Казанский (Приволжский) Федеральный университет
Институт Вычислительной математики и информационных технологий

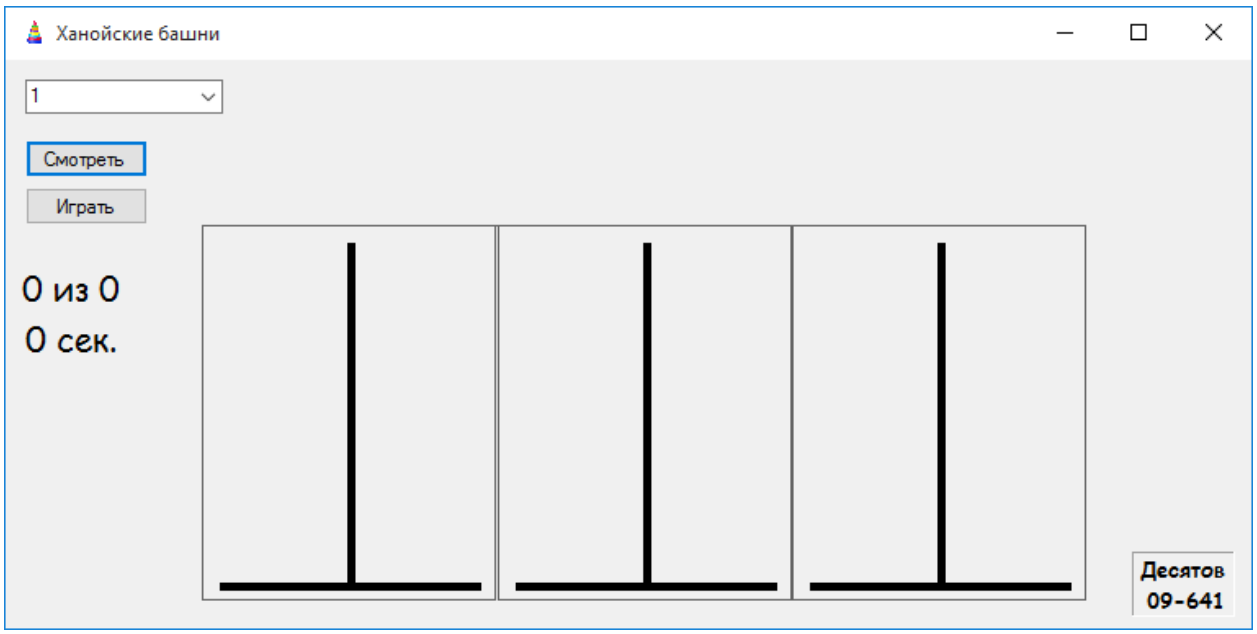
Лабораторная работа. Реализация игры «Ханойские башни» и разработка оптимальной стратегии

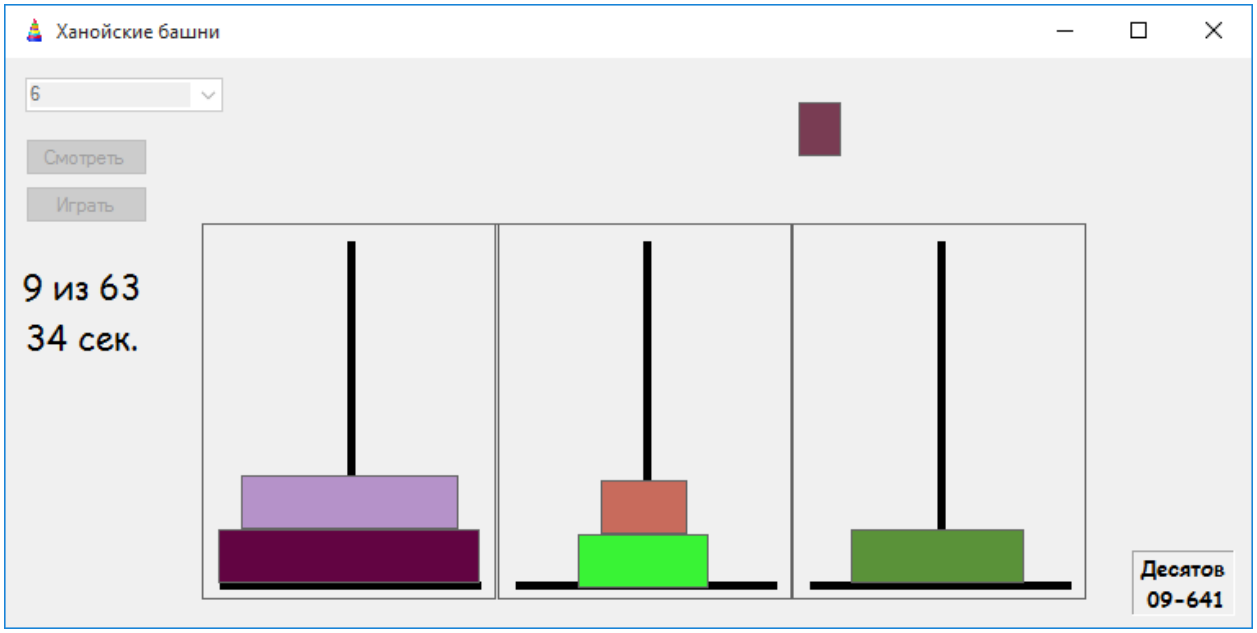
Студент 09-641 Десятов Александр

Казань – 2017

Цель работы: реализовать игру «Ханойские башни» и разработать оптимальную стратегию.

Скрины:





Код:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace Ханойские_башни
{
    public partial class Form1 : Form
    {
        public Form1()
```

```
        {
            InitializeComponent();
        }
        System.Media.SoundPlayer sw;
        int x0 = 30, y0 = 30,
            xk, yk, x, y;
        PictureBox[] Disk = new PictureBox[10];//Диски
        PictureBox[] Tower = new PictureBox[3];//Башни
        Stack<int>[] tow = new Stack<int>[3];//стэк для хранения номеров дисков для каждой башни
        int time = 0;
        bool up_end = true, vbok_end = true, down_end = true;//перемещение
        int AA, BB;//AA откуда BB куда для k = 0
        int A, B, k; // A откуда B куда k номер диска
        int number = 0; // число дисков
        int step = 0, stepMax; // шаг
        int w; // как переместить по горизонтале
        int MiN;// -11
        int MaX;// +11

        bool Game = false; //Играть
        bool perenos = false;

        private void Form1_Load(object sender, EventArgs e)
        {
            sw = new System.Media.SoundPlayer("D:\\Users\\Desyatov Alexander\\Документы\\Visual
Studio 2012\\Projects\\Ханойские башни\\IBelieve.wav");//добавление фоновой музыки
            pictureBox1.Enabled = false;
            pictureBox2.Enabled = false;
            pictureBox3.Enabled = false;
            sw.PlayLooping();//зацикленное проигрывание музыки
            xk = Width - 30;
            yk = Height - 30;
            for (int i = 0; i < 3; i++)//создаем массив башень
            {
                PictureBox pic0 = new PictureBox();
                pic0.Size = new Size(180, 230);
                pic0.Location = new Point(x0 + (i + 1) * (xk - x0) / 4 - pic0.Width / 2, yk -
pic0.Height - y0);
                pic0.BorderStyle = BorderStyle.FixedSingle;
                Tower[i] = pic0;
                Controls.Add(Tower[i]);
                Tower[i].Visible = false;
            }
            comboBox1.SelectedItem = "1";
            label1.Text = "0 из 0";
            label2.Text = "0" + " сек.";
        }
        private void button1_Click(object sender, EventArgs e)//реализация автоматической игры
        {
            comboBox1.Enabled = false;
            button1.Enabled = false;
            button2.Enabled = false;
            timer2.Start();
            timer2.Interval = 1000;
            timer2.Enabled = true;
            number = Convert.ToInt32(comboBox1.Text);
            stepMax = (int)Math.Pow(2, number) - 1;
            AA = 0;
            BB = (number % 2) + 1;
            MiN = -(number + 1);
            Random rnd = new Random();
            Stack<int> tw0 = new Stack<int>();
            tw0.Push(MiN);
            for (int i = 0; i < number; i++)//инициализируем диски, укладываем по порядку в
//первую башню
            {
                PictureBox disk0 = new PictureBox();
                disk0.Size = new Size((i + 1) * (Tower[0].Width - 20) / number,
```

```
                    (Tower[0].Height - 30) / number);
                disk0.Location = new Point(Tower[0].Location.X + Tower[0].Width / 2 - disk0.Width
/ 2,
                    Tower[0].Location.Y + Tower[0].Height - 10 - (number - i) * disk0.Height);
                disk0.BorderStyle = BorderStyle.FixedSingle;
                disk0.BackColor = Color.FromArgb(rnd.Next(0, 255), rnd.Next(0, 255), rnd.Next(0,
255));
                Disk[i] = disk0;
                Controls.Add(Disk[i]);
                Disk[i].BringToFront();
                tw0.Push(number - i - 1);

            }
            tow[0] = tw0;
            Stack<int> tw1 = new Stack<int>();
            tw1.Push(MiN);
            Stack<int> tw2 = new Stack<int>();
            tw2.Push(MiN);
            tow[1] = tw1;
            tow[2] = tw2;
            timer1.Start();
            timer1.Interval = 1;
            timer1.Enabled = true;
        }
        private void timer1_Tick(object sender, EventArgs e)
        {
            timer1.Enabled = false;
            if (up_end && vbok_end && down_end)//работа с перемещением дисков
            {
                if ((tow[1].Count() - 1) == number || (tow[2].Count() - 1) == number)
                {
                    sw.Stop();
                    timer2.Enabled = false;
                    MessageBox.Show("Успех!\nВремя: " + label2.Text);
                    Close();
                }
                else
                {
                    step++;
                    label1.Text = step.ToString() + " из " + stepMax.ToString();
                    if (step % 2 == 1)
                    {
                        A = AA;
                        B = BB;
                        AA = B;
                        if (number % 2 == 0)
                            BB = (BB + 1) % 3;//для четного;
                        else
                            BB = (BB + 2) % 3;
                        up_end = false;
                    }
                    else
                    {
                        for (int i = 0, max = -1, min = number + 1; i < 3; i++)
                        {
                            int value = tow[i].Peek();
                            if (value > max && value != 0)
                            {
                                B = i;
                                max = value;
                            }
                            if (value < min && value != 0 && value != MiN)
                            {
                                A = i;
                                min = value;
                            }
                            if (value == MiN)
                            {
```

```
                            B = i;
                            max = number + 1;
                        }
                    }
                    up_end = false;
                }
                w = Tower[B].Location.X - Tower[A].Location.X;
            }
        }
        else
        {
            int dh = 6;
            k = tow[A].Peek();
            if (!up_end)
            {
                Disk[k].Location = new Point(Disk[k].Location.X,
                    Disk[k].Location.Y - dh);
                if (Disk[k].Location.Y <= y0)
                {
                    up_end = true;
                    vbok_end = false;
                }
            }
            if (!vbok_end)
            {
                if (w >= dh)
                {
                    Disk[k].Location = new Point(Disk[k].Location.X + dh,
                    Disk[k].Location.Y);
                    w -= dh;
                }
                if (w <= (-dh))
                {
                    Disk[k].Location = new Point(Disk[k].Location.X - dh,
                    Disk[k].Location.Y);
                    w += dh;
                }
                if (Math.Abs(w) < dh)
                {
                    vbok_end = true;
                    down_end = false;
                }
            }
            if (!down_end)
            {
                Disk[k].Location = new Point(Disk[k].Location.X,
                    Disk[k].Location.Y + dh);
                if (Disk[k].Location.Y >= Tower[B].Location.Y + Tower[B].Height - 10 -
Disk[k].Height * (tow[B].Count()))
                {
                    down_end = true;
                    tow[B].Push(tow[A].Pop());
                }
            }
        }
        timer1.Enabled = true;
    }
    private void Form1_FormClosed(object sender, FormClosedEventArgs e)//отключение таймеров
при закрытии формы
    {
        timer1.Enabled = false;
        timer2.Enabled = false;
    }
    private void timer2_Tick(object sender, EventArgs e)
    {
        time++;
        label2.Text = time.ToString() + " сек.";
    }
```

```
        private void button2_Click(object sender, EventArgs e)//реализация ручной игры
        {
            Game = true;
            pictureBox1.Enabled = true;//перетаскивать будем их
            pictureBox2.Enabled = true;
            pictureBox3.Enabled = true;
            comboBox1.Enabled = false;
            button1.Enabled = false;
            button2.Enabled = false;
            timer2.Start();
            timer2.Interval = 1000;
            timer2.Enabled = true;
            number = Convert.ToInt32(comboBox1.Text);
            MaX = number + 1;
            Random rnd = new Random();
            Stack<int> tw0 = new Stack<int>();
            tw0.Push(MaX);
            for (int i = 0; i < number; i++)//создание и задание положение дисков
            {
                PictureBox disk0 = new PictureBox();
                disk0.Size = new Size((i + 1) * (Tower[0].Width - 20) / number,
                    (Tower[0].Height - 30) / number);
                disk0.Location = new Point(Tower[0].Location.X + Tower[0].Width / 2 - disk0.Width
/ 2,
                    Tower[0].Location.Y + Tower[0].Height - 10 - (number - i) * disk0.Height);
                disk0.BorderStyle = BorderStyle.FixedSingle;
                disk0.BackColor = Color.FromArgb(rnd.Next(0, 255), rnd.Next(0, 255), rnd.Next(0,
255));
                disk0.Name = "diskN" + (number - i - 1).ToString();//
                Disk[i] = disk0;
                Controls.Add(Disk[i]);
                Disk[i].BringToFront();
                tw0.Push(number - i - 1);
            }
            Prisvoit(pictureBox1, Disk[0]);
            tow[0] = tw0;
            Stack<int> tw1 = new Stack<int>();
            tw1.Push(MaX);
            tow[1] = tw1;
            Stack<int> tw2 = new Stack<int>();
            tw2.Push(MaX);
            tow[2] = tw2;
            label1.Text = "0" + "-й шаг";
        }
        private void Prisvoit(PictureBox A, PictureBox B)//задание координат, размеров, цвета
одного диска другому(потому что перетаскиваем не элементы массива дисков, а вспомогательные)
        {
            A.BackColor = B.BackColor;
            A.Size = B.Size;
            A.Location = B.Location;
            A.Visible = true;
            A.BringToFront();
            //B.BackColor = Color.White;
        }
        private void MouseDown_Disk()//нажатие на диск, все нажатия на диски направляются сюда
        {
            if (!perenos)
            {
                k = tow[A].Peek();
                Disk[k].Visible = false;
                perenos = true;
            }
        }
        private void MouseMove_Disk(MouseEventArgs e)движение мышки по диску, аналогично нажатию
        {
            x += e.Location.X - Disk[k].Width / 2;
            y += e.Location.Y - Disk[k].Height / 2;
            if (perenos)
```

```
            {
                switch (A)
                {
                    case 0:
                        pictureBox1.Location = new Point(x, y);
                        break;
                    case 1:
                        pictureBox2.Location = new Point(x, y);
                        break;
                    case 2:
                        pictureBox3.Location = new Point(x, y);
                        break;
                }
            }
        }
        private void MouseUp_Disk()//отпускание диска, аналогично двум предыдущим методам
        {
            //не диск а пикчабокс
            if (perenos)
            {
                int xDisk = 0, yDisk = 0;
                switch (A)
                {
                    case 0:
                        xDisk = pictureBox1.Location.X;
                        yDisk = pictureBox1.Location.Y;
                        break;
                    case 1:
                        xDisk = pictureBox2.Location.X;
                        yDisk = pictureBox2.Location.Y;
                        break;
                    case 2:
                        xDisk = pictureBox3.Location.X;
                        yDisk = pictureBox3.Location.Y;
                        break;
                }
                bool popal = false;
                for (int i = 0; i < 3; i++)
                {
                    if (xDisk > Tower[i].Location.X && xDisk < Tower[i].Location.X +
Tower[i].Width &&
                        yDisk > Tower[i].Location.Y && yDisk < Tower[i].Location.Y +
Tower[i].Height)
                    {
                        B = i;
                        popal = true;
                    }
                }
                if (tow[A].Peek() >= tow[B].Peek())
                {
                    popal = false;
                }
                if (popal)
                {
                    tow[B].Push(tow[A].Pop());

                    switch (A)
                    {
                        case 0:
                            if (tow[A].Peek() != MaX)
                            {
                                Prisvoit(pictureBox1, Disk[tow[A].Peek()]);
                            }
                            else
                            {
                                pictureBox1.Visible = false;
                            }
                            break;
```

```
                        case 1:
                            if (tow[A].Peek() != MaX)
                            {
                                Prisvoit(pictureBox2, Disk[tow[A].Peek()]);
                            }
                            else
                            {
                                pictureBox2.Visible = false;
                            }
                            break;
                        case 2:
                            if (tow[A].Peek() != MaX)
                            {
                                Prisvoit(pictureBox3, Disk[tow[A].Peek()]);
                            }
                            else
                            {
                                pictureBox3.Visible = false;
                            }
                            break;
                    }
                    Disk[k].Location = new Point(Tower[B].Location.X + Tower[B].Width / 2 -
Disk[k].Width / 2,
                        Tower[B].Location.Y + Tower[B].Height - 10 - (tow[B].Count() - 1) *
Disk[k].Height);
                    switch (B)
                    {
                        case 0:
                            Prisvoit(pictureBox1, Disk[k]);
                            break;
                        case 1:
                            Prisvoit(pictureBox2, Disk[k]);
                            break;
                        case 2:
                            Prisvoit(pictureBox3, Disk[k]);
                            break;
                    }
                    step++;
                    label1.Text = step.ToString() + "-й шаг";
                    if ((tow[1].Count() - 1) == number || (tow[2].Count() - 1) == number)
                    {
                        sw.Stop();
                        timer2.Enabled = false;
                        MessageBox.Show("Успех!\nВремя: " + label2.Text + "\nШагов: " +
step.ToString());
                        Close();
                    }
                }
                else//если не попал
                {
                    switch (A)
                    {
                        case 0:
                            pictureBox1.Location = Disk[k].Location;
                            break;
                        case 1:
                            pictureBox2.Location = Disk[k].Location;
                            break;
                        case 2:
                            pictureBox3.Location = Disk[k].Location;
                            break;
                    }
                }
            }
            Disk[k].Visible = true;
            perenos = false;
        }
        //MouseDown
```

```
        private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
        {
            A = 0;
            MouseDown_Disk();
}
        private void pictureBox2_MouseDown(object sender, MouseEventArgs e)
        {
            A = 1;
            MouseDown_Disk();
        }
        private void pictureBox3_MouseDown(object sender, MouseEventArgs e)
        {
            A = 2;
            MouseDown_Disk();
        }
        private void Form1_MouseDown(object sender, MouseEventArgs e)
        {
        }
        //MouseMove
        private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
        {
            MouseMove_Disk(e);
        }
        private void pictureBox2_MouseMove(object sender, MouseEventArgs e)
        {
            MouseMove_Disk(e);
        }
        private void pictureBox3_MouseMove(object sender, MouseEventArgs e)
        {
            MouseMove_Disk(e);
        }
        private void Form1_MouseMove(object sender, MouseEventArgs e)//запоминаем положение мыши
на форме
        {
            if (Game)
            {
                x = e.Location.X;
                y = e.Location.Y;
            }
        }
        //MouseUp
        private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
        {
            MouseUp_Disk();
        }
        private void pictureBox2_MouseUp(object sender, MouseEventArgs e)
        {
            MouseUp_Disk();
        }
        private void pictureBox3_MouseUp(object sender, MouseEventArgs e)
        {
            MouseUp_Disk();
        }
        private void Form1_MouseUp(object sender, MouseEventArgs e)
        {
        }
    }
}
```