# Gymnázium, Praha 6, Arabská 14

Obor programování

ROČNÍKOVÝ PROJEKT

Oren Holiš 1.E

Šachový algoritmus

Květen 2021

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, Arabská14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V …. dne ……  Oren Holiš …….

Název práce: Algoritmus šachovnice

Autor: Oren Holiš

Abstract: Cílem projektu bylo, aby uživatel položil na šachovnici 2 figury dle svého výběru a ty hledaly co nejkratší cestu, aby se vyhodily. Tato cesta je zvýrazněna, aby uživatel poznal kudy figury šly a mohl to zkontrolovat. Na obrazovku je po vykreslení cesty vypsán počet kol nutných k vyhození figur.

# Obsah

# 1 Úvod

Tento dokument se zabývá řešením výpočtu nejkratší cesty dvou libovolných figur z šachů a její vykreslením.

## 1.1 Zadání

Uživatel položí na šachovnici dvě libovolné figurky a ty potom budou hledat nejkratší cestu k sobě, aby se vyhodily. Dostupné figurky budou pěšec, věž, dáma, král, kůň, střelec. Jako výstup budou dvě různě vybarvené cesty s políčkem kde se vyhodí.

# 2 Technologie a řešení aplikace

## 2.1 Technologie

Tato kapitola se zabývá použitými technologiemi, kterými byl projekt vytvořen.

### 2.1.1 Java

Jako vývojový jazyk byla použita java 1.8 (Oracle Corporation, 2021).

### 2.1.2 GUI[1]

Uživatelské rozhraní bylo vytvořeno v JavaFX 1.8 (Oracle Corporation, 2021). JavaFX byla použita pro svou snadnou implementaci a větší možnosti oproti java Swing.

## 2.2 Řešení aplikace

Aplikace je rozdělena na Obrazovku – kap. 3Chyba! Nenalezen zdroj odkazů. a Algoritmus – kap. 4. Uživatel ovládá panely na obrazovce a po položení dvou figur na šachovnici se spouští algoritmus, který nijak nezasahuje do Obrazovky neboli části viditelné uživateli. Až algoritmus dokončí výpočet tak Obrazovce předá informace o tom, co má vykreslit.

[1] GUI – Grafic user interface neboli uživatelské rozhraní

# 3 Obrazovka aplikace

Tato kapitola popisuje vnější vhled aplikace a použití jejích jednotlivých částí. Každá tato část je stručně popsána v jednotlivé kapitole. V aplikaci se nachází jediná obrazovka složená ze čtyř panelů, se kterými uživatel pracuje nebo je mu na nich zobrazována nápověda, popřípadě výsledek hry.

## 3.1 Výběr figurek

V horní části obrazovky nad šachovnicí se nachází panel s figurami. Tento panel obsahuje dohromady 12 figur. Tyto figury jsou vždy v černé a bílé variantě. Figurami jsou pěšec, dáma, kůň, věž, střelec a král. Každá figura se dá přesunout pomocí DND[2] na šachovnici.

Při přetahování se zobrazuje na levé části obrazovky pod tlačítkem vyčistit šachovnici její jméno, barva a nápověda, jak se figurka pohybuje. Na pravé části se zobrazí obrázek šachovnice s danou figurou a graficky se ukazuje, jak se figura pohybuje.

Po přesunutí dvou figur na šachovnici se automaticky spouští výpočet nejkratší cesty.

[2] Drag and drop – Systém přetahování a přesouvání položek pomocí myši.

## 3.2 Šachovnice

V prostřední části obrazovky se nachází panel s Šachovnicí. Pokud jsou na šachovnici přetaženy dvě figury tak se automaticky spustí výpočet jejich trasy k vyhození. Po konci výpočtu se na šachovnici zobrazí cesty obou figur.

- Cesta první figury je výrazněna modře
- Cesta druhé figury je zvýrazněna zeleně
- Políčko vyhození je zvýrazněno oranžově

Figury se dají z šachovnice odstranit zase pomocí DND[3] a po odstranění jedné z nich se automaticky vymaže jejich cesta. Nebo se dá celá šachovnice vyresetovat pomocí tlačítka v Informačním panelu kap. 3.3.

## 3.3 Informační panel s ovládacími prvky

Na levé části obrazovky se nachází panel s ovládacími a informačními prvky.

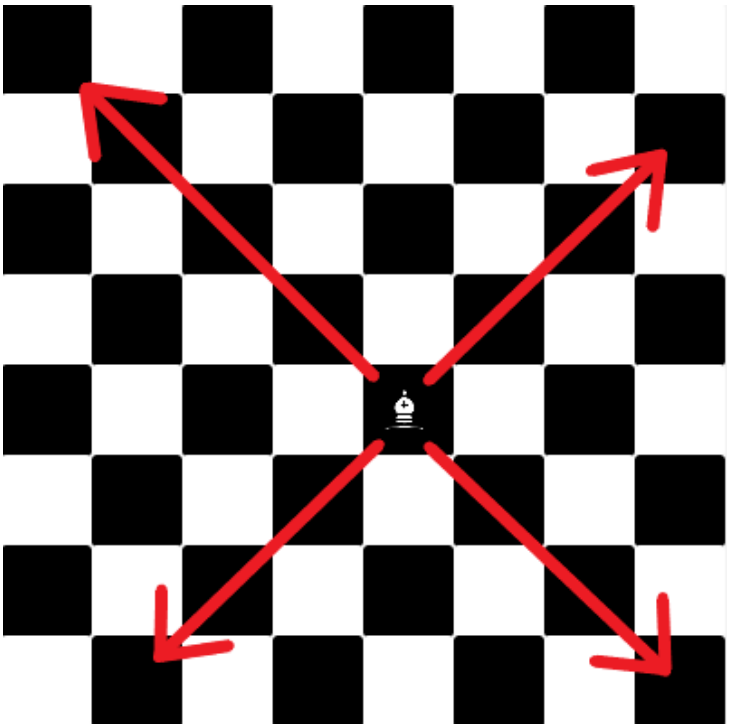
- Tlačítko pro vyresetování šachovnice dá figury na svá původní místa a vyčistí šachovnici od vykreslených výsledků.
- Informační prvek pro sdělení počtu potřebných kol k tomu, aby se figury vyhodili.
- Informační prvek o tom, jakou figurou je pohybováno a jaká je její barva se zobrazuje při přetahování figury.

[3] Drag and drop – Systém přetahování a přesouvání položek pomocí myši.

- Informační prvek zobrazující nápovědu při přetahování figur. Ukazuje, jak se jednotlivé figury pohybují například „Kůň je figurka, která jde o dvě pole vpřed a poté o jedno vlevo.“ Tato nápověda je pak graficky ukázána uživateli na Panelu s grafickou ukázkou pohybu figur kap. 3.4.

## 3.4 Panel s grafickou ukázkou pohybu figur

Na pravé části obrazovky se nachází panel, kde si při přetahování figury uživateli zobrazuje šachovnice a na ní přetahovaná figura. U této figury je šipkami ukázáno, jak se figura pohybuje. Jako například u střelce:



*Obrázek 1 - Pohyby figury střelec po šachovnici*

# 4 Algoritmus

Výpočet nejkratší cesty a její vykreslení je počítáno pomocí algoritmu. Tento algoritmus pracuje s virtuální šachovnicí pro vyšší rychlost a nižší nároky na výkon. Po vypočítání cesty je tato cesta vykreslena na šachovnici.

## 4.1 Virtuální šachovnice

Po spuštění algoritmu je vždy vygenerována nová virtuální šachovnice. Do této šachovnice se ukládají veškerá výpočetní data jako například návštěvy figur, na kolikátý tah tam figury vstoupili, která figura to byla a z jakého pole tam figura vstoupila.

## 4.2 Výpočet kol

Výpočet počtu kol se skládá z pěti hlavních opakujících se funkcí. V pěti následujících kapitolách vám popíši, jak každá z nich funguje a co dělá. Tyto kapitoly jsou pojmenovány stejně jako funkce pro jejich přehlednost.

### 4.2.1 Count figure moves

Tato funkce spouští a ukončuje celý výpočet. V této funkci je while loop[4], který se opakuje, dokud se může figura jedna i dva mají dostupná pole kam se mohou pohnout. Po vyčerpání všech dostupných polí nebo nalezení políčka vyhození se výpočet ukončuje a Informační prvek s počtem kol potřebných k vyhození figur kap. 3.3 oznámí uživateli, že

[4] While loop – cyklus opakující se dokud je platná podmínka

zvolené figury se nikdy nevyhodí nebo mu oznámí přesný počet kol potřebných k vyhození.

### 4.2.2 Go throw visited fields

Tato funkce prochází všechna dostupná pole figurou a opakuje to, dokud neprojde všechny. V situaci, kdy se figury potkají tak se před započnutím dalšího opakování ukončuje.

### 4.2.3 Calculate available fields for figure

Tato funkce vypočítává všechna dostupná pole pro figuru na základě jejího typu. O rychlý výpočet, kam na šachovnici se mohou figury pohnout se stará funkce blíže popsaná v kap. Optimalizace, která výrazně zrychluje a usnadňuje výpočet.

### 4.2.4 Mark field figure entered

Tato funkce označuje vstup figury. Na dané políčko zapíše, že bylo navštíveno, kterou figurou a z jakého předchozího pole bylo navštíveno.

### 4.2.5 Evaluate figure ways cross

Vyhodnocuje setkání figur na daném políčku, pokud se figury potkají tak se zapíše, která figura udělala setkání a zapíší se na políčko její informace. Poté se spustí výpočet předchozích polí z políčka setkání, aby bylo možné vykreslení cesty na šachovnici – kap. 3.2, po dokončení výpočtu cesty se a ukončuje se celý algoritmus.

## 4.3 Výpočet projitých polí

Toto je poslední funkce, která se spouští a spouští se pouze tehdy kdy je nalezeno pole vyhození. Tato funkce z pole vyhození prochází cesty první a druhé figury a ukládá si jejich cestu. Tato cesta je znázorněna jako ArrayList of Array[5] a je použito jako soubor souřadnic pro vykreslení výsledku na šachovnici.

## 4.4 Testy

K algoritmu jsou napsány testy pro většinu možných kombinací figur. Implementace těchto testů je velmi jednoduchá a díky nim se dá algoritmus velmi snadno rozšiřovat a zároveň kontrolovat, že nebyla vytvořena žádná chyba.

## 4.5 Optimalizace

Funkce calculateFigureAvailableFields se stará o veškeré obecné informace o pohybu figur. Tak u figur střelece, věže a nebo kombinace věžě a střelce dáma mohou vypočítává kam se mohou pohnout. To výrazně urychluje výpočet, protože se u těchto figur nemusí vyhodnocovat zda by při přičtění tohoto obecného tahu nepohnuly mimo šachovnici.

[5] ArrayList of Array – 2D pole polí

# 4.6 Výhody

Výhody algoritmu jsou, velmi vysoká rychlost díky práci s virtuální šachovnicí, algoritmus by fungoval na libovolně velkou šachovnici a je obecně použitelný na libovolné figury, které se hýbou předem definovaným způsobem. Takže pokud by bylo třeba implementovat novou figuru tak pokud a nebude mít volitelná pole jako pěšec, který se hýbe šikmo do stran pouze v případě vyhození. Tak stačí do aplikace do přidat obecné tahy figury jako například obecné tahy krále:

```
case "king":
    createPosition( x: +1,  y: +0, availableFields);
    createPosition( x: +0,  y: +1, availableFields);
    createPosition( x: -1,  y: +0, availableFields);
    createPosition( x: +0,  y: -1, availableFields);
    createPosition( x: +1,  y: +1, availableFields);
    createPosition( x: +1,  y: -1, availableFields);
    createPosition( x: -1,  y: +1, availableFields);
    createPosition( x: -1,  y: -1, availableFields);
    break;
```

*Obrázek 2 - Obecné tahy figury král*

# 5 Instalace

Instalace aplikace je velmi jednoduchá na githubu se v projektu sekci code nachází v pravé části panel releases. Kliknete na poslední release a vyberete si formát zip nebo gz a stáhnete ho. Po stažení soubor extrahujete a aplikaci spustíte přes soubor Sachovnice1.jar. Ke spuštění je nutná java verze 1.8.

# 6 Závěr

Vytvořená aplikace se vyvinula velmi dobře. Splňuje zadání - 1.1 a má až nad očekávání velmi dobře vyvinutý algoritmus. Největším problémem byla figura pěšce, která má velmi specifické chování a pohyb po šachovnici.

Ve vývoji se kromě vytvoření správného algoritmu pro pěšce nevyskytovali problémy, a tudíž jsem se svou aplikací velmi spokojen. Jediné, co by se ještě dalo vylepšit je grafický design aplikace.

# Bibliografie

Oracle Corporation. (2021). *https://docs.oracle.com/javase/8/docs/api/*. Načteno z Java 1.8 documentation.

Oracle Corporation. (2021). *https://docs.oracle.com/javase/8/javafx/api/toc.htm*. Načteno z JavaFX 1.8 documentation.

*Schachfiguren -Aufgabe: Ideen für didaktisches Design mit H5P entwickeln*. (25. listopad 2020). Načteno z https://peter.baumgartner.name/: https://i0.wp.com/peter.baumgartner.name/wp-content/uploads/2020/11/1920px-Chess_Pieces_Sprite.svg-min.png

Wikipedia Foundation, Inc. (2021). *https://cs.wikipedia.org/wiki/%C5%A0achy*. Načteno z Šachy.