**Gymnázium, Praha 6, Arabská 14**

Obor programování

TODO obrázek

Ročníkový projekt

Oren Holiš 1.E

Šachový algoritmus //TODO

Květen 2021

//TODO odsazeníProhlašuji, 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 tahů nutných k vyhození figur.

Klíčová slova: TODO

Obsah

[Obsah 1](#_Toc70148422)

[1. Úvod 2](#_Toc70148423)

[1.1 Zadání 2](#_Toc70148424)

[2. Technologie 3](#_Toc70148425)

[2.1 Technologie 3](#_Toc70148426)

[2.1.1 GUI 3](#_Toc70148427)

[2.1.2 IDE 3](#_Toc70148428)

[3. Aplikace 4](#_Toc70148429)

[3.1 Obrazovka 4](#_Toc70148430)

[3.1.1 Výběr figurek 4](#_Toc70148431)

[3.1.2 Šachovnice 5](#_Toc70148432)

[3.1.3 Informační panel s ovládacími prvky 5](#_Toc70148433)

[3.1.4 Panel s grafickou ukázkou pohybu figur 6](#_Toc70148434)

[4. Algoritmus 7](#_Toc70148435)

[4.1 Virtuální šachovnice 7](#_Toc70148436)

[4.2 Výpočet tahů 7](#_Toc70148437)

[4.2.1 Count figure moves 7](#_Toc70148438)

[4.2.2 Go throw visited fields 8](#_Toc70148439)

[4.2.3 Calculate available fields for figure 8](#_Toc70148440)

[4.2.4 Mark field figure entered 8](#_Toc70148441)

[4.2.5 Evaluate figure ways cross 8](#_Toc70148442)

[4.3 Výpočet projitých polí 9](#_Toc70148443)

[4.4 Optimalizace TODO 9](#_Toc70148444)

[4.5 Výhody 9](#_Toc70148445)

[Závěr 10](#_Toc70148446)

[Bibliografie 11](#_Toc70148447)

1. Úvod

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

* 1. Zadání

Uživatel položí na šachovnici 2 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 2 různě vybarvené cesty s políčkem kde se vyhodí.

1. Technologie
   1. Technologie

Tato kapitola se zabývá použitými technologiemi a vývojářským prostředím, ve kterém byl projekt vytvořen.

* + 1. Java

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

* + 1. GUI

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

* + 1. IDE

Jakožto IDE (TODO vysvětlivka) bylo zvoleno InteliJ od JetBrains (TODO odkazy) kvůli pohodlnému užití a velmi pokročilím kotrolám kódu od IDE.

1. Aplikace

Tato kapitola popisuje vnější vhled aplikace a použití jejích jednotlivých částí. Každá tato část je stručně popsána v jednotlivé kapitole.

* 1. Obrazovka

V aplikaci se nachází jediná obrazovka složená ze 4 panelů, se kterými uživatel pracuje nebo je mu na nich zobrazována nápověda popřípadě výsledek hry.

* + 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. Figury: pěšec, dáma, kůň, věž, střelec a král. (TODO odkazy na figury na konec a obrázky) Každá figura se dá přesunout pomocí DND[[1]](#footnote-1) 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.

* + 1. Šachovnice

V prostřední části obrazovky se nachází panel s Šachovnicí. Pokud jsou na šachovnici přetaženy 2 figury tak se automaticky spustí výpočet jejich trasy k vyhození. Po konci výpočtu se na šachovnici zobrazí cesty obou figur.

* Cesta figury 1 je výrazněna modře
* Cesta figury 2 je zvýrazněna zeleně
* Políčko vyhození je zvýrazněno oranžově

Figury se dají z šachovnice odstranit zase pomocí DND[[2]](#footnote-2) a po odstranění jedné z nich se automaticky vymaže jejich cesta.

* + 1. 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
* informační prvek pro sdělení počtu tahů k vyhození figur
* informační prvek pro nápovědu, která se chytře zobrazuje při tahání figurek, položení figurek a před uděláním jakékoli akce
* informační prvek o tom jakou figurou je pohybováno a jaká je její barva
  + 1. 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.

1. 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.

* 1. Virtuální šachovnice

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

* 1. Výpočet tahů

Výpočet počtu tahů se skládá z 5 hlavních opakujících se funkcí v 5 následujících kapitolách vám popíši jak každá z nich funguje a co dělá. Tyto kapitoly jsou pojmenovány stejně jako funkce pro jejich přehlednost.

* + 1. Count figure moves

Tato funkce spouští a ukončuje celý výpočet. V této funkci je while loop[[3]](#footnote-3), který se opakuje dokud se může figura 1 i 2 mají dostupná pole kam se mohou pohnout. Po vyčerpání všech dostupných polí nebo nalezení políčka vyhození se výpočet ukončuje.

* + 1. 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.

* + 1. Calculate available fields for figure

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

* + 1. 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. (todo odkaz na výpočet …)

* + 1. Evaluate figure ways cross

Vyhodnocuje setkání figur na daném políčku pokud se figury potkají tak se spustí výpočet předchozích polí pro vykreslení cesty na šachovnici - 3.1.2, uloží se pole setkání a ukončuje se celý algoritmus.

* 1. 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 1 a 2 figury a ukládá si jejich cestu. Tato cesta je znázorněna jako ArrayList of Array [[4]](#footnote-4) a je použito jako soubor souřadnic pro vykreslení výsledku na šachovnici.

* 1. Optimalizace TODO
  2. Výhody

Obsah obrázku text, elektronika

Popis byl vytvořen automatickyVýhody algoritmu jsou, že je obecně použitelný na veškeré 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 přidat obecné tahy figur jako například král:

Závěr

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.

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

1. Drag and drop – Systém přetahování a přesouvání položek pomocí myši. [↑](#footnote-ref-1)
2. Drag and drop – Systém přetahování a přesouvání položek pomocí myši. [↑](#footnote-ref-2)
3. While loop – cyklus opakující se dokud je platná podmínka [↑](#footnote-ref-3)
4. ArrayList of Array – 2D pole polí [↑](#footnote-ref-4)