Dílenská praxe

A4	4. Program gravitační piškvorky		
Petřík Vít		1/7	Známka:
27.11. 2019	Datum odevzdání:	8.1. 2020	Odevzdáno:

Zadání:

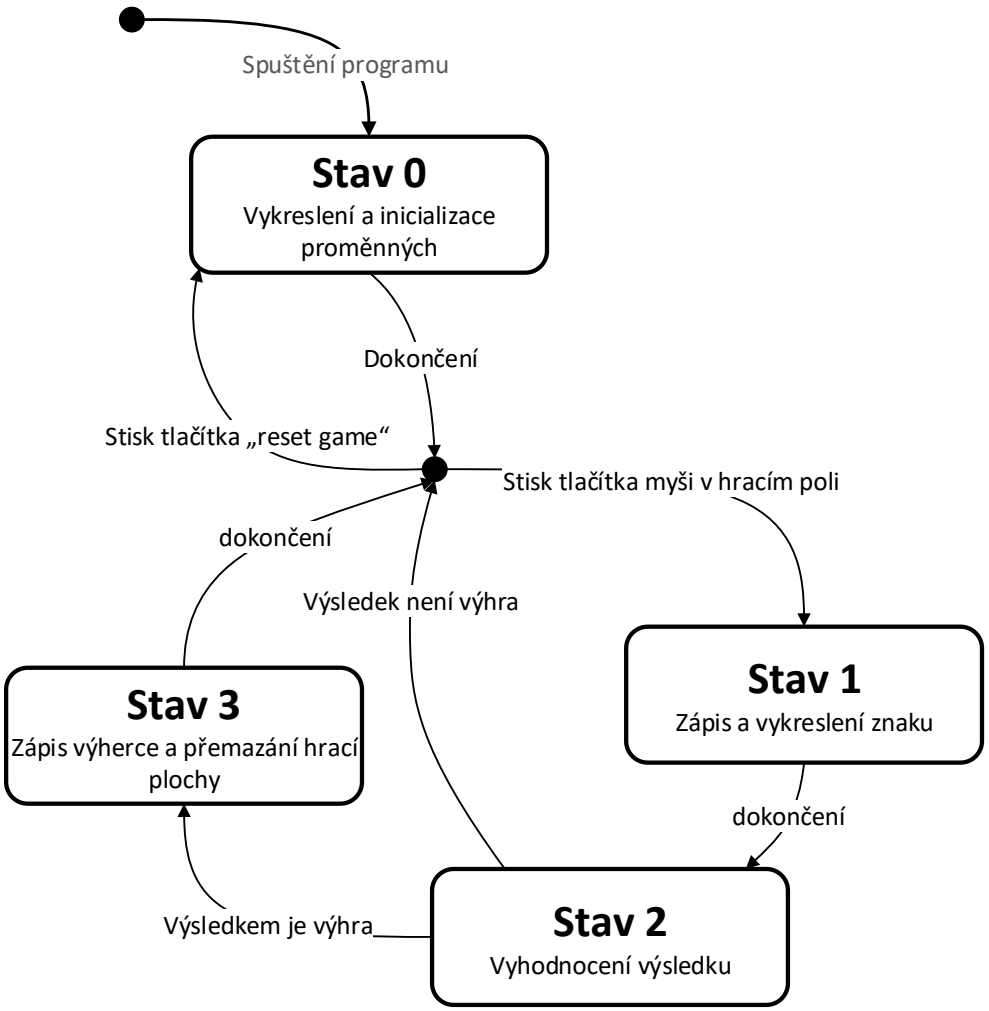
Zpracujte program (hru) v programovacím jazyce C# tak, aby obsahoval nejméně tyto funkce:

1) hra je pro dva hráče, jednoho hráče může volitelně simulovat program
2) hráči střídavě vkládají své hrací kameny na horní straně hrací plochy. Tyto „působením gravitace“ propadají ke spodní straně hrací plochy na první volnou pozici od spodní hrany hrací plochy.
3) pravidla hry jsou stejná jako u „klasických“ piškvorek
4) sleduje skóre obou hráčů a určí případného vítěze

Postup:

- Vykreslení čtverečkové sítě na začátku programu.
- Reakce na stisk tlačítka myši.
- Vyhodnocení všech možných výherních kombinací, které stisk tlačítka mohl vyvolat.
- Výpis výsledků do messageBoxu a do herního okna.

Vývojový diagram:



Výpis programu:

Příloha 1 – C++

Závěr:

Program funguje tak jak má. Jak jsem při testování zjistil, gravitační piškvorky se hrají docela špatně a vyžadují implementaci nových herních strategií. Proto jsem přidal checkbox, kterým se dá gravitace přepínat v průběhu hry.

Přílohy:

- Příloha 1 – 4 strany

Příloha 1

```
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 tic_tac_toe
{
    public partial class Form1 : Form
    {
        //proměnné
        int boxSize = 20;
        int[,] grid;
        int player = 1;
        int[] score;

        public Form1()
        {
            InitializeComponent();
        }

        //vyhodnocení zda tah byl vítězný
        private void evaluate(int[] position)
        {
            int result = 0;
            for(int i = 0; i < count.Value; i++)
            {
                //horizontální čára
                for (int x = 0; x < count.Value; x++)
                {
                    try
                    {
                        if (grid[position[0] + x - i, position[1]] != player)
                            break;
                    }
                    catch { break; }
                    if (x == count.Value - 1)
                        result = player;
                }

                //vertikální čára
                for (int y = 0; y < count.Value; y++)
                {
                    try
                    {
                        if (grid[position[0], position[1] + y - i] != player)
                            break;
                    }
                    catch { break; }
                    if (y == count.Value - 1)
                        result = player;
                }

                //uhlopříčka
                for (int j = 0; j < count.Value; j++)
                {
```

```
                    try
                    {
                        if (grid[position[0] + j - i, position[1] + j - i] != player)
                            break;
                    }
                    catch { break; }
                    if (j == count.Value - 1)
                        result = player;
                }

                //uhlopřička na druhou stranu
                for (int j = 0; j < count.Value; j++)
                {
                    try
                    {
                        if (grid[position[0] + j - i, position[1] - j + i] != player)
                            break;
                    }
                    catch { break; }
                    if (j == count.Value - 1)
                        result = player;
                }
            }

            //vypsání výsledku do message boxu
            //a přepsání gridu
            if(result != 0)
            {
                score[result-1]++;
                MessageBox.Show("Vyhrál hráč " + (result == 1 ? "O" : "X"));
                int[] gridSize = drawGrid(play_ground.CreateGraphics());
                grid = new int[gridSize[0], gridSize[1]];
            }
            //aktualizace výpisu výsledků
            updateScore();
        }

        //vykreslí čtverečkovou síť
        //a vrátí rozměr sítě
        private int[] drawGrid(Graphics g)
        {
            g.Clear(Color.White);
            Pen pen = new Pen(Color.Green);
            int boxSize = 20;
            int[] dimension_Array = new int[2];
            //g.Clear(Color.Blue);
            for (dimension_Array[0] = 1; dimension_Array[0] < play_ground.Size.Width / boxS
ize; dimension_Array[0]++)
            {
                g.DrawLine(pen, dimension_Array[0] * boxSize, 0, dimension_Array[0] * boxSi
ze, play_ground.Size.Height - play_ground.Size.Height % boxSize);
            }
            for (dimension_Array[1] = 1; dimension_Array[1] < play_ground.Size.Height / box
Size; dimension_Array[1]++)
            {
                g.DrawLine(pen, 0, dimension_Array[1] * boxSize, play_ground.Size.Width -
 play_ground.Size.Width % boxSize, dimension_Array[1] * boxSize);
            }
            return dimension_Array;
        }

        //zakreslí symbol do sítě
```

```
        private void drawSymbol(char symbol, int[] position, Pen pen)
        {
            Graphics g = play_ground.CreateGraphics();
            switch (symbol)
            {
                case 'o':
                    g.DrawEllipse(pen, position[0] * boxSize, position[1] * boxSize, boxSiz
e, boxSize);
                    break;
                case 'x':
                    g.DrawLine(pen, position[0] * boxSize, position[1] * boxSize, position[
0] * boxSize + boxSize, position[1] * boxSize + boxSize);
                    g.DrawLine(pen, position[0] * boxSize, position[1] * boxSize + boxSize,
 position[0] * boxSize + boxSize, position[1] * boxSize);
                    break;
                default:
                    break;
            }
        }

        //zmáčknutí tlačítka myši v čtvrečkové síti
        private void Play_ground_MouseClick(object sender, MouseEventArgs e)
        {
            MouseEventArgs eM = (MouseEventArgs)e;
            Graphics g = play_ground.CreateGraphics();

            //zjistíme do jaké pozice na síti myš klikla
            int[] position = { (int)Math.Floor((double)eM.X / boxSize), (int)Math.Floor((do
uble)eM.Y / boxSize) };
            Pen[] pens = { new Pen(Color.Red, 2), new Pen(Color.Blue, 2) };

            //pokud máme oktivovanou gravitaci necháme propadnou Y souřadnici
            if(gravity.Checked)
            {
                position[1] = grid.GetLength(1)-1;
                for(int i = 0; i < grid.GetLength(1); i++)
                {
                    if(grid[position[0], i] != 0)
                    {
                        position[1] = i - 1;
                        break;
                    }
                }
            }

            //když je zapnutá gravitace, může se stát, že se Y souřadnice dostane
            //do bodu -1, tak to vošéfujeme try-catchem
            try
            {
                //pokud je pozice prázdná zakreslíme znak a vyhodnotíme
                if(grid[position[0], position[1]] == 0)
                {
                    grid[position[0], position[1]] = player;
                    drawSymbol(player == 1 ? 'o' : 'x', position, pens[player - 1]);
                    evaluate(position);

                    //přepneme hráče
                    player = player == 1 ? 2 : 1;

                    //deaktivujeme textbox na změnu výherního počtu políček
                    count.Enabled = false;
                }
```

```
            }
            catch
            {
                Console.Write("Něco se trochu pokazilo nebo máme souřadnice mimo hrací ploc
hu :)");
            }
        }

        //vykreselní sítě na začátku programu
        private void Play_ground_Paint(object sender, PaintEventArgs e)
        {
            int[] gridSize = drawGrid(e.Graphics);
            grid = new int[gridSize[0], gridSize[1]];
            score = new int[] {0, 0};
            updateScore();
        }

        //zmáčknutí resetovacího tlačidla
        private void resetGrid(object sender, EventArgs e)
        {
            int[] gridSize = drawGrid(play_ground.CreateGraphics());
            grid = new int[gridSize[0], gridSize[1]];
            score = new int[] { 0, 0 };
            updateScore();
            count.Enabled = true;
        }

        //aktualizování výpisu výsledku
        private void updateScore()
        {
            score_text.Text = $"{score[0].ToString()} : {score[1].ToString()}";
        }
    }
}
```