

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

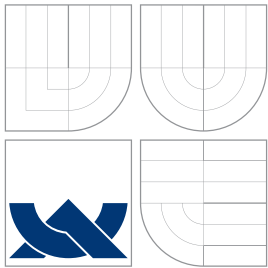
MANAŽER STOLNÍCH DESKOVÝCH HER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

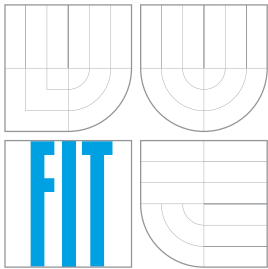
AUTOR PRÁCE
AUTHOR

JIŘÍ KUSÁK

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

MANAŽER STOLNÍCH DESKOVÝCH HER

MANAGER FOR BOARD GAMES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ KUSÁK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAROSLAV ROZMAN

BRNO 2007

Zadání bakalářské práce

Manažer stolních deskových her

Manager for Board Games

Vedoucí:

Rozman Jaroslav, Ing., UITS FIT VUT

Oponent:

Jurka Pavel, Ing., UITS FIT VUT

Přihlášen:

Kusák Jiří

Zadání:

1. Seznámit se s nejčastějšími hrami (piškvorky, reversi, dáma, šachy, GO).
2. Navrhnout program pro řízení hry dvou programů proti sobě.
3. Navržený program implementovat včetně grafického uživatelského rozhraní.
4. Zhodnotit dosažené výsledky.

Kategorie:

Umělá inteligence

Literatura:

Internet

Licenční smlouva

Licenční smlouva je uložena v archivu Fakulty informačních technologií Vysokého učení technického v Brně.

Abstrakt

Hlavním cílem práce je návrh a realizace aplikace pro řízení dvou programů (hráčů) proti sobě. Z počátku v textu pojednávám obecněji o deskových hrách (zvláště jejich pravidlech) a systémech hodnocení výkonnosti hráčů. Po tomto bloku následuje krátké zhodnocení aktuálního stavu zpracovávané problematiky. Stěžejní částí práce je ale popis návrhu a realizace vytvořené aplikace. Nato je uveden stručný popis ovládání aplikace. V závěru jsou zhodnoceny dosažené výsledky a načrtnuty možné další směry vývoje programu.

Klíčová slova

manažer, deskové hry, šachy, dáma, reversi, piškvorky, Go, C#, .NET, umělá inteligence, ELO

Abstract

The main goal of this work is to design and realize an application for managing two programs playing against each other. At the beginning I'm discussing board games (especially their rules) and player's skill rating systems. Then there is short analysis of the actual situation in this area. Fundamental part of the work is the description of the designing and realization of the manager. After that there is short description how to control the application. In the end I'm evaluating achieved results and proposing additional directions for the program development.

Keywords

Manager, Board Games, Chess, Checkers, Reversi, Gomoku, Go, C#, .NET, Artificial Intelligence, ELO

Citace

Jiří Kusák: Manažer stolních deskových her, bakalářská práce, Brno, FIT VUT v Brně, 2007

Manažer stolních deskových her

Prohlášení

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

.....

Jiří Kusák
12. května 2007

Poděkování

Chtěl bych poděkovat vedoucímu práce, Ing. Jaroslavu Rozmanovi, za inspiraci a spolehlivé vedení. Rovněž bych chtěl ocenit jeho přístup k vzájemné spolupráci.

© Jiří Kusák, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Deskové hry	4
2.1	Historie her	4
2.2	Dělení her	4
2.3	Pravidla	5
2.3.1	Reversi	5
2.3.2	Šachy	6
2.3.3	Dáma	7
2.3.4	Gomoku	8
2.3.5	Go	8
2.4	Hráči	8
2.5	Počítačové hraní	8
3	Turnaje a hodnotící systémy	9
3.1	Systém ELO	9
3.1.1	Matematické detaily	10
3.2	Turnaje	11
4	Současný stav problematiky	12
4.1	Brainy	12
4.2	Rozhraní	13
4.3	Komunikace	13
4.4	Manažery, GUI	13
4.5	Turnaje	14
5	Návrh aplikace	15
5.1	Funkčnost programu	15
5.2	Komunikace	16
5.3	Turnaj	18
5.4	Záznamy her	18
5.5	Nápověda	19
5.6	GUI	19
6	Realizace	20
6.1	Brain	20
6.2	Datové struktury	20
6.3	Popis řešení	22

6.3.1	Komunikace	22
6.3.2	Záznamy her	22
6.3.3	Nápověda	23
6.3.4	Turnaj	23
6.4	Poznámky	24
7	Ovládání programu	25
7.1	Samostatná hra	25
7.2	Turnaj	26
7.3	Záznamy her	26
8	Závěr	29

Kapitola 1

Úvod

Soupeření na poli deskových her má velmi dlouhou tradici. Jde nejčastěji o souboje dvou lidí, ale stále častěji můžete zachytit zprávu o hře člověka proti počítači nebo dokonce počítačů proti sobě. Právě poslední zmiňovaná možnost je předmětem zájmu této práce. Inspirací pro vznik této práce byla aktivita *Gomocup*, o které se více dozvíte na webové adrese [3].

Oficiální zadání práce znělo:

1. Seznámit se s nejčastějšími hrami (piškvorky, reversi, dáma, šachy, GO).
2. Navrhnout program pro řízení hry dvou programů proti sobě.
3. Navržený program implementovat včetně grafického uživatelského rozhraní.
4. Zhodnotit dosažené výsledky.

K těmto oficiálním požadavkům se pak přidávají mé osobní cíle, kterých bych chtěl dosáhnout řešením projektu. Jde především o seznámení se s programovacím jazykem C# a platformou .NET.

V následující kapitole se nachází obecné pojednání o deskových hrách. Jde především o pravidla jednotlivých her, ale najdete tam i krátký pohled do historie, informace o dělení her a počítačovém hraní. Ve třetí kapitole je něco málo o turnajích, ale především se zde nalézá detailní rozbor systému hodnocení výkonnosti hráčů ELO. Následuje kapitola, kde se zabývám současným stavem v oblasti hraní deskových her na počítači.

Dále je pak kapitola zabývající se popisem návrhu aplikace. Zde můžete najít, jaké vlastnosti by měl manažer splňovat a jaké funkce by měl mít. Zde jsou návrhy řešení, které byly následně v programu realizovány. Najdete tu volby použitých technologií a jejich zdůvodnění. V další kapitole pak je vlastní technický popis řešení.

Osmá kapitola pak obsahuje popis ovládání programu, včetně ilustračních obrázků. V závěru pak hodnotím dosažené výsledky a navrhuji směry, kterými by se mohl ubírat další vývoj aplikace.

Kapitola 2

Deskové hry

Deskové hry jsou takové hry, které se hrají s herními kameny nebo jinými figurami, jež se umisťují na (přemisťují po, odebírají z) herní desku. Tato deska bývá velice často jednoznačně dána a pro konkrétní hru vytvarována. Některé hrací desky jsou velice jednoduché (za všechny uveďme třeba šachovnici), některé jsou naopak velmi komplikované a rozsáhlé (toto se týká především novějších her, příkladem mohou být třeba *Osadníci z Katanu*). V této práci se však zabývám jen klasickými deskovými hrami (právě do této kategorií spadají Go, gomoku, šachy, dáma a reversi), proto je následující pojednání zaměřeno právě na tyto hry.

2.1 Historie her

Zmínku o deskových hrách můžeme nalézt již u nejstarších civilizací. Možná nejstarší deskovou hrou je *Senet*. Je to egyptská hra datovaná do období 3500 př.n.l. Velmi stará je i hra *Go*, která vznikla v Číně a první zmínka o ní je ze 6.století př.n.l., gomoku je pak ještě mladší.

Šachy, tak, jak je dnes známe, se objevily až v 15.století n.l. Přesto (nebo možná i kvůli tomu), že je tato hra relativně mladá, získala si vysokou popularitu a v současné době patří mezi nejpopulárnější deskové hry. Zjednodušenou verzí šachů je pak dáma, která se hraje na stejné herní desce, avšak s jinými pravidly (o pravidlech jednotlivých her se ještě zmíním).

Nejmladší hrou z mnou zpracovávané pětice je reversi. Často se můžete také setkat s označením *Othello*. Ostatně fakt, že jedna konkrétní hra je známa pod několika názvy není nijak ojedinělý. Hra reversi původně vznikla v Anglii na konci 19.století, ovšem její současná pravidla byla vyhlášena až v roce 1970 a to právě pod názvem *Othello*.

2.2 Dělení her

Deskové hry můžeme dělit podle několika kritérií. Jedním ze samozřejmých je počet hráčů. Většina tradičních deskových her je pro dva hráče – vzpomeňme již zmíněné Go, šachy, dámu, reversi, ale třeba i *backgammon* nebo *Stratego*. Naopak novější hry (často nezaložené na principu abstrakce) bývají pro více než dva hráče - příkladem mohou být třeba hry „německého typu“, které jsou v současnosti velice oblíbené. (Jsou více strategicky zaměřené, ale ne natolik, aby to bránilo méně strategicky založeným hráčům v jejich hraní, obvykle jsou středně časově náročné – jejich partie nejsou z nejkratších, nejsou ani neúnosně dlouhé a jako základ, na kterém staví, si berou nějaký aspekt reálného života, takže nevyžadují

nijak zvlášť vysokou úroveň abstrakce.) Zástupci tohoto proudu deskových her jsou třeba *Osadníci z Katanu* nebo *Carcassonne*.

Jiným možným dělením je rozdělení her na ty, které jsou postaveny na prvku náhody a na ty, jež jsou založeny na čistě strategických principech. Hry založené na náhodě jsou populární především u dětí. Takovéto hry mají především funkci zábavnou a soutěžní složka je až druhořadá. Nicméně neplatí to zcela stoprocentně - výjimkou mohou být již zmínění *Osadníci z Katanu*. V této hře je sice vše podřízeno náhodě (přidělování surovin závisí na hodě šestistěnnou kostkou), ale strategická složka je také velice důležitá.

Vyšší oblibu u dospělé části hráčské populace si však získaly hry, které neponechávají žádný nebo jen minimální podíl náhodě. U takových her je možno plně využít strategický potenciál daného hráče, aniž by byl jakkoliv srážen prvkem náhody. Do této kategorie také spadají hry, pro které je manažer navrhován a jejichž zpracování má značnou tradici v oboru umělé inteligence.

Pro úplnost ještě uvedu dělení na hry, které staví na reálném základu a hry abstraktní. Do té první kategorie patří kupříkladu hra *Monopoly*, která je z prostředí obchodu s nemovitostmi. Mnou zpracovávané hry patří do kategorie druhé a to konkrétně do její podoblasti her strategických. Do této kategorie ale taky spadají hry se slovy (třeba *Scrabble*) a kvízové hry.

Pokud se o těchto věcech chcete dozvědět více, doporučuji anglický materiál [14], ze kterého jsem čerpal i já.

2.3 Pravidla

Pravidla většiny zde zpracovávaných her jsou velice dobře známa i široké veřejnosti, přesto zde některá základní (nebo naopak některá neznámá či nedodržovaná) pravidla uvedu. Podrobné seznámení s pravidly je ostatně nutnou podmínkou k úspěšné realizaci tohoto projektu. Jen pokud jste si dokonale vědomi všech pravidel dané hry a jejich důsledků, můžete vytvořit program, který se bude chovat správně.

2.3.1 Reversi

Reversi se tradičně hraje na desce o velikosti 8x8 polí. Takovou desku si můžete prohlédnout například zde [18] nebo na obrázku 7.1. Na začátku partie se na hrací desku položí čtyři kameny, jak je znázorněno na výše uvedené webové adrese. Někdy se ale hraje i na desce menší o rozměrech 6x6 nebo na desce větší o rozměrech 10x10. Realizovaný program počítá jen s velikostí 8x8, při troše úsilí a práce by šel rozšířit i o podporu pro hrací pole jiných velikostí.

Hra se hraje s 64 kameny, které jsou z jedné strany černé a z druhé bílé (za předpokladu, že barvy hráčů jsou právě tyto, což je tradiční přístup, který je v mém programu zachován). Začíná černý hráč (toto pravidlo není ovšem v programu dodrženo a to proto, aby se mohli hráči střídát při začínání, aniž by měnili svou identitu), hráči se při tazích střídají v sérii tahů. Každý tah musí splňovat tato kritéria:

- Hráč, který je na tahu, umístí nový kámen své barvy tak, aby tímto tahem zajal minimálně jeden kámen spoupeře.
- Zajetím se myslí obestoupení soupeřových kamenů kameny vlastní barvy a jejich následné otočení na druhou stranu (budou tedy mít barvu hráče, který zajetí provádí).

- Toto zajímání se provádí ve všech směrech, tedy svisle, vodorovně i diagonálně. A to ve všech směrech zároveň.
- Pokud hráč nemůže provést žádný tah podle pravidel, vzdá se tahu.
- Pokud hráč naopak může provést tah podle pravidel, provést jej musí.

Hra končí položením všech 64 kamenů na hrací plochu. Vítězí hráč, který má na desce více kamenů.

Pokud máte zájem o oficiální české znění pravidel, mohu doporučit [7]. Z tohoto zdroje jsem čerpal informace i já.

2.3.2 Šachy

Šachy se hrají na šachovnici o velikosti 8x8 polí, pole mají střídavě světlou a tmavou barvu. Šachovnice i se základním postavením figur je ukázána na obrázku 7.4 na straně 28. Nebudu zde popisovat, jak vypadají figury, to si ostatně můžete prohlédnou zde [11] spolu se spoustou dalších informací o šachu (stránky jsou v angličtině).

Cílem hry je *ohrožit* soupeřova krále takovým způsobem, aby protihráč nemohl provést žádný tah, kterým by tuto hrozbu odvrátil. Hráč, který tohoto docílí dal *mat* soupeři a vyhrává hru.

Striktně podle pravidel začíná vždy bílý hráč, ale jak již bylo zmíněno výše toto pravidlo není v programu dodrženo z důvodu zachování identity hráče. Pak se hráči v tazích střídají, v každém tahu hráč přemístí jednu svoji figuru. Pravidla pro povolené tahy jsou zde:

- Figuru nelze přemístit na pozici, kde již má hráč jinou svoji figuru. Přemístit ji lze jen na volné pole nebo na pole obsazené protihráčem (v tomto případě provádí takzvané *braní* a odstraní soupeřovu figuru ze šachovnice).
- Dámu lze přemístit na kterékoliv pole v řadě, ve sloupci nebo v úhlopříčce. Nesmí však přeskočit žádnou figuru (to platí i pro ostatní figury s výjimkou koně).
- Střelce je možno přesunout jen po úhlopříčce.
- Tah věží se provádí jen ve sloupci nebo v řadě, na kterých stojí.
- Kůň (jezdec) se pohybuje dvě pole ve sloupci nebo řadě, na kterých stojí, a pak jedno pole doleva či doprava ze směru pohybu. Jezdec přeskakuje figury v cestě.
- Pro pěšce platí hned několik pravidel:
 1. Může se posunout dopředu o jedno pole, pokud není obsazeno.
 2. Ze své startovní pozice je možno provést i tah dopředu o dvě pole.
 3. Může se pohnout i o jedno pole dopředu ve směru úhlopříčném, pokud je na něm figura soupeře a tím ji brát.
 4. Může také provést takzvané *braní mimochodem*, kdy bere soupeřova pěšce, který provedl tah o dvě pole a první z nich bylo ohrožováno tímto pěšcem. Takový tah je možné provést jen těsně po onom tahu soupeře.
 5. Pokud se pěšec svým pohybem dostane až do nejbližší řady, *promění* se v dámu, věž, střelce nebo jezdce.

- Král může jet o jedno pole ve všech směrech, pokud ono pole není ohrožováno soupeřem. Alternativně může provést takzvanou *rošádu* (více se o této možnosti můžete dozvědět zde [5]). Král je *v šachu*, jestliže je ohrožován alespoň jednou figurou protivráče.
- Hráč nesmí provést tah, kterým by vystavil svého krále šachu.

Partie končí buď vítězstvím hráče, který dal mat, nebo může skončit nerozhodně. Nerozhodná je pokud hráč na tahu nemůže provést tah podle pravidel (situace se nazývá *pat*). Partie může být nerozhodná, pokud posledních 50 tahů nebyl brán žádný kámen a současně nebylo taženo pěšcem.

Na webové stránce s oficiálními pravidly šachu (naleznete zde [5]) jsou zmíněna ještě jiná pravidla, která jsou ovšem aplikovatelná pouze na lidské hráče. Pro hru dvou programů nemají význam.

2.3.3 Dáma

Hned na úvod je třeba zmínit, že pravidla dámy se v jednotlivých zemích liší. Zde uvedená pravidla jsou pravidla takzvané *české dámy*. Těmito pravidly se také řídí navržený program.

Hra se hraje na šachovnici o velikosti 8x8 polí podobně jako šachy. Deska má i stejné značení. Malé vyjasnění pojmů: dáma se hraje se dvěma typy figur – kameny a dámami. Kámen se stává dámou, jakmile dosáhne nejvzdálenější řady.

Ani u této hry není dodrženo pravidlo začínajícího hráče. Pro pohyb figur platí následující pravidla:

- Všechny figury se pohybují jen po diagonálách.
- Kámen se pohybuje jen vpřed po diagonále na volné sousední pole.
- Dáma se pohybuje po diagonále libovolně. Libovolně daleko a libovolným směrem.
- Kámen provádí *braní*, jestliže je před ním kámen cizí barvy a za tímto kamenem volné pole. Pokud je za tímto volným polem další kámen soupeře, opakuje se braní. To může pokračovat tak dlouho, než se kámen ocitne v poslední řadě.
- Pokud se kámen ocitne v poslední řadě, stává se dámou a tah končí.
- Dáma provádí braní, pokud je na stejné diagonále jako figura soupeře a za ní alespoň jedno volné pole.
- Zabrané figury se odstraňují z desky.
- Žádná figura nemůže být brána dvakrát.
- Braní figur je povinné. Pokud hráč může braní provést, musí jej uskutečnit.
- Platí taky pravidlo o přednosti braní dámou. Pokud má hráč možnost provést braní dámou a současně kamenem, musí jej provést dámou.

Podrobnější pravidla naleznete na tomto webu [6]. Nachází se tam i pravidla takzvané *mezinárodní dámy*. Při dalším možném vývoji programu by jistě nebylo špatné rozšířit jej o možnost hrát tuto verzi dámy.

2.3.4 Gomoku

Hra je v českém prostředí známa spíše pod názvem piškvorky a hraje se většinou na čtverečkováném papíře pouze s tužkou. Pravidla jsou velice jednoduchá. Cílem hry je vytvořit na hrací ploše souvislou řadu pěti vlastních kamenů (těmito „kameny“ jsou většinou kolečka a křížky) a to ve všech směrech (horizontálně, vertikálně i diagonálně). Podrobná pravidla i s rozbohem hry naleznete například zde [15].

2.3.5 Go

Pravidla hry Go jsou poněkud složitější než u předchozích her a k jejich pochopení jsou nutné zejména diagramy znázorňující různé herní situace. Takové vysvětlení by bylo prostorově náročné, proto Vás odkáží na internetové stránky [10] a [20], kde naleznete všechny potřebné informace.

2.4 Hráči

Ve standardním chápání deskových her si člověk jistě představí jiné dva lidi sedící u hrací desky a hrající nějakou hru. Z pohledu umělé inteligence je hráčem počítačový program (nemusí jít vždy o samostatnou aplikaci, může se jednat kupříkladu o nějaký skript v určitém jazyce), který komunikuje s jiným podobným programem. Představu, jak takový *brain* (program řízený umělou inteligencí, který hraje určitou hru; tento název bude dále v tomto významu používán a lze jej nalézt v cizojazyčné literatuře, která se tímto zabývá) vypadá můžete získat po přečtení sekce 4.1.

2.5 Počítačové hraní

Počítačové hraní je již poměrně stará záležitost a existuje velmi mnoho postupů a metod jež se zde využívají. Některé jednoduché hry jsou v dnešní době (dostatek výpočetního výkonu) řešitelné pomocí poměrně snadných metod, které ale s reálným myšlením nemají mnoho společného. Jde hlavně o prohledávání stavového prostoru. Používají se takové metody jako *mini-max* nebo *alfa-beta*.

Tyto techniky mohou ale v rozumném čase fungovat jen v případě, že prohledávaný stavový prostor není příliš velký. Dobře se proto hodí pro šachy, dámu nebo reversi. Pokud se hra hraje na větší desce (kupříkladu standardní Go na desce o velikosti 19x19 polí), roste velikost stavového prostoru značnou rychlostí. V těchto případech je nutné použít jiné, pokročilejší techniky.

Prvním krokem bývá zavedení určitých heuristik s cílem omezit počet možností k prohledávání. Dále se zde uplatňují postupy jako hledání vzorů nebo začlenění nějaké znalostní báze. Trochu jiným přístupem je aplikování metody *Monte-Carlo*. Dochází i k užití prostředků strojového učení – uplatní se zde neuronové sítě a genetické algoritmy.

Jiným problémem je vůbec nalezení dostatečně kvalitní vyhodnocovací funkce, tedy funkce pro ohodnocení vhodnosti tahu. Někdy je taková funkce zřejmá (např. u hry reversi), v jiných případech bylo třeba intenzivního testování. Někde není optimální funkce známa ani dnes (Go).

Zatímco některé hry se z hlediska umělé inteligence považují za kompletně vyřešené, jiné zatím odolávají – především zmíněné Go.

Kapitola 3

Turnaje a hodnotící systémy

Abychom mohli nějakým způsobem měřit výkonnost jednotlivých hráčů, musíme si stanovit nějaký hodnotící systém. Existuje velmi mnoho různých hodnotících systémů, které jsou používány různými organizacemi k různým účelům. Téměř každá významná sportovní organizace provozuje svůj vlastní systém hodnocení. Jmenuji například systém, který používá organizace FIFA, nebo systém provozovaný profesionální tenisovou organizací ATP.

Nejpopulárnější a také nejvíce oceňované systémy jsou dnes ty, jež nějakým způsobem stavějí na systému *ELO*. Tento systém bude podrobněji popsán v následující kapitole. Vylepšení právě tohoto systému profesorem Markem Glickmanem pak vznikly systémy *Glicko* a *Glicko-2*, které ho rozšířily o proměnnou, která měří spolehlivost hodnocení. Více je rozebírat nebudu, další informace můžete získat na této stránce [16]. Jiným systémem, který staví na ELO, je *TrueSkill*. Tento systém byl navržen společností *Microsoft* pro jeho službu *Xbox Live* a i jeho základem je systém ELO. Snaží se o to, aby se daly rychle identifikovat hráčovy schopnosti (v některých případech jen během 3-6 zápasů) a pak jeho schopnosti náležitě monitorovat. Pro více informací navštivte tuto stránku [9].

3.1 Systém ELO

Hodnotící systém ELO byl navržen Arpadem Elem, americkým profesorem fyziky pocházejícím z Maďarska. Název systému je tedy odvozen od jména jeho tvůrce. Ten byl nespokojen s tehdejšími hodnotícími systémy, který používala šachová organizace USCF (United States Chess Federation), a proto navrhl svůj systém, který se snažil nedostatky předchozího systému řešit.

Systém je založen na statistickém základu (narozdíl od svého předchůdce, který byl založen na systému odměn). Systém založený na principu odměn hráče odměňuje za každou výhru, remízu nebo prohru udělením bodů, které se následně přičtou k jejich aktuálnímu bodovému hodnocení. Celkové bodové hodnocení není počítáno od začátku kariéry daného hráče, ale má jisté časové omezení (například hodnocení organizace FIFA je v současné době prováděno za období uplynulých čtyř let).

Profesor Elo navrhl systém, který se snaží hodnotit skutečné schopnosti hráče. Usuzoval, že výkonnost hráče se řídí normálním rozdělením a změna výkonnosti probíhá jen velmi pomalu. V šachu (stejně jako v jiných hrách a sportech) je však obtížné nějakým způsobem určit přímo hráčovu výkonnost, stejně musí být použito hodnocení odvíjející se od výher, proher a remíz.

Pro ilustraci uvedu jednoduchý a názorný příklad hodnocení v tomto systému, snáze se

pak dají pochopit další fakta. Předpokládejme, že spolu hrají dva hráči nestejně výkonnosti. Jestliže vyhraje silnější (ten, který měl před zápasem vyšší hodnocení), pak se nové hodnocení hráčů příliš nezmění. Silnější hráč sice může získat pár bodů hodnocení navíc (stejně jako slabší pár bodů ztratit; to závisí na rozdílu původních hodnocení), ale bude to změna spíše kosmetická.

Jestliže ale silnější hráč prohraje, pak je jeho současné hodnocení považováno za nadhodnocené a on ztratí relativně více bodů než kolik by získal v případě výhry. Podobně slabší hráč v případě výhry získá relativně vysoký počet bodů, protože je zřejmě jeho hodnocení nižší než by odpovídalo skutečnosti. Tento systém ohodnocení vychází z předpokladu, že hráč, který vyhrál by měl mít vyšší hodnocení než hráč, který prohrál. Pokud tomu tak není, upravují se hodnocení hráčů patřičným směrem, jak bylo vysvětleno.

K aplikování bodových přírůstků a ztrát může být přikročeno po každém hodnoceném zápase nebo může být prováděno v předem stanovených časových intervalech, případně po jistých událostech.

Systém samozřejmě není ideální a, jak bylo zmíněno výše, existují i jeho vylepšené varianty. Přináší například některé nechtěné sociální faktory, které se ale týkají především lidských hráčů. Jelikož je z globálního pohledu na věc tento systém poměrně přesný a splňuje i jiné požadavky (jako je relativní jednoduchost a názornost), byl vybrán i pro hodnocení brainů v realizovaném programu.

3.1.1 Matematické detaily

Již bylo zmíněno výše, že výkonnost nelze měřit absolutně. Každé hodnocení má význam jen v porovnání s jiným. Parametry a konstanty si proto můžeme libovolně zvolit. U systému ELO, který je používán organizací FIDE (Fédération Internationale des Échecs), jsou nastaveny tak, aby hráč, který má o 200 bodů vyšší hodnocení než protihráč, měl očekávanou pravděpodobnost výhry 0,75 %. V realizovaném programu jsou tyto parametry nastaveny obdobně.

Fungování celého systému je následující:

1. Pro každého hráče se vypočte očekávaný výsledek zápasu podle následujícího vzorce:

$$H_o = \frac{1}{1 + 10^{\frac{H_B - H_A}{400}}} \quad (3.1)$$

- H_o ... očekávaný výsledek
- H_A ... původní ELO hodnocení hráče
- H_B ... původní ELO hodnocení protihráče

2. Reálný výsledek hry je 1 pro výhru, 0,5 pro remízu a 0 pro prohru.

3. Nové ELO hodnocení se pak vypočte podle tohoto vzorce:

$$H_n = H_A + K * (H_s - H_o) \quad (3.2)$$

- H_n ... nové hodnocení
- H_A ... původní hodnocení
- K ... faktor změny

- H_s ... skutečný výsledek
- H_o ... očekávaný výsledek

Zbývá ještě osvětlit parametr K , nazývaný faktor změny. Víme, že hodnoty očekávaného výsledku H_o a skutečného výsledku H_s se pohybují v rozmezí 0 až 1, a proto koeficient K určuje maximální možný přírůstek nebo úbytek bodů po jednom zápase. Tento koeficient bývá někdy různý pro nováčky (hráči, kteří odehráli méně zápasů než je stanovený limit), někdy se mění podle aktuální hodnoty ELO každého hráče. Žádné takové modifikace koeficientu ve svém programu neprovádím (specifikaci mých nastavení systému se dočtete v části 5.3).

Nové ELO hodnocení může být počítáno po každém zápase zvlášť, což je ale třeba při turnajích diskutabilní a nepříliš spravedlivé. Proto je možné aktualizaci hodnocení provést až po odehrání všech her turnaje. V takovém případě se sečtou očekávané výsledky (stejně tak i ty reálné) a dají tak jednu hodnotu, která se pak dosadí do rovnice 3.2. Nové hodnocení je pak vypočteno standardním způsobem. Tento způsob hodnocení ostatně používám ve svém programu.

Systém ELO může být aplikován na všechny hry a sporty, ve kterých jsou možnými výsledky výhra, prohra nebo remíza. Detailní informace můžete najít na webu [12].

3.2 Turnaje

Turnajová klání představují vždycky možnost zahrát si s kvalitními hráči a otestovat svoje schopnosti. Toto platí pro lidské hráče i pro počítačové programy (potažmo jejich tvůrce). Rovněž je to příležitost posunout se v případě úspěšných her značně vzhůru na žebříčku.

Systémy, jakými je možné organizovat turnaj jsou jistě všeobecně známé. Jen pro přehled některé uvedu:

- Každý s každým – asi nejspravedlivější systém
- Na jednu porážku – taky nazývaný K.O.
- Na dvě porážky – každý zápas se hraje na dvě vítězné hry
- Švýcarský systém – specifický systém, lze znát z šachu

Vzhledem k tomu, že turnaj se odehrává na počítači, kde je možné hrát několik zápasů paralelně a rychlost hraní zápasů bývá u brainů zpravidla drasticky vyšší než u lidských hráčů, je nejrozumnější volbou systém každý s každým. Jeho nevýhodou je velký počet zápasů, které je nutné odehrát, ale to při počítačovém zpracování není na překážku.

Výše zmíněný přehled byl převzat z webu [2]. Pořádání turnajů umělých inteligencí v hraní deskových her je poměrně pěkným zvykem a něco více se o tomto můžete dočíst v sekci 4.5.

Kapitola 4

Současný stav problematiky

Současný stav na poli umělých inteligencí hrajících nějakou deskovou hru je velice různorodý. Pro každou hru existují programy, které zpravidla umožňují hru člověka proti umělé inteligenci (brainu), která se za oním programem skrývá. Někdy tyto programy také umožňují spojení s manažerem (program, který umožňuje hraní dvou brainů proti sobě, obvykle poskytuje grafický výstup z takové hry) nebo s jiným kompatibilním programem. Existuje několik různých způsobů realizace vzájemné komunikace, některé jsou jedinečné (defacto jsou vymyšleny v rámci aktivity jednoho autora), některé se snaží poskytnout robustnější a univerzálnější řešení.

4.1 Brainy

Z pohledu řešení projektu není nijak zásadní mít přehled o všemožných programech, které byly kdy vytvořeny pro hraní nějaké deskové hry, přesto je zajímavé sledovat, jaké programy a kdy byly vytvářeny, jaké byly jejich možnosti a provedení.

Seznam programků, které hrají reversi (také známo pod názvem Othello, jak bylo zmíněno výše) můžete nalézt například zde [8]. Na stránce naleznete mnoho takových programů. Nejnovější jsou sice skoro pět let staré, ale pro představu to stačí. Je vidět, že se tvorbou takového programu zabývali a zabývají různí autoři po celém světě již poměrně dlouhou dobu. Převážně se jedná o amatérské práce nebo školní výtvořky. (Taková je ostatně situace u každé zde zmíněné hry.) Můžeme tam nalézt textové programy převážně pro operační systém DOS, ale i pro starší systémy a platformy. Stejně tam jsou programy novější s účelným grafickým uživatelským rozhraním, pracující pod systémem Windows nebo na počítačích Mac.

Možná nejznámějším reversi programem je *Logistello* (oficiální stránku naleznete zde [1]). Jedná se o velmi sofistikovaný software, který prošel dlouhým vývojem, ale dále již není intenzivně vyvíjen a jeho zdrojový kód je volně k dispozici. Každý zájemce o tuto problematiku má možnost do tohoto kódu nahlédnout a nechat se inspirovat. Program komunikuje buď prostřednictvím souborového systému, nebo prostřednictvím socketů a protokolu IOS, který je ale v současné době již zastaralý.

Šachy jsou hra populárnější, takže jste o nějakém počítači hrajícím šachy určitě museli slyšet (kupříkladu *Deep Blue*). Seznam takových programů a strojů naleznete na webu [19]. Nejsilnější šachové programy již porážejí lidské šachové velmistry, takže výkonnost těchto programů je zjevně velká. To ovšem nebrání hraní dvou programů proti sobě, zvláště na amatérské úrovni. V šachovém odvětví vznikají i velice profesionální programy s propracovanými

vaným grafickým rozhraním, které třeba umožňují nahrávání různých brainů.

Dáma má v oblasti umělé inteligence podstatně menší popularitu, přesto i zde existuje několik velmi kvalitních programů. Na webu [13] v sekci *Non-human players* naleznete jejich seznam a odkazy na další informace.

Piškovorky (v originále *Gomoku*) se z pohledu umělé inteligence považují na ploše 15x15 za vyřešené, proto je zájem o ně z profesionálního hlediska velmi slabý. Na amatérské úrovni ale jistý zájem nalézt můžeme. Za zmínku například stojí aktivita zmíněná na těchto stránkách [3], která byla ostatně jistým vzorem i pro tvorbu tohoto programu. V sekci *download* naleznete celou řadu takových programů ke stažení.

Situace u hry Go je podobná jako u předchozích her, nicméně hra samotná je pro počítačové zpracování poměrně náročná a úspěšně odolává metodě *brutální síly*, která je například velmi úspěšná u šachu. Některé z těchto problémů jsem již rozvedl v sekci 2.5. Seznam takových programů naleznete například na webu [17].

4.2 Rozhraní

Zde není míněno rozhraní komunikace s jinými brainy (toto je rozebráno v následující sekci), ale rozhraní komunikace s uživatelem. Některé brainy jsou zabaleny do vlastního programu, který je určen pro hraní s lidským oponentem. Takové programy mají (zvláště v dnešní době) velice často propracované grafické uživatelské rozhraní. Pouze starší programy tohoto typu jsou založeny textově.

Některé brainy jsou ale určeny čistě k hraní proti jiným podobným brainům a jsou obvykle tvořeny jen *myslicím jádrem* a komunikačním rozhraním.

4.3 Komunikace

Jak jste se mohli přesvědčit v sekci 4.1, existuje opravdu velké množství brainů, které komunikují různými způsoby. Objevuje se zde především komunikace prostřednictvím souborového systému a prostřednictvím socketů, ale lze nalézt i komunikaci skrze *roury* (angl. pipes).

V poslední době se více prosazuje komunikace pomocí socketů (nebo jiný způsob síťové komunikace). Tento způsob je zřejmě nejpružnější a poskytuje největší funkčnost. Lze takto komunikovat lokálně i síťově stejným způsobem. I z těchto důvodů byl tento způsob vybrán pro tento projekt.

4.4 Manažery, GUI

Pro řadu her existují programy, které samy o sobě neobsahují žádnou umělou inteligenci, ale umožňují graficky znázorňovat průběh zápasu. Většinou také umožňují hry zaznamenávat a následně opakovaně přehrávat. (Ostatně pro některé hry existují záznamy her v některém z herních formátů – třeba SGF (Smart Game Format) – volně ke stažení.) Takovým programům můžeme říkat manažery.

Takovou funkčnost by měl mít realizovaný program.

4.5 Turnaje

I pro umělé inteligence se pořádají turnaje. Prostřednictvím svých brainů vlastně soutěží jejich autoři a proto turnajová klání neztrácejí soutěživý charakter. S organizováním turnajů pro počítačové programy se můžeme setkat čím dál častěji. Dokonce existuje událost nazvaná *Computer Olympiad* (více se můžete dočíst zde [\[4\]](#)).

Takové turnaje se ale pořádají častěji a existuje jich celá řada. Některé mají opravdu mezinárodní jméno, některé tak známé nejsou. Souvisí to i s hrou, ve které je turnaj pořádán. Šachové turnaje mají značně větší popularitu i obsazovanost než turnaje v piškvorkách. O tom, že existují i malé a mezinárodně téměř neznámé turnaje, svědčí mimo jiné i existence již zmíněného *Gomocupu*, jehož stránky můžete nalézt zde [\[3\]](#). Právě pro pořádání podobného turnaje by mohla být určena mnou navržená aplikace.

Kapitola 5

Návrh aplikace

Když jsme se seznámili se všemi důležitými fakty a ujasnili jsme si všechny požadavky na vytvářenou aplikaci, můžeme se pustit do jejího podrobnějšího návrhu.

První důležitou volbou, od které se další odvíjejí, je volba operačního systému, pod kterým aplikace poběží. Byl zvolen systém `Windows` vzhledem k jeho rozšířenosti (a tedy vyšší koncentraci amatérských brainů) a s přihlédnutím k tomu, že *spřátelený* projekt (bakalářská práce Tomáše Labaje – Reversi) je také pod tímto systémem. Svou roli hrály i osobní cíle, které jsou popsány v úvodu. V dnešní době je preferovaným způsobem vývoje aplikací pod `Windows` tvorba pomocí tzv. *.NET frameworku*. Takovou volbu jsem zvolil i já. Volba programovacího jazyka padla na `C#`.

Ze zadání na první pohled jasně vyplývá jediná funkčnost: tou je řízení dvou programů. Při bližším pohledu na problém (a přihlédnutí k prostudovaným materiálům) by však měla obsahovat ještě minimálně dva další funkční celky. Možnost zaznamenávat hraná utkání (a možnost jejich následného přehrávání) a nějakou podporu pro organizaci turnaje. Samozřejmě pak je nápověda přímo v programu.

Hned na úvod kapitoly bych chtěl říct, že aplikace byla realizována podle návrhu. V následující kapitole – 6 – uvedu jen některé technické detaily a poznámky.

5.1 Funkčnost programu

Základní funkční celky již byly nastíněny, je čas je postupně probrat trochu více do hloubky. Podrobné informace o návrhu komunikačního rozhraní, záznamů her, turnajové podpory, GUI (grafické uživatelské rozhraní) a nápovědy se dočtete v následujících sekcích. Zde uvádím návrh části, týkající se samotného zobrazování a ovládání průběhu hry.

Tady by se měly objevit následující prvky: zobrazení hrací plochy, vybírání brainů, kteří budou hrát partii, informace o brainech a jejich hře, možnosti nastavení hry a panel s ovládáním. Vzhledem k faktu, že hry si jsou navzájem velice podobné, mělo by GUI této části vypadat jednotně pro všechny hry (více v sekci 5.6).

Zobrazení desky s probíhající hrou je jednoznačné, nebudu dále rozepisovat. Vybrání brainu se provádí tlačítkem, po jehož stisknutí se objeví nabídka s možnostmi (lokální `x` na síti, vybrání souboru, zvolení portu apod.).

Informace o brainu by měly obsahovat: název, barvu, za kterou hraje, počet výher, proher a remíz, aktuální počet bodů v probíhající hře (pouze u některých her), informaci o tazích, zobrazení délky trvání tahu.

Nastavení hry by mělo obsahovat: nastavování počtu zápasů (abychom nemuseli každý

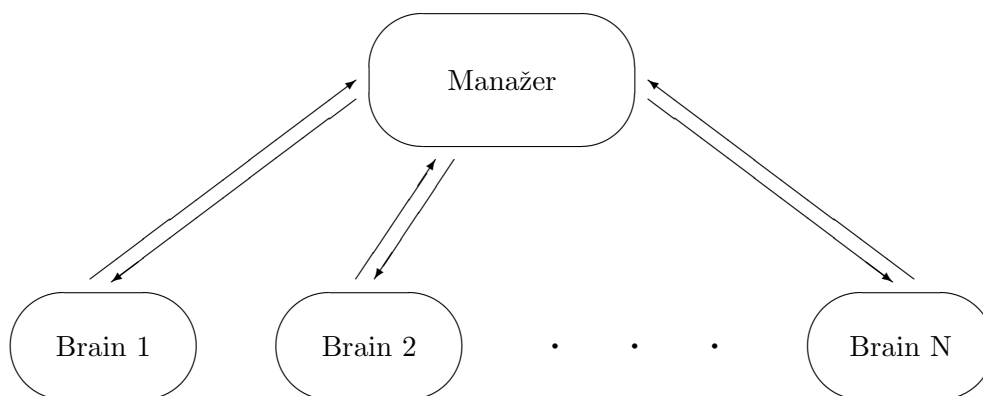
zvlášť startovat), možnost volby střídání se v začínání, nastavení maximální povolené délky tahu. Dále by mohlo obsahovat možnost stanovit konstantní délku tahu (tahy by se prováděly v předem nastavených intervalech nezávisle na délce tahů brainů) a případně zobrazení nějakých statistik po skončení hry.

Panel s ovládáním by měl být spíše jednoduchý, postačí tlačítka pro start hry, její pozastavení a okamžité ukončení.

5.2 Komunikace

Jako prostředek realizace komunikace byly vybrány *sockets*. Jedná se o způsob síťové komunikace. Výhody byly popsány v sekci 4.3. Komunikace probíhá prostřednictvím architektury TCP/IP, je tedy zajištěno spolehlivé doručování zasílaných zpráv.

Program je navržen tak, aby umožňoval několik vzájemně hraných utkání (například reversi a zároveň šachy). To je umožněno multi-vláknovým zpracováním. Schéma navržené komunikace vidíte na obrázku 5.1.



Obrázek 5.1: Schéma komunikace manažera s brainy

Z obrázku je zřejmé, že veškerá komunikace probíhá prostřednictvím manažera, brainy navzájem přímo vůbec nekomunikují. Z pohledu síťového se manažer chová jako *klient*, brainy jako *server*. Manažer je inicializátorem – posílá dotazy, brainy odpovídají.

Aby se brainy s manažerem domluvily, je nutné ustanovit nějaký komunikační protokol – sadu příkazů a reakcí na ně. Bylo možné zvolit nějaký z již používaných, nebo vymyslet vlastní (případně kombinaci těchto přístupů). S ohledem na to, že existují různé protokoly pro různé hry, bylo přistoupeno k vymyšlení vlastního protokolu. Přidání podpory jiného protokolu by se ovšem dalo poměrně lehce realizovat a je to rozhodně oblast, kterou by se mohl další možný vývoj aplikace ubírat.

Vytvořený protokol se skládá z několika příkazů a oznámení a příslušných reakcí na ně. Jejich seznam:

- **hello msg** – Dotaz má určit, zda je brain připraven komunikovat, v budoucnu zamýšleno využít k zjišťování odezvy. Jedinou možnou odpovědí je **hello ok**. Pokud brain odpoví jinak, považuje se to za selhání komunikace.

- **com msg** – Zjišťuje maximální podporovanou verzi komunikačního protokolu. Manažer by měl podporovat i starší verze protokolu, dokud nedojde k nějaké nekompatibilní změně. Brain odpoví verzí, kterou používá – pošle například `com 0.1`. Pokud verze podporována není, manažer ukončí komunikaci.
- **game [hra]** – Má zjistit, zda brain hraje konkrétní hru. Nyní podporované hry jsou: `reversi`, `chess`, `checkers`, `gomoku`, `go`. Odpověď od braina může být buď kladná (`game ok`), nebo záporná (`game failed`).
- **board [velikost]** – Nastavuje velikost herní desky (posílá se i v případě, že je možná jen jedna velikost). Parametrem je jedno celé číslo, zpráva tedy může vypadat např. takto: `board 8`. Pokud brain danou velikost podporuje, pošle odpověď `board ok`, v opačném případě `board failed`. Jakákoliv jiná odpověď než `board ok` je ale stejně považována za selhání.
- **player [barva]** – Nastavuje identitu brainu, tedy za jakou barvu bude hrát. Možné varianty: `white`, `black`, `cross`, `round` (poslední dvě zmíněné platí pro piškvorky). Akceptování brain provede zasláním odpovědi `player ok`, jiná odpověď je považována za selhání.
- **name msg** – Požadavek na název brainu. Brain odpovídá ve formátu `name [jmeno]`, tedy třeba `name brain_hulu_0.6.20`. Toto jméno pak textově identifikuje brain.
- **timeout [čas]** – Nastavuje maximální možnou dobu pro odpověď. Parametrem je přirozené číslo větší než nula představující délku timeoutu v sekundách. Brain neodpovídá, bere však timeout navědomí.
- **move [tah]** – Tato zpráva oznamuje brainu tah oponenta. Každá hra má jiné formátování tahu (tyto formáty uvádím za tímto seznamem), pro `reversi` je to například `move F4`. Zpráva `move start` má speciální charakter, oznamuje začátek zápasu a tedy to, že má brain provést první tah. Standardní odpovědí brainu je jiná `move` zpráva, tentokrát s parametrem svého tahu. Může však odpovědět i zprávou `forfeit`, která znamená, že se vzdává tahu. Poslání jiných odpovědí než těchto (případně ještě `end` – zatím bez využití) má za následek selhání komunikace a konec hry.
- **new game** – Informace o začátku nové hry. Pokud již nějaká hra probíhá, je ukončena. Brain neodpovídá, ale je povinen resetovat svůj stav (především hrací plochu) do počáteční pozice pro hru.
- **end game** – Ukončuje hru. Při normální hře nemá smysl. Používá se jen pro testovací účely. Brain neodpovídá, ukončí však hraní hry (bez toho, aby resetoval svůj stav).
- **quit msg** – Oznamuje ukončení komunikace. Brain neodpovídá, program braina se ukončí. Pokud běží brain jako síťová služba, ukončí se pouze komunikace s manažerem.

Zde následují formáty parametru zprávy `move`:

- **Reversi: X0** – za `X` se dosadí velká písmena A-H, za `0` číslice od 1 do 8. Příklad: `F4`.
- **Šachy: X0FY1** – za `X` se dosadí velká písmena A-H, za `0` číslice od 1 do 8, `Y` a `1` jsou na tom stejně. `X0` označuje pole, ze kterého se táhne, `Y1` pole, na které se táhne. `F` označuje figuru, kterou se táhne (anglické značky K, Q, R, B, N, P). Existují ale dvě výjimky: `0-0` pro malou rošádu a `0-0-0` pro rošádu velkou.

- Dáma: $X0[\text{oddelovač}]Y1*\{[\text{oddelovač}]Z2\}$ – $X0$, $Y1$ a $Z2$ mají stejný význam jako u předchozích her, označují postupně pole, na která se táhne. Oddělovačem může být buď $-$ (označuje prostý tah), nebo x (označuje skákání). Příklady: $B6-A5$, $E3xG5xE7$.
- Piškvorky: $X0$ – za X se dosadí velká písmena A-O, za 0 číslice od 1 do 15. Příklad: $K2$.
- Go: $X0$ – za X se dosadí velká písmena A-S, za 0 číslice od 1 do 19. Příklad: $H7$.

Takováto pravidla poměrně jasnou formou definují, jaká kritéria by měl kompatibilní brain splňovat, aby mohl bez problémů hrát zápasy prostřednictvím manažeru.

5.3 Turnaj

Co se týká podpory pro pořádání turnajů, ta by v programu měla být určitě zahrnuta. GUI by mělo obsahovat seznam brainů v turnaji, tlačítka pro přidání dalšího a odebrání označeného brainu, nastavení parametrů turnaje, panel s ovládáním, statistické a přehledové informace.

Seznam brainů bude obsahovat jejich názvy. Pod ním budou obě tlačítka pro manipulaci se seznamem. Přidávání brainů otevře okno obdobné jako v případě normálního výběru brainu. Druhé tlačítko odebere aktuálně označený brain ze seznamu.

V nastavení turnaje jsou dvě základní položky: hra a turnajový mód. Ze seznamu podporovaných her si uživatel vybere požadovanou; výběr módu turnaje se také provede ze seznamu podporovaných systémů. Systémy, které by měly být podporovány, jsou zmíněny v sekci 3.2. Přednastavené módy mají pevně nastavené parametry (počet her v rámci zápasu, střídání začátků, timeout...), ale mělo by být umožněno i zadání jiných hodnot.

Panel s ovládáním by měl umožnit, podobně jako normální hra, odstartování, pozastavení a ukončení turnaje. Přehledové informace o turnaji jsou představovány tabulkou zápasů. Po odehrání každého zápasu, se zde objeví jeho výsledek, takže uživatel vždy ví, kde se turnaj nachází. Po ukončení celého turnaje se zobrazí statistické informace a aktualizuje se ELO (pokud toto nebylo vypnuto). Statistika obsahuje seznam brainů s informacemi o jejich umístění a počtu získaných bodů.

K zjišťování výkonnosti jednotlivých brainů slouží jejich hodnocení v rámci systému ELO. Každý brain má své hodnocení, které udává, jak je výkonný (nováčkům je uděleno ELO 2000). Systém hodnocení je detailně popsán v sekci 3.1. Při turnajích (a právě jen při nich) se toto hodnocení může měnit. Hodnocení se uchovávají v souboru `elo.xml` v adresáři programu. Pro uchovávání informací byla tedy zvolena technologie XML a využívá se i pro ukládání nastavení programu či záznamů her.

5.4 Záznamy her

Zaznamenávání probíhající hry a následné umožnění její archivace pro budoucí přehrávání by v manažeru také nemělo chybět. Nejdříve ke GUI: okno by mělo obsahovat seznam uložených záznamů (nejlépe tříděný podle her) a možnost přehrát vybraný záznam. Po označení záznamu se zobrazí základní informace o zaznamenané hře – názvy brainů, čas začátku hraní. Dále by rozhraní mohlo umožnit vyhledávání záznamů podle různých kritérií a výběr adresáře se záznamy.

Záznamy budou ukládány ve formátu XML. Volba na tuto technologii padla, protože se jedná o moderní a široce podporovanou záležitost, soubory jsou lehce zpracovatelné jak

počítačem, tak člověkem (jedná se o formát čitelný pro člověka). Ukázku takového souboru se záznamem můžete vidět v části [6.3.2](#).

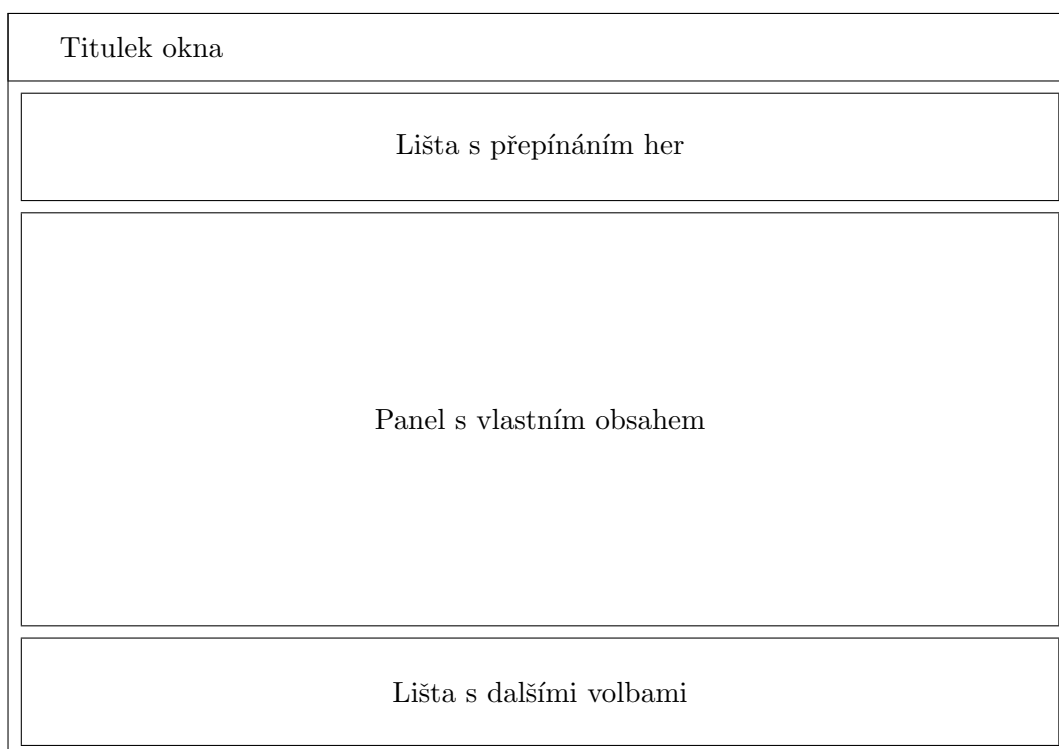
5.5 Nápověda

Nápověda přímo v programu patří ke každé slušné aplikaci. Zde bude realizována prostřednictvím internetového prohlížeče zabudovaného přímo do programu. Měla by obsahovat sekci s pravidly jednotlivých her (případně odkazy na pravidla na internetu), informace o ovládání a nastavení programu, seznam verzí a s tím spojený výčet provedených úprav programu.

Realizace nápovědy pomocí dokumentů v jazyce HTML má tu hlavní výhodu, že se dá snadno použít i jinde (na webových stránkách, převést do formátu PDF). Autor tak může psát jen jedny dokumenty a má zajištěn relativně stejný vzhled a stejný obsah na všech místech, kde je nápověda prezentována. Další výhodou spočívá v tom, že z nápovědy je možno odkazovat i externí internetové zdroje.

5.6 GUI

Většina požadavků na GUI již byla zmíněna v předchozích sekcích, zde již uvedu jen návrh celkového vzhledu aplikace. V okně aplikace by se měly nalézat dvě lišty (na jedné přepínání jednotlivých her a turnaje, na druhé ostatní volby) a panel s vlastním obsahem. Tento návrh je schematicky znázorněn na obrázku [5.2](#). Výslednou podobu aplikace si můžete prohlédnout v kapitole [7](#).



Obrázek 5.2: Schéma navrhovaného GUI aplikace

Kapitola 6

Realizace

Tato kapitola neobsahuje popis funkčnosti programu, toto najdete v kapitole předchozí. Zde se zabývám vlastním technickým zpracováním aplikace, a to jen přehledově se zdůrazněním podstatných prvků. Kompletní technická dokumentace by zabrala mnohem více prostoru než by bylo vhodné. Zájemce si může prohlédnout přímo zdrojový kód programu, který je snad dostatečně komentovaný a přehledný.

Program byl vytvořen v jazyce C# na platformě .NET. Jedná se o okenní aplikaci řízenou událostmi. Využívá standardních tříd a jmenných prostorů, které poskytuje platforma .NET. Bylo však nutné vytvořit několik programově specifických tříd, aby tak byla celá realizace více objektová.

6.1 Brain

Jelikož se jedná o zcela nové řešení problému, nebylo možné použít pro testování žádný již existující brain. Z tohoto důvodu jsem naprogramoval aplikaci, která by mohla s manažerem komunikovat. Jedná se o velmi prostou aplikaci, která slouží jen k ověřování správnosti zvolených řešení.

Program hraje hru reversi, což je stěžejní hra, na které byly odzkoušeny všechny použité postupy. Více v sekci 6.4. Brain umožňuje jak lokální spuštění (s parametrem portu, na kterém bude naslouchat), tak je možné jej spustit někde na síti a začít naslouchání ručně. Je možné zobrazit historii komunikace a nastavit délku tahu; tedy funkce důležité pro testování.

Možné tahy ohodnocuje podle počtu zabíraných kamenů. Náhodně zvolí mezi dvěma tahy, které jsou z tohoto hlediska nejvýhodnější. Rozhodně to není důstojný algoritmus pro hraní této hry, ale pro požadované účely to postačuje.

6.2 Datové struktury

Byly navrženy a vytvořeny čtyři nové třídy tak, aby poskytovaly potřebné datové složky a požadovanou funkčnost. Jsou to: třída *Brain*, zajišťující komunikaci a uchovávající nezbytné informace o brainu a jeho hře; třída *Move* představující tah; třída *Replay*, ve které jsou uchovávána data potřebná pro záznam hry a nakonec třída *Game* poskytující potřebné informace o právě hrané partii. Jejich veřejné vlastnosti a metody jsou zde:

```

class Game
{
    public Int32 GamesToPlay;
    public Int32 GamesPlayed;
    public Int32 OnTheMove;
    public Double MoveTimeout;
    public bool AlternateStarts;
    public DateTime MatchStart;
    public string LastMove;
    public DateTime MoveStartTime;
    public int[,] Board;
    public Replay Replay;
    public int GameType;

    public Game(Int32 gtp, Int32 otm, Double mto, bool alter,
        int[,] brd, int gametype);
    public string Info();
}

class Move
{
    public Int32 Player;
    public string Move;
    public TimeSpan MoveTime;

    public Move(Int32 pla, string tah, TimeSpan mt)
}

class Replay
{
    public string Player1;
    public string Player2;
    public DateTime TimeStart;
    public List<Move> ListOfMoves;
    public string GameType;
    public Int32 BoardDimensions;
    public int Winner;

    public Replay(string p1, string p2, DateTime ts, string gt, Int32 bd);
}

class Brain
{
    public IPAddress Ip;
    public int Port;
    public string FileName;
    public NetworkStream Stream;
    public bool Connected;
}

```

```

public string CurrMove;
public string PrevMove;
public Int32 Wins;
public Int32 Loses;
public Int32 Ties;
public Int32 MoveTime;
public Int32 AvgTime;
public Int32 CurrPoints;
public string Name;

public Brain(string ho, int po);
public Brain();
public bool Connect();
public string RecvMsg();
public void SendMsg(string msg);
}

```

Myslím, že další popis těchto tříd již není nutný, vlastnosti i metody jsou pojmenovány tak, že je jejich účel zřejmý.

6.3 Popis řešení

V této části popíši realizaci nejdůležitějších prvků programu. Zmíním se jen o řešení důležitých částí programu. Spoustu práce zabrala tvorba GUI a programování reakcí na vyvolávané události, což je ale přirozené a není třeba toto nijak zvlášť popisovat.

6.3.1 Komunikace

Nejpodstatnější a nejvíce komplikovanou součástí programu je zajištění funkční komunikace. Síťová komunikace (neustálá čekání na odpovědi serveru – brainu) jsou blokuující operace. Bylo tudíž nutné tuto komunikaci umístit do vlastních vláken, aby bylo vůbec možné rozumně pracovat s GUI aplikací. Tato vlákna jsou vytvářena třídou *BackgroundWorker*.

Vlastní komunikace je zařízena pomocí tříd *TcpClient*, *StreamReader* a *StreamWriter*. První ustaví spojení s brainem, další dvě slouží ke čtení a zápisu zpráv. Příjem a odesílání zpráv je ale zajištěno metodami *RecvMsg* a *SendMsg* třídy *Brain*, které v sobě zapouzdřují také ošetření výjimek, které mohou nastat při síťové komunikaci.

Průběh hry z pohledu komunikace pak vypadá následovně: manažer postupně kontaktuje oba brainy a zjistí informace nutné pro pokračování ve hře (dotazy `hello`, `com`, `game`, `board`, `name` a `player`); nastaví a odstartuje partii (příkazy `timeout`, `new game`); následuje dotaz `move` pro začínající brain. Jakmile dorazí odpověď, je provedena kontrola, zda je tah povolen a kontrola konce hry. Pokud je tah povolen a hra pokračuje, manažer opět pošle dotaz `move` (tentokrát druhému hráči) a tento cyklus se opakuje až do konce partie.

6.3.2 Záznamy her

Jakmile hra skončí je o ní uložen záznam. Strukturu takového záznamu je uvedena níže. K ukládání veškerých dat byla zvolena technologie XML. Záznamy se ukládají do složky `replays` v adresáři programu.

K reprezentaci záznamu slouží třída *Replay*. K vytvoření instance dochází hned po zahájení hry, postupně se do ní přidávají prováděné tahy. Po skončení hry je tam zaznamenán vítěz a skóre (pokud je při hře užíváno). Následně se záznam uloží s patřičným pojmenováním (název obsahuje jména obou brainů a čas zahájení hry) do souborového systému.

K ukládání slouží třídy *XmlDocument*, *XmlNode*, *XmlElement* a *XmlTextWriter*. První je takovou paměťovou reprezentací souboru XML, další dvě jsou prvky vkládané do onoho dokumentu. Poslední třída poskytuje prostředek zápisu na diskový prostor.

K zobrazení záznamů v GUI aplikace slouží třída *TreeView*, která poskytuje potřebnou funkčnost. Při označení nějakého záznamu v tomto seznamu dojde k načtení důležitých informací a jejich zobrazení (příklad můžete vidět na obrázku 7.3 na straně 28). Ještě uvedu ukázkou onoho XML souboru se záznamem:

```
<?xml version='1.0'?'> <Replay>
  <GameType>reversi</GameType>
  <BoardDimensions>8</BoardDimensions>
  <StartTime>4.5.2007 17:15</StartTime>
  <Player1>brain_hulu_0.6.20</Player1>
  <Player2>brain_hulu_0.6.20</Player2>
  <Winner>Player 1</Winner>
  <Score p1=„38\ p2=„26\ />
  <ListOfMoves>
    <Move player=„1\ time=„00:00:01.4062500\>D6</Move>
    <Move player=„2\ time=„00:00:01.3281250\>E6</Move>
    <Move player=„1\ time=„00:00:00.3125000\>F6</Move>
    ...
    <Move player=„2\ time=„00:00:00.1562500\>forfeit</Move>
    <Move player=„1\ time=„00:00:00.2968750\>B1</Move>
    <Move player=„2\ time=„00:00:00.3125000\>H1</Move>
    <Move player=„1\ time=„00:00:00.3125000\>H8</Move>
  </ListOfMoves>
</Replay>
```

Uvedená ukázkou představuje část skutečného záznamu hry a lze si z ní udělat představu, co v takovém záznamu můžete nalézt. Rovněž je zřejmé, že záznam není nijak svázán jen s tímto konkrétním manažerem, lze lehce udělat jiný zobrazovač záznamů, je dokonce snadno čitelný a srozumitelný pro člověka.

6.3.3 Nápověda

Většina podstatného v souvislosti s nápovědou byla již zmíněna v sekci 5.5. Technicky nejde o nijak složitou záležitost. Platforma .NET poskytuje třídu *WebBrowser*, která obstarává veškerou funkčnost. Pro snadnější navigaci byla přidána tlačítka *Vpřed* a *Zpět*. Pro potřeby této nápovědy to téměř nebylo nutné, ale uživatel je na takové možnosti zvyklý ze své práce s klasickými webovými prohlížeči.

6.3.4 Turnaj

Podpora pořádání turnajů za pomoci tohoto manažeru je v současné době realizována prostým naplánováním série zápasů, které je nutné odehrát. Zápas se pak odehrávají

normálním způsobem, jak je popsáno o tři podseky výše.

Takové řešení je samozřejmě zcela přijatelné, ale zejména z časového hlediska není optimální. Při velkém počtu brainů v turnaji by se dal celkový čas hraní turnaje značně zkrátit, kdyby se některé zápasy hrály paralelně. Takové řešení by zahrnovalo vypracování nějakého algoritmu, který by flexibilně startoval zápasy volných brainů. Rovněž by to vyžadovalo některé změny do GUI. Z časových důvodů se toto nepodařilo realizovat. Jde ale o oblast, které by mohla být v budoucnu věnována pozornost.

6.4 Poznámky

Některé z poznámek uváděných v této podkapitole jsem již diskutoval na jiných místech práce. Logicky však patří k sobě, takže budou zmíněny i zde.

Přestože aplikace nevypadá příliš složitě, realizace některých věcí tak, jak byly naplánovány se ukázalo jako časově velice náročné. Některé ze stanovených požadavků proto nebyly splněny. Zde se samozřejmě otevírá cesta pro další zdokonalování programu.

Předně je zde problém s neexistencí kompatibilních brainů. Toto bylo vyřešeno vytvořením vlastního testovacího programu, který by obstaral funkčnost braina. Protože tvorba takového programu pro všechny podporované hry není reálná vzhledem k vymezenému času řešení, hraje vytvořený brain pouze Reversi.

Program byl tedy vyvíjen a testován pro tuto hru, hraní dalších her staví na nalezených postupech pro hru Reversi. Některé věci mohou fungovat beze změny (kupříkladu schéma komunikace, ukládání záznamů, pořádání turnaje), některé by ke své stoprocentní funkčnosti vyžadovaly vysokou míru testování. Jedná se především o kontrolu správnosti tahu, určení existence možného tahu a kontrolu konce hry. Tyto kontroly nejsou prováděny, což je jistou nedokonalostí manažeru, a je nutné se spoléhat na brainy, že provádějí v těchto hrách validní tahy. To není optimální, ale v současné době defacto neexistuje brain, který by byl s manažerem kompatibilní (samozřejmě s výjimkou mého a hry Reversi Tomáše Labaje), a tak se toto dá přejít.

Vývoj manažeru by ostaně měl pokračovat, je zde spousta prostoru pro vylepšování. O věcech, které by se daly vylepšit či přidat se ještě zmíním v závěru práce.

Zmíním se ještě o grafické podobě herních desek. Všechny desky her – kromě šachu – jsou generovány přímo v programu pomocí funkcí Windows API. Pro vzhled šachovnice a šachových figur byly využity obrázky z *Wikimedia Commons*. Jedná se o iniciativu, která se snaží distribuovat některé soubory k volnému užití (jedná se o obrázky, zvukové a obrazové nahrávky apod.) s cílem jejich jednotného vzhledu v nekomerčních aplikacích.

Kapitola 7

Ovládání programu

Ovládání programu je do značné míry intuitivní, zde uvedený popis tedy bude stručný. Postupy pro spuštění samostatné hry, turnaje a práce se záznamy jsou uvedeny v samostatných podkapitolách, protože se jedná o stěžejní funkce aplikace.

Kromě výše zmíněného se v programu nachází nápověda a nastavení programu. S nápovědou se pracuje stejným či podobným způsobem jako s kterýmkoliv jiným prohlížečem. Uživatel se pohybuje mezi stránkami prostřednictvím odkazů, případně pomocí tlačítek *Vpřed* a *Zpět*. Nápověda je dostupná pod záložkou *Help*.

Na záložce *Settings* se pak nachází možnost nastavit některé vlastnosti programu. Zatím jich není mnoho – zobrazování informačních zpráv, možnost spustit brainy bez případného GUI, výběr složky pro ukládání záznamů turnajových her, volba jazyka nápovědy.

7.1 Samostatná hra

K úspěšnému zahájení hry je nutné provést několik kroků:

1. **Vybrat brainy.** To se provede kliknutím na tlačítko *Brain 1*, resp. *Brain 2*. Po kliknutí se zobrazí dialog, kde uživatel vybere, zda se jedná o lokální aplikaci, nebo zda je brain na síti. V prvním zmíněném případě vybere spustitelný soubor, ve druhé zadá IP adresu nebo doménové jméno. V obou případech pak nastaví port, na kterém se bude komunikovat. Následně může stisknout tlačítko *OK*. Dojde ke kontrole braina, stav se zobrazí na patřičných místech.
2. **Nastavit parametry zápasu.** Jedná se o: počet her v zápase, volbu střídání se v začínání, zadání timeoutu na tah, volba sledování zápasu v reálném čase a určení konstantní doby tahu.
3. **Odstartovat zápas.** Provede se snadno kliknutím na *Start*.

Po provedení těchto úkonů již brainy začnou partii. Na herní ploše můžete sledovat průběh, na pravé straně okna se zobrazují průběžné informace. Hraní můžete pozastavit tlačítkem *Pause* a analyzovat situaci. Stiskem tlačítka *Stop* se hra ukončí. V takovém případě se však neukládá záznam. Představu o ovládání si můžete udělat i z obrázku [7.1](#), kde vidíte rozehranou partii hry Reversi.

7.2 Turnaj

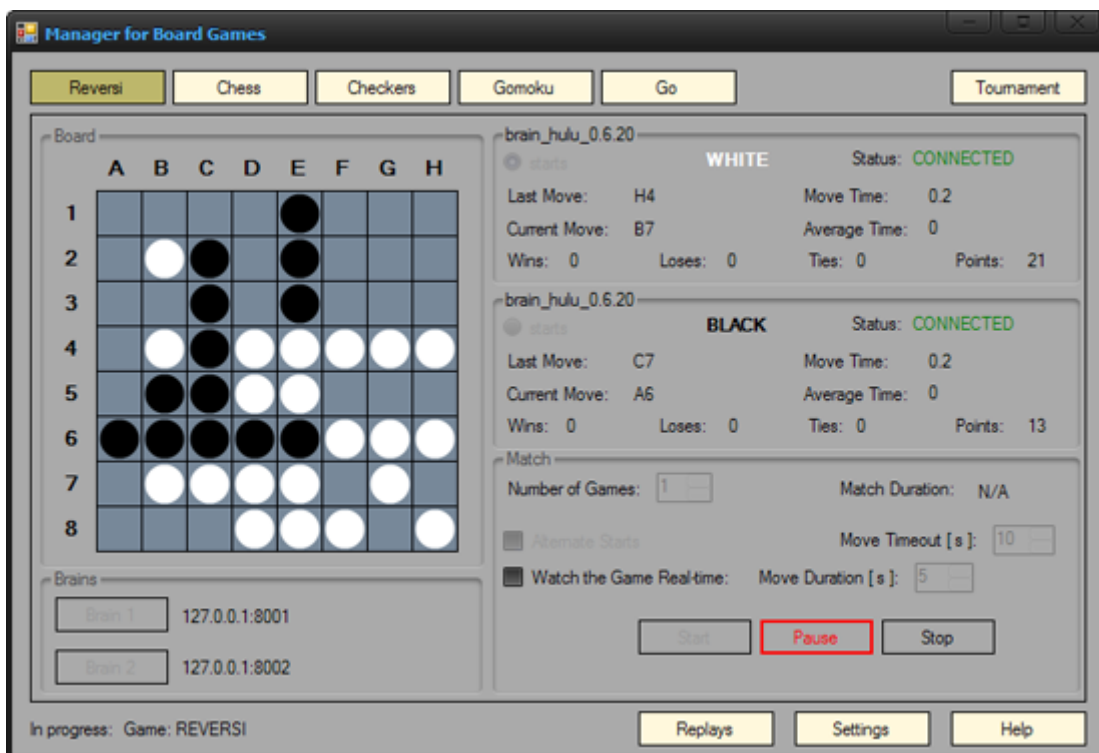
Postup při zakládání turnaje se do jisté míry podobá tomu předchozímu. Je nutné opakovaně přidávat brainy pomocí tlačítka *Add brain*. Je nutné si ale dávat pozor při zadávání portu. U samostatné hry jsou porty pro oba brainy předvoleny, takže se o toto uživatel nemusí příliš starat. Při přidávání brainů do turnaje je nutné dbát na to, aby uživatel zadal jiný port pro každý brain. Nejvíce problémů to může způsobovat při vybrání brainu lokálně, protože zadávat komunikační port při spouštění lokálních procesů není zvykem. Pro zjednodušení tam bylo přidáno malé tlačítko *+*. Odebírání brainu se provádí tlačítkem *Delete brain*.

Při každém přidání či odebrání hráče se aktualizuje tabulka turnaje, statistiky i záložka s hodnocením ELO. Následuje nastavení turnaje (podobně jako u samostatné hry). Až je uživatel spokojen se vším nastavením, odstartuje turnaj tlačítkem *Start*. Vzhled okna se zadáváním turnaje je na obrázku [7.2](#).

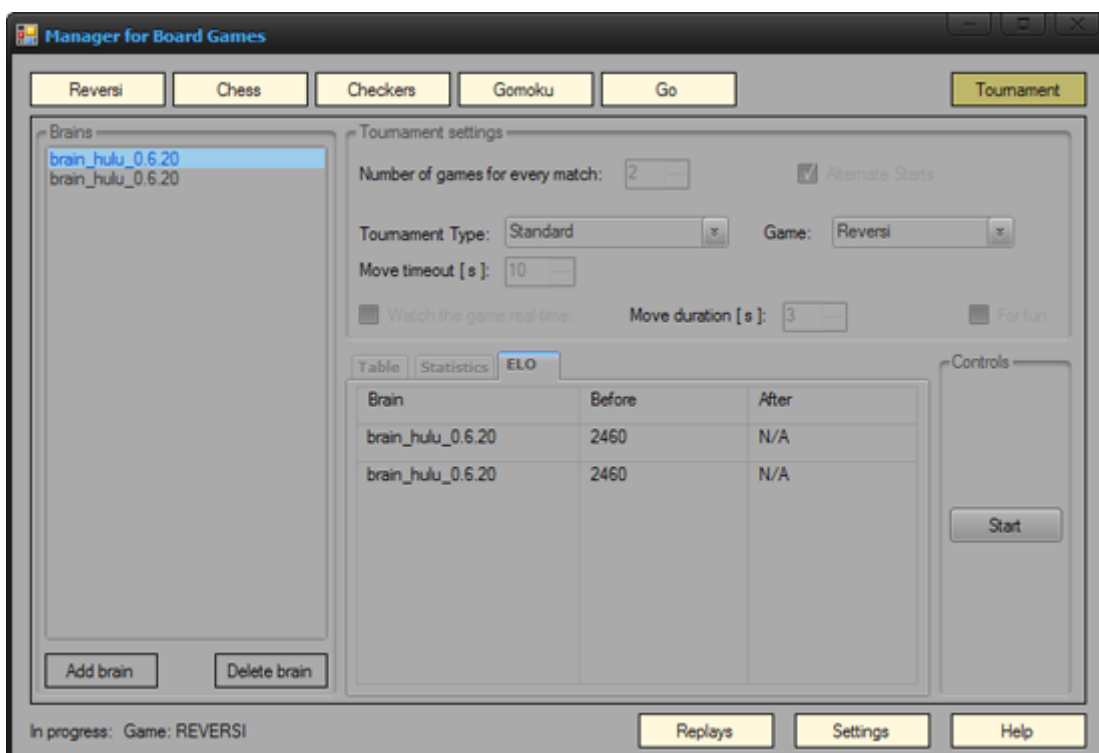
7.3 Záznamy her

Práce s herními záznamy je pak již snadná. Levý box obsahuje seznam záznamů v aktuálně vybrané složce. Název složky je zobrazen v dolní části panelu a lze zvolit jinou složku tlačítkem *Change folder*. Po označení konkrétního záznamu se vpravo zobrazí vybrané podstatné informace poskytující bližší představu o charakteru záznamu.

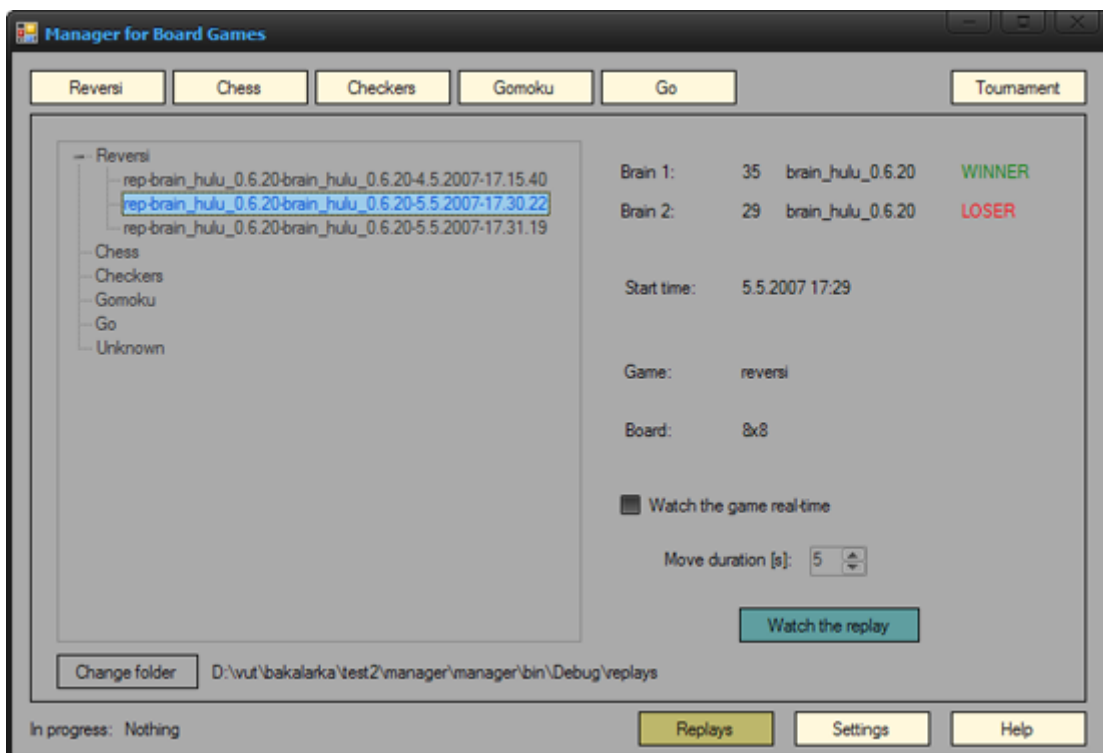
Pod těmito daty lze nastavit konstantní dobu přehrávání tahu. Pokud tato volba není zaškrtnuta, přehrává se záznam rychlostí, jež byl hrán. Přehrávání se odstartuje tlačítkem *Watch the game*. Situace je zobrazena na obrázku [7.3](#).



Obrázek 7.1: Snímek programu – hra Reversi



Obrázek 7.2: Snímek programu – turnaj



Obrázek 7.3: Snímek programu – záznamy



Obrázek 7.4: Snímek programu – hra šachy

Kapitola 8

Závěr

V rámci této práce se podařilo vytvořit aplikaci, která je schopna řídit hru dvou brainů. Jde především o grafické zobrazování průběhu hry, možnost hru pozastavit a studovat situaci, zpomalovat či zrychlovat průběh hraní. Program umí tyto hry zaznamenávat a následně znova přehrávat a prohlížet. Je zde také přítomna podpora pro pořádání turnajů a zaveden systém pro hodnocení výkonnosti jednotlivých brainů.

Z pohledu autora bylo hlavním přínosem důkladné seznámení se s programovacím jazykem C# a platformou .NET. Cenná je rovněž zkušenost s vývojem takto rozsáhlé aplikace. Projekt mi umožnil nahlédnout do oblasti počítačového hraní, zvláště pak do té amatérské části tvorby brainů a pořádání „turnajků“ umělých inteligencí.

Některé věci se podařilo vyřešit k mé spokojenosti. Jiné by si zasloužily zlepšení. Jde o validaci tahů a kontrolu konce hry u jiných her než reversi. Toto by vyžadovalo především vytvoření brainů pro ony hry a řádnou dávku následného testování. Existuje také celá řada možných dalších rozšíření programu.

Jde třeba o umožnění změnit velikost hrací desky u těch her, kde to je běžné (reversi, piškvorky, Go). Mohlo by se provést začlenění jiných pravidel pro hry. Nyní jsou u každé hry přesně specifikována pravidla, jimiž se musí hráči řídit. U některých her ale existuje celá řada variant těchto pravidel. Umožněním hrát i podle jiných pravidel by se manažer stal jistě použitelnějším.

V současné době už existují komunikační protokoly a různé formáty pro záznamy her. Dalo by se uvažovat o jejich podpoře v manažeru. To by umožnilo hru i starším brainům a rovněž brainům, které jsou vytvářeny pro takové protokoly a formáty. Dalším vylepšením by bylo umožnit vyhledávání v záznamech podle zadaných kritérií. Při delším používání se totiž může počet záznamů nakupit a ani pojmenování záznamu jmény zúčastněných brainů a časem začátku hry nemusí postačovat.

Jinou možností je pak upravit kód programu tak, aby bylo možné hrát více zápasů stejné hry najednou. To by se hodilo při pořádání turnajů. Velké množství přihlášených brainů totiž znamená mnoho zápasů, které je nutné odehrát (zvláště pak v systému každý s každým). Takto by se tento čas dal značně zkrátit. Dalším krokem by mohlo být přidání dalších turnajových systému nebo dokonce umožnění definovat si vlastní systém nastavením potřebných parametrů. A možností dalších směrů vývoje je jistě více.

Souběžně s tímto projektem byl vytvářen také program kolegy Tomáše Labaje *Reversi*. Jde o samostatnou aplikaci, jejíž hlavní funkcí je umožnění hraní člověka proti počítači. Program ale umožňuje i komunikaci s mým manažerem.

Literatura

- [1] Buro, M.: Logistello. [online], Naposledy modifikováno 18:08, 26.6.2003, naposledy navštíveno 2.5.2007.
URL <http://www.cs.ualberta.ca/~mburo/log.html>
- [2] deskovehry.cz: Turnajové systémy. [online], Naposledy navštíveno 1.5.2007.
URL http://deskovehry.cz/index.php/Turnajov%C3%A9_syst%C3%A9my
- [3] gomocup.wz.cz: Programátorská soutěž Gomocup. [online], Naposledy navštíveno 2.5.2007.
URL <http://gomocup.wz.cz/cz/index.htm>
- [4] ICGA: ICGA Tournaments. [online], Naposledy navštíveno 2.5.2007.
URL <http://www.grappa.univ-lille3.fr/icga/>
- [5] Klub deskových her Paluba: Šachy. [online], Naposledy navštíveno 1.5.2007.
URL <http://hrejssi.cz/sachy/pravidla.htm>
- [6] Klub deskových her Paluba: Dáma. [online], Naposledy navštíveno 1.5.2007.
URL <http://www.hrejssi.cz/dama/pravidla.htm>
- [7] Klub deskových her Paluba: Othello. [online], Naposledy navštíveno 30.4.2007.
URL <http://hrejssi.cz/othello/pravidla.htm>
- [8] Lauppert, T.: Othello. [online], Naposledy modifikováno 27.4.2003, naposledy navštíveno 2.5.2007.
URL <http://members.chello.at/theodor.lauppert/games/othello.htm>
- [9] Microsoft Research: What is the TrueSkill Ranking system? [online], Naposledy modifikováno 18:45, 21.2.2006, naposledy navštíveno 1.5.2007.
URL <http://research.microsoft.com/mlp/trueskill/>
- [10] Wikipedia: Go (board game). [online], Naposledy modifikováno 04:55, 5.5.2007, naposledy navštíveno 7.5.2007.
URL http://en.wikipedia.org/wiki/Go_%28board_game%29
- [11] Wikipedia: Chess. [online], Naposledy modifikováno 08:13, 1.5.2007, naposledy navštíveno 1.5.2007.
URL <http://en.wikipedia.org/wiki/Chess>
- [12] Wikipedia: Elo rating system. [online], Naposledy modifikováno 09:17, 27.4.2007, naposledy navštíveno 1.5.2007.
URL http://en.wikipedia.org/wiki/ELO_rating_system

- [13] Wikipedia: List of draughts players. [online], Naposledy modifikováno 09:30, 14.3.2007, naposledy navštíveno 2.5.2007.
URL http://en.wikipedia.org/wiki/List_of_draughts_players
- [14] Wikipedia: Board Games. [online], Naposledy modifikováno 09:55, 30.4.2007, naposledy navštíveno 30.4.2007.
URL http://en.wikipedia.org/wiki/Board_games
- [15] Wikipedia: Gomoku. [online], Naposledy modifikováno 18:17, 21.4.2007., naposledy navštíveno 7.5.2007.
URL <http://en.wikipedia.org/wiki/Gomoku>
- [16] Wikipedia: Glicko rating system. [online], Naposledy modifikováno 19:01, 4.1.2007, naposledy navštíveno 1.5.2007.
URL http://en.wikipedia.org/wiki/Glicko_rating_system
- [17] Wikipedia: List of free Go programs. [online], Naposledy modifikováno 19:02, 19.3.2007, naposledy navštíveno 2.5.2007.
URL http://en.wikipedia.org/wiki/List_of_free_Go_programs
- [18] Wikipedia: Reversi. [online], Naposledy modifikováno 19:26, 14.4.2007, naposledy navštíveno 30.4.2007.
URL <http://en.wikipedia.org/wiki/Reversi>
- [19] Wikipedia: Chess engine. [online], Naposledy modifikováno 21:17, 22.4.2007, naposledy navštíveno 2.5.2007.
URL http://en.wikipedia.org/wiki/Chess_engine
- [20] www.czech-go.net: Pravidla GO. [online], Naposledy navštíveno 7.5.2007.
URL <http://www.czech-go.net/pravidla/pravidla.php>