|  |
| --- |
| Piškvorky |
| Ivan Kratochvíl |
| Prohlašuji, že jsem projekt vypracoval samostatně pouze s využitím zdrojů uvedených v tomto dokumentu. |

## Zadání projektu

Vytvořit hru piškvorky pro dva hráče. Hra umožní hráčům nastavit jejich jména, velikost herního pole a počtu značek za sebou potřebného k výhře. Ovládá se klikáním myši na jednotlivá políčka. Pokud dojde k výhře některého s hráčů, hra to oznámí. Hráči můžou uložit a nahrát rozehranou hru a přehrát již dohranou partii.

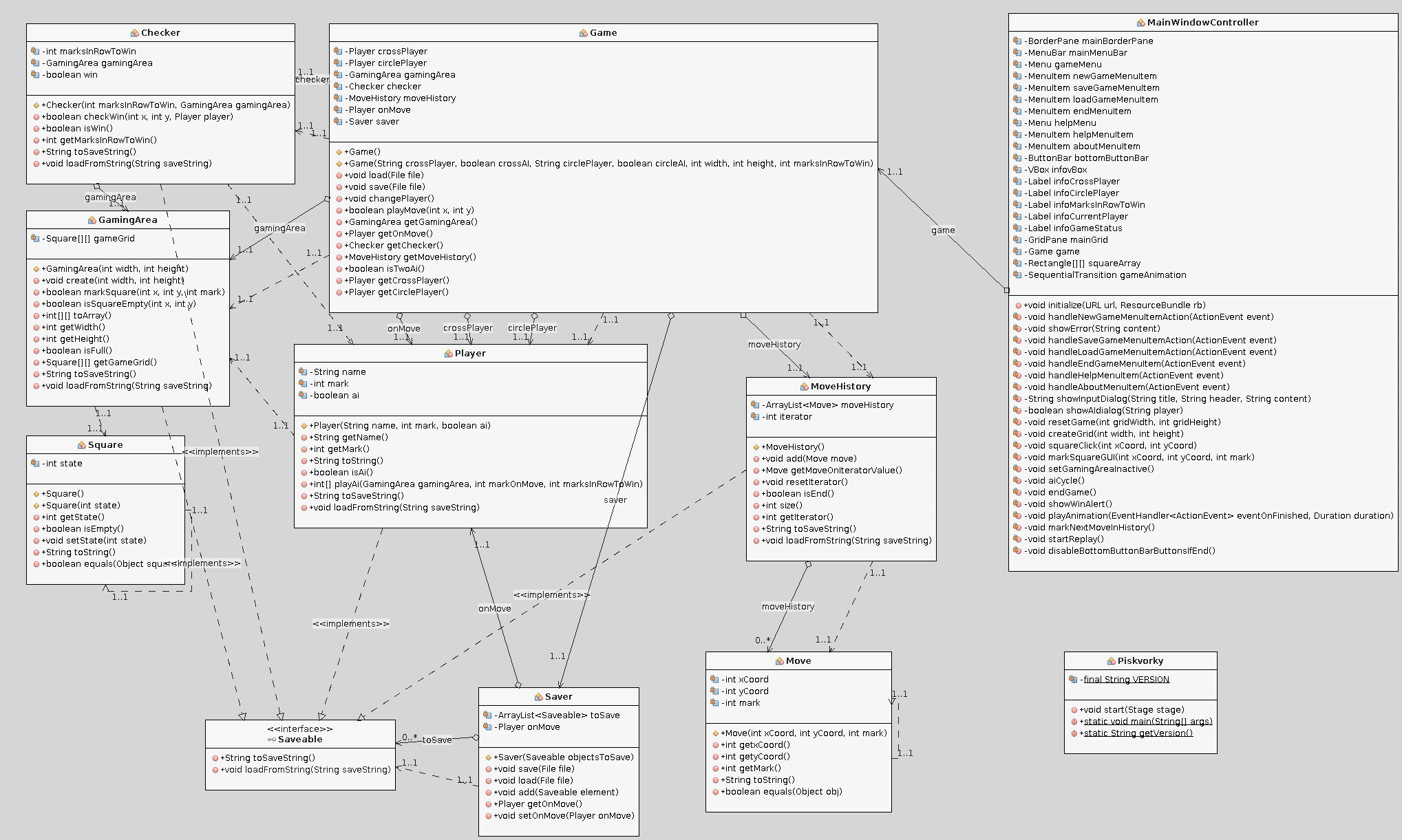
# Funkcionalita projektu

Při spuštění hry se objeví prázdné okno s menu. Uživatel si může zobrazit nápovědu, ve které jsou jednoduchá pravidla hry a návod na ovládání programu. Také si může zobrazit informace o verzi programu.

Pokud uživatel spustí novou hru, vyskočí několik dialogů, ve kterých postupně nastaví jména obou hráčů, zda za některého z nich má hrát umělá inteligence, velikost pole a počet značek, které jsou třeba na výhru. Pokud uživatel nahraje již rozehranou hru, pak může pokračovat ve hře. Pokud uživatel nahraje již dohranou hru, pak se zobrazí ovládání přehrávání hry. Zde může uživatel spustit animaci tahů, procházet tahy krok po kroku nebo přeskočit na konec hry. Pokud uživatel spustí hru dvou umělých inteligencí, zobrazí se také ovládání přehrávání hry.

Když je uživatel ve hře, ovládá jí klikáním myši na jednotlivé čtverečky. Pokud hrají dva lidé, po každém tahu se střídají. Pokud hraje uživatel proti umělé inteligenci, po každém jeho tahy odehraje počítač. Uživatel může dále uložit již rozehranou hru do souboru. Tento soubor by měl jít přečíst člověkem. Uživatel také může kdykoliv vytvořit či nahrát novou hru. Na konci hry se objeví vyskakovací okno s výhercem.

# Třídy projektu



(čitelný obrázek je v souboru PiskvorkyUML.png)

## Třída Game

Tato třída má na starost základní herní logiku. Obsahuje odkazy na instance tříd Player (hráči), GamingArea (herní pole), Checker (zjišťuje, zda nenastala výhra), MoveHistory (uložené tahy), Saver (ukládání hry). Dále obsahuje metody, které využívají těchto tříd. Hlídá, jestli je tah možný a střídá hráče.

## Třída Player

Tato třída obsahuje informace o hráči – jeho jméno, značku, a zda je AI. Také obsahuje rozsáhlou metodu, která vrací souřadnice nejvhodnějšího tahu (umělá inteligence). Implementuje rozhraní Saveable.

## Třída GamingArea

Tato třída má na starost herní pole. Má metody na jeho inicializaci, označení jednotlivých čtverečků a vrací, zda je pole zaplněno. Samotné herní pole je tvořeno polí odkazů na instance třídy Square. Implementuje rozhraní Saveable.

## Třída Square

Tato třída reprezentuje jeden čtvereček. Mění a vrací jeho hodnotu (0 – prázdné, 1 – křížek, 2 – kolečko).

## Třída Checker

Tato třída obsahuje metodu na zjištění, zdali došlo k výhře. Implementuje rozhraní Saveable.

## Třída MoveHistory

Tato třída reprezentuje historii tahů, které byly zahrány. Obsahuje metody na přidání tahu a procházení historií. Implementuje rozhraní Saveable.

## Třída Move

Tato třída reprezentuje jeden tah. Obsahuje jeho souřadnice a značku.

## Třída Saver

Tato třída má za úkol uložit hru do souboru. Ukládá všechny instance tříd, které implementuj rozhraní Saveable. Také nahrává ze souboru vlastnosti těchto objektů.

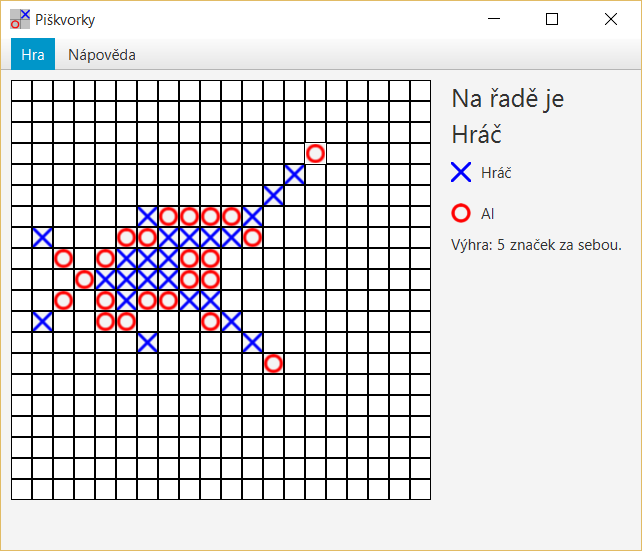
## Třída MainWindowController

Kontroler okna. Stará se o grafické rozhraní.

## Rozhraní Saveable

Toto rozhraní obsahuje hlavičky dvou metod. První má uložit do proměnné typu String všechny důležité vlastnosti a druhá je má z této proměnné načíst. Tyto metody jsou pak využívání třídou Saver na ukládání a načítání hry.

# Grafické rozhraní

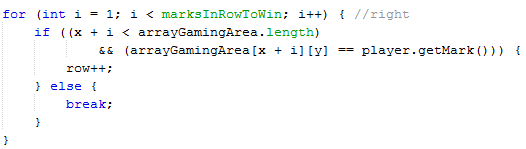


Nahoře se nachází menu, kde je možno spustit novou hru, uložit, načíst či ukončit hru. Zároveň je zde možné nechat si zobrazit nápovědu. Vpravo se nachází informace o konkrétní hře. Jména hráčů, kdo je zrovna na tahu, zda již nedošlo k výhře či úplnému zaplnění pole a kolik je třeba značek k ukončení hry. Vlevo dochází ke hraní samotné hry. Poslední označené políčko je zvýrazněno. Stejně zvýrazněné je i políčko, na kterém se zrovna nachází myš.

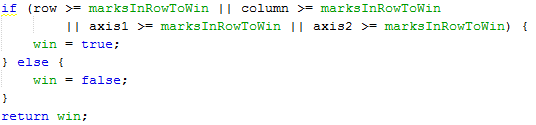
# Výběr některých metod

## Metoda boolean checkWin(int x, int y, Player player) třídy Checker

Tato metoda zjišťuje, zda již došlo k výhře. Zkoumá čtverečky kolem políčka, jehož souřadnice jsou v parametrech. Tato metoda je volána po každém tahu. Obsahuje 8 krát podobný cyklus, každý pro jiný směr. Ten vypadá nějak takhle:



V každém z těchto cyklů se zvýší určitá proměnná o 1, pokud se od zadaného políčka daným směrem opakují značky naposledy hrajícího hráče (také v parametrech). V tomto je to zrovna proměnná row, která představuje jeden řádek (směry vpravo a vlevo). Na konci této metody je podmínka:

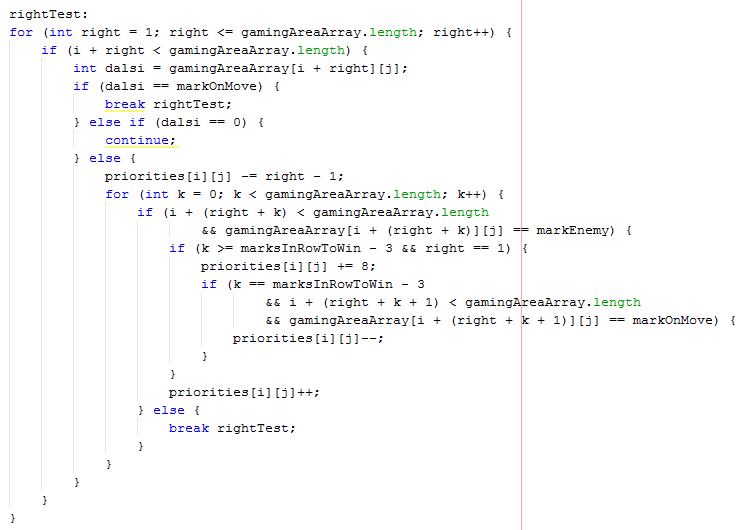


Pokud je některé z proměnných row, column, axis1, axis2 větší nebo rovna požadovanému počtu značek v řadě, metoda vrátí true. To znamená, že došlo k výhře.

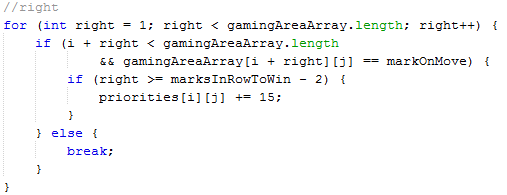
## Metoda int[] playAi(GamingArea gamingArea, int markOnMove, int marksInRowToWin) ze třídy Player

Tato metoda vrací souřadnice nejvhodnějšího tahu, na základě umělé inteligence. V parametrech dostane herní pole k analýze, značku hráče (AI) na tahu a počet značek za sebou potřebných k výhře. Metoda nastavuje prioritu každému políčku na základě tří věcí:

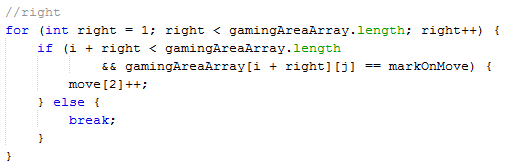
1. Obrana – nejdůležitější – počítá prioritu podle vzdálenosti nepřátelských značek a podle toho, kolik jich je za sebou. V případě, že má protihráč určitý počet značek za sebou (např. při hře na 5 značek na výhru jsou to 3), pak se priorita na blokování této řady rapidně zvýší. Toto je dokázáno osmi podobnými cykly (osm směrů):



1. Útok při šanci na výhru – Pokud by AI měla šanci na výhru (počet značek za sebou umělé inteligence je vhodný – např. při hře na 5 značek na výhru jsou to 3), tak se zvýší priorita na pokračování v této řadě, avšak ne o víc, než je u blokování soupeřovy výhry. Toto je zase dokázáno 8 podobnými cykly:



1. Nejbližší pole značkám AI – Pokud má více políček podobnou prioritu, pak se zvýší priorita těm políčkám, které jsou blíže značkám umělé inteligence (a je jich více za sebou). Toto má též na starost 8 podobných cyklů:



# Zhodnocení výsledku

Funkcionalita mého projektu je širší, než byla zadána. Do projektu jsem totiž implementoval možnost, hrát proti jednoduché umělé inteligenci. Tato AI se snaží především bránit protihráči, ale pokud má možnost jednoduše vyhrát (má za sebou označené určité množství políček), tak se této možnosti snaží využít.

Grafické prostředí a návrh tříd je trochu jiný, než jsem posílal v počáteční specifikaci mého projektu. V průběhu psaní mě totiž napadli některé věci, které by šly vytvořit lépe.

# Použité zdroje

1. Piškvorky. *Wikipedie: otevřená encyklopedie* [online]. [cit. 2016-05-14]. Dostupné z: <https://cs.wikipedia.org/wiki/Pi%C5%A1kvorky>
2. *Stackoverflow* [online]. Dostupné z: <https://stackoverflow.com/>
3. *Java Platform, Standard Edition (Java SE) 8* [online]. Dostupné z: <http://docs.oracle.com/javase/8/>