

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Počítačový hráč šachu



2021

Vedoucí práce: Mgr. Tomáš Kühn,
Ph.D.

Vojtěch Matras

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Vojtěch Matras
Název práce: Počítačový hráč šachu
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2021
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Tomáš Kühn, Ph.D.
Počet stran: 29
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Vojtěch Matras
Title: Computer Chess Player
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2021
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Tomáš Kühn, Ph.D.
Page count: 29
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

Program Chess umožňuje uživateli hrát partii proti počítači využívající algoritmus alpha-beta, zároveň si uživatel může nastavovat, do jaké hloubky má počítačový hráč propočítávat tahy. V aplikaci je také možné zkusit si nastavit vlastní ohodnocovací hodnoty pro jednotlivé figurky a porovnávat jednotlivé ohodnocovací funkce proti sobě v partii dvou počítačů. V neposlední řadě je v programu možné sehrát i partii dvou lidských hráčů.

Synopsis

Chess allows the user to play a game against the computer using the alpha-beta algorithm, while the user can set the depth to which the computer player should calculate moves. In the application, it is also possible to try to set your own evaluation values for individual pieces and compare individual evaluation functions against each other in a game of two computers. Last but not least, it is possible to play a game of two human players in the program.

Klíčová slova: šachy; logická hra; minimax; alfa-beta; C#

Keywords: chess; logic game; minimax; alpha-beta; C#

Děkuji Mgr. Tomáši Kührovi, Ph.D. za vedení, ochotu a pomoc při zpracování této práce. Také děkuji rodině a svým přátelům za podporu po celou dobu studia.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1 Úvod	7
1.1 Historie šachu	7
1.2 Šachová pravidla	7
1.3 O bakalářské práci	10
2 Počítačový šach	12
2.1 Proč zrovna šachy?	12
2.2 Historie počítačového šachu	12
3 Použité technologie	14
3.1 C#	14
3.2 WPF	14
4 Programátorská dokumentace	15
4.1 Computer Player	15
4.2 Game	15
4.3 Move	15
4.4 PgnNotation	15
4.5 Pieces	16
4.6 Resources	16
4.7 Tools	16
4.8 Nezařazené soubory	16
4.9 Šachové algoritmy	17
4.10 Porovnání ohodnocovacích funkcí	18
4.11 Možné rozšíření programu	19
5 Uživatelská dokumentace	20
5.1 Hlavní menu	20
5.2 Šachovnice a její ovládání	21
5.3 Partie dvou lidských hráčů	22
5.4 Partie člověka proti počítači	22
5.5 Partie dvou počítačových hráčů	22
5.6 Odehrání více partií počítačových hráčů	23
5.7 Ukládání a přehrávání odehraných partií	23
5.8 Nastavení vlastních ohodnocovacích hodnot	23
Závěr	25
Conclusions	26
A Obsah příloženého CD/DVD	27
Literatura	28

Seznam obrázků

1	Počáteční postavení.	8
2	Pohyb jezdce.	9
3	Braní mimo chodem.	10
4	Turek. Zdroj: [8]	13
5	Minimaxový vyhledávací strom.	17
6	Vyhledávací strom s Beta-odsekáváním.	18
7	Hlavní menu.	20
8	Šachovnice v průběhu partie.	21
9	Vytváření vlastní ohodnocovací funkce.	23

1 Úvod

1.1 Historie šachu

Podle legendy šachy [1] vynalezl před více jak 2 000 lety moudrý učenec Sissa ben Dahir. V Indii tehdy vládl mladý král Šahram, jeho vláda ale nebyla zdaleka ideální, tak se učenec rozhodl s tím něco udělat. Vynalezl právě šachy, díky kterým panovníkovi ukázal, že i ta nejmenší figurka ve hře, pěšec, je důležitá pro výhru hráče. Pěšci symbolizují obyčejný lid a i s ním bylo nutné podle Sissy ben Dahira počítat. Králi se hra velmi zalíbila a dokonce se z ní i poučil. Moudrého učence chtěl proto odměnit a zeptal se ho na jeho přání. Sissa si přál pouze zrnka rýže. Na první políčko šachovnice jedno zrnko a na každé další políčko dvojnásobek. Královi se to zdálo příliš skromné, ale učenec trval na svém. O to víc byl král překvapen, když zjistil, že přání nemůže vyhovět. Ukázalo se totiž, že chytrý Sissa si řekl o osmnáct trilionů zrněk rýže. Matematikové Šahramovi vysvětlili, že by Sissovi musel dát úrodu za 10 let po celém povrchu země včetně moře.

To je konec legendy, skutečnost bude pravděpodobně trochu jiná. Šachy [2] nejspíše vznikly v indické Guptovské říši v 6. století. Název šachy je odvozen ze sanskrtského čaturanga, to zahrnuje čtyři základní součásti tehdejšího vojska, mezi které patřila pěchota, jízda, sloni a válečné vozy. To představuje dnešní pěšce, jezdce, střelce a věže.

V Persii byla hra upravena a přejmenována na šatrandž. [2] Postupně se dostala do Číny a muslimských zemí. V 9. století se šachy dostaly i do Evropy a poměrně rychle se staly známou hrou. Pravidla se neustále vyvíjela, až konečně kolem roku 1475 získala hra podobu, kterou, až na pár drobných odchylek, známe dodnes. Hra neustále získávala větší a větší popularitu. Začala vznikat teorie, ta se zapisovala do knih a v 19. století se začaly hrát i zápasy a pořádat turnaje.

První zmínka o šachách na českém území pochází z 12. století.[2] Ve svatovojtěšské legendě jsou latinsky popsána pravidla a historie. Ze známých panovníků hráli rádi šachy například Jan Lucemburský nebo Jiří z Poděbrad. Nadšeným šachistou byl také Jan Ámos Komenský. Na konci 19. století vznikl první český tým Český spolek šachový a začal také vycházet první časopis zabývající se šachovou problematikou. Na začátku minulého století se i u nás začaly pořádat různé šachové soutěže.

1.2 Šachová pravidla

Šachy [3] nejsou hrou, která by byla náročná na vybavení. Potřebujeme desku, jíž se říká šachovnice, 32 figurek a pro závodní partii šachové hodiny. Pro jistotu ještě doplním, že v nejběžnější formě se jedná o hru pro dva hráče.

Šachovnice ve tvaru čtverce se skládá ze 64 políček. Sloupce se označují písmeny a–h a řady číslicemi 1–8. Jednotlivá políčka na šachovnici mají buď černou nebo bílou barvu. Pole, která spolu sousedí vertikálně nebo horizontálně,

mají vždy odlišnou barvu. Pole, jež spolu sousedí diagonálně, mají barvu stejnou. Podle pravidel je známo, že pole a1 má černou barvu. Díky tomu nemohou vzniknout neshody jak správně natočit neoznačenou šachovnici.

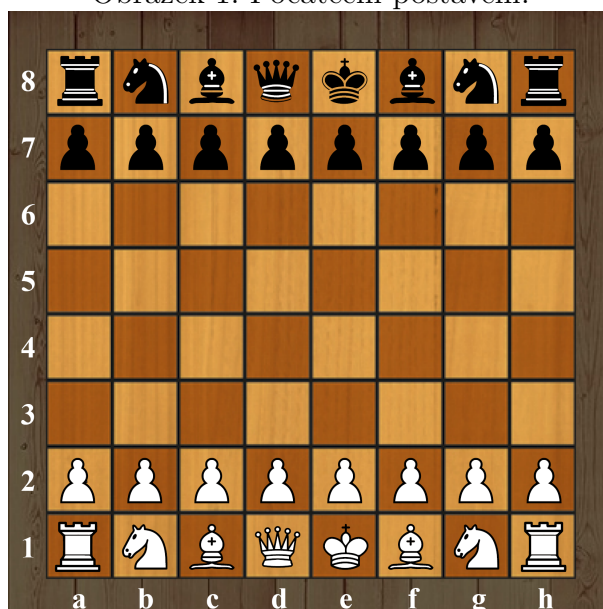
Před začátkem partie hráči buď losem nebo domluvou určí, kdo bude mít bílé a kdo černé figurky. Bílý hráč má své figurky v počátečním postavení na prvních dvou řadách. Černý zpočátku okupuje sedmou a osmou řadu. Každý hráč má 8 pěšců, 2 jezdce, 2 střelce, 2 věže, 1 dámu a 1 krále. Před partií je jasné dáno, jak jsou jednotlivé figurky rozmístěné. Počáteční postavení můžete vidět na obrázku 1. (Ikony šachových figur pro všechny obrázky z této kapitoly jsou ze zde uvedeného zdroje. [4])

Hru začíná bílý. Hráč vždy provede jeden tah a poté následuje tah protihráče. Na každém políčku šachovnice může být maximálně jedna figurka. Pokud figurka jedné barvy vstoupí na pole, kde je figurka druhé barvy, je figurka druhé barvy odstraněna ze hry. Každý typ figurky má přesně stanovený způsob, jak se může pohybovat.

Pěšec se pohybuje pouze dopředu. Ze základního postavení může o jedno nebo dvě políčka. Z jakékoliv jiné pozice může pouze o jedno políčko. Pěšec je zároveň jediná figurka ve hře, která bere soupeřovu figurku jiným způsobem, než je její normální pohyb. Soupeřovu figurku bere vždy pouze o jedno políčko diagonálně dopředu.

Věž se může pohybovat všemi směry kromě šikmého. Věže společně se střelci, dámou a králem patří mezi figurky, jež se v prvním tahu nemohou pohnout ze svého počátečního postavení.

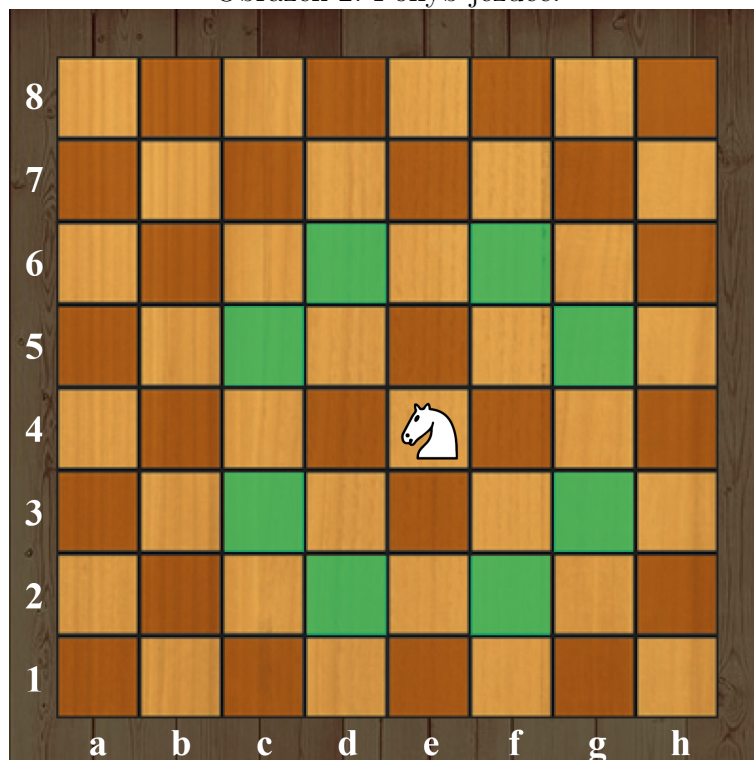
Obrázek 1: Počáteční postavení.



Jezdec je jediný kámen na šachovnici, který může přeskakovat jiné figurky. Jeho skok připomíná písmeno L a skočí vždy z bílého pole na černé a naopak.

Jezdec se pohybuje buď o dvě pole do strany a jedno pole vertikálně, nebo o jedno pole do strany a dvě pole vertikálně. Nejlépe je to vidět na obrázku 2.

Obrázek 2: Pohyb jezdce.



Další figurkou je střelec, ten se po šachovnici pohybuje pouze šikmo. Proto se střelec, jenž se na začátku partie položí na černé políčko, na pole bílé nikdy nedostane.

Dáma se může pohybovat všemi možnými směry, proto je to nejcennější útočná figura. Král může pro pohyb využívat taky všechny směry ale vždy pouze o jedno políčko. Král je jediná figurka ve hře, která nemůže být sebrána.

V partii mohou nastat tři situace, kterým každý šachista musí rozumět. Jedná se o šach, pat a mat.

Šach nastane, pokud hráč ohrozí protivníkova krále tak, že by mu ho mohl v dalším tahu sebrat. Druhý hráč musí na tuto skutečnost zareagovat, protože po odehrání svého tahu nikdy nemůže zůstat jeho král v šachu. Pokud neexistuje možnost, jak šach odvrátit, tedy po jakémkoliv tahu je král druhého hráče stále v šachu, znamená to, že se jedná o mat, což je zároveň cílem hry, a první hráč vyhrál. Pat nastane, pokud hráč sice nemá krále v šachu, ale nemůže zahrát žádný tah, buď proto, že mu to nedovolují pravidla pro pohyb ostatních figurek, nebo proto, že by se každým jiným tahem dostal pod šach. Pat znamená, že partie skončí remízou, i když jeden z hráčů může mít i výraznou materiální převahu.

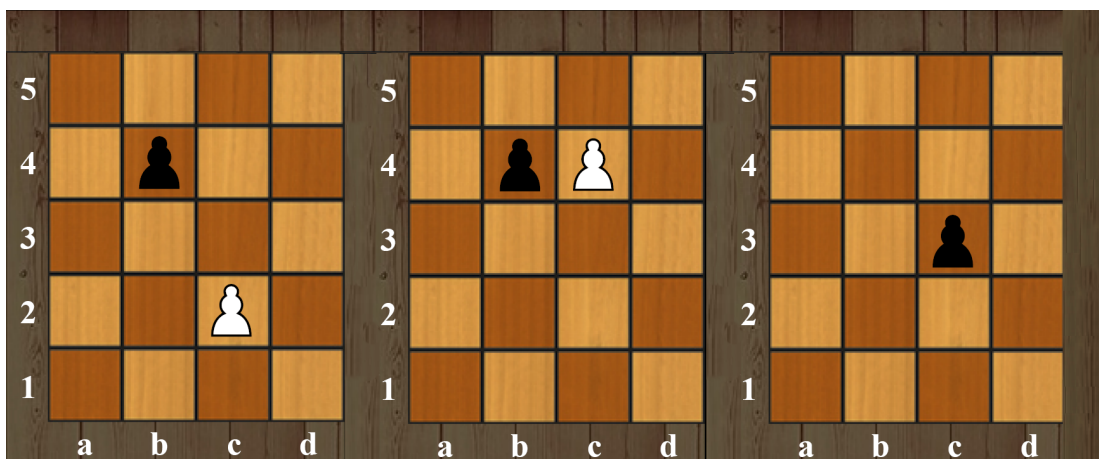
K tomu, abychom měli potřebný základ pro hraní šachů, je potřeba znát ještě tři další způsoby, jak provést tah.

Prvním speciálním tahem je rošáda, společný tah krále a jedné z věží. Důležitým pravidlem je to, že rošádu lze provést pouze v případě, jestli král a věž, se kterou rošádu provádíme, neopustily počáteční postavení. Jsou dvě možnosti rošády, dlouhá a krátká. Krátká rošáda se provede tak, že král se ze sloupce e dostane na sloupec g a věž ze sloupce h na sloupec f. Při dlouhé rošádě král táhne ze sloupce e na sloupec c a věž ze sloupce a na pole d. Důležité je také to, že rošádu nemůžeme zahájit, pokud je král v šachu a také pokud by král přešel nebo skončil na poli, kde by byl v šachu.

Druhou možností je proměna pěšce za jinou figuru. Pokud pěšec dojde na poslední řadu, tak si hráč musí vybrat jednu ze čtyř možností, na co se pěšec promění. Možnosti jsou jezdec, střelec, věž a dáma.

Poslední speciální tah je braní mimochodem. Předpokládejme, že bílý pěšec z počátečního postavení postoupil o dvě pole, například z a2 na a4, pak pokud má černý hráč hned na vedlejším poli, tedy na b4, svého pěšce, může hned v dalším tahu (později už ne) uskutečnit braní mimochodem. To se provede tak, že černý odstraní bílého pěšce z pole a4 a svého pěšce postaví na pole b3. Přehledně to je ukázáno na obrázku 3.

Obrázek 3: Braní mimochodem.



1.3 O bakalářské práci

Tato bakalářská práce se věnuje počítačovému šachu. Primárním cílem bylo vytvoření aplikace, kde uživatel bude moci porovnávat sílu jednotlivých algoritmů. Zároveň bylo zamýšleno, aby si uživatel mohl nastavit vlastní ohodnocení jednotlivých herních situací. Své vlastní ohodnocení by pak mohl testovat proti defaultnímu ohodnocení přímo v aplikaci. Program by dále měl umožňovat partii přímo proti jak defaultnímu počítači, tak právě vlastnímu nastavení. Aplikace by také měla umožňovat partii dvou lidských hráčů.

V prvních kapitolách jsou popsány samotné šachy a historie počítačového šachu. Následuje část o technologiích, jež jsem ke svému programu využíval. Poté již přichází hlavní části bakalářské práce, tedy programátorská dokumentace a uživatelská dokumentace.

2 Počítačový šach

2.1 Proč zrovna šachy?

Šachy [6] se od počátku umělé inteligence jevíly jako ideální objekt k programování a to z několika různých důvodů. Nejdůležitějším předpokladem je, že šachy disponují pevně ohraničenou množinou objektů (pole, figurky, tahy), i přesto, se ale jedná o hru, která má nepřehledné množství možností.

Mezi další důvody patří fakt, že pro počítač je reprezentace tahů a i jejich grafická podoba poměrně snadno a dobře uchopitelná. Šachová partie se dá také jednoduše zaznamenat, proto lze odehrané partie dobře analyzovat a poučit se tak z chyb, které program dělá. Díky tomu programátor může svůj projekt neustále zdokonalovat.

Královská hra je sport [5], tudíž jednotliví programátoři (nebo celé skupiny programátorů) můžou porovnat svoje programy a hodnotit, který je lepší. V neposlední řadě je důležitý i obchodní důvod. Šachy [6] hraje obrovské množství lidí a jak závodník, tak rekreační hráč v šachovém programu nalezne protihráče, nebo i pomocníka k vlastnímu rozvoji, i při analýze vlastních partií.

2.2 Historie počítačového šachu

Za první šachový stroj byl dlouho považován tzv. Turek. [7] Zařízení, které vytvořil v roce 1771 Wolfgang von Kempelen, bylo tvořeno skříní se šachovnicí, za kterou „sedělo“ torzo lidského těla, jak můžeme vidět na obrázku 4. V levé části obrázku je vidět rekonstrukce Turka, která je umístěna v Los Angeles. Autorem je John Gaughan. V pravé části je rytina uvedena v knize Karla Gottlieba. Jelikož postava měla turecké oblečení a turban začalo se stroji říkat Turek. Ruka Turka dokázala ovládat figurky, a to dokonce velmi dobře. Stroj dosahoval velmi dobrých výsledků. Díky tomu, že Kempelen otevíral před partií různá dvířka, jimiž se dalo nahlížet do skříně, tak se zdálo, že ve skříní není nikdo schován a jedná se skutečně o stroj, který dovede hrát šachy. V roce 1854 při požáru byl ovšem Turek zničen a následně odhaleno tajemství, že se ve skříní dokázal ukrýt člověk a i přes postupné otvírání dvířek se ho nepodařilo odhalit.

První pravý šachový stroj [6] byl zkonstruován v roce 1890 jistým Torres y Quevedom. Tento Španěl byl specialistou na elektromechanické řídicí systémy. Svoje vědomosti dokázal přetavit ve stroj, který zvládl sehrát koncovku krále a věže proti věži.

V polovině dvacátého století učinil velký pokrok slavný matematik Alan Turing. Vytvořil „Papírový stroj“ [6], který byl v podstatě plnohodnotný šachový program, bohužel v té době neexistoval takový počítač, který by mohl program vykonat. Turing všem figurkám dal hodnotu, prověřil všechny tahy bílého a protitahy černého a na základě Turingem vytvořené ohodnocovací funkce, „Papírový stroj“ vždy vybral podle něj nejlepší možný tah. V roce 1952 se uskutečnila partie mezi „Papírovým strojem“ a rekreačním hráčem Alickem Glenniem. Papírový stroj sice prohrál, i tak se ale jednalo o velký progres v této problematice.

Pouze o 4 roky později se na volně programovatelném počítači [6] uskutečnila první skutečná partie počítače proti člověku. Hrál se tehdy na zmenšené šachovnici 6x6 bez střelců. K tomuto okleštění se autor rozhodl pro zjednodušení hry, díky kterému mohl program rychleji vykonávat své tahy. Zbylá šachová pravidla fungovala beze změny. Program MANIAC [6], jehož hlavním tvůrcem byl John van Neumann, dokonce v jedné ze své prvních partií zvítězil nad lidským protivníkem, jenž byl sice začátečník, ale i tak se jednalo o významný milník.

Obrázek 4: Turek. Zdroj: [8]



Počítače brzy dokázaly hrát na normální šachovnici a jejich tvůrci je postupně zdokonalovali. [6] Také se časem začaly konat různé soutěže, kde se jednotlivé programy proti sobě utkávaly. Tato soutěživost postupně hnala tvůrce dopředu, zároveň snad všichni programátoři snili o tom, že jednou dokáže jejich program porazit i mistra světa v šachu. Díky stále se zlepšujícímu hardware a neutu-
chající snaze programátorů tento den skutečně nastal. V květnu v roce 1997 program Deep Blue porazil tehdejšího mistra světa Garryho Kasparova 3,5-2,5. [9] V dnešní době už jsou počítače o hodně dál a ani ten nejlepší hráč na světě proti nim nemá sebemenší šanci. [10]

3 Použité technologie

3.1 C#

C# [11, 12] vznikl v roce 2000, jeho vývojářem je společnost Microsoft. C# je vysokoúrovňový a typově bezpečný jazyk. Patří také jako jazyky Java a C++ do rodiny objektově orientovaných jazyků. Jedná se o jazyk, jenž čerpá svoji syntaxi z jazyku C. Je mimo jiné vhodný k tvorbě databázových systémů, webových aplikací a formulářových aplikací ve Windows. Aplikace naprogramované v jazyce C# jsou spuštěné na platformě .NET

Jelikož je C# objektově orientovaný jazyk, tak podporuje základní rysy těchto programovacích jazyků, mezi které patří zapouzdření, polymorfismus a dědičnost. C# neobsahuje vícenásobnou dědičnost, tudíž každá třída může být potomkem pouze jedné třídy. Absence vícenásobné dědičnosti je nahrazena rozhraním, kterých lze implementovat v každé třídě libovolný počet.

V C# neexistují globální proměnné ani metody. Náhradou za ně jsou statické metody a proměnné veřejných tříd. Jedná se o case-sensitive jazyk, tedy rozlišuje malá a velká písmena.

3.2 WPF

Windows Presentation Foundation (Zkráceně WPF) [13, 14] je knihovna tříd pro tvorbu grafického rozhraní. Součástí .NET frameworku je od verze 3.0, tedy od listopadu 2006. WPF je nástupcem Windows Forms.

Mezi klíčové vlastnosti WPF patří:

1. Grafika: Funguje pomocí Direct3D knihoven. Je možné využít i vektorovou grafiku.
2. Podpora starších knihoven: Je možná spolupráce s Windows Forms i Windows API
3. Práce s médii: Podpora nejpopulárnějších obrazových formátů. Také podpora videoformátů WMV, MPEG a AVI.

4 Programátorská dokumentace

Celý projekt je pro větší přehlednost a možnosti rychlejší práce rozdělen do jednotlivých složek. Ty jsou popsány v abecedním pořadí. Zde je popis nejdůležitějších součástí programu.

4.1 Computer Player

- `ComputerPlayer` – Abstraktní třída ze které prozatím dědí pouze třída `ComputerPlayerAlpha`. Při případném rozšíření programu je ale možné přidat další algoritmy.
- `ComputerPlayerAlpha` – Třída reprezentující Alpha-beta algoritmus implementovaný na základě tohoto zdroje. [15]
- `ValuesForPieceType` – Třída, která předává algoritmu hodnoty figurek.
- `ValuesForPiecesPositions` – Třída, která předává algoritmu hodnoty figurek závislé na jejich dané pozici.

4.2 Game

- `Game` – Hlavní třída celého programu, jež má za úkol řídit celý program.
- `Render` – Třída dědí ze třídy `Grid` a stará se především o vykreslení šachovnice.
- `Tile` – Třída dědí ze třídy `Frame` a slouží hlavně k barevné reprezentaci políčka při určitých herních situacích. Například označení figurky, šach a další.
- `Turn` – Třída jejímž hlavním úkolem je střídání tahů mezi protihráči a informování o počtu odehraných tahů.

4.3 Move

Tato složka obsahuje struktury a výčtový typ pro reprezentaci tahu.

4.4 PgnNotation

- `PgnNotation`. Jediná třída ve složce. Ta se stará o zápis dění ze šachovnice do přehledné šachové notace.

4.5 Pieces

Složka obsahující veškeré třídy zabývající se vlastnostmi figurek.

- `Piece` – Abstraktní třída, která obsahuje základní vlastnosti, jako jsou například pozice figurky nebo její barva, proto všechny třídy pro reprezentaci jednotlivých figurek dědí právě z této třídy.
- `Pawn` – Třída reprezentující pohyb pěšce.
- `Knight` – Třída reprezentující pohyb jezdce.
- `Bishop` – Třída reprezentující pohyb střelce.
- `Rook` – Třída reprezentující pohyb věže.
- `Queen` – Třída reprezentující pohyb dámy.
- `King` – Třída reprezentující pohyb krále.
- `PieceControl` – Třída, jež má na starost vykreslení figurky na šachovnici.

4.6 Resources

Tato složka obsahuje pouze všechny ikonky figurek, které jsou potřeba ke hře. Použil jsem ikonky ze zde uvedeného zdroje. [4]

4.7 Tools

- `GameState` – Třída, jež je využívána při možnosti automatického sehrání více partií za sebou mezi dvěma počítačovými hráči. Udržuje informace o aktuální hře.
- `PieceType` – Třída, která reprezentuje výčtový typ s řetězcovými hodnotami (protože v programovacím jazyce `C#` neexistují řetězcové výčtové typy) a slouží pro rozlišení typů figurek.
- `TeamColor` – Typově stejná třída jako `PieceType` akorát slouží pro rozlišení barev figurek.

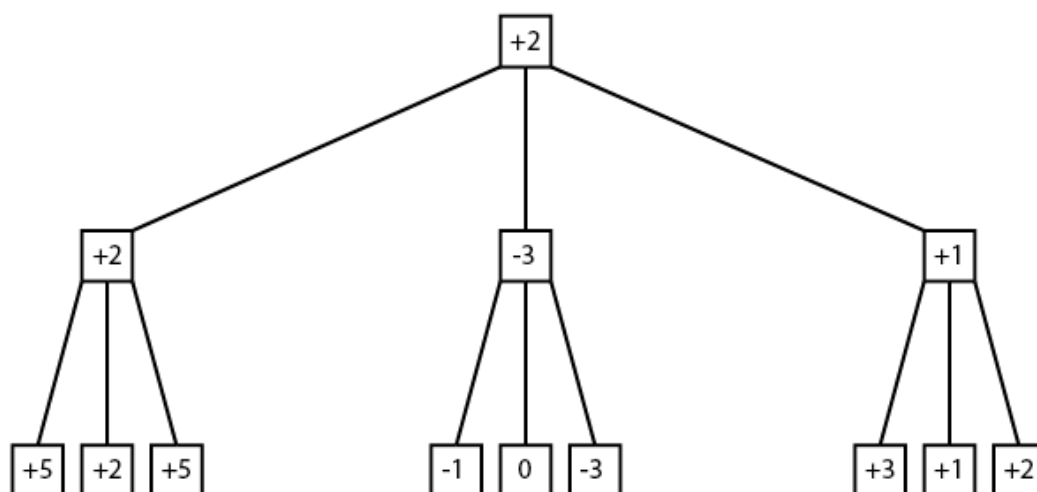
4.8 Nezařazené soubory

Mimo složky jsou v projektu další soubory, které se starají o grafiku aplikace. Tedy o rozvržení okna a o jednotlivé popisky.

4.9 Šachové algoritmy

Úkolem počítačového hráče je na základě algoritmu, který daný počítačový hráč využívá, vybrat nejlepší možný tah. Nejednoduší šachový program by mohl fungovat tak, že při dané pozici, postupně probere všechny své možné tahy a pamatuje si nejlepší tah. Poté, co projde všechny tahy, tak nejlepší tah vybere. Takový program pracuje pouze do hloubky 1, to znamená, že počítá pouze s jedním půltahem (jeden tah černého nebo jeden tah bílého). Tedy takový program klidně vezme dávou krytého pěšce. Teď ale nastává otázka, jak počítačový hráč pozná tah, jenž je nejlepší? K tomu využije počítač svoji ohodnocovací funkci, jež přiřazuje ke každému tahu číslo, které odpovídá stavu na šachovnici. Ohodnocovací funkce jsou různě kvalitní. Nejprimitivnější zajímá pouze hodnota figurek, které jsou stále ve hře. Propracovanější zajímá, kde jsou figurky umístěny. Ty nejlepší se pak zabývají různými kombinacemi, které může více figurek na šachovnici vytvořit. [6, 16]

Obrázek 5: Minimaxový vyhledávací strom.



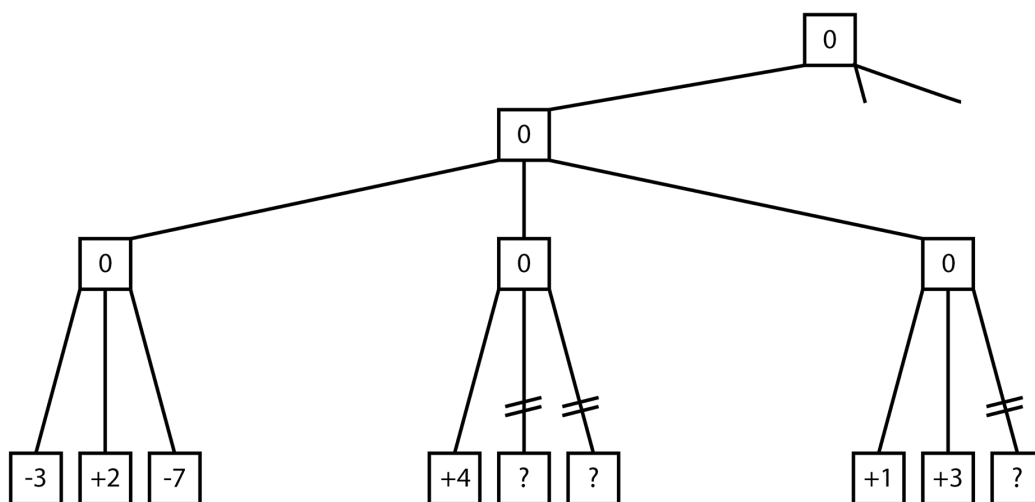
Vylepšením předchozího programu je Minimax, [6] ten za použití rekurze pro počítává svoje i soupeřovy tahy do větší hloubky, která je stanovena programátorem nebo uživatelem. Název Minimax pochází z toho, že počítač hledá maximální hodnoty pro své tahy a zároveň minimální hodnoty pro tahy soupeře. Minimax totiž vždy počítá s nejsilnější možnou odpovědí soupeře. Jedná se o klasický příklad stromového vyhledávání jak je možné vidět na obrázku 5. Tento příklad jde do hloubky 2 půltahy. V nejvrchnějším uzlu je vidět ohodnocení aktuální pozice. Čáry vedoucí z tohoto uzlu reprezentují 3 možné tahy počítače. Každý další uzel reprezentuje ohodnocení pozice po počítačovém tahu. Na každý počítačový tah existují další 3 odpovědi a tím vzniká 9 ohodnocení koncových pozic. Počítač si pro tah z levé části stromu pamatuje hodnotu +2, protože to je hodnota nej-

lepší odpovědi protihráče. Ve střední části stromu je to hodnota -3 a v pravé části +1. Z toho vychází, že si počítač vybere tah z levé části.

Dalším vylepšením je algoritmus Alfa-beta. [6] Tento algoritmus vychází z minimaxu, ale je rychlejší díky ořezávání. To znamená, že se počítač neprobírá všemi možnostmi. Alfa-odsekávání se provádí, pokud se do koncové pozice počítač dostane po soupeřově tahu. Můžeme si ho představit na obrázku 5. Levá část stromu proběhne stejně jako u minimaxu, tudíž si počítač pamatuje hodnotu 2. V prostřední části stromu po ohodnocení levé pozice dostane počítač hodnotu -1, která je pro něj horší jak hodnota 2, kterou si pamatuje. Počítač tedy následující dvě pozice již nemusí ohodnocovat a jde do pravé části stromu, kde ze stejných důvodů nemusí ohodnocovat poslední hodnotu.

Beta-odsekávání funguje na stejném principu, ale s tím rozdílem, že tentokrát se počítač dostane do koncové pozice po svém tahu. Na obrázku 6 ohodnocuje počítač tahy do hloubky 3. Z levé části získá hodnotu 2. Následně ve střední části hned v první pozici získá hodnotu 4, tím zjistí, že tato varianta by pro něj dopadla určitě lépe, takže další tahy už nemusí prozkoumávat, a protože počítá s nejlepší možnou soupeřovou odpovědí, pamatuje si stále hodnotu 2 z levé části. Následně v pravé části stromu to dopadne úplně stejně.

Obrázek 6: Vyhledávací strom s Beta-odsekáváním.



4.10 Porovnání ohodnocovacích funkcí

Šachový počítač je implementovaný algoritmem alpha-beta a má k dispozici tři základní ohodnocovací funkce. První z nich je funkce Basic, která pro svůj výpočet využívá jen hodnoty jednotlivých figur. Druhou funkcí je funkce, jež kromě hodnoty figur započítává do svého ohodnocení i rozmístění jednotlivých figu-

rek po šachovnici v podobě různých bonusů a penalizací. Obě tyto funkce jsou inspirovány ze zdroje [15].

Poslední funkcí je moje funkce, kde jsem se snažil o aplikování mých šachových zkušeností a upravoval jsem tak hodnoty pro jednotlivé figurky. Mezi hlavní změny patří snížení penalizace za krále ve středu šachovnice, protože během koncovky je tam král často velmi silný. A také zmenšení bonusů za jezdce ve středu šachovnice, aby v úvodu partie netáhl často jen jezdec, jak se některým algoritmům stává.

Funkce, kterou jsem se inspiroval `values.json` i funkce, jež jsem vytvořil `matras.json` jsou obě v adresáři `bin` k nahlédnutí.

4.11 Možné rozšíření programu

Program jsem se snažil vytvořit tak, aby byl dobře připravený na případné změny či možná rozšíření.

Nejzajímavějším rozšířením by podle mě bylo přidat nějaké další sofistikovanější vylepšení alpha-bety a porovnat její sílu proti aktuálnímu algoritmu. Také bych se při případné další práci na tomto programu chtěl zaměřit na větší rychlost a dosažení větší hloubky.

Dalším možným rozšířením by bylo přidat databázi knihovny zahájení, do které by si sám uživatel mohl přidávat tahy. Stejně tak by bylo vhodné zauvažovat nad přidáním databáze koncovek, která by mohla zvýšit kvalitu počítačového hráče.

V neposlední řadě by stálo za to upravit šachovou notaci a zápis do PGN souboru, aby byla možné partie z mého programu přehrávat v jiných programech a naopak.

5 Uživatelská dokumentace

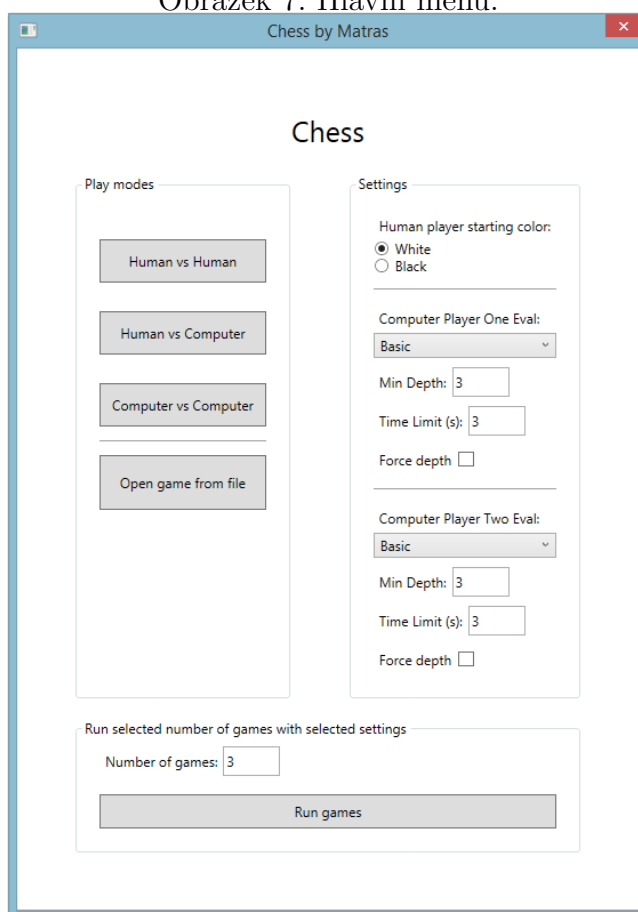
Uživatelská dokumentace slouží uživateli jako manuál pro používání a ovládání programu Chess. Postupně zde bude popsána celá aplikace a její funkce. Ke snadnějšímu pochopení zde budou i screeny přímo z aplikace.

5.1 Hlavní menu

Po spuštění programu se objeví jednoduché menu (obr.7), jež je rozděleno do tří částí. V levé části si může uživatel vybrat ze 4 hlavních možností, co se v programu dá dělat.

První možností je sehrání partie dvou lidských hráčů. Druhou možností je nahrazení jednoho lidského hráče počítačovým hráčem. Třetí možností je spustit partii dvou počítačů a poslední možností je načtení již dříve odehrané partie. Pravá část menu se věnuje nastavení počítačových hráčů. V dolní části je možné nastavit odehrání několika partií počítačových hráčů s předem vybraným nastavením.

Obrázek 7: Hlavní menu.



5.2 Šachovnice a její ovládání

Než se podíváme na jednotlivé módy programu, tak popíšeme ovládání šachovnice (obr. 8). Pokud je lidský hráč na tahu, tak kurzorem najede na figurku své barvy, se kterou chce hrát, a kliknutím na ni ji označí. Zelenou barvou se ukážou všechna políčka, kam může figurku umístit, červeně se ukážou všechna políčka, kam může figurku také umístit a zároveň vezme soupeřovu figurku. Jestliže si uživatel rozmyslí svůj tah a už označenou figurkou nechce hrát, jednoduše na označenou figurku klikne znovu a tím ji odznačí. Rošáda se provede tak, že se klikne na krále a pak na věž se kterou chce uživatel rošádu provést. Pokud ovšem políčko s věží není zelené, znamená to, že s touto věží už rošádu nelze provést. Pokud v partii nastane šach je lidský hráč upozorněn tak, že se červeně rozsvítí políčko s králem, který je v šachu.

Pod šachovnicí jsou dvě tlačítka. Tlačítko `Main Menu` zruší rozehranou partii a vrátí se do menu. Tlačítko `Rotate Board` otočí šachovnici a uživatel tak může nahlížet na partii z pozice druhého hráče.

Obrázek 8: Šachovnice v průběhu partie.



5.3 Partie dvou lidských hráčů

Tento mód se nastaví velmi snadno. V hlavním menu v levé části uživatel klikne na tlačítko `Human vs. Human` a může zahájit partii. Pro pohodlnost obou hráčů, můžou uživatelé šachovnici otáčet, jak bylo popsáno výše. V partii lze také vracet tahy kliknutím do historie, jež je v pravé části, a pak následně pomocí tlačítka `Play as White`, nebo `Play as Black` vybrat jiný tah.

5.4 Partie člověka proti počítači

Pokud si chce uživatel aplikace zahrát partii proti počítači, tak si nejprve v pravé části menu musí vybrat, jestli chce hrát za bílé nebo černé figury, a následně nastavit vlastnosti počítačového hráče, ty se v tomto módu nastavují pod popiskem `Computer Player One Eval`. Na výběr je základní algoritmus nebo pokročilý algoritmus. Pokud si uživatel chce zvolit základní algoritmus, tak přeskočí na 4. krok.

1. Ve scrollbaru si uživatel vybere možnost `Advanced`.
2. Následně klikne na tlačítko `Open Values File` a přes okno procházení se dostane do složky `bin`, kde jsou uloženy soubory `values.json` a `matras.json` reprezentující dvě odlišné ohodnocovací funkce. Popřípadě tam může být i uživatelská ohodnocovací funkce.
3. Uživatel si vybere vhodný soubor a potvrdí svoji volbu.
4. Uživatel má na výběr ze dvou možností, jak chce aby počítač odpovídal svými tahy. První možností je, že uživatel nastaví hodnotu `Min Depth`: na hloubku do které chce, aby počítač propočítával své tahy, a také zaškrtně volbu `Force depth`. Druhou možností je nastavit hodnotu `Time Limit (s)` za kolik sekund chce, aby počítač odehrál svůj tah.
5. Uživatel klikne na tlačítko `Human vs Computer` a partie bude zahájena.

5.5 Partie dvou počítačových hráčů

Pokud chce uživatel pouze pozorovat partii dvou počítačových hráčů postupuje obdobně jako u partie proti počítači, ale s tím rozdílem, že si v pravé části hlavního menu logicky nevybírání nastavení svých figur, ale naopak nastaví i druhý počítač podle postupu výše.

Do partie pak může uživatel kdykoliv vstoupit buď kliknutím na tlačítko `Pause` nebo kliknutím do historie. Obdobně jako v předchozích módech může do partie zasáhnout pomocí tlačítek `Play as White` a `Play as Black`, a převzít tak roli jednoho z počítačů, kterou pak může opět přes tlačítka `Pause` a `Play` vrátit zpět počítačovému hráči.

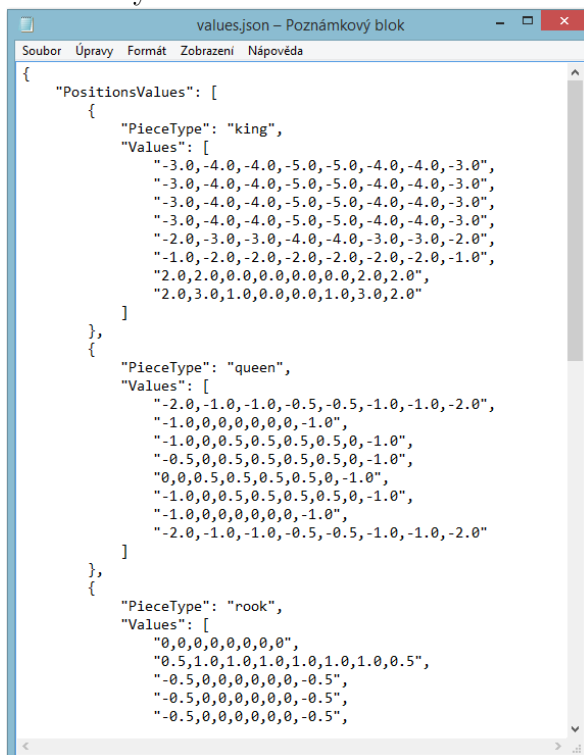
5.6 Odehrání více partií počítačových hráčů

U tohoto módu je potřeba nastavit počítačové hráče stejně jako v předešlých módech. Pak stačí pouze v dolní části hlavního menu nastavit hodnotu, kolik partií se má odehrát, a kliknout na tlačítko `Run games`. Program začne partie vykonávat a po jejich dokončení upozorní uživatele dialogovým oknem. Partie se automaticky uloží do složky `bin`.

5.7 Ukládání a přehrávání odehraných partií

Po každé skončené partii dostane uživatel možnost uložit si ve formátu `nazev.pgn` právě odehranou partii. Pokud v hlavním menu uživatel klikne na `Open game from file`, tak si přes okno procházení najde odehranou partii a může si ji kdykoliv přehrát klikáním do historie na jednotlivé tahy. Uživatel se taky může rozhodnout pokračovat jinak a vyzkoušet si jiné varianty tlačítka `Play as White` nebo `Play as Black`. Také může nechat vybrat jiný tah počítačového hráče, jehož vlastnosti si ale musí nastavit ještě před načtením partie.

Obrázek 9: Vytváření vlastní ohodnocovací funkce.



```
values.json - Poznámkový blok
Soubor Úpravy Formát Zobrazení Nápověda
{
  "PositionsValues": [
    {
      "PieceType": "king",
      "Values": [
        "-3.0,-4.0,-4.0,-5.0,-5.0,-4.0,-4.0,-3.0",
        "-3.0,-4.0,-4.0,-5.0,-5.0,-4.0,-4.0,-3.0",
        "-3.0,-4.0,-4.0,-5.0,-5.0,-4.0,-4.0,-3.0",
        "-3.0,-4.0,-4.0,-5.0,-5.0,-4.0,-4.0,-3.0",
        "-2.0,-3.0,-3.0,-4.0,-4.0,-3.0,-3.0,-2.0",
        "-1.0,-2.0,-2.0,-2.0,-2.0,-2.0,-2.0,-1.0",
        "2.0,2.0,0.0,0.0,0.0,0.0,2.0,2.0",
        "2.0,3.0,1.0,0.0,0.0,1.0,3.0,2.0"
      ]
    },
    {
      "PieceType": "queen",
      "Values": [
        "-2.0,-1.0,-1.0,-0.5,-0.5,-1.0,-1.0,-2.0",
        "-1.0,0,0,0,0,0,-1.0",
        "-1.0,0,0.5,0.5,0.5,0.5,0,-1.0",
        "-0.5,0,0.5,0.5,0.5,0.5,0,-1.0",
        "0,0,0.5,0.5,0.5,0.5,0,-1.0",
        "-1.0,0,0.5,0.5,0.5,0.5,0,-1.0",
        "-1.0,0,0,0,0,0,-1.0",
        "-2.0,-1.0,-1.0,-0.5,-0.5,-1.0,-1.0,-2.0"
      ]
    },
    {
      "PieceType": "rook",
      "Values": [
        "0,0,0,0,0,0,0",
        "0.5,1.0,1.0,1.0,1.0,1.0,0.5",
        "-0.5,0,0,0,0,0,-0.5",
        "-0.5,0,0,0,0,0,-0.5",
        "-0.5,0,0,0,0,0,-0.5"
      ]
    }
  ]
}
```

5.8 Nastavení vlastních ohodnocovacích hodnot

Poslední funkcí tohoto programu je možnost vytvoření vlastní ohodnocovací funkce. Uživatel si ze složky `bin` může duplikovat soubor `values.json` a v po-

známkovém bloku upravit funkci.(obr.9) Pro každou figurku je možné nastavit plusové i minusové hodnoty za to, když se dostane na nějaké pole. Čísla jsou uspořádána do tvaru šachovnice, tudíž v dolních dvou řadách má počítač na začátku svoje figurky a v horních dvou řadách jsou soupeřovy figurky Svoji upravenou funkci může pak uživatel testovat v různých módech.

Závěr

Výsledkem práce je program, který umožňuje uživateli sehrání šachové partie proti jednomu z připravených algoritmů. Uživatel si navíc může nastavit, do jaké hloubky má počítačový hráč propočítávat své tahy, popřípadě za jak dlouho je musí počítačový hráč odehrát.

Aplikace dále umí sehrát i partii dvou počítačových hráčů. Uživatel také může zadat i sehrání více počítačových her najednou. V Uživatelské dokumentaci je popsán návod, jak upravit hodnoty pokročilé ohodnocovací funkce. Díky tomu si tak uživatel může vyzkoušet své vlastní nápady, jak by počítačový hráč mohl být ještě silnější nebo se tak i dokonce zamýšlet nad tím jak hraje on.

Program disponuje i možností sehrání partie dvou lidských hráčů. Jakákoliv partie jde po odehrání uložit a v budoucnu je možné si ji opět přehrát. V aplikaci je také možné přecházet mezi jednotlivým nastavením počítačových a lidských hráčů.

Za nejproblematičtější část práce považuji rychlost výpočtu jednotlivých tahů. Aplikace ale své hlavní cíle plní a je do budoucna připravena na změny, které popisují v podkapitole Možné rozšíření programu.

Conclusions

The result of the work is a program that allows the user to play a chess game against one of the prepared algorithms. In addition, the user can set the depth to which the computer player should calculate his moves, or for how long the computer player must play them.

The application can also play a game of two computer players. Alternatively, the user can specify more than one game of computer games at once. The User Documentation describes how to adjust the values of the advanced evaluation function. Thanks to this, the user can try out his own ideas of how a computer player could be even stronger, or even think about how he plays.

The program also has the option of playing a game of two human players. Any game can be saved after the game and can be replayed in the future. In the application, it is also possible to switch between individual settings of computer and human players.

I consider the speed of calculation of individual moves to be the most problematic part of the work. However, the application fulfills its main goals and is ready for changes in the future, which I describe in the subchapter Possible extension of the program.

A Obsah přiloženého CD/DVD

bin/

Program CHESS, spustitelný přímo z CD/DVD. Adresář obsahuje i všechny runtime knihovny a další soubory potřebné pro bezproblémový běh programu z CD/DVD.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletní zdrojové texty programu CHESS.

readme.txt

Instrukce pro spuštění programu CHESS, včetně všech požadavků pro jeho bezproblémový provoz.

U veškerých cizích převzatých materiálů obsažených na CD/DVD jejich zahrnutí dovoluují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy na CD/DVD, je uveden jejich zdroj (např. webová adresa) v bibliografii nebo textu práce nebo v souboru `readme.txt`.

Literatura

- [1] 1. lekce: Úvod. Šachovnice a označení polí [online]. [cit. 2020-06-21]. Dostupné z: <https://www.chess.cz/1-lekce-uvod-sachovnice-a-oznaceni-poli>
- [2] Dějiny šachové hry [online]. [cit. 2020-06-21]. Dostupné z: https://cs.wikipedia.org/wiki/D%C4%9Bjiny_%C5%A1achov%C3%A9_hry
- [3] Pravidla šachů [online]. [cit. 2020-06-21]. Dostupné z: https://cs.wikipedia.org/wiki/Pravidla_%C5%A1ach%C5%AF
- [4] File Pieces Sprite Wikimedia - Chess Pieces Png Clipart [online]. [cit. 2020-06-21]. Dostupné z: https://www.pinclipart.com/pindetail/hRbwim_file-pieces-sprite-wikimedia-chess-pieces-png-clipart/?fbclid=IwAR2fm5tqTeGMR3KjwTBZKjpoFWuHEEf4eDcDxthBaqVfuaDd9IHTMJsz4wk
- [5] Šachy [online]. [cit. 2020-06-21]. Dostupné z: <https://cs.wikipedia.org/wiki/%C5%A0achy>
- [6] Steinwender, Dieter. Šachy na PC. 1997. 486 s. [cit. 2020-06-21].
- [7] Turek (stroj) [online]. Dostupné z: [https://cs.wikipedia.org/wiki/Turek_\(stroj\)](https://cs.wikipedia.org/wiki/Turek_(stroj))
- [8] The Turk: Wolfgang von Kempelen's Fake Automaton Chess Player [online]. [cit. 2020-06-21]. Dostupné z: <https://interestingengineering.com/the-turk-fake-automaton-chess-player>
- [9] Deep Blue [online]. [cit. 2020-06-21]. Dostupné z: https://cs.wikipedia.org/wiki/Deep_Blue
- [10] Proti počítači nemají šachisté šanci. SkyNetu se prý ale zatím bát nemusíme [online]. [cit. 2020-06-21]. Dostupné z: <https://svetchytre.cz/a/STZUk/proti-pocitaci-nemaji-sachiste-sanci-skynetu-se-pry-ale-zatim-bat-nemusime>
- [11] Prohlídka jazyka C# [online]. [cit. 2020-06-21]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/csharp/tour-of-csharp/>
- [12] C Sharp [online]. [cit. 2020-06-21]. Dostupné z: https://cs.wikipedia.org/wiki/C_Sharp
- [13] Začínáme s WPF (Windows Presentation Foundation) [online]. [cit. 2020-06-21]. Dostupné z: <https://docs.microsoft.com/cs-cz/visualstudio/designers/getting-started-with-wpf?view=vs-2019>

- [14] Windows Presentation Foundation [online]. [cit. 2020-06-21]. Dostupné z:
https://cs.wikipedia.org/wiki/Windows_Presentation_Foundation
- [15] A step-by-step guide to building a simple chess AI [online]. [cit. 2020-06-21].
Dostupné z:
<https://www.freecodecamp.org/news/simple-chess-ai-step-by-step-1d55a9266977/>
- [16] Šachové myšlení (1) - nejjednodušší program [online]. [cit. 2020-06-21]. Do-
stupné z:
http://text.linuxsoft.cz/article.php?id_article=1109