, с пробелом сразу под ней. Далее двигайте части в следующих направлениях:

Вниз, вниз, вправо, вверх, влево, вверх, вправо, вниз, вниз, влево, вверх.

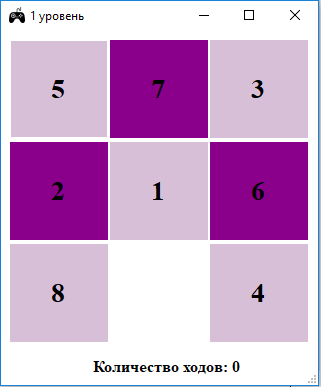
Это должно поместить костяшку с нужным числом на место. Заметим, что это временно нарушает последовательность частей, собранных ранее.

***Стадия 2: Сборка остальных частей.***

Используйте технику, описанную в стадии 1, чтобы последовательно собрать каждую строку, кроме двух последних.

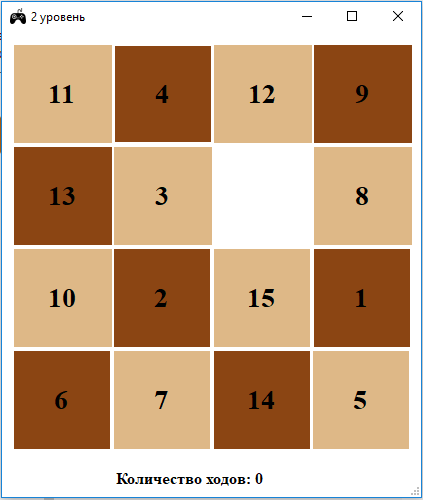
Поверните головоломку на четверть поворота вправо. Левая колонка из двух строк теперь стала верхней строкой.

Используйте технику из стадии 1, чтобы последовательно собрать каждую строку, пока их не останется две. Это значит, что осталось собрать квадрат 2 на 2.  
Двигайте части оставшегося квадрата по кругу, пока одна из частей не встанет на своё место, и пробел не окажется на правильном месте. Две другие части так же должны автоматически выстроиться на свои места.

****

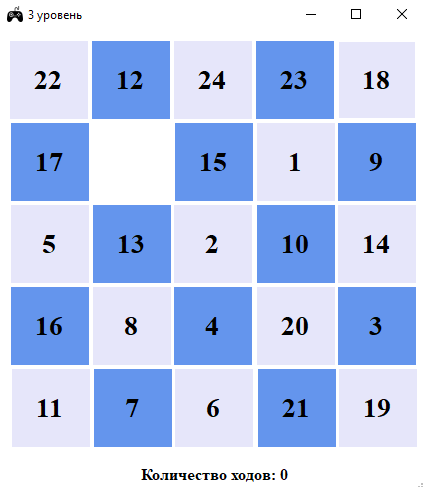
**1 уровень**

- простой вариант игры «Пятнашки»



**2 уровень**

- оригинальный вариант игры «Пятнашки»



**3 уровень**

- усложнённый вариант игры «Пятнашки»

# Исходный код классов

## **Класс StartGame**

using System;

using System.Threading;

using System.Windows.Forms;

namespace WindowsFormsApplication2

{

public partial class StartGame : Form

{

public StartGame()

{

InitializeComponent();

}

private void Exit\_Click(object sender, EventArgs e)

{

SOUND.playExit();

Thread.Sleep(1000);

this.Close();

}//Exit\_Click()

public void Chechikov\_CheckedChanged(object sender, EventArgs e) // легкий режим

{

// если было изменение галочки

if (Chechikov.Checked)

{

Chechikov.Text = "Режим Чечиков";//on

SOUND.SoundOn(); // звук на смену галочки включить

FormGame9.chOn();

}

else

{

Chechikov.Text = "Режим Чечиков";//off

SOUND.SoundOn(); // звук на смену галочки включить

FormGame9.chOff();

}

SOUND.playKey();

}//Chechikov\_CheckedChanged()

private void Start\_Click(object sender, EventArgs e)

{

SOUND.playStart();

//start game from 1st level

Hide();

StartLevel();

Close();

}//Start\_Click()

private void StartLevel()

{

// создать форму для лабиринта 1го уровня

FormGame9 Level = new FormGame9();

Level.ShowDialog();

}//StartLevel()

}//StartGame

}//WindowsFormsApplication2

## **Класс FormGame9**

using System;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication2

{

public partial class FormGame9 : Form

{

int size = 3;

public Game starting;

public static bool chEnable = false;//режим выкл

public FormGame9()

{

InitializeComponent();

starting = new Game(size);

}

public static void chOn() { chEnable = true; }

public static void chOff() { chEnable = false; }

private void button0\_Click(object sender, EventArgs e)

{

int position = Convert.ToInt16(((Button)sender).Tag);

Clicking(position,size,1);

}//button0\_Click()

protected void Clicking(int position,int size,int lvl)

{

starting.offset(position);

Renew(size);

labelMoves.Text = "Количество ходов: " + starting.moves;

if (starting.check\_numbers())

{

if (lvl != 3)

{

SOUND.playYouWin();

DialogResult vibor\_1 = MessageBox.Show("Вы победили!!!" + "\nКоличество шагов: " + starting.moves +

"\nХотите перейти на следующий уровень?", "ПОЗДРАВЛЕНИЯ",

MessageBoxButtons.YesNo, MessageBoxIcon.None);

if (vibor\_1 == DialogResult.Yes) { Hide(); StartLevel(); }

else

{

DialogResult vibor\_2 = MessageBox.Show("Хотите повторить?", "",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (vibor\_2 == DialogResult.Yes) Launch(size);

else this.Close();

}

}//if

else

{

SOUND.playYouWin();

DialogResult vibor\_1 = MessageBox.Show("Вы победили!!!" + "\nКоличество шагов: "

+ starting.moves + "\nИГРА ОКОНЧЕНА" + "\nХотите начать заново?", "ПОЗДРАВЛЕНИЯ",

MessageBoxButtons.YesNo, MessageBoxIcon.Asterisk);

if (vibor\_1 == DialogResult.Yes) { Hide(); StartLevel(); }

else this.Close();

}//else

}

}//Clicking()

protected virtual Button button(int position)

{

switch (position)

{

case 0: return button0;

case 1: return button1;

case 2: return button2;

case 3: return button3;

case 4: return button4;

case 5: return button5;

case 6: return button6;

case 7: return button7;

case 8: return button8;

default: return null;

}

}//button()

private void FormGame9\_Load\_1(object sender, EventArgs e)

{

Launch(size);

}//FormGame9\_Load\_1()

protected void Launch(int size)

{

int a;

starting.start();

//случайное перемешивание

if (chEnable) a = 10;//ремиж вкл

else a = 1024 + DateTime.Now.DayOfYear;//режим выкл

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

starting.offset\_random();

Renew(size);

labelMoves.Text = " ";

}//Launch()

protected void Renew(int size)//обновляет содержимое кнопок

{

for (int position = 0; position < size\*size; position++)

{

int number = starting.get\_number(position);

button(position).Text = number.ToString();

button(position).Visible = (number > 0);//если номер больше 0-копочка видна

}

}//Renew()

protected virtual void StartLevel()

{

// создать форму для лабиринта 2го уровня

FormGame15 Level = new FormGame15();

Level.ShowDialog();

}//StartLevel2()

}//FormGame9

}//WindowsFormsApplication2

## **Класс FormGame15**

using System;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication2

{

public partial class FormGame15 : WindowsFormsApplication2.FormGame9

{

int size = 4;

public FormGame15()

{

InitializeComponent() ;

starting = new Game(size);

}

private void button0\_Click(object sender, EventArgs e)

{

int position = Convert.ToInt16(((Button)sender).Tag);

Clicking(position,size,2);

}//button0\_Click()

protected override Button button(int position)

{

switch (position)

{

case 0: return button0;

case 1: return button1;

case 2: return button2;

case 3: return button3;

case 4: return button4;

case 5: return button5;

case 6: return button6;

case 7: return button7;

case 8: return button8;

case 9: return button9;

case 10: return button10;

case 11: return button11;

case 12: return button12;

case 13: return button13;

case 14: return button14;

case 15: return button15;

default: return null;

}

}//button()

private void FormGame15\_Load\_1(object sender, EventArgs e)

{

Launch(size);

}//FormGame15\_Load\_1();

protected override void StartLevel()

{

// создать форму для лабиринта 3го уровня

FormGame25 Level = new FormGame25();

Level.ShowDialog();

}//StartLevel3()

}//FormGame15

}// WindowsFormsApplication2

## **Класс FormGame25**

using System;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication2

{

public partial class FormGame25 : WindowsFormsApplication2.FormGame9

{

int size = 5;

public FormGame25()

{

InitializeComponent();

starting = new Game(size);

}

private void button0\_Click(object sender, EventArgs e)

{

int position = Convert.ToInt16(((Button)sender).Tag);

Clicking(position, size,3);

}//button0\_Click()

protected override Button button(int position)

{

switch (position)

{

case 0: return button0;

case 1: return button1;

case 2: return button2;

case 3: return button3;

case 4: return button4;

case 5: return button5;

case 6: return button6;

case 7: return button7;

case 8: return button8;

case 9: return button9;

case 10: return button10;

case 11: return button11;

case 12: return button12;

case 13: return button13;

case 14: return button14;

case 15: return button15;

case 16: return button16;

case 17: return button17;

case 18: return button18;

case 19: return button19;

case 20: return button20;

case 21: return button21;

case 22: return button22;

case 23: return button23;

case 24: return button24;

default: return null;

}

}//button()

private void FormGame25\_Load(object sender, EventArgs e)

{

Launch(size);

}//FormGame25\_Load()

protected override void StartLevel()

{

// создать форму для лабиринта 3го уровня

FormGame9 Level = new FormGame9();

Level.ShowDialog();

}//StartLevel3()

}//FormGame25

}//WindowsFormsApplication2

## **Класс SOUND**

using System;

using System.Media;

namespace WindowsFormsApplication2

{

public static class SOUND// чтобы не надо было инициализировать и задавать экземпляр класса

{

// подключаем звуковые файлы ВАВ

static SoundPlayer soundKey = new SoundPlayer(Properties.Resources.KeyGame);

static SoundPlayer soundStart = new SoundPlayer(Properties.Resources.StartGame);

static SoundPlayer soundYouWin = new SoundPlayer(Properties.Resources.youWinGame);

static SoundPlayer soundExit = new SoundPlayer(Properties.Resources.ExitGame);

// галочка на включение звука

static bool soundEnable = true;

public static void SoundOn()

{

soundEnable = true;

}//SoundOn()

public static void playKey()

{

// если флаг позволяет - играем

if (soundEnable == true)

soundKey.Play();

}//playKey()

public static void playStart()

{

// если флаг позволяет - играем

if (soundEnable == true)

soundStart.Play();

}//playStart()

public static void playYouWin()

{

// если флаг позволяет - играем

if (soundEnable == true)

soundYouWin.Play();

}//playYouWin()

public static void playExit()

{

// если флаг позволяет - играем

if (soundEnable == true)

soundExit.Play();

}//playExit()

}//SOUND

}//WindowsFormsApplication2

## **Класс Game**

using System;

namespace WindowsFormsApplication2

{

public class Game

{

Coord coord;

int size;//размер

protected int[,] card;

protected int space\_x, space\_y;//координаты пустой ячейки

static Random rand = new Random();

public int moves { get; private set; }

public Game(int size)

{

this.size = size;

card = new int[size, size];//массив

coord = new Coord(size);

}//Game()

public void start()

{

//заполняем массив значениями

for (int x = 0; x < size; x++)

for (int y = 0; y < size; y++)

card[x, y] = coord.position(x, y) + 1;// +1, т.к. позиция начинается с 0

space\_x = size - 1;

space\_y = size - 1;

card[space\_x, space\_y] = 0;

}//start()

public void offset(int position)

{

int x, y;

coord.coords(position, out x, out y);

int steps = Math.Abs(x - space\_x) +

Math.Abs(y - space\_y);

if (Math.Abs(space\_x - x) + Math.Abs(space\_y - y) != 1)//1-если справо,слево,наверху,внизу;2-по диоганали

return;

card[space\_x, space\_y] = card[x, y];

card[x, y] = 0;

space\_x = x;

space\_y = y;

moves += steps;

}//offset()

public void offset\_random()

{

int a = rand.Next(0, 4);//max 4 хода

int x = space\_x;

int y = space\_y;

switch (a)

{

case 0: x--; break;//влево

case 1: x++; break;//вправо

case 2: y--; break;//вверх

case 3: y++; break;//вниз

}

offset(coord.position(x, y));

moves = 0;

}// offset\_random()

public bool check\_numbers()

{

if (!(space\_x == size - 1 &&

space\_y == size - 1))

return false;

for (int x = 0; x < size; x++)

for (int y = 0; y < size; y++)

if (!(x == size - 1 && y == size - 1))

if (card[x, y] != coord.position(x, y) + 1)

return false;

return true;

}//check\_numbers()

public int get\_number(int position)

{

//возрващает число, которое находится в нужном месте

int x, y;

coord.coords(position, out x, out y);

if (x < 0 || x >= size) return 0;//чтобы не было ошибки переполнения массива

if (y < 0 || y >= size) return 0;

return card[x, y];

}//get\_number()

}//Game

}//WindowsFormsApplication2

## **Класс Coord**

using System;

namespace WindowsFormsApplication2

{

public class Coord

{

int size;

public Coord(int size)

{

this.size = size;

}

public int position(int x, int y)

{

//переводит координаты (х,у) в позицию

if (x < 0) x = 0;

if (x > size - 1) x = size - 1;

if (y < 0) x = 0;

if (y > size - 1) y = size - 1;

return y \* size + x;

}//c->p

public void coords(int position, out int x, out int y)

{

//переводит позицию в координаты(х,у)

if (position < 0) position = 0;

if (position > size \* size - 1) position = size \* size - 1;

x = position % size;

y = position / size;

}//p->c

}//Coord

}//WindowsFormsApplication2

## **Класс Program**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApplication2

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

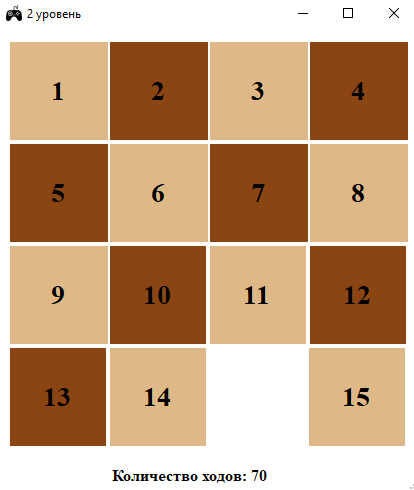
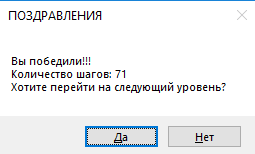
Application.Run(new StartGame());

}

}

}

# Пример работы

****

.

