

Gymnázium, Praha 6, Arabská

Programování

ROČNÍKOVÁ PRÁCE



2022

František Pešula

Prohlášení

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené.

Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

Obsah

PROHLÁŠENÍ.....	2
OBSAH.....	3
1. ANOTACE.....	4
2. ABSTRACT.....	4
3. ZADÁNÍ.....	5
4. HRA	6
4.1. PRAVIDLA HRY	6
4.2. TVORBA HRY.....	6
4.3 VYHODNOCENÍ HRY.....	8
4.3.1 HORIZONTÁLNÍ KONTROLA.....	9
4.3.2 VERTIKÁLNÍ KONTROLA	10
4.3.3 DIAGONÁLNÍ KONTROLA – KRAJE	11
4.3.4 DIAGONÁLNÍ KONTROLA – Z POSLEDNÍHO ŘÁDKU	12
4.3.5 KONTROLA REMÍZA	13
5. ZÁVĚR.....	14
5.1. ZHODNOCENÍ PROJEKTU	14
5.2 ZDROJE	15

1. Anotace

Mým projektem byla hra Connect Four. Ve hře Connect Four se dva hráči střídají ve vhazování žetonů do hracího pole, které je velké 7 x 6 políček. Žetony padají do hracího pole směrem dolů. Vítězí hráč, který spojí se svým žetonem čtyři políčka horizontálně, vertikálně nebo diagonálně.

2. Abstract

My project was game Connect Four. In Connect Four, two players take turns throwing tokens into the 7 x 6 court. The tokens fall down into the playing field. The winner is the player, who connects with his token the four spots horizontally, vertically or diagonally.

3. Zadání

Zadáním bylo vytvořit hru Connect Four, kterou mohou hrát dva hráči. Hra musí barevně odlišovat oba hráče. Hra musí kontrolovat počet políček spojené horizontálně, vertikálně i diagonálně na každém místě, a zároveň jakým hráčem. Hra musí zjišťovat, zda se ve hracím poli, spojili čtyři políčka. Dále hra musí určit vítěze hry, a také ho zobrazit.

4. Hra

4.1. Pravidla Hry

Connect Four je pro dva hráče. Hraje se ve hracím poli 7 x 6. Při každém tahu hráč vhodí svůj žeton do sloupce, který byl vybrán (žetony padají směrem dolů). Hráči se navzájem střídají. Cílem hry je spojit čtyři žetony své barvy horizontálně, vertikálně nebo diagonálně. Hra končí, jakmile jeden z hráčů spojí své čtyři žetony dříve zmiňovanými způsoby.

4.2. Tvorba hry

Na úvod bylo potřeba inicializovat prázdný projekt s názvem „ConnectFour“ a v něm vytvoření prázdné aplikace „HelloApplication“. Pro celé fungování hry využívám knihoven JavaFX.

Následně jsem vytvořil potřebné objekty, s kterými budu v průběhu hry pracovat či jinak manipulovat. Nejpodstatnější je třída „Kruh“ která extenduje vlastnosti a rozšiřuje třídu JavaFX.Circle. Slouží k reprezentaci jednotlivých žetonů. Další důležitou částí hry je dvourozměrné pole „hraciPole“ objektů „Kruh“ (žetonů), které má velikost počtu sloupců a řádků a reprezentuje hrací desku. Každý žeton v hracím poli má výchozí bílé pozadí a tím není vidět, tudíž není obsazen.

Dále bylo potřeba využít eventHandler `setOnMouseClicked` díky kterému mohu sledovat do jakého místa hrací desky hráč klikl. Pokud hráč klikne mimo hrací desku anebo do sloupce, který je plně obsazen, neděje se nic a čeká se na kliknutí do volného sloupce v hracím poli. Při kliknutí na sloupec, který ještě není plně obsazený, najdu nejnižše umístěný volný žeton „Kruh“. Přiřadím jej danému hráči a změním hráče na tahu.

Pro informování hráčů, který je na řadě využívám textového objektu vycentrovaným nad hrací deskou.

Zároveň při každém umístění žetonu je potřeba zkontrolovat, zda nedošlo ke spojení čtyř stejně barevných žetonů v různých směrech anebo k situaci, že už není kam žeton umístit, čímž by došlo k remíze. K tomuto využívám funkci „kontrola“.

4.3 Vyhodnocení hry

K vyhodnocení hry využívám funkce „kontrola“, která se volá vždy při umístění nového žetonu.

Funkci jsem rozdělil na celkem pět částí. Každá část, pokud najde v řadě za sebou čtyři jdoucí stejně barevné žetony, ihned funkci ukončí a vrátí číslo hráče, který vyhrál.

Funkce může vracet celkem čtyři hodnoty:

- -1: Ve hře došlo k remíze
- 0: Nikdo zatím nespojil čtyři po sobě jdoucí žetony
- 1: Vyhrál první hráč
- 2: Vyhrál druhý hráč

4.3.1 Horizontální kontrola

Kontrola hracího pole ve vodorovném směru. Zde procházím postupně jednotlivé sloupce v daném řádku a v pomocném „čítači“ si ukládám počet stejných žetonů v řadě. Při změně barvy žetonu „čítač“ vynuluji a pokračuji opět od jedné. Hledám výskyt čtyř stejných „vlastníků“ žetonů v řadě. V případě, že dojdou k poslednímu sloupci a součet v řadě stejných žetonů nedosahuje hodnoty čtyři, přesunu se na další řádek, vynuluji „čítač“ a začínám kontrolu v daném řádku.

```
/*
Kontrola horizontálního spojení.
*/

for(int r = 0; r < radky; r++) {
    posledniHrac = 0;
    pocetSpojeni = 0;
    for(int s = 0; s < sloupce; s++) {

        if (hraciPole[s][r].getHrac() == 0) {
            posledniHrac = 0;
            pocetSpojeni = 0;
            continue;
        }
        if (posledniHrac == 0 || posledniHrac != hraciPole[s][r].getHrac()) {
            posledniHrac = hraciPole[s][r].getHrac();
            pocetSpojeni = 1;
        } else {
            pocetSpojeni++;
        }
        if (pocetSpojeni == 4) {
            return posledniHrac;
        }
    }
}
```

4.3.2 Vertikální kontrola

Kontrola hracího pole ve svislém směru. Zde procházím postupně jednotlivé řádky v daném sloupci a v pomocném „čítači“ si ukládám stejně jako u horizontální kontroly, počet stejných žetonů v řadě. Při změně barvy žetonu „čítač“ vynuluji a pokračuji opět od jedné. Hledám výskyt čtyř stejných „vlastníků“ žetonů v řadě. V případě, že dojdou k poslednímu řádku a součet v řadě stejných žetonů nedosahuje hodnoty čtyři, přesunu se na další sloupec, vynuluji „čítač“ a začínám kontrolu v daném sloupci.

```
/*  
Kontrola vertikálního spojení.  
*/  
for(int s = 0; s < sloupce; s++) {  
    posledniHrac = 0;  
    pocetSpojeni = 0;  
  
    for(int r = 0; r < radky; r++) {  
  
        if (hraciPole[s][r].getHrac() == 0) {  
            posledniHrac = 0;  
            pocetSpojeni = 0;  
            continue;  
        }  
        if (posledniHrac == 0 || posledniHrac != hraciPole[s][r].getHrac()) {  
            posledniHrac = hraciPole[s][r].getHrac();  
            pocetSpojeni = 1;  
        } else {  
            pocetSpojeni++;  
        }  
        if (pocetSpojeni == 4) {  
            return posledniHrac;  
        }  
    }  
}
```

4.3.3 Diagonální kontrola – kraje

Kontrola hracího pole v diagonálním směru začínající v krajních sloupcích. Kontroluji tedy diagonální spojení z prvního sloupce vedoucí vpravo nahoru a z posledního sloupce vlevo nahoru.

V obou případech procházím řádek po řádku shora dolů a pokud existuje žeton na dalším/předchozím sloupci o jeden řádek výše, tak pokračuji dál. V momentě, kdy dojdou k prvnímu řádku a žádný předchozí řádek není, přecházím na další řádek

v prvním/posledním sloupci a celý proces opakuji. Využití pomocného čítače je stejné jako u předešlých částí funkce, tedy při posunu na další řádek jej nulují.

```
/*  
Kontrola diagonálních spojení ze stran (levá, pravá)  
*/  
  
int[] prava;  
int[] leva;  
  
for(int r = 2; r < radky; r++) {  
    prava = new int[2];  
    leva = new int[2];  
  
    for(int s = 0; s <= r; s++) {  
  
        leva = kontrolaPolicka(leva, hraciPole[s][r - s]);  
        if (leva[1] == 4) {  
            return leva[0];  
        }  
  
        prava = kontrolaPolicka(prava, hraciPole[sloupce - 1 - s][r - s]);  
        if (prava[1] == 4) {  
            return prava[0];  
        }  
    }  
}
```

4.3.4 Diagonální kontrola – z posledního řádku

Kontrola hracího pole v diagonálním směru začínající na posledním řádku v obou směrech. Kontroluji diagonální spojení z posledního řádku vedoucí vpravo nahoru a vlevo nahoru. V obou případech procházím řádek po řádku zespoda nahoru a pokud existuje žeton na dalším/předchozím sloupci o jeden řádek výše, tak pokračuji dál. V momentě, kdy dojdou k prvnímu řádku nebo ke krajnímu sloupci a žádný předchozí řádek nebo sloupec není, přecházím na další/předchozí sloupce v posledním řádku a celý proces opakuji. Využití pomocného čítače je stejné jako u předešlých částí funkce, tedy při posunu na další sloupce jej nulují.

```
/*
Kontrola diagonál zespoda oběma směry
*/

for(int s = 0; s < sloupce; s++) {
    prava = new int[2];
    leva = new int[2];

    for(int r = 0; r < radky; r++) {
        if (s + r <= sloupce - 1) {
            leva = kontrolaPolicka(leva, hraciPole[s + r][radky - 1 - r]);
            if (leva[1] == 4) {
                return leva[0];
            }
        }
        if (sloupce - 1 - s - r >= 0) {
            prava = kontrolaPolicka(prava, hraciPole[sloupce - 1 - s - r][radky - 1 - r]);
            if (prava[1] == 4) {
                return prava[0];
            }
        }
    }
}

return 0;
}
```

4.3.5 Kontrola remíza

Zde ověřím, zda je počet tahů roven násobku počtu sloupců a řádků. V tomto případě již hráč na tahu nemůže umístit svůj žeton do hracího pole, a pokud nedošlo ke spojení čtyř žetonů v jakémkoliv směru, dojde k remíze.

5. Závěr

5.1. Zhodnocení projektu

Psaní projektu byla pro mě velká zkušenost, to samé i psaní této dokumentace. Tento program je můj první, který byl zaměřený na herní tematiku. Jsem zvědavý, jakou podobnou hru udělám příště.

5.2 Zdroje

- https://en.wikipedia.org/wiki/Connect_Four – veškeré informace o hře
- <https://docs.oracle.com/javase/8/javafx/api/toc.htm> - seznam kódů (některé použity v kódu)