



# Díleňská praxe

<b>A4</b>	I) Program „Gravitační“ piškvorky		
Denis John		1/9	Známka:
26.10.2023	Datum odevzdání:	9.11.2023	Odevzdáno:



### **Zadání:**

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

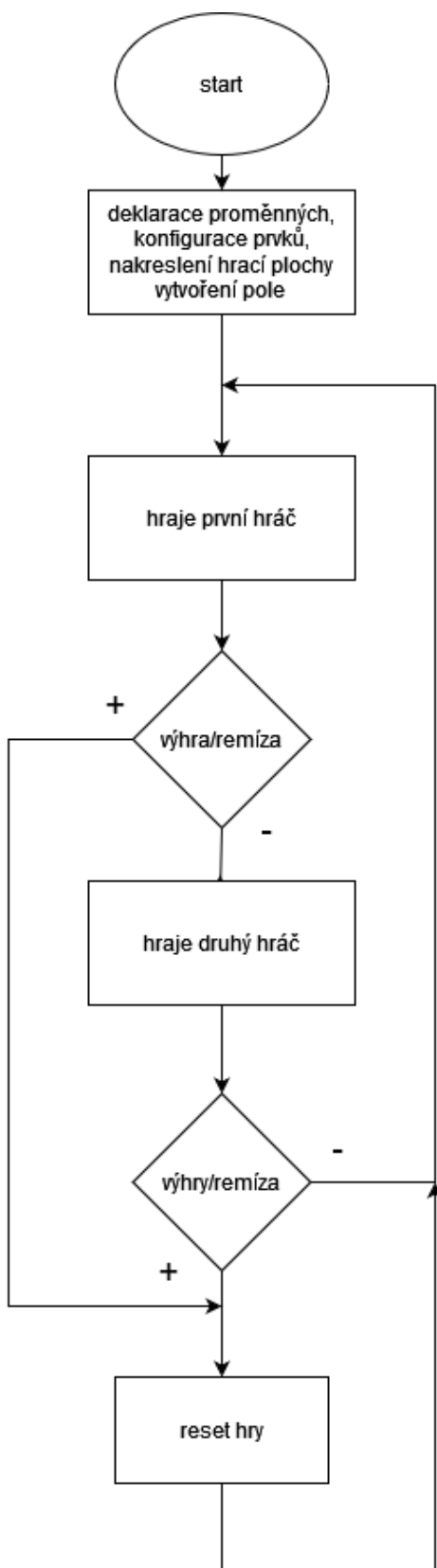
- 1) hra je pro dva hráče, jednoho hráče může volitelně simulovat program
- 2) hráči střídavě vkládají své hrací kameny na horní straně hrací plochy. Tyto „gravitačním působením“ propadnou ke spodní straně hrací plochy na první volnou pozici od spodní hrany hrací plochy.
- 3) pravidla hry jsou stejná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

### **Postup řešení:**

- 1) Vytvořím projekt v jazyce C# ve Visual Studiu.
- 2) Definuji herní pole o určeném počtu řádků a sloupců.
- 3) Vytvořím vizuální rozhraní pole z tlačítek.
- 4) Vytvořím smyčku, kdy se po každém tahu aktualizuje stav pole, vizuální rozhraní a kontroluje výhru nebo remízu.
- 5) Přidám kontrolu na výherní kombinace a na remízu.
- 6) Vytvořím funkci pro resetování hry v případě výhry nebo remízy.



### Vývojový diagram:





### Výpis programu:

```
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 Piskvorkyjohn
{
    public partial class Form1 : Form
    {
        private const int Rows = 6;           // počet řádků
        private const int Columns = 7;         // počet sloupců
        private int Player = 1;                // deklarace hdonoty pro hráče
        private int[,] board = new int[Rows, Columns]; // deklarace pole vytvořené z řádků a
                                                    sloupců, kde se uchovávají informace
                                                    políček hrací plochy
        private Button[,] buttonGrid = new Button[Rows, Columns]; // deklarace pole vytvořené
                                                                    z tlačítek k uchování
                                                                    odkazů na tlačítka

        public Form1()
        {
            InitializeComponent(); // vytvoří a konfiguruje prvky, nastaví vlastnosti prvků a přidá
                                    k nim události
            PoleHracíPlochy(); // vytvoří prázdné pole
            Tlačítka(); // vytvoří hrací pole tlačítek
            KresleníPlochy(); // vykreslí prázdnou hrací plochu
        }
        private void PoleHracíPlochy()
        {
            for (int row = 0; row < Rows; row++)
            {
                // cyklus prochází řádky a slopce
                for (int col = 0; col < Columns; col++)
                {
                    board[row, col] = 0; // každému políčku nastaví hodnotu 0
                }
            }
        }
    }
}
```



```
private void Tlačítka()           // funkce pro vytvoření hracího pole z tlačítek
{
    for (int row = 0; row < Rows; row++)
    {
        // cyklus prochází řádky a sloupce a vytváří hrací pole z tlačítek
        for (int col = 0; col < Columns; col++)
        {
            Button button = new Button();    // vytvoří tlačítko tak, aby mohlo být použito
                                              ve hře
            button.Width = 50;               // nastavení šířky tlačítka
            button.Height = 50;              // nastavení výšky tlačítka
            button.Top = row * 50;           // nastavení tlačítka v ose Y 50 pixelů *
                                              momentální řádek od horní hrany
            button.Left = col * 50;          // nastavení tlačítka v ose X 50 pixelů *
                                              momentální sloupec od boční hrany
            button.Tag = col;                // ukládání sloupce do Tag, aby bylo jasné, kam hráč
                                              kliknul.
            button.Click += Click;           // nastavuje obsluhu funkce "Click"
            buttonGrid[row, col] = button;   // odkaz z vytvořeného tlačítka se bude ukládat
                                              do pole
            Controls.Add(button);            // přidá tlačítko do seznamu ovládacích prvků, tím
                                              se tlačítko zobrazí ve formuláři
        }
    }
}

private void KresleníPlochy()      // funkce kreslení políček v hracím poli
{
    for (int row = 0; row < Rows; row++)
    {
        // cyklus prochází řádky a sloupce
        for (int col = 0; col < Columns; col++)
        {
            if (board[row, col] == 1)      // pokud kliknul na tlačítko 1. hráč
            {
                buttonGrid[row, col].Text = "X";    // v tlačítku se zobrazí "X"
                buttonGrid[row, col].Font = new Font("Arial", 17, FontStyle.Bold); // nastaví
                                                         se font a velikost písma
                buttonGrid[row, col].BackColor = Color.Red;    // tlačítko se vybarví
                                                         červeně
            }
            else if (board[row, col] == 2)
            {
                buttonGrid[row, col].Text = "O";    // v tlačítku se zobrazí "O"
                buttonGrid[row, col].Font = new Font("Arial", 17, FontStyle.Bold); //
                                                         nastaví se font a velikost písma
                buttonGrid[row, col].BackColor = Color.Blue;    // tlačítko se vybarví
                                                         modře
            }
        }
    }
}
```



```
    }
    else
    {
        buttonGrid[row, col].BackColor = System.Drawing.Color.White;    // tlačítko
                                                                           se vybarví bíle
        buttonGrid[row, col].Text = "";    // tlačítko bude prázdné
    }
}
}
}

private void Click(object sender, EventArgs e)    // funkce kliknutí
{
    Button button = (Button)sender;    // odkaz ze zmáčknutého tlačítka se přepíše na
                                     proměnnou button
    int col = (int)button.Tag;    // přepíše hodnotu vlatnosti Tag na int deklaraci
    int row = Gravitace(col);    // vyvolá funkci Gravitace a zapíše na proměnnou
                                řádek

    if (row != -1)    // pokud není sloupec plný
    {
        board[row, col] = Player;    // políčko v poli se přepíše na aktuálního hráče
        KresleníPlochy();    // vyvolá funkci která vyplní políčko podle aktuálního
                             hráče, buď "X" nebo "O"

        if (Výhra())    //vyvolá funkci výhry, pokud funkce vrátí pravdu, provede se akce
        {
            MessageBox.Show(+Player + ". Hráč vyhrál!");    // zobrazí se okno se zprávou
                                                            výhry

            Resethry();    //vyvolá funkci resetování hry
        }
    }
    else
    {
        {
            Player = 3 - Player;    // přepne hráče
        }
    }
}

if (JeRemiza())    // vyvolá funkci remízi,pokud funkce vrátí pravdu
{
    MessageBox.Show("Hra skončila remízou!");    // zobrazí se okno se zprávou remízi
    Resethry();    // vyvolá funkci resetování hry
}
}
```



```
private int Gravitace(int col)    // funkce zajišťuje padání symbolů ve sloupci
{
    for (int row = Rows - 1; row >= 0; row--) // cyklus prochází řádky v daném sloupci
                                                od spoda nahoru
    {
        if (board[row, col] == 0)    // pokud je políčko prázdné
        {
            return row;    // vrátí hodnotu řádku
        }
    }
    return -1;    // vrátí -1 = sloupec je plný a nelze zahrát symbol
}

private bool Výhra()    // funkce pro kontrolu výhry
{
    // Kontrola vodorovných kombinací
    for (int row = 0; row < Rows; row++)    // cyklus prochází všechny řádky
    {
        for (int col = 0; col < Columns - 3; col++)    // cyklus prochází pouze 4 sloupce
        {
            int player = board[row, col];    // načte hodnotu hráče v políčku z pole
            if (player != 0 && player == board[row, col + 1] && player == board[row, col +
2] && player == board[row, col + 3])
            {
                return true;    // pokud se 4 políčka v řadě rovnají stejné hodnotě hráče,
                                hráč vyhrál
            }
        }
    }

    // Kontrola svislých kombinací
    for (int row = 0; row < Rows - 3; row++)    // cyklus prochází pouze 4 řádky
    {
        for (int col = 0; col < Columns; col++)    // cyklus prochází všechny sloupce
        {
            int player = board[row, col];    // načte hodnotu hráče v políčku z pole
            if (player != 0 && player == board[row + 1, col] && player == board[row + 2,
col] && player == board[row + 3, col])
            {
                return true;    // pokud se 4 políčka ve sloupci rovnají stejné hodnotě hráče,
                                hráč vyhrál
            }
        }
    }
}
```



```
// Kontrola diagonálních kombinací (sleva doprava)
for (int row = 0; row < Rows - 3; row++) // cyklus prochází pouze 4 řádky
{
    for (int col = 0; col < Columns - 3; col++) // cyklus prochází pouze 4 sloupce z
                                                // leva
    {
        int player = board[row, col]; // načte hodnotu hráče v políčku z pole
        if (player != 0 && player == board[row + 1, col + 1] && player == board[row +
2, col + 2] && player == board[row + 3, col + 3])
        {
            return true; // pokud se 4 políčka v diagonále rovnají stejné hodnotě
                        hráče, hráč vyhrál
        }
    }
}

// Kontrola diagonálních kombinací (zprava doleva)
for (int row = 0; row < Rows - 3; row++) // cyklus prochází pouze 4 řádky
{
    for (int col = 3; col < Columns; col++) // cyklus prochází pouze 4 sloupce z prava
    {
        int player = board[row, col]; // načte hodnotu hráče v políčku z pole
        if (player != 0 && player == board[row + 1, col - 1] && player == board[row +
2, col - 2] && player == board[row + 3, col - 3])
        {
            return true; // pokud se 4 políčka v diagonále rovnají stejné hodnotě
                        hráče, hráč vyhrál
        }
    }
}

return false;
}

private bool JeRemiza() // funkce pro kontrolu remízi
{
    for (int col = 0; col < Columns; col++) // prochází všechny řádky
    {
        if (board[0, col] == 0) // pokud je první políčko ve sloupci volné
        {
            return false; // vrátí nepravdu a nic se nestane
        }
    }

    return true; // všechny sloupce jsou plné
}
```





```
private void Resethry() // funkce resetování hry
{
    PoleHracíPlochy(); // vymaže hrací pole
    Player = 1;        // nastaví prvního hráče jako aktuálního hráče
    KresleníPlochy();  // vykreslí prázdnou hrací plochu
}
}
```

### **Závěr:**

Přestože výsledek nevypadá vizuálně nejlépe, tak funguje. Místo vkládání kamenů do hracího pole stačí kliknout na libovolné políčko ve sloupci a symbol se objeví na první volné pozici odspoda.