

# Konzolové piškvorky

Jako mojí ročníkovou práci/projekt jsem si vybral vytvořit piškvorky v C#

jak kód funguje:

Program začne tím, že vyzve uživatele k zadání jmen hráčů, symbolů a velikosti hrací desky.

Vytvoří novou instanci třídy Deska se zadanou velikostí a nastaví aktuálního hráče na hráče 1.

Spustí se herní smyčka, která pokračuje, dokud některý z hráčů nevyhraje nebo dokud hra neskončí remízou.

V každé iteraci smyčky se vytiskne hrací deska a aktuální hráč je vyzván, aby zadal řádek a sloupec a provedl tah.

Kód kontroluje, zda je tah platný, tj. zda je zadané místo prázdné. Pokud není platný, je hráč vyzván k provedení dalšího tahu.

Pokud je tah platný, je symbol aktuálního hráče nastaven na zadané místo na hrací ploše. Kód kontroluje, zda aktuální tah vede k výhře aktuálního hráče, voláním metody CheckWin. Pokud aktuální hráč vyhrál, hra končí a je vyhlášen vítěz.

Pokud již nejsou žádné platné tahy a žádný hráč nevyhrál, hra končí remízou.

Pokud hra neskončila, vymění se aktuální hráč a hra pokračuje.

Celkově kód funguje tak, že sleduje stav hry ve dvourozměrném poli a kontroluje výhru kontrolou řádků, sloupců a úhlopříček pole. Střídá se mezi oběma hráči, dokud některý z nich nevyhraje nebo dokud hra neskončí remízou.

Otevřete program v editoru kódu C#

Program a spustěte.

Program vás vyzve k zadání jmen dvou hráčů a velikosti herní desky.

Zadejte jméno prvního hráče a stiskněte klávesu Enter.

Zadejte jméno druhého hráče a stiskněte klávesu Enter.

Zadejte velikost herní desky, která by měla být kladné celé číslo, a stiskněte klávesu Enter.

Zobrazí se herní deska a na řadě je hráč 1.

Hráč 1 by měl zadat řádek a sloupec, kam chce umístit svůj symbol. Řádky a sloupce jsou očíslovány od 1 do velikosti hrací desky. Pokud je například velikost desky 3, jsou řádky a sloupce očíslovány od 1 do 3.

Poté, co hráč číslo 1 provede svůj tah, se hrací deska opět zobrazí a na řadě je hráč číslo dvě.

Hráč 2 by měl zadat řádek a sloupec, kam chce umístit svůj symbol.

Hra bude pokračovat tímto způsobem, přičemž se oba hráči budou střídat, dokud jeden z nich nevyhraje nebo dokud hra neskončí remízou.

Pokud některý z hráčů vyhraje, program zobrazí zprávu, který hráč vyhrál, a hru ukončí.

Pokud hra skončí remízou, program zobrazí zprávu, že hra skončila nerozhodně, a hru ukončí.

Chcete-li hrát znovu, můžete program spustit znovu a postupovat stejně jako předtím.

Rozvoj tohoto projektu do budoucna:

Dokážu si představit vytvořit robotického soupeře aby se tato hra dala hrát i v jednom hráči a teoreticky celý program přesunout do forms

Použité technologie:

```
public bool CheckWin(int row, int col, char symbol)
{
    for (int i = 0; i < Size; i++)
    {
        if (board[row, i] != symbol) break;
        if (i == Size - 1) return true;
    }

    for (int i = 0; i < Size; i++)
    {
        if (board[i, col] != symbol) break;
        if (i == Size - 1) return true;
    }

    if (row == col)
    {
        for (int i = 0; i < Size; i++)
        {
            if (board[i, i] != symbol) break;
            if (i == Size - 1) return true;
        }
    }

    if (row + col == Size - 1)
    {
        for (int i = 0; i < Size; i++)
        {
            if (board[i, Size - 1 - i] != symbol) break;
            if (i == Size - 1) return true;
        }
    }

    return false;
}
```

Metoda pro kontrolování výhry

```
2 references
public class Board
{
    public char[,] board;
    16 references
    public int Size { get; }

    1 reference
    public Board(int size)
    {
        Size = size;
        board = new char[Size, Size];
    }
}
```

Zadávání velikosti pole

```
1 reference
public bool Set(int row, int col, char symbol)
{
    if (board[row, col] != '\0')
    {
        Console.WriteLine("toto místo je již použité vyber si jiné");
        return false;
    }
    board[row, col] = symbol;
    return true;
}
```

Ochrana proti vybrání si již zabraného pole