VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

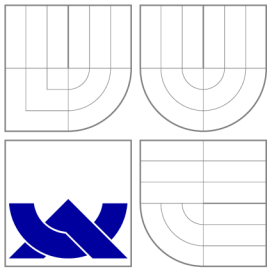
ŠACHOVÝ PROGRAM S RŮZNÝMI VARIANTAMI ŠACHŮ OBSAHUJÍCÍ NOVOU FIGURU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

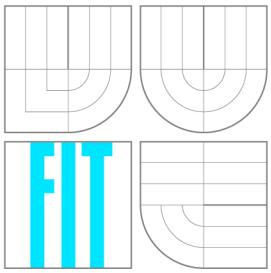
AUTOR PRÁCE
AUTHOR

MARTIN DOSTÁL

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

ŠACHOVÝ PROGRAM S RŮZNÝMI VARIANTAMI ŠACHŮ OBSAHUJÍCÍ NOVOU FIGURU

CHESS PROGRAM WITH DIFFERENT CHESS VARIATIONS INCLUDING NEW FIGURE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN DOSTÁL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAROSLAV ROZMAN, Ph.D.

BRNO 2013

Abstrakt

Tato práce se zabývá analýzou a zhodnocením metod a algoritmů potřebných k úspěšné implementaci šachového programu. Uvedeny jsou jak základní principy těchto metod a algoritmů, tak i další dnes využívaná rozšíření s cílem implementovat kvalitní šachový program hrající různé varianty šachů obsahující novou figuru. Práce představuje základní pilíře každého šachového programu, kterými jsou reprezentace šachovnice, prohledávání stavového prostoru, ohodnocování stavu hry a nutné změny pro implementaci nových šachových variant. Pro přehled jednotlivých metod je v práci uvedeno porovnání jejich náročnosti.

Abstract

This thesis deals with an analysis and evaluation of methods and algorithms needed for successfull implementation of a chess program. Both the basic principles and further currently used extensions of these methods and algorithms are stated with a goal to implement a quality chess program playing different chess variations including a new figure. This thesis also introduces the basic pillars of every chess program - these pillars include board representation, state space search, evaluation of a game state and changes required for implementation of new chess variations. A comparison of difficulties of individual methods is included for an overview.

Klíčová slova

Šachy, různé varianty šachů, umělá inteligence, ohodnocovací funkce, prohledávání stavového prostoru, Alfa-Beta

Keywords

Chess, different chess variations, artificial intelligence, evaluation function, state space search, Alpha-Beta

Citace

Martin Dostál: Šachový program s různými variantami šachů obsahující novou figuru, bakalářská práce, Brno, FIT VUT v Brně, 2013

Šachový program s různými variantami šachů obsahující novou figuru

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doktora Jaroslava Rozmana. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Martin Dostál
15. května 2013

Poděkování

Na tomto místě bych rád poděkoval panu doktoru Rozmanovi za ochotu a trpělivost. Dále bych chtěl poděkovat rodině a přítelkyni za projevenou podporu.

Obsah

Úvod

Jak vyplývá z názvu, bakalářská práce se zabývá implementací šachového programu pro varianty šachů obsahující nové figury. Práce rozebírá a porovnává prostředky pro řešení zadaného úkolu, poskytuje základní přehled v problematice šachu a implementačních řešení šachových programů. Seznámí čtenáře s aktuálně využívanými principy pro klasické šachy, představí řešení nutné k implementaci nových variant a popíše problémy s nimi spojené, jako je například vypořádání se s různou velikostí šachovnice.

První kapitola se věnuje seznámení s klasickými šachy pro případ, že se čtenář této práce doposud s šachy nesetkal. Popisuje šachovnici, hrací figury, princip a pravidla hry samotné. Pro přehled tato kapitola obsahuje i historii hry šachy. V druhé kapitole jsou vyjmenovány a popsány jednotlivé uvažované herní varianty šachů s novou figurou, které jsou následně v praktické části bakalářské práce zanalyzovány, naprogramovány a otestovány. Varianty se z větší části hrají podle standardních pravidel, až na specifické výjimky uvedené u každé varianty.

Ve třetí kapitole jsou uvedeny současné metody používané pro programování šachů, které se dělí podle okruhů činností, na které se zaměřují. Kapitola je rozdělena na čtyři okruhy. Prvním je reprezentace šachovnice, která je základním kamenem každého šachového programu. V dalším okruhu jsou popsány základní metody pro prohledávání stavového prostoru, jehož efektivita určuje z větší části úspěšnost programu. Poté následuje okruh zaměřující se na různá vylepšení základních metod a efektivity programu. Poslední okruh se zabývá ohodnocováním stavu hry, na jehož základě se program rozhoduje jaké tahy bude provádět.

Čtvrtá kapitola popisuje změny v základních metodách a ohodnocení, které jsou nutné pro úspěšnou implementaci nových variant šachů. Pátá kapitola představuje implementační řešení tohoto problému, celkový vzhled a funkčnost programu. Také je zde uvedeno srovnání efektivity metod pro prohledávání stavového prostoru. Šestá kapitola se věnuje testování a ověřování efektivity a kvality programu proti různým soupeřům. Jedním ze soupeřů je program kolegy Zbyňka Jadrníčka, který má zadání bakalářské práce pro rok 2012/2013 "Šachový program pro různé varianty šachů"[15].

Kapitola 1

Klasické šachy

Šachy, nebo také hra v šach, jsou desková hra dvou hráčů známá již mnoho století. Dnes jsou šachy velmi populární hrou, v níž se pořádají turnaje od začátečníků až po velmistrovké. Úkolem hry je dát soupeřově králi mat. To je situace, kdy je král napaden soupeřovou figurou a neexisuje prázdné pole, kam by mohl ustoupit. Matu je možno také zabránit zajmutím útočící figury králem, pokud není ve směru tahu jiné soupeřovi figury, zajmutím útočící figury jinou figurou nebo posunutím vlastní figury na pole, které je mezi králem a útočící figurou. Partie ovšem může dospět do situace, kdy již není možné dát mat. Tento stav se nazývá pat a hra končí remízou. Patu je možné dosáhnout i tehdy, kdy hráč na tahu nemá žádný platný tah, který by mohl zahrát. Dále hra končí patem, pokud se třikrát po sobě opakují tahy figur vedoucí ke stejnému rozestavení na šachovnici (pohybováním figur tam a zpět), nebo během posledních padesáti tahů nebyla zajmuta žádná figura. Hráči se v hraní střídají a partii vždy zahajuje bílý hráč. V následujících částech bude popsána hrací deska, tahy figur a původ šachu.

Hrací deska se skládá ze 64 polí uspořádaným do čtverce s osmi řadami a sloupci. Sousední pole jsou barevně odděleny, aby nedošlo k chybnému diagonálnímu pohybu. Oba hráči mají stejnou sadu figur obsahující:

- 8 pěšců v přední řadě,
- 2 věže po okrajích,
- 2 jezdce umístěné vedle věží,
- 2 střelce umístěné vedle jezdců,
- 1 královnu stojící na poli, které má její barvu (světlé nebo tmavé),
- 1 krále.

V klasickém šachu jsou figury při začátku partie rozestaveny do výchozího postavení, které je vidět na obrázku 1.1.

Pěšci se mohou pohybovat pouze o jedno pole dopředu, popřípadě ze základního postavení o dvě pole. Pěšec útočí na pole, která jsou ve vzdálenosti jednoho pole diagonálně před ním. Navíc má pěšec možnost zajmout soupeřova pěšce, který se v minulém tahu pohnul ze základního postavení o dvě pole vpřed a nyní stojí na stejné řadě jako pěšec patřící hráči

Obrázek 1.1: Výchozí postavení figur v klasickém šachu (Obrázek ze stránek brainking [22])

na tahu, jen ve vedlejším sloupci. Tento způsob zajmutí se nazývá "braní mimochodem" nebo také francouzsky "en passante". Více o tomto tahu je možné se dočíst na stránce wikipedia.org [23]. Poslední vyjímečnou vlastností pěšců je povyšování. Pokud pěšec dojde na poslední řadu na opačném konci herní desky, je možné ho vyměnit za jednu z následujících figur:

- dáma,
- věž,
- střelec,
- jezdec.

Věž se pohybuje a útočí pouze vertikálně nebo horizontálně o neomezený počet polí, střelec naopak pouze po diagonálách. Při tahu věže nebo střelce není možné přeskakovat vlastní ani soupeřovy figury. Jezdec má pohyb složitější, skládá se totiž ze dvou částí. První část je vertikální nebo horizontální o dvě pole. Pokud se v první části vertikálně druhá část pohybu bude horizonální o jedno pole a naopak. Tímto složeným pohybem může jezdec přeskakovat ostatní figury. Hráč ovšem nemůže pohnout figurou na pole, které je jíž obsazené jednou z jeho figur. Dáma se může pohybovat jako věž a střelec dohromady. Král se také může hýbat jako věž a střelec dohromady, ale vždy pouze o jedno pole, a nikdy nesmí skončit na poli, které je ve směru pohybu některé ze soupeřových figur.

1.1 Vznik šachu

Tato kapitola čerpá z knihy *Historie šachu* [3]. První obdoba hry šachy se objevila pravděpodobně v Indii ke konci 6. století našeho letopočtu. Jmenovala se "čaturanga", hrála se na desce se 64 poli a byla to hra pro 4 hráče, kde každý hráč měl 4 pěšce, slona, koně, válečný vůz a krále. Při hře se používala i kostka, a podle čísla, které padlo na kostce se muselo hrát určitou figurou. Vždy hráli dva hráči v týmu proti druhým dvěma a cílem hry bylo obsadit pozici soupeřova krále nebo mu sebrat figury. Postupem času se z čaturangy stala hra pro dva hráče, pravděpodobně kvůli nedostatku hráčů. Na nátlak hinduistických kněží byla

odebrána kostka. Poté se čaturanga rozšířila do Persie a s perskými válečnými výpravami začátkem 7. století dále do Egypta pod jménem "čatrang"a do arabského světa pod jménem "šatrandž". Arabové si hru šatrandž osvojili a všestranně rozvinuli. Z této doby pochází i termín "šach mat", protože šach je persky "král" a mat v arabštině znamená "bezbranný". Pravidla ovšem ještě ani zdaleka nepřipomínala dnešní. Jediné figury, které si zachovaly původní pohyb, jsou král, věž a jezdec. Arabové si šatrandž oblíbili, zdokonalovali se v něm a pořádali turnaje. Při střetech křesťanské a arabské kultury často docházelo k osvojování znalostí druhé strany, a tak se šachy dostaly z arabského světa do Evropy. První písemný doklad o hře šachy v Evropě se objevuje kolem roku 1000 našeho letopočtu, ale pravděpodobně byly šachy známy již dříve. Postupem času se rozvinulo šachové myšlení (cílené tahy figurami) namísto pouhého posunování figur. K tomu z velké části přispěl vynález knihtisku, kdy mohly být znalosti hráčů tištěny a zachovány. Postupem času se měnily pravidla i figury do dnešní podoby (arabský slon byl nahrazen křesťanským knězem).

Kapitola 2

Vybrané varianty šachů

V této kapitole jsou uvedeny jedntolivé herní varianty šachů, jejich základní popis, příslušná pravidla a změny oproti klasickým šachům. Informace a obrázky byly čerpány ze stránky brainking.com [22].

2.1 Maharajah Chess

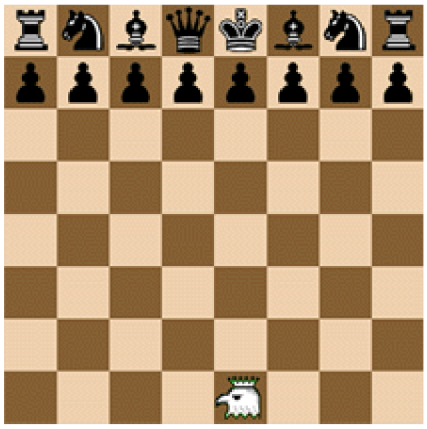
Maharajah chess je varianta šachů, kde černý hráč má klasické figury ve standardním postavení a bílý pouze figuru zvanou Maharajah nebo česky také Maharadža. Úkolem Maharadži je dát soupeřovu králi mat a nenechat se zajmout. Hráč s Maharadžou se tedy musí snažit sebrat co nejvíce soupeřových figur, než si vybuduje formaci. Naproti tomu soupeř se snaží maximálně omezit Maharadžův pohyb a neztratit příliš mnoho vlastních figur. Maharadža má výhodu velké pohyblivosti a nebude lehké omezit jeho pohyb bez obětování figur. Černý hráč musí postupovat opatrně, protože střední a těžké zajmuté figury není možné nahradit povýšením pěšců. Podle statistik uvedených na stránce brainking.com vítězí Maharadža ve 30,61 % her. Při testování této varianty se mi ovšem za bílého nepovedlo proti počítači vyhrát ani jednou. Výchozí postavení figur je na obrázku 2.1.

Charakteristiké vlastnosti varianty Maharajah Chess:

- velikost šachovnice 8x8,
- konec hry: mat soupeře nebo pat,
- bílý: Maharadža kombinuje krále, královnu a jezdce,
- bílý: výchozí pozice Maharadžy je E1 (pozice krále),
- černý: pěšce nelze vyměnit za jiné figury při příchodu na řadu 1 a zůstávají zde do konce hry.

2.2 Janus Chess

Janus chess je varianta šachů, kde obě strany mají novou figuru zvanou Janus. Hraje se na rozšířené šachovnici. Oba hráči mají dva Janusy umístěné mezi vežemi a jezdci. Výhodou Januse je zvýšená pohyblivost, protože kombinuje pohyb střelce a jezdce. Vzhledem k manévrovacím schopnostem se hráčům vyplatí umístit Januse do středu šachovnice, odkud může



Obrázek 2.1: Maharajah Chess výchozí pozice

podporovat ostatní figury nebo sám zaútočit na nechráněná křídla soupeře. Janus nabízí úplně nové možnosti kombinační hry, díky jeho schopnosti rychle se přemísit a zaútočit za obranu soupeře. V tomto ohledu bude ze začátku pravděpodobně úspěšnější počítač, protože lidský hráč bude potřebovat čas než se naučí plně využívat potenciál Januse. Další výhodou je, že je možné dát mat pouze králem a Janusem. Výchozí postavení figur je na obrázku 2.2.

Charakteristiké vlastnosti varianty Janus Chess:

- velikost šachovnice 8x10,
- konec hry: mat soupeře nebo pat,
- oba: Janus kombinuje jezdce a střelce,
- oba: výchozí pozice januse je B1 a I1/ B8 a I8,
- oba: pěšce lze vyměnit za královnu, věž, jezdce, střelce nebo Januse,
- rošáda bílý: král se přemístí na B1 nebo I1 a vež vedle krále směrem do středu šachovnice C1/H1,
- rošáda černý: král na B8 nebo I8, věž na C8/H8.

2.3 Embassy Chess

Embassy chess je varianta šachů, kde obě strany mají nové figury zvané Kardinál a Maršál. Kardinál kombinuje jezdce a střelce, Maršál jezdce a věž. Šachovnice je rozšířená o tyto dvě figury soustředěné na pravé polovině šachovnice. Převaha materiálu na pravé straně porušuje rovnováhu existující v klasickém šachu. Na hru budou mít zásadní dopad pěšcové struktury, které jednotliví hráči vybudují, aby omezily pohyb Maršálovi s Kardinálem. Při dobré kombinační hře je možné jen těmito dvěma figurami rozložit obranu soupeře nebo zajmout celé křídlo. Šance na výhru jsou vyrovnané a bude záležet na schopnostech každého z hráčů, jak si dokáže s novými figurami poradit. Navíc má každá z nových figur schopnost dát mat soupeři pouze s králem. Výchozí postavení figur je na obrázku 2.3.

Obrázek 2.2: Janus Chess výchozí pozice

Charakteristiké vlastnosti varianty Embassy Chess:

- velikost šachovnice 8x10,
- konec hry: mat soupeře nebo pat,
- oba: Kardinál kombinuje jezdce a střelce,
- oba: Maršál kombinuje jezdce a věž,
- oba: pěšce lze vyměnit za královnu, věž, jezdce, střelce, Kardinála nebo Maršála,
- bílý: výchozí pozice Kardinála je G1 a Maršála F1,
- černý: výchozí pozice Kardinála je G8 a Maršála F8,
- rošáda oba: klasickým způsobem, ale král ujde 3 pole (výchozí 2).

Obrázek 2.3: Embassy Chess výchozí pozice

2.4 Amazon Chess

Amazon chess je varianta šachů, kde obě strany mají novou figuru zvanou Amazonka. Amazonka přidává královně možnost pohybu jezdce, čímž z ní činí ještě nebezpečnější figuru

s téměř neomezeným pohybem. Hráči se budou snažit dostat svoji Amazonku do pozice, kde bude největší hrozbou pro soupeře. Ideální jsou krajní sloupce u výchozího postavení nebo střed šachovnice. Správně umístěná Amazonka může podporovat několik figur stojících třeba i na opačných koncích šachovnice a zároveň ohrožovat soupeřovy figury. Pohyblivost Amazonky tak silně převyšuje ostatní figury, že její ztráta téměř jistě znamená prohru, pokud soupeř svoji Amazonku stále má. Výchozí postavení figur je na obrázku 2.4.

Charakteristiké vlastnosti varianty Amazon Chess:

- velikost šachovnice 8x8,
- konec hry: mat soupeře nebo pat,
- oba: Amazonka kombinuje jezdce a královnu,
- oba: výchozí pozice Amazonky je D1/D8 (nahrazuje královnu),
- oba: pěšce lze vyměnit za věž, jezdce, střelce nebo Amazonku.

Obrázek 2.4: Amazon Chess výchozí pozice

2.5 Grand Chess

Grand chess je varianta šachů, která přidává stejné figury jako varianta Embassy Chess popsáná výše. Jedním z hlavních rozdílů je velikost šachovnice, která je rozšířena na 10x10, kde na první nebo desáté řadě stojí pouze věže. Varianta dává prostor věžím, aby využily svůj potenciál dříve než v klasickém šachu. Na druhou stranu není možné provádět rošádu, a tak král zůstává v nebezpečném středu šachovnice. V kombinaci s figurami Kardinál a Maršál s rozšířenou pobyblivostí se otevírá nový prostor pro kombinace, obzvláště pak při hře s pěšci, kteří se povyšují do figur sebraných soupeři, a to volitelně na osmé a deváté řadě a povinně na desáté řadě. Výchozí postavení figur je na obrázku 2.5.

Charakteristiké vlastnosti varianty Grand Chess:

- velikost šachovnice 10x10,

- konec hry: mat soupeře nebo pat,
- oba: Kardinál kombinuje jezdce a střelce,
- oba: Maršál kombinuje jezdce a věž,
- oba: pěšce lze vyměnit pouze za figuru předem zajmutou soupeři, kromě pěšce (odstraní se ze zajmutých figur),
- oba: pokud soupeř nemá žádné dostupné zajmuté figury, pěšec nemůže postoupit na poslední řadu,
- oba: rošáda nepovolena,
- bílý: výchozí pozice pěšců je řada 3,
- černý: výchozí pozice pěšců je řada 8,
- bílý: na řadě 8 a 9 může pěšce vyměnit, na řadě 10 pěšce vyměnit musí,
- černý: na řadě 3 a 2 může pěšce vyměnit, na řadě 1 pěšce vyměnit musí,
- bílý: výchozí pozice Kardinála je G2 a Maršála F2,
- černý: výchozí pozice Kardinála je G9 a Maršála F9.

Obrázek 2.5: Grand Chess výchozí pozice

2.6 Capablanca Random Chess

Capablanca Random chess je varianta šachů, kde obě strany mají nové figury zvané Arcibiskup a Kancléř. Arcibiskup je kombinací jezdce a střelce, Kancléř kombinací jezdce a věže. Šachovnice je rozšířena o tyto figury na rozměry 8x10. Figury mají náhodně vygenerovanou počáteční pozici, která je stejná pro oba hráče. To dodává hře rozmanitost, protože není jednoduše možné natrénovat různé výchozí strategie. Každá hra je originálem a tento fakt může zvýhodnit počítač před lidským hráčem. Celkem takto vygenerovaných počátečních

postavení může být 12118. Tato šachová varianta byla vyvinuta speciálně pro verifikaci nových i stávajících metod a algoritmů používaných při programování šachových programů. Výchozí postavení figur je na obrázku 2.6.

Charakteristiké vlastnosti varianty Capablanca Random Chess:

- velikost šachovnice 8x10,
- konec hry: mat soupeře nebo pat,
- oba: pěšci mají výchozí pozice na 2 a 7 řadě,
- oba: arcibiskup kombinuje jezdce a střelce,
- oba: kancléř kombinuje jezdce a věž,
- počáteční pozice: každý pěšec musí být chráněn jinou figurou,
- počáteční pozice: král musí být umístěn mezi věžemi,
- počáteční pozice: střelci musí být jeden na bílém a druhý na černém poli,
- počáteční pozice: královna a arcibiskup musí být jeden na bílém a druhý na černém poli,
- počáteční pozice: pozice figur obou hráčů jsou symetrické,
- rošáda: jako u klasických šachů, stejné cílové pozice.

Obrázek 2.6: Capablanca Random Chess výchozí pozice

2.7 Behemoth Chess

Behemoth chess je varianta šachů, při které se ve hře objevuje nová neutrální figura zvaná Behemoth. Behemoth se hýbe náhodně a ničí vše, co má v cestě. Behemoth se pohybuje stejnými směry jako královna. Směr a délka kroku (1 - 4 pole) se vybere náhodně. Pokud by pohybem Behemoth vypadl ze šachovnice, objeví se ve směru pohybu na druhé straně šachovnice. Například z pozice G8, směr doprava nahoru, krok délky 2. Pozice budou postupně G8 → H1 → A2. Při této variantě je nejdůležitější držet vlastního krále mimo možný

směr tahu Behemotha, protože o něj lze velmi rychle přijít. Možná je efektivnější vyčkávat až Behemoth zajme soupeřova krále, než se snažit dát mat. Výchozí postavení figur je na obrázku 2.7.

Charakteristiké vlastnosti varianty Behemoth Chess:

- velikost šachovnice 8x8,
- konec hry: mat soupeře nebo Behemoth zajme soupeřova krále,
- oba: klasická pravidla šachů kromě neexistence šachu (matu).

Pro Behemotha platí následující pravidla:

- výchozí pozice D4,
- táhne po každém tahu některého z hráčů,
- délka kroku 1-4 (náhodně), směr náhodný,
- nemůže být zajmut,
- zajímá všechny figury v cestě,
- pokud by měl vykročit ze šachovnice, objeví se na druhé straně (závislé podle směru),
- pokud zničí oba krále najednou hra končí remízou.

Obrázek 2.7: Behemoth Chess výchozí pozice

Kapitola 3

Posupy a metody využívané v současnosti při programování šachů

Kapitola je rozdělena do tří základních okruhů, kde se každý okruh zabývá nezbytnou součástí šachového programu. Těmito okruhy jsou:

- reprezenace šachovnice,
- prohledávání stavého prostoru,
- ohodnocování stavu hry.

Jednotlivé okruhy se věnují příslušným problémům a metodám. Na závěr je doplněn specifický okruh věnující se potřebným změnám, které je nutné v základních okruzích provést pro úspěšnou implementaci nových šachových variant.

3.1 Reprezentace šachovnice

Prvním základním okruhem je reprezentace šachovnice. Šachovnice a její reprezentace je srdcem každého šachového programu. Podle výběru, jakým budeme reprezentovat šachovnici, se odvíjí celý program. Pokud zvolíme špatnou reprezentaci, program bude pomalý a nebude poskytovat dostatečně kvalitní výsledky. Obzvláště záleží na rychlosti operací "zahrát tah"a "vrátit tah", protože těchto operací proběhnou miliony během jedné partie šachu. Pro reprezentaci šachovnice existují dva hlavní směry [6]:

- zaměřené na figury (Piece Centric),
- zaměřené na pozici (Square Centric).

Dále existuje kombinace těchto dvou směrů zvaná hybridní přístup. Poslední zmíněnou formou reprezentace šachovnice jsou Zobristické klíče, které poskytují téměř unikátní identifikátor v podobě celého čísla, a využívají se k identifikaci stavu hry a hashování.

3.1.1 Zaměřené na figury (Piece Centric)

Jak vypovídá název, tento směr se zaměřuje na figury, šachovnice je až druhotná [6]. Figury nebo matice figur samy nesou informaci, kde se na šachovnici nacházejí. Matice šachovnice je pouze očíslované pole a nemá žádnou informaci o figurách stojících na ní. Výhodou je,

pokud hledáme konkrétní figuru, víme přesně, na kterém indexu se nachází. Ovšem při generování tahů musíme projít vždy celý seznam, abychom ověřili, že na dané pozici se žádná figura nenachází. Existují tři nejrozšířenější varianty této reprezentace:

- seznam figur,
- množina figur,
- bitové pole.

Seznam figur je pole nebo seznam obsahující pozici pro každou figuru. Figury hráčů jsou odlišeny indexem.

Množina figur jsou 32b slova pro každou figuru, která obsahují index v seznamu figur.

Bitové pole reprezentuje pozici pro každou skupinu figur (střelci, koně, atd.), rozdělené podle barvy. Hlavní výhodou je, že se celá šachovnice vejde do jednoho 64b registru.

3.1.2 Zaměřené na pozici (Square Centric)

Tento přístup je přesným opakem předchozího. Šachovnice obsahuje odkazy na figury stojící na ní [13]. Absence seznamu nebo množiny figur ovšem způsobuje, že na první pohled není vidět, kolik figur každý hráč aktuálně vlastní. Na druhou stranu generování tahů probíhá rychle, protože každá pozice o sobě ví, zda je prázdná nebo obsazená popřípadě, jaká figura se na daném místě nachází. I pro tento přístup existují tři nejrozšířenější varianty:

- 8x8 šachovnice,
- poštovní schránka,
- 0x88.

Šachovnice 8x8 je nejjednodušší reprezentací šachovnice orientované na pozice. Je tvořena dvoudimenzionálním polem o rozměrech 8x8 a číslováním polí 0 - 63.

Poštovní schránka je 8x8 šachovnice umístěná v poli 12x10. Tato obálka umožnuje jednodušší detekci tahů mimo šachovnici.

Varianta 0x88 využívá k indexaci jeden Byte, kde vrchní 4 bity jsou řádky a spodní jsou sloupce. Ovšem celková velikost pole pro uložení šachovnice je 128 polí. Aby nevznikaly mezery v indexech (08 - 0F,18 - 1F,...), je druhá šachovnice neadresovaná.

3.1.3 Hybridní přístup

Hybridní metody kombinují výhody předchozích variant za cenu větší paměťové náročnosti. Pro generování pohybu je výhodnější procházet šachovnici v žádaném směru, než u každé figury kontrolovat, jestli se nachází v cestě plánovaného pohybu (jak je tomu u variant zaměřených na pozici). Pro výběr figur, u kterých se bude generovat pohyb, je zase jednodušší procházet jejich seznam, než celou šachovnici (jak je tomu u variant zaměřených na figury). Vhodnou kombinací obou přístupů je možné zcela eliminovat nedostatky za cenu

vyšší paměťové náročnosti, která ovšem při kapacitách dnešních počítačů nepředstavuje žadný problém.

3.1.4 Zobristické klíče (Zobrist keys)

Jelikož se při programování šachů často používá hashování, je nutné zvolit reprezentaci šachovnice v takové podobě, aby umožnila rychlé vygenerování indexu do tabulek. První dostatečně unikátní variantou je řetězec, který by obsahoval pozici a typy všech figur. Ovšem vytvořit takovýto řetězec a vygenerovat z něj hash by bylo hodně náročné. Naopak velmi efektivní je reprezentace pomocí Zobristických klíčů. Tento způsob reprezentace použil pan J. Pettersson pro svůj šachový program Mediocre [21]. Myšlenka zní

> Reprezentovat stav pomocí 64b čísla, které budeme měnit v závislosti na tom, jaký tah provádíme nebo odstraňujeme pomocí předem vygenerovaných náhodných klíčů. Rozsah čísel je dostatečný, takže kolize jsou velmi vyjímečné. Nový klíč je poté použit jako vstup hashovací funkce [21].

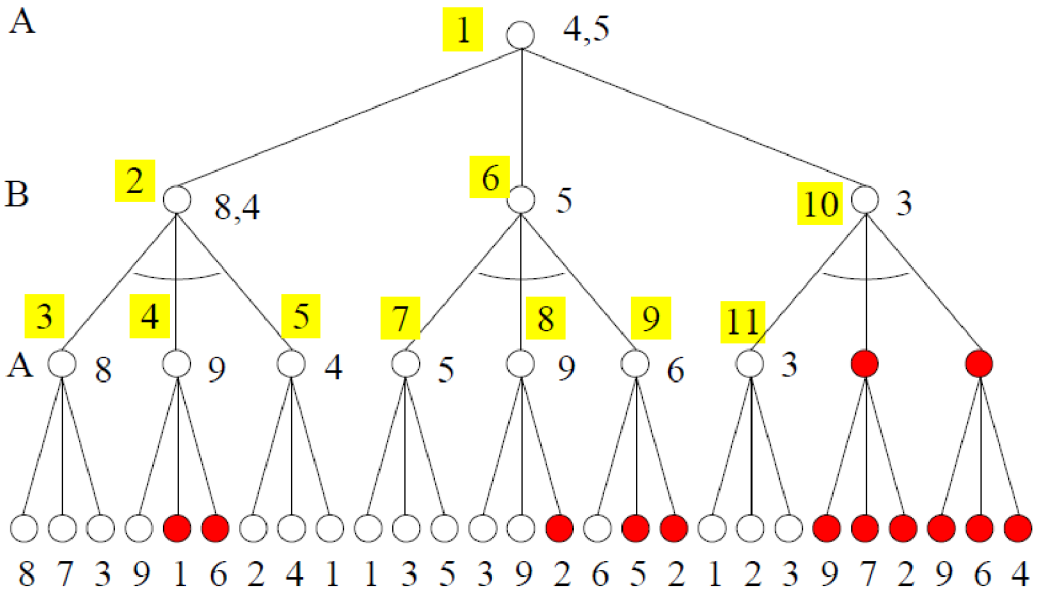
3.2 Prohledávání stavového prostoru

Druhým základním okruhem je prohledávání stavového prostoru. Stavový prostor je množina všech možných stavů, kde stav reprezentuje stav hry. Při hraní her dvou hráčů je stavový prostor uspořádán do stromu. Prohledávání stromu je jedním ze základních pilířů hraní šachů. Umělá inteligence pro hraní šachů většinou nedokáže taktizovat na takové úrovni, aby například vlákala soupeře do pasti. Její největší a zřejmě jedinou zbraní je rychlejší analýza situace a prozkoumání následných tahů do větší hloubky (více tahů dopředu). Proto velice záleží na rychlosti s jakou je program schopen prozkoumat určitou množinu možných tahů z dané pozice a vybrat z ní ten nejvýhodnější. Vzhledem k dnešním výpočetním možnostem počítačů a množině všech možných kombinací tahů a reakcí na ně nikdy nemůžeme z výchozí pozice dojít k výsledku partie výpočtem. Protože máme jen omezené množství času, zkoumáme pouze několik tahů dopředu a vybíráme pozice, které se jeví v nastalé situaci nejslibnější. Pro lepší odhad situace je výhodné prozkoumat co nejvíce tahů dopředu. Následuje představení jednotlivých metod prohledávání stavového prostoru.

3.2.1 MiniMax

MiniMax je algoritmus pro hraní tahových her dvou hráčů [24]. Pro reprezentaci tahů využívá stromovou strukturu, kterou následně prochází a hledá nejlepší řešení pomocí metody prohledávání do hloubky (Depth First Search). Kořen stromu je uzel reprezentující aktuální stav hry. Synovské uzly stromu reprezentují všechny přípustné tahy hráčů z daného stavu hry, kteří se střídají na tahu. Listové uzly stromu reprezentují konečný stav hry nebo stav, který je vyhovující. Příklad průchodu stromem pomocí metody MiniMax je uveden na obrázku 3.1. Z důvodu náročnosti výpočtu a jeho omezené době nebyl ještě nikdy pro hru šachy sestaven úplný strom od výchozího postavení až po výhru jednoho z hráčů. V listových uzlech se ohodnotí stav hry v daném bodě a hodnoty se předávají otcovským uzlům. Hráč A se poté snaží vybírat tahy s maximálním ohodnocením a hráč B s minimálním z pohledu hráče A. Až se prozkoumají všechny uzly, je jasné, který tah z aktuální pozice vede k nejlepšímu stavu hry za určitý počet tahů. MiniMax hledá řešení hrubou silou, prozkoumáním všech variant. Tento přistup je velice nákladný. Uvažujme, že prohledáváme do hloubky stromu 8 (4 tahy hráče) a průměrný větvící faktor je 40 - tak se dostáváme k číslu

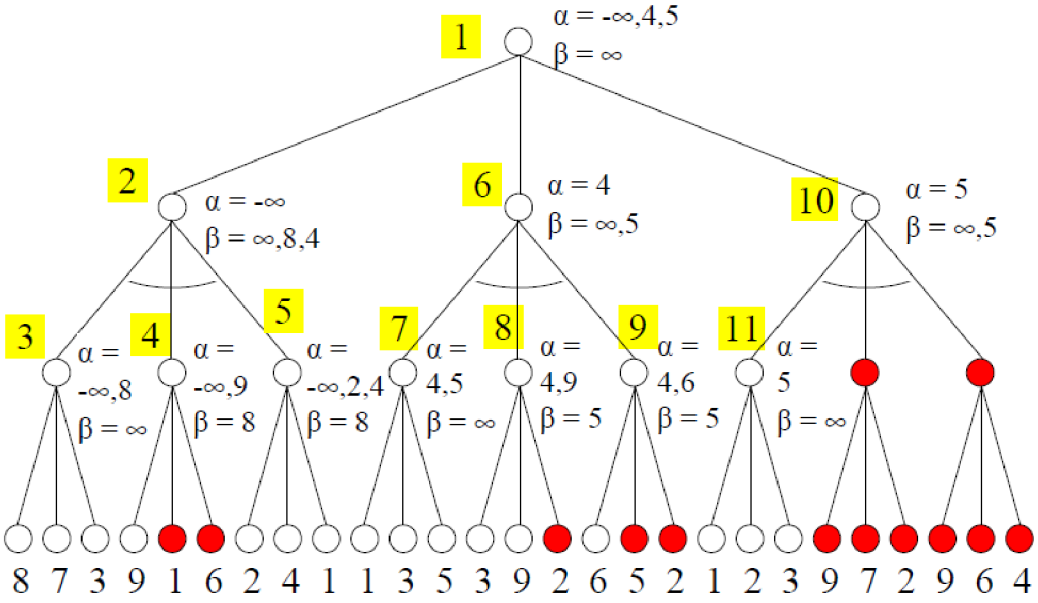
6 553 600 000 000 uzlů, které musíme prozkoumat. Názorně vidíme, že miniMax samotný je nedostačující pro účely hraní her, proto byly zavedeny heuristiky a různá vylepšení pro zefektivnění prohledávání. Jedním z nich je Alfa-beta prořezávání.



Obrázek 3.1: Průchod stromem pomocí algoritmu miniMax (Obrázek ze skript k předmětu IZU [24])

3.2.2 Alfa-beta prořezávání

Alfa-beta je rozšířením algoritmu MiniMax [24]. K vyhodnocování přidává krajní hodnoty alfa a beta, jejichž použitím lze některé uzly vypustit z výpočtu. Jejich ohodnocení je natolik špatné, že na hru nebudou mít žádný efekt. Hodnoty alfa a beta se mění v průběhu výpočtu. Na začátku jsou nastaveny na maximální krajní hodnotu. Alfa je nastavena na $-\infty$ a beta na $+\infty$. Tyto hodnoty se předávají synovským uzlům. Na listové úrovni se vypočítá ohodnocení daného uzlu a tato hodnota je vrácena otcovskému uzlu. Pokud je to uzel hráče A, porovná se vrácená hodnota od syna s hodnotou alfa, a pokud je alfa menší, bude přepsána vrácenou hodnotou. Pokud je to uzel hráče B, porovná se vrácená hodnota od syna s hodnotou beta, a pokud je beta větší, bude přepsána vrácenou hodnotou. Takže alfa obsahuje minimální nalezenou hodnotu pro hráče A, beta maximální nalezenou hodnotu pro hráče B. Vyšetřování dalších uzlů podstromu se zastaví, jakmile je v aktuálním uzlu hodnota alfa $\geq$ beta. Vrátí se nalezená hodnota a otcovký uzel si ji uloží podle jeho typu buď do alfy, nebo do bety. Takto je možné vyhnout se prozkoumávání zbytečných uzlů (variant tahů), které by člověk vyloučil pouhým pohledem. V nejhorším možném případě bude Alfa-beta prohledávat stejný počet uzlů jako MiniMax, ale v nejlepším možném případě pouhých 5 119 999. To je úspora až o 99 %. Příklad průchodu stromem pomocí metody Alfa-beta je na obrázku 3.2. Aby jsme prohledávali co nejméně uzlů, musíme hned ze začátku najít jeden z nejlepších možných tahů. Pro tento účel slouží prohledávání iterativním zanořováním a řazení tahů, které jsou zmíněné v následující části 3.3.1.



Obrázek 3.2: Průchod stromem pomocí algoritmu Alfa-beta (Obrázek ze skript k předmětu IZU [24])

3.2.3 Prohledávání hlavních variant (Principal Variation Search)

Hlavní varianta je dnes zřejmě nejpoužívanější metodou na prohledávání stromových struktur [20]. Hlavní varianta je rozšířením metody Alfa-beta. Je založena na rozdělení uzlů na PV uzly a non-PV uzly.

PV uzel definujeme jako uzel, kde došlo ke zlepšení hodnoty alfa.

PV (Hlavní varianta) je sekvence tahů, které program považuje za nejlepší. Pro rozhodnutí, zda je uzel PV existuje více způsobů:

- tabulka přesunů (Transposition Table) - ohodnocení pozic uložených v tabulkách může odhalit PV uzel,
- oddělená tabulka přesunů (Separate Transposition Table) - oddělení tabulky pro PV uzly,
- sběr v průběhu prohledávání (Collection During Search) - uzly vyhodnocené jako PV jsou ukládány do tabulky, v dalším průchodu jsou preferovány [18].

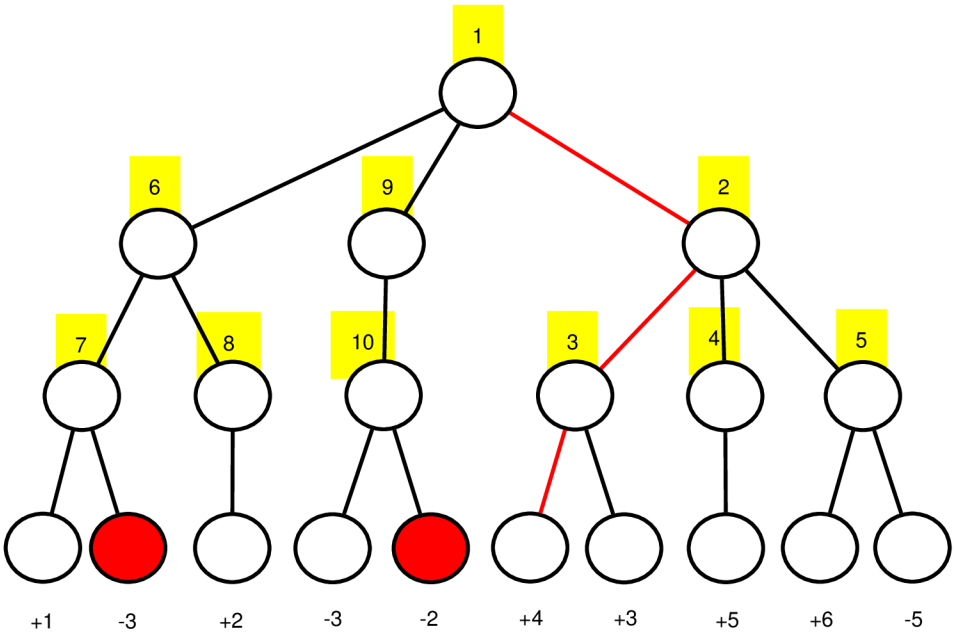
Při prohledávání non-PV uzlu nás nezajímá konkrétní hodnota uzlu, ale pouze zda je horší než nalezené PV uzly [1]. Na základě porovnání můžeme uzel vyříznout ze stromu. Při použití kvalitního řazení tahů a postupného zanořování lze vyříznout až 90 % ostatních uzlů. Myšlenkou je hledat řešení přijatelné pro oba hráče.

> Příliš vysoká hodnota uzlu naznačuje, že jsme prohledáváním našli něco příliš dobrého - to znamená, že protihráč pravděpodobně objeví nějaký způsob, jak se vyhnout této pozici, takže je nutné předpokládat, že tak učiní. Pokud se může

vyhnout dané pozici, nemá smysl prohledávat synovské uzly, protože k tomuto stavu hry nikdy nedojde [19].

Příliš nízká hodnota uzlu naznačuje, že tato pozice pro nás nebyla dost dobrá, tudíž se jí nebudeme snažit dosáhnout, protože jsme schopni nalézt lepší pozice. Nebudeme provádět tah, který umožnil protihráči dostat nás do této pozice [19].

Prohledání prvního uzlu v dané hloubce se děje s otevřeným oknem (alfa = $-\infty$ a beta = $+\infty$) a u dalších uzlů nulovým nebo minimálním oknem.



Obrázek 3.3: Průchod stromem pomocí hlavní varianty s využitím znalostí nejlepšího tahu z předchozí iterace

3.3 Rozšíření používané v moderních programech

Další používaná rozšíření při prohledávání stromu jsou:

- řazení tahů,
- algoritmus NegaMax,
- tabulka přesunů,
- klidové hledání,
- cílené okno.

3.3.1 Postupné zanořování (Iterative Deepening)

Postupné zanořování je vylepšení pro metodu Depth First Search (DFS) využívanou algoritmem MiniMax. Toto vylepšení pracuje s postupně se zvyšující hloubkou zanoření [24]. V prvním průchodu se prozkoumají synovské uzly do hloubky 1, v dalším průchodu do hloubky 2, a tak dále. Sice se několikrát budou prohledávat stejné uzly, ale výhody předčí vynaložené úsilí. Při iterativním zanořování známe pro každou hloubku nejlepší řešení, čili známe nejlepší nalezený tah. To se nám při DFS nemusí povést, pokud od začátku prozkoumáváme uzly vedoucí k nevýhodným tahům. Obzvláště se tato výhoda projeví při omezeném času výpočtu, protože při vypršení času vyhrazenému pro výpočet známe nejlepší tah z předchozí iterace. Efektivitu je možné dále vylepšit použitím techniky řazení tahů.

3.3.2 Řazení tahů (Move Ordering)

Řazení tahů je technika pro zvýšení účinnosti Alfa-beta řezů [20]. Pokud se jako první tah podaří vybrat řešení, které se blíží optimálnímu, pak bude možné provézt více Alfa-beta řezů a nevyšetřovat zbytečné uzly. Tím ve výsledku urychlíme výpočet a můžeme počítat do mnohem větší hloubky. Tahy se seřadí podle výsledku předchozí iterace tak, že nejvýhodnější uzel se přesune na nejlevější větěv stromu, která se prohledává jako první. Takto bude hodnota alfa hned od začátku vysoká a dojde k většímu prořezávání stromu.

3.3.3 Algoritmus NegaMax

NegaMax je varianta MiniMaxu, kde místo dvou procedur Min a Max je pouze jediná procedura NegaMax() [9]. Procedury Min a Max jsou chováním téměř totožné, liší se pouze tím, že vrací hodnoty z pohledu zkoumaného hráče. Procedura NegaMax sdružuje toto chování a vrací relativní hodnoty bez příslušnosti k hráči. Za cenu implementace pouze jediné procedury musíme vyřešit pár následujících implementačních detailů:

- jak inicializovat první volání procedury,
- jak poznat, který tah je nejlepší, když procedura vrací pouze optimální ohodnocení.

Řešením je vracet relativní řešení vždy z pohledu hráče, pro kterého je procedura volána. Ohodnocení musí být symetrické (násobit 1 nebo -1 podle hráče na tahu).

3.3.4 Tabulka přesunů (Transposition Table)

Často se stává, že se stejné stavy šachovnice počítají znovu a znovu. K této situaci dojde záměnou pořadí tahů nebo návratem na původní pozice [14]. Počítání stále stejných stavů je neefektivní, proto si ohodnocení každého stavu uložíme do tzv. hash tabulky a při zkoumání nového stavu pouze zkontrolujeme, zda už jsme ho jednou nepočítali. Výsledky předchozích hledání jsou uloženy v hash tabulce s informacemi o hloubce, ohodnocení pozice, hráčem na tahu, atd. Dojde sice ke zvýšení paměťových nároků, ale urychlí se prohledávání stromu, protože není potřeba znovu počítat pozice již jednou prozkoumané. Nejčastěji se využívá Zobrish hashování 3.1.4, které dosahuje nejlepších výsledků pro šachy.

3.3.5 Klidové hledání (Quiescence Search)

Po skončení hlavního prohledávání je nutné provést klidové hledání [2], protože se může stát, že v dané hloubce se nám tah jeví jako dobrý, ale ve skutečnosti tomu tak nemusí být. Například dáma bere pěšce, ale v hloubce o 1 vyšší by mohl protihráč naši dámu sebrat jiným pěšcem, takže jsme jasně prodělali. K tomuto jevu dochází, protože z nedostatku času není možné prozkoumat následky všech tahů. Z tohoto důvodu se provádí omezené hledání do klidového stavu, kdy už není možné měnit figury, a to z pozice, která se v dané hloubce jeví jako nejlepší. Pokud se narazí na klidový stav a stav hry by byl pro nás nepříznivý, musí se vybrat jiný tah.

3.3.6 Cílené okno (Aspiration Window)

Cílené okno je technika, která slouží ke zvýšení počtu Alfa-beta řezů [16]. Z předchozí iterace známe nejlepší ohodnocení a podle něj se nastaví hodnoty alfa a beta ještě dříve, než začne samotné prohledávání do větší hloubky - to znamená, že hodnoty alfa a beta nastavíme místo $-\infty$ a $+\infty$ v kořeni stromu na hodnoty v určité toleranci od vráceného výsledku předchozí iterace. Tímto omezíme zbytečné prozkoumávání nezajímavých tahů ještě dříve, než začneme prohledávat strom.

3.4 Ohodnocování stavu hry

Ohodnocování je třetí základní okruh hraní šachů. Správné zhodnocení situace je klíčové pro rozhodnutí jak táhnout dále. Existuje spousta možností a heuristik, jak velmi dobře zhodnotit situaci při neomezeném času. Bohužel při hraní šachů máme omezené množství času a ohodnocovací funkce musí být dostatečně rychlá a účinná, aby zbylo dost času na prohledání stromu. Ohodnocování může zahrnovat různé aspekty hry. Nejzákladnější ohodnocovací funkce bere v úvahu materiální hodnotu figur.

3.4.1 Hodnoty jednotlivých figur

Běžně se ve hře lidských hráčů používají hodnoty (král = 20, dáma = 9, věž = 5, střelec = 3, jezdec = 3 a pěšák = 1). Toto dělení je pro výpočty na počítačích nevýhodné, protože by se musela použít aritmetika plovoucí desetinné čárky, která je mnohem náročnější než aritmetika celých čísel [12]. Z tohoto důvodu se používají hodnoty stopěšcové (král = 20000, dáma = 900, věž = 500, střelec = 300, jezdec = 300 a pěšák = 100). Hodnoty jsou pouze orientační a v každém programu mohou být lehce pozměněny, čímž dojde k preferování určitých typů výměn figur.

3.4.2 Dělení hodnocení a základní části

Hodnocení se může provádět z pohledu konkrétního hráče po celou dobu výpočtu nebo vždy z pohledu hráče, který je právě na tahu [10]. Při ohodnocení z pohledu hráče na tahu je nutné provést korekci vrácené hodnoty, protože prohledávaný strom je orientován vždy na jednoho hráče.

Nejjednodušší ohodnocení pozice spočívá v součtu hodnot materiálu (figur) na šachovnici. Při sofistikovanějším ohodnocení se bere do úvahy aktuální pozice figur, jejich pohyblivost

a celková struktura, kterou figury tvoří - to zahrnuje, jak se figury vzájemně kryjí. Nejvýznamnější je struktura pěšců, protože tvoří první linii obrany a určuje další možnou stategii a vývoj hry.

3.4.3 Vlastnosti ohodnocovací funkce

Obecné vlastnosti poskytují základní přehled o požadavcích kladených na ohodnocovací funkce, které by se každá taková funkce měla dodržet. V podstatě existují tři možné varianty [11]:

- Silná ohodnocovací funkce beroucí do úvahy více aspektů pozice a možných následků tahu. Hlavní nevýhodou je časová náročnost.
- Slabá ohodnocovací funkce poskytující minimální dostačující znalost pozice. Výhodou je časová nenáročnost.
- Poslední varianta je někde uprostřed, kdy chceme mít rychlou funkci, která na druhou stranu poskytne maximum možných informací.

Dnes se šachové programy spoléhají spíše na slabší a rychlé ohodnocovací funkce. Pro dosažení rychlého ohodnocení nejvíce se blížícího realitě je vhodné držet se určitých zásad:

- Ortogonalita - ohodnocení různých aspektů pozice by se nemělo zakládat na stejném výpočtu, který by se musel opakovat.
- Spojitost - pokud se rozhodneme zvýhodnit například věž ve volném prostoru, pak by jsme měli přidělit i menší bonus figurám, které zabraňují nepřátelské věži v pohybu na takové pole.
- Smysl pro pokrok - zvýhodňovat tahy, které podporují předem zvolenou strategii (například rošáda na stranu, kde máme více figur).
- Raději špatné chování - tato zásada říká, že je lepší nepatrně podhodnotit všechny pozice, než 99 % ohodnotit naprosto správně a 1 % špatně.

3.4.4 Pohyblivost

Vedle součtu hodnot materiálu je pohyblivost druhou základní částí ohodnocovací funkce [8]. Cena figur sama o sobě nevypovídá o tom, kde figury stojí a kam můžou táhnout. Pro ilustraci můžeme říci, že máme dvě silné věže na šachovnici, ale ty jsou nám k ničemu pokud stojí blokovány v rozích šachovnice. Podle výzkumů je pohyblivost jistým měřítkem úspěchu. Hráči mající více možností kam táhnout, mají také vyšší šanci na výhru. Pohyblivost je možné počítat pouze jako součet možných tahů pro všechny figury. Sofistikovanější počítání pohyblivosti může zahrnovat i pseudo-legální tahy (tahy kde na cílovém poli stojí jiná spřátelená figura), pomocí kterých můžeme kontrolovat, zda a jak máme pokryté figury.

3.4.5 Urychlení výpočtu

Pro urychlení výpočtu se nejčastěji používají předvyplněné hashovací tabulky pro různé účely. Nejvýznamnější tabulkou je hashovací tabulka materiálu [5]. Předpokladem je předpočítání všech kombinací figur na šachovnici, které můžou při hře nastat. Při dnešních pamětech počítačů je tato metoda velmi oblíbená zejména kvůli úspoře času při vhodně

zvolené hashovací funkci. Pro zjednodušení je možné vyloučit velmi nepravděpodobné kombinace jako například 10 jezdců. Druhou variantou je hodnotu jednou vypočítat a uložit do tabulky. V tomto případě je nutné počítat hodnotu pouze při výměnách nebo povýšení pěšce.

Pomocí hash tabulek materiálu se dají detekovat tahy vedoucí k remíze, protože s určitými kombinacemi figur není možne dát soupeři mat. Je proto výhodné vyhnout se výměnám, po kterých by nám zbyly právě tyto kombinace figur.

3.4.6 Poziční tabulky pro jednotlivé figury

Poziční tabulky slouží pro zlepšení ohodnocení figur stojících na správných pozicích a penalizování špatně stojících. Pro každý typ figury je sestavena matice s bonusy a postihy ke každému poli. V následujících tabulkách budou názorně předvedeny bonusy a postihy pro figury [4].

Pro pěšce je výhodné postupovat vpřed a nejlépe na sedmou řadu. Ovšem nejlepší je začít postup se středovými pěšci, zatímco krajní pěšce je lepší pro začátek ponechat v základním postavení, aby bylo možné provést rošádu do chráněné části šachovnice. Proto mají krajní pěšci ve výchozím postavení bonus a středoví mají postih. Bonus ovšem nemůže být takový, že by pěšci šli na jistou smrt. Ukázku pozičních bonusů a postihů pro pěšce je možné vidět v tabulce 3.1.

8	0	0	0	0	0	0	0	0
7	50	50	50	50	50	50	50	50
6	10	10	20	30	30	20	10	10
5	5	5	10	25	25	10	5	5
4	0	0	0	20	20	0	0	0
3	5	-5	-10	0	0	-10	-5	5
2	5	10	10	-20	-20	10	10	5
1	0	0	0	0	0	0	0	0
X	A	B	C	D	E	F	G	H

Tabulka 3.1: Tabulka pro pěšce (Tabulka ze stránky chessprogramming [4])

Jezdce je ideální mít umístěné ve středu šachovnice, kde mají největší manévrovací prostor. U kraje šachovnice je pohyblivost jezdce redukována na polovinu. Ukázku pozičních bonusů a postihů pro jezdce je možné vidět v tabulce 3.2.

Ani pro střelce není výhodné stát u kraje šachovnice, nýbrž uprostřed. I když není postih tak výrazný díky zvýšené pohyblivosti, stále je to nevýhodná pozice. Ukázku pozičních bonusů a postihů pro střelce je možné vidět v tabulce 3.3.

Pozice věže může být víceméně libovolná. Nejlepší ale je vyhnout se sloupcům A a H a různě podporovat postupující figury. Vliv věže narůstá až ke konci partie, kdy je jednou z klíčových figur pro mat. Ukázku pozičních bonusů a postihů pro věž je možné vidět v tabulce 3.4.

8	-50	-40	-30	-30	-30	-30	-40	-50
7	-40	-20	0	0	0	0	-20	-40
6	-30	0	10	15	15	10	0	-30
5	-30	5	15	20	20	15	5	-30
4	-30	0	15	20	20	15	0	-30
3	-30	5	10	15	15	10	5	-30
2	-40	-20	0	5	5	0	-20	40,
1	-50	-40	-30	-30	-30	-30	-40	-50
X	A	B	C	D	E	F	G	H

Tabulka 3.2: Tabulka pro jezdce (Tabulka ze stránky chessprogramming [4])

8	-20	-10	-10	-10	-10	-10	-10	-20
7	-10	0	0	0	0	0	0	-10
6	-10	0	5	10	10	5	0	-10
5	-10	5	5	10	10	5	5	-10
4	-10	0	10	10	10	10	0	-10
3	-10	10	10	10	10	10	10	-10
2	-10	5	0	0	0	0	5	-10
1	-20	-10	-10	-10	-10	-10	-10	-20
X	A	B	C	D	E	F	G	H

Tabulka 3.3: Tabulka pro střelce (Tabulka ze stránky chessprogramming [4])

8	0	0	0	0	0	0	0	0
7	5	10	10	10	10	10	10	5
6	-5	0	0	0	0	0	0	-5
5	-5	0	0	0	0	0	0	-5
4	-5	0	0	0	0	0	0	-5
3	-5	0	0	0	0	0	0	-5
2	-5	0	0	0	0	0	0	-5
1	0	0	0	5	5	0	0	0
X	A	B	C	D	E	F	G	H

Tabulka 3.4: Tabulka pro věž (Tabulka ze stránky chessprogramming [4])

Královna, jako nejsilnější figura, by měla být v místě, odkud může podporovat maximum ostatních figur. Ukázku pozičních bonusů a postihů pro dámu je možné vidět v tabulce 3.5.

8	-20	-10	- 10	-5	-5	-10	-10	-20
7	-10	0	0	0	0	0	0	-10
6	-10	0	5	5	5	5	0	-10
5	-5	0	5	5	5	5	0	-5
4	0	0	5	5	5	5	0	-5
3	-10	5	5	5	5	5	0	-10
2	-10	0	5	0	0	0	0	-10
1	-20	-10	-10	-5	-5	-10	-10	-20
X	A	B	C	D	E	F	G	H

Tabulka 3.5: Tabulka pro dámu (Tabulka ze stránky chessprogramming [4])

Pro krále je situace snažší. V prostřední části hry by se měl držet skrytý za pěšci pro provedení rošády. V koncovce by se měl snažit dostat do středu šachovnice, aby mohl podporovat přeživší pěšce rozmístěné různě po šachovnici.

Nastavením tabulek hodnot pro pozice a různých cen figur můžeme prioritizovat výměnu určitých figur v určitých situacích. Například uvažujeme hodnotu jezdce 320 a střelce 330 - pokud jezdec stojí na poli s bonusem +10 a střelec s postihem -5, je jezdec cennější figurou i přesto, že má nižší vlastní hodnotu. Zvýhodňování se nejvíce projeví při přetlačování o střed šachovnice, který je obecně považován za strategicky nejvýhodnější.

3.4.7 Pěšcové struktury

Klíčovým prvkem pro hraní šachů jsou pěšcové struktury [7]. Jelikož je pěšců nejvíce a je možné je povýšit na libovolnou figuru, má jejich rozestavění zásadní dopad na hru. V první části hry mají pěšci velmi silnou pozici jako obranné figury a jejich rozestavění může úplně znehybnit některé figury (věže nebo střelce). V koncovce je cílem dojít si s pěšcem pro dámu nebo jinou silnou figuru. Schopnost dosáhnout s pěšcem výměny závisí na poměru pěšců stojících proti sobě na šachovnici, a pozici podpůrných figur, nejčastěji krále.

Při zkoumání pěšcové struktury se ignorují všechny ostatní figury a uvažují se pouze pěšci obou hráčů. Podle vzájemného postavení se pěšci kategorizují. Pěšci můžou být blokovaní, zdvojení, izolovaní, potenciální běžci, předstíraní běžci a další. Na základě této klasifikace se rozhodne, zda se s pěšci bude postupovat kupředu nebo budou blokovat soupeřovy pěšce. Klasifikace může být časově náročná, proto se pěšcové struktury hashují do tabulek. Pěšcová struktura se mění velmi zřídka ve střední části hry, takže úspěšnost nalezení v tabulce dosahuje přes 90 %.

3.4.8 Zhodnocení a výběr metod

Během několikatiletého snažení o vytvoření šachového programu, který by porazil člověka, bylo navrhnuto mnoho různých řešení pro prohledávání stromu a ohodnocovací funkce. Některá řešení byly jen pokusy o zrychlení nebo zefektivnění výpočtu, jiná osvědčená řešení byla

dále vylepšována. Výzkum a pokusy v tomto směru dále pokračují, protože ve hře v šach je možné odzkoušet různé nové myšlenky a nápady, které se při zobecnění dají aplikovat i na další problémy umělé inteligence.

Následuje seznam vybraných variant pro jednotlivé části herního jádra.
Pro reprezentaci šachovnice:

- Hybridní přístup - zvýšená paměťová náročnost je převážena jednoduchým přístupem k šachovnici a figurám.

Pro prohledávání:

- Hlavní varianta - efektivnější prohledávání než Alfa-beta nebo MiniMax, nutnost implementovat tabulky přesunů, postupné zanořování a řazení tahů.
- Řazení tahů - vyžadováno prohledávacím mechanismem, ale i tak velice dobrý prostředek pro zvýšení počtu řezů ve stromě.
- Postupné zanořování - vyžadováno prohledávacím mechanismem, samotné nemá příliš velký význam.
- Tabulka přesunů - vyžadováno prohledávacím mechanismem, odpadá nutnost ohodnocovat jednou již vypočítané pozice.
- Klidové hledání - významné pro inteligenci programu, odstraňuje nedostatky plynoucí z omezené hloubky prohledávání.

Pro ohodnocení:

- Tabulky hodnot materiálu - výhodné pro rychlost výpočtu, není nutné počítat znovu jednou již ohodnocené pozice.
- Pěšcové struktury - výhodné pro koncovku a celkovou strategii hry.
- Ohodnocovací funkce.

Kapitola 4

Nutné změny pro implementaci různých variant šachů

Zásadní změnou v implemenaci nových variant oproti klasickým šachům je nemožnost použít většinu optimalizovaných řešení pro reprezentaci šachovnice. Dnes nejvyužívanějším řešením je reprezentace pomocí bitboardů, které ovšem počítají s využitím 64b registru. V nových variantách se vyskytují i šachovnice o větších rozměrech než pouze 64 polí. Z tohoto důvodu není tento způsob vhodný a šachovnici je nutné reprezentovat jiným způsobem. Nejjednodušším řešením se zdá být reprezentace pomocí matic a seznamů figur. Jedna matice bude obsahovat ukazatele na figury stojící na šachovnici a druhá matice bude obsahovat typy těchto figur. Pro urychlení generování tahů by měly být matice obklopeny extra sloupci a řádky, které značí oblast mimo šachovnici. Každý hráč bude mít i seznam figur obsahující všechny jeho figury, i ty zajmuté. Reprezentace pomocí takto provázaných stylů se nazývá hybridní a byla popsána v kapitole 3.1 o reprezentaci šachovnice.

Co se týče prohledávání stavového prostoru, není metody třeba zvláště upravovat, pouze je nutné zajistit bezchybnou komunikaci se šachovnicí, která musi uchovávat kozistentní a synchronizovaná data v maticích a seznamech. Tento požadavek klade vysoký nárok na operace "zahrát tah" a "vrátit tah", které jako jediné mění data v šachovnici a které musí být dostatečně obecné, aby mohly být použity ve všech nových variantách.

Ohodnocovací funkce musí nutně projít úpravou, aby byla schopna počítat s cenou a dalšími aspekty nových figur. Výpis nutných rozšíření pro ohodnocování je následující:

- ohodnotit cenu materiálu všech nových figur,
- změnit tabulky hodnot pro všechny stávající figury, protože některé varianty se hrají na větší šachovnici než 8x8 (například Capablanca nebo Grand Chess),
- přidat tabulky hodnot pro nové figury,
- hashované tabulky materiálu rozšířit o kombinace s novými figurami,
- předělat pěšcové struktury v hash tabulkách.

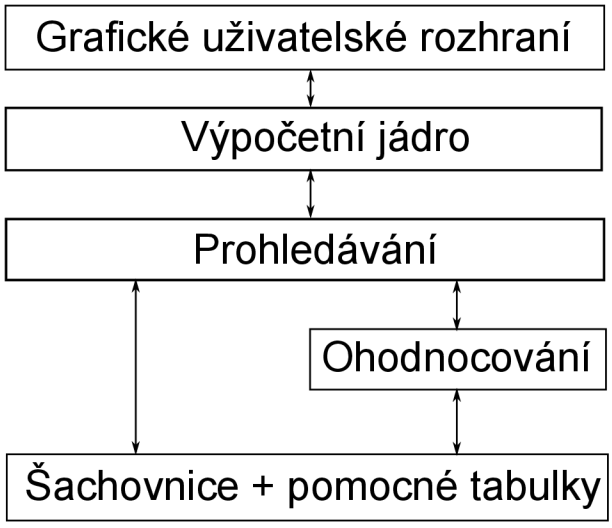
Kapitola 5

Implementace

Tato kapitola popisuje rozdělení programu na moduly, jejich obsah a řešení jednotlivých okruhů uvedených v kapitole 3. Program je návrhově rozdělen do následujících pěti modulů:

- grafické uživatelské rozhraní (GUI),
- knihovna pro komunikaci mezi uživatelským rozhraním a výpočetními moduly,
- knihovna na prohledávání stromu,
- knihovna ohodnocovacích funkcí,
- knihovna pomocných tabulek a tříd.

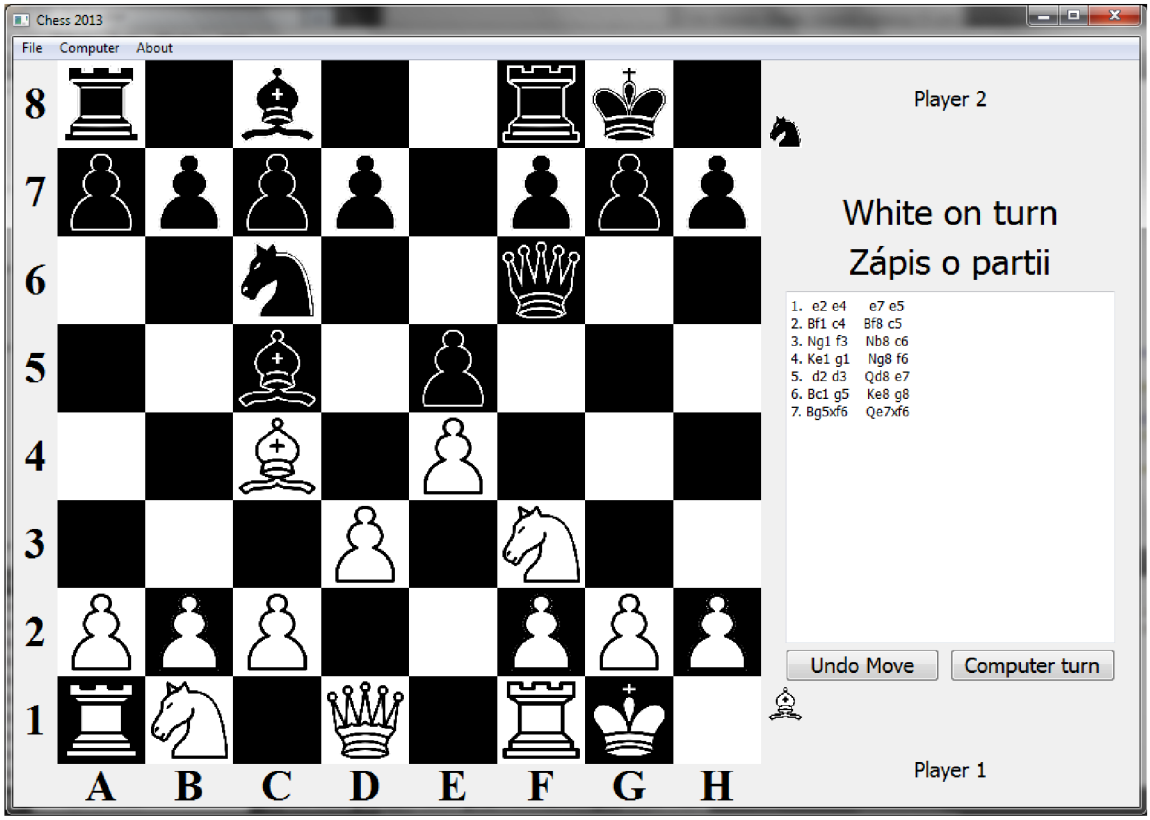
Tento dekomponovaný přístup je vhodný pro oddělenou implementaci modulů a pozdější externí modifikaci programu v případě, že studenti následujících ročníků budou rozšiřovat již hotové šachové implementace ze starších bakalářských prací. Hierarchie modulů je ukázána na obrázku 5.1.



Obrázek 5.1: Hierarchie modulů

5.1 Grafické uživatelské rozhraní

Grafické uživatelské rozhraní je složeno z hlavního okna pro zobrazení šachovnice, jmen hráčů a záznamu tahů. Velikost okna je proměnlivá v závislosti na velikosti aktuálně používané šachovnice ve vybrané variantě šachů. Maximální podporovaná velikost je 16x16 polí. Pro budoucí účely nebo varianty šachů je velikost stanovena na 4 čtvercovitě uspořádané šachovnice. Ukázka uživatelského rozhraní je na obrázku 5.2



Obrázek 5.2: Hlavní okno aplikace

Hra se spustí výběrem File → New Game, který zobrazí dialog pro výběr nové hry. Dialog je ukázán na obrázku 5.3.

Konec hry, detekovaný výpočetním jádrem, je uživateli oznáme pomocí dialogu, ukázaném na obrázku 5.4.

5.2 Výpočetní jádro

V následujících kapitolách je popsána implementace základních prvků a dalších zajímavých detailů hry. Sekce jsou rozděleny podle modulů.

Obrázek 5.3: Dialog pro výběr nové hry

Obrázek 5.4: Dialog po konci hry

5.2.1 Reprezentace šachovnice

Šachovnice je realizována pomocí hybridního přístupu. Pro univerzálnost a rozšiřitelnost má matice šachovnice 16x16 polí (čtyři šachovnice uspořádané do čtverce). Pole jsou číslovány celočíselně po řádcích a 0 je v levém spodním rohu matice. Tato fixní velikost je vhodná pro kontrolu tahů mimo šachovnici.

Figury hráčů jsou uchovávány ve dvou seznamech, jeden pro bílého a druhý pro černého hráče. Každá figura ma svoji celočíselnou pozici a typ (jezdec, střelec, ...). Pro urychlení výpočtu a zpětné mapování z pole na figuru slouží dvě matice. První udává, jaký typ figury se na dané pozici vyskytuje, a druhá matice obsahuje ukazatele na tyto figury. Dále si program uchovává matice ohrožených polí (pro každého hráče zvlášť). Z těchto matice lze jednoduše zjistit, zda je král v šachu nebo v matu, zda lze provést rošádu (pole nejsou napadena soupeřovými figurami), a také slouží k vyloučení nelegálních tahů.

Tento způsob reprezentace zajisté není nejvhodnější, protože použití bitboardů je limitováno na šachovnice 8x8 a moje varianty mají i větší šachovnice. Volil jsem tedy univerzální způsob reprezentace. Cenou za univerzálnost je pomalejší generování tahů, zahrání tahu a zrušení tahu, jelikož je nutné udržet hodnoty všech matic a seznamů aktualizované. Ovšem mnohem výraznějším problémem je výpočetní čas. Pro dosažení podobné odezvy, jaká je u šachů s reprezentací pomocí bitboardů, je nutné omezit maximální hloubku prohledávání na tři tahy.

5.2.2 Prohledávání stromu

Různé varianty prohledávání stromu jsou spolu s generátorem tahů a historií tahů umístěny v oddělené knihovně. Program obsahuje následující varianty prohledávání stromu:

- MiniMax,
- Alfa-beta,
- Prohledávání hlavní variantou.

Hlavní varianta je uvažována jako výchozí metoda pro prohledávání. Ostatní metody jsou implementovány pro experimentální a statistické účely a především pro srovnání efektivity jednotlivých metod. Dále jsou implementovány metody nezbytné pro varianty prohledávání. Uvažované a nezbytné metody jsou vypsány v kapitole 3.4.8 a zobrazeny na diagramu 5.5. Knihovna bude ovládána obecnou třídou, jejíž veřejné metody (”najdi další tah”, ”ověř tah hráče”apod.) budou mapovány na konkrétní metody implementované k různým variantám prohledávání. Toto chování umožní rozšířit implementaci pouhým doplněním nových variant a přemapováním volaných metod.

Základní metoda prohledávání MiniMax implementuje tyto veřejné metody a ostatní metody prohledávání je od třídy MiniMax dědí. Prohledávání do omezené hloubky potřebné pro prohledávání MinMax a Alfa-beta je implementováno rekurzivně se střídajícími se metodami Max a Min. Prohledávání Alfa-beta těmto metodám přídává dva parametry alfa a beta na základě, kterých se provádí prořezávání stromu.

Obrázek 5.5: Metody pro Prohledávání hlavní variantou s pokročilým ohodnocením

Prohledávání Hlavní variantou využívá jiný typ prohledávání stromu, a to postupné zanořování, kde se strom prohledává od hloubky 1 postupně až do maximální zadané hloubky. V každé hloubce si program zapamatuje pousloupnost tahů vedoucí k nejlepšímu tahu v dané hloubce. V následující iteraci je tato posloupnost přesunuta na nejlevější větev stromu, aby se vyhodnotila jako první a nastavila vysoké hodnoty alfa a beta. Tímto se prořezávání stromu dále zefektivní. Tato heuristika se nazývá historická a je často využívána pro urychlení výpočtu. V programu je realizována dvěma zásobníky. První pracovní zásobník si ukládá tahy, jejichž ohodnocení je vyšší než aktuální hodnota alfa a nižší jak beta. Pouze tyto tahy mají šanci ovlivnit hru. Druhý finální zásobník si uchovává cestu vedoucí k nejlepší pozici. Pokud je tah přidán do pracovního zásobníku v kořenovém uzlu, je cesta uložená v pracovním zásobníku přenesena do finálního zásobníku. Níže jsou pro názornost uvedeny pseudokódy pro metody PVSMax 1 a PVSMin 2.

Dalším významným prvkem je klidové hledání. K volání klidového hledání dochází v listovém uzlu a pouze pokud byla zahráním tahu vyhozena figura. Klidové hledání prohledává podstrom s kořenem v listovém uzlu a uvažuje pouze tahy, ve kterých dochází k výměně figur. Prohledávání je implementováno pomocí metod Max a Min a předáváním hodnot alfa a beta. Hodnoty alfa a beta jsou převzaty z klasického prohledávání, ale jejich modifikace se nijak do klasického prohledávání nepromítne. Slouží pouze k prořezáni stromu, protože při složitějších výměnách může prohledávaná hloubka dosahovat až několikanásobku nastavené maximální hloubky. K ohodnocení uzlu dochází až v případě, kdy nejsou žádné další tahy, ve kterých by došlo k vyhození figury. Výsledné ohodnocení celého podstromu je poté vráceno jako ohodnocení listového uzlu. Prohledávání všech podstromů v listových uzlech vede k znatelnému zlepšení hratelnosti, ale také ke zpomalení výpočtu. V některých případech může program strávit více času v klidovém hledání než v klasickém. Proto je nutné řadit tahy podlé určité heuristiky, aby i v podstromu klidového hledání docházelo k prořezávání. Zvolil jsem heuristiku MVV/LVA (Most Valuable Victim - Least Valuable Aggressor). Tato heuristika řadí tahy podle materiálního rozdílu figur, jak vyplývá z názvu. Prvně se prohledávají tahy, kde slabá figura bere silnou. Tento přístup nastaví alfu a betu do ideálních hodnot pro prořezávání. Pro znázornění jsou průměrné počty uzlů z prvních 11 tahů uvedeny v následující tabulce 5.1.

Algorithm 1 Pseudokód pro metodu PVSMax

```
function PVSMax(hloubka)
    if hloubka je 0 then
        if posledni tah sebral figuru then return Klidové hledání
        else if náš král v šachu then return PVSMax(2)
        else return Ohodnoť pozici
        end if
    end if
    Vygeneruj tahy
    for pro každý tah do
        Zahraj tah
        if náš král v šachu then
            Vrať tah
            Pokračuj dalším tahem
        end if
        skóre ← PVSMin(hloubka - 1)
        Vrať tah
        if skóre < alfa then
            alfa ← skóre
            if alfa < beta then
                ulož tah do zasobníku nejlepších tahů
            end if
        end if
        if beta ≤ alfa then return alfa
        end if
    end for
return alfa
end function
```

Algorithm 2 Pseudokód pro metodu PVSMin

```
function PVSMin(hloubka)
    if hloubka je 0 then
        if posledni tah sebral figuru then return Klidové hledání
        else if náš král v šachu then return PVSMin(2)
        else return Ohodnoť pozici
        end if
    end if
    Vygeneruj tahy
    for pro každý tah do
        Zahraj tah
        if náš král v šachu then
            Vrať tah
            Pokračuj dalším tahem
        end if
        skóre ← PVSMax(hloubka - 1)
        Vrať tah
        if skóre < beta then
            beta ← skóre
            if alfa < beta then
                ulož tah do zasobníku nejlepších tahů
            end if
        end if
        if beta ≤ alfa then return beta
        end if
    end for
return beta
end function
```

hloubka	Minmax	Alfa-beta	PVS	PVS + klidové hledání
1	14 104	207	244	1 579
2	5 443 843	13 298	15 015	28 513
3	1 795 477 151	757 238	876 746	1 907 077

Tabulka 5.1: Průměrný počet prohledávaných uzlů pro jednotlivé typy prohledávání po 11 tazích

Z tabulky je patrné, že prohledávání pomocí hodnot alfa a beta výrazně snížilo počet prozkoumaných uzlů. Po prořezání se prozkoumalo pouze 0,042 % původních uzlů při hloubce 3. Dále je vidět, že prohledávání postupným zanořováním nepřidalo příliš mnoho dalších uzlů, konkrétně 15,78 % původních uzlů. Klidové hledání naopak výrazně zvedlo počet uzlů, a to až na 151,85 % původní hodnoty. Tento fakt je ovšem tolerovatelný vzhledem k obrovskému nárůstu hratelnosti.

Knihovna dále obsahuje generátor tahů. Tahy všech klasických figur i figur v implementovaných variantách je možné vygenerovat z pěti základních tahů:

- tah bílého pěšce,
- tah černého pěšce,
- tah jezdce,
- tah střelce,
- tah věže.

Vygenerované tahy jsou pseudolegální (po zahrání tahu se může král hrající strany dostat do šachu, nebo již v šachu je a tah tomuto šachu nezabrání) a samotná legalita tahu je kontrolována v průběhu prohledávání stromu. Pokud se generuje nejlevější větev stromu, generátor využívá sekvenci tahů z předchozí iterace.

5.2.3 Knihovna pro ohodnocovací funkce

Knihovna pro ohodnocování obsahuje jedinou třídu Evaluation, která realizuje ohodnocení aktuálního stavu šachovnice na základě předaného typu ohodnocení. Ohodnocení uzlu bere v úvahu cenu figur a jejich pozici. Pohyblivost ani pěšcové struktury nejsou implementovány, i když by jejich implementace zlepšila herní vlastnosti programu. Dále knihovna detekuje koncovku na základě hodnoty materiálu a počtu zbývajících figur na šachovnici. Na základě této detekce nahraje jiné poziční tabulky hodnot pro krále a pěšce, aby změnili své chování. V závěru partie je výhodné směřovat králem do středu šachovnice nebo dovést pěšce do proměny. Pěšci místo defenzivních struktur začnou směřovat k povýšení a následnému matu soupeře.

5.2.4 Rozhraní pro Manažer stolních deskových her

Knihovna s herním jádrem je upravena tak, aby i místo klasické hry obsahovala metody umožnující integraci do "Manažeru stolních deskových her". Tento manažer je výsledkem bakalářské práce pana Kouřila [17] a umožnuje hru dvou programů s různou implementací šachů proti sobě při dodržení komunikačního protokolu. Složka **bin** obsahuje spustitelný program s daným rozhraním pro tento manažer s názvem **Dostal_manazer**.

Kapitola 6

Testování

Pro účely testování a optimalizaci programu je nutné ověřit výpočetní schopnosti proti reálnému soupeři. Je sice možné sputit hru počítače proti počítači, ale ta nemá vypovídající hodnotu o reálné konkurenceshopnosti programu.

Pro testování bylo zvoleno následující pořadí protivníků:

- autor,
- předchozí školní programy z minulých bakalářských prací,
- dobrovolní lidští protihráči různé úrovně,
- volně šířitelné programy.

Jedním z volně šířitelných programů je ChessV vytvořený Gregory Strongem. Aktuální počet variant, které program umí hrát, je přes 50 různých variant šachů. Dalším programem je SMIRF od Reinharda Scharnagla, který je autorem varianty Capablanca Random Chess.

Pro automatizované ověření konkurenceschopnosti je na základě doporučení pana Rozmana implementováno rozhraní pro "Manažer stolních deskových her" a testováno proti šachovému programu kolegy Zbyňka Jadrníčka [15]. Kvůli různosti implementovaných variant bylo možno otestovat pouze hru klasických šachů. Po zanalyzováním výsledků tohoto testování lze říci, že program hraje klasické šachy na středně pokročilé úrovni, která odpovídá rozsahu a účelu práce.

Testování zahrnovalo 10 partií klasických šachů se střídaním barev hráčů. Výsledky jsou shrnuty v tabulce 6.1.

autor	počet her	vítězství	remíz	proher
Martin Dostál	10	0	10	0
Zbyněk Jadrníček	10	0	10	0

Tabulka 6.1: Shrnutí výsledků automatizovaného testování ze dne 12.05.2013

Pouhých deset opakování je dostačující, protože oba programy hrají deterministicky a výsledky partií jsou vždy stejné. Oba programy prohledávají do hloubky 3 tahy dopředu a

po určitém počtu tahů se partie dostanou do vyrovnané pozice, kde by bylo potřeba prohledávat více tahů dopředu, aby bylo nalezeno přijatelné řešení, nebo změnit ohodnocení patu po trojím opakování pozic. Jinak programy vyhodnotí tah vedoucí k patu jako nejlepší možný v dané situaci. V průběhu testování bylo zjištěno, že "Manažer stolních deskových her" nepodporuje braní mimochodem, což značně komplikuje testování i když tento tah není příliš častý.

Závěr

Vytvořit kvalitní šachový program není triviální záležitost. Již jen samotný výběr použitých metod velmi výrazně ovlivnil celkovou úspěšnost programu. Použití základních metod zaručilo jistou úrověň hratelnosti, ale jistě ne na úrovni pokročilých hráčů. Pokročilé metody podaly mnohem uspokojivější výsledky. Jejich nevýhodou ale byla nutnost implementovat další podpůrné prostředky a metody pro úspěšný výpočet. Program se tímto rozrostl a byl více náchylný k chybám, které mohly celý výpočet překazit. Uvažovaná dekompozice programu přinesla určitou režiji, která je ovšem v daném rozsahu a účelu práce akceptovatelná. Vzhledem k možnosti dalšího rozvoje práce bylo důležitým cílem dosáhnutí snadné rozšiřitelnosti a modifikovatelnosti programu.

Vytvořený šachový program vykazuje konkurenceschopnost na úrovni středně pokročilého hráče. Objevily se ovšem situace, se kterými si program nedokázal poradit - toto bylo způsobeno chybějící implementací volitelných metod, které nebylo možné doplnit z důvodu již vysoké komplexnosti programu. Mezi tyto volitelné metody patří tabulky přesunů, pěšcové struktury a pohyblivost figur na šachovnici. Tyto metody ovšem nejsou klíčové pro funkčnost programu a jejich absence nemá zásadní dopad na hratelnost.

Program splnil požadavky vyplývající ze zadání práce, ale zcela jistě se nedá prohlásit za kompletní. Při dalším rozvoji práce bych rád doplnil chybějící části zmíněné v předchozím odstavci, tzn. volitelné metody, které mohou vylepšit celkovou hratelnost programu. Kompletní změnou by měla projít reprezentace šachovnice, kde by na místo šachovnicové matice byla mnohem vhodnější bitboardová reprezentace. Pro využití bitboardů na rozšířených šachovnicích, které musí být nutně rozděleny do více registrů, je nutné vyvinout vhodnou formu bitových operací mezi jednotlivými částmi šachovnice. Dalším zajímavým prvkem je také jistě paralelizace výpočtu, pro kterou již existuje řada navrhovaných řešení a postupů.

Literatura

[1] Akl, S.; Newborn, M.: *The Principal Continuation and the Killer Heuristic*. Seattle, WA: ACM Annual Conference Proceedings, 1977, 466–473 s.

[2] Beal, D.: A Generalized Quiescence Search Algorithm. *Artificial Intelligence*, ročník 3, č. 1, 1990: s. 85–98, ISSN 0004-3702.

[3] Chalupa, I.: *Historie šachu*. Praha: LIKA KLUB, 2012, ISBN 978-80-86069-76-0, 12–18 s.

[4] ChessProgramming: *Simplified evaluation function*. [online], Poslední modifikace: 08.05.2012, [cit. 07.12.2012]. Dostupné z: http://chessprogramming.wikispaces.com/Simplified+evaluation+function.

[5] ChessProgramming: *Material Tables*. [online], Poslední modifikace: 09.04.2013, [cit. 07.12.2012]. Dostupné z: http://chessprogramming.wikispaces.com/Material+Tables.

[6] ChessProgramming: *Board Representation*. [online], Poslední modifikace: 11.04.2013, [cit. 25.10.2012]. Dostupné z: http://chessprogramming.wikispaces.com/Board+Representation.

[7] ChessProgramming: *Pawn Structure*. [online], Poslední modifikace: 11.06.2012, [cit. 10.12.2012]. Dostupné z: http://chessprogramming.wikispaces.com/Pawn+Structure.

[8] ChessProgramming: *Mobility*. [online], Poslední modifikace: 15.03.2013, [cit. 25.11.2012]. Dostupné z: http://chessprogramming.wikispaces.com/Mobility.

[9] ChessProgramming: *Negamax*. [online], Poslední modifikace: 22.07.2012, [cit. 10.11.2012]. Dostupné z: http://chessprogramming.wikispaces.com/Negamax.

[10] ChessProgramming: *Evaluation*. [online], Poslední modifikace: 23.04.2013, [cit. 23.11.2012]. Dostupné z: http://chessprogramming.wikispaces.com/Evaluation.

[11] ChessProgramming: *Evaluation Philosophy*. [online], Poslední modifikace: 27.03.2013, [cit. 25.11.2012]. Dostupné z: http://chessprogramming.wikispaces.com/Evaluation+Philosophy.

[12] ChessProgramming: *Point Value*. [online], Poslední modifikace: 28.03.2013, [cit. 23.11.2012]. Dostupné z: http://chessprogramming.wikispaces.com/Point+Value.

[13] Eppstein, D.: *ICS 180, April 8, 1997*. [online], Poslední modifikace: 14.04.1997, [cit. 27.10.2012]. Dostupné z: http://www.ics.uci.edu/~eppstein/180a/970408.html.

[14] Greenblatt, R.; Eastlake, D.; Crocker, D. S.: The Greenblatt Chess Program. In *Proceedings of the AfiPs Fall Joint Computer Conference*, 1967, s. 801–810.

[15] Jadrníček, Z.: *Šachový program pro různé varianty šachů, bakalářská práce*. Brno: FIT VUT v Brně, 2013.

[16] Kaindl, H.; Shams, R.; Horacek, H.: Minimax Search Algorithms with and without Aspiration Windows. *Pattern Analysis and Machine Intelligence*, ročník 13, č. 12, 1991: s. 1225–1235, ISSN 0162-8828.

[17] Kouřil, J.: *Manažer stolních deskových her, bakalářská práce*. Brno: FIT VUT v Brně, 2009.

[18] Moreland, B.: *Collecting the Principial Variation*. [online], Poslední modifikace: 04.11.2002, [cit. 02.11.2012]. Dostupné z: http://web.archive.org/web/20040427013839/brucemo.com/compchess/programming/pv.htm.

[19] Moreland, B.: *Glossary*. [online], Poslední modifikace: 24.01.2003, [cit. 05.11.2012]. Dostupné z: http://web.archive.org/web/20040512194831/brucemo.com/compchess/programming/glossary.htm.

[20] Pettersson, J.: Mediocre Chess. [online], 2007, [cit. 03.11.2012]. Dostupné z: http://mediocrechess.sourceforge.net/guides/aspirationwindows.html.

[21] Pettersson, J.: *Mediocre Chess*. [online], 2007, [cit. 13.02.2013]. Dostupné z: http://mediocrechess.sourceforge.net/guides/zobristkeys.html.

[22] Rachůnek, F.: *Brainking - Pravidla her*. [online], 2013, [cit. 23.10.2012]. Dostupné z: http://brainking.com/cz/GameRules.

[23] Wikipedia: *Braní mimochodem*. [online], Poslední modifikace: 14.03.2013, [cit. 02.05.2013]. Dostupné z: http://cs.wikipedia.org/wiki/Bran%C3%AD_mimochodem.

[24] Zbořil, F.; Zbořil, F.: *Základy umělé inteligence - Studijní opora*. VUT FIT Brno, třetí vydání, 2006.

Seznam obrázků

Seznam tabulek

Příloha A

Obsah CD

Přiložené CD obsahuje zkompilovaný program, zdrojové kódy a technickou zprávu. Dále složky k Manažeru stolních deskových her a poslední verzi programu pana Zbyňka Jadrníčka.