

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ŠACHOVÝ SERVER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

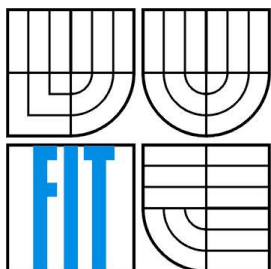
AUTOR PRÁCE
AUTHOR

VIKTOR BÁRCZI

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ŠACHOVÝ SERVER

CHESS SERVER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VIKTOR BÁRCZI

VEDOUCÍ PRÁCE
SUPERVISOR

ING. OTA JIRÁK

BRNO 2008

Zadání bakalářské práce

Řešitel: **Bárczi Viktor**
Obor: Informační technologie
Téma: **Šachový server**
Kategorie: Web

Pokyny:

1. Seznamte se s technologiemi pro vývoj webových aplikací a základními pravidly hry šachy.
2. V jazyce UML specifikujte model systému pro podporu šachového herního serveru podporující offline hru, počítající ELO hráčů a vedoucí základní statistiky.
3. Navržený systém implementujte.
4. Zhodnoťte dosažené výsledky a diskutujte další možné rozšíření tohoto projektu.

Literatura:

- Kolektiv autorů: PHP, MySQL, Apache. Computer Press, 2006. ISBN: 80-251-1073-7.

Při obhajobě semestrální části projektu je požadováno:

- Bez požadavků.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese
<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Jiráček Ota, Ing., UIFS FIT VUT**
Datum zadání: 1. listopadu 2007
Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Viktor Bárczi**
Id studenta: 84442
Bytem: Bátorove Kosihy 770, 946 34 Bátorove Kosihy
Narozen: 15. 06. 1984, Komárno
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Šachový server
Vedoucí/školitel VŠKP: Jiráček Ota, Ing.
Ústav: Ústav informačních systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě	počet exemplářů: 1
elektronické formě	počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel

Babara Viktor
.....
Autor

Abstrakt

Táto práca sa zaoberá vytvorením a riešením problematiky šachového servera. Jedná sa o vytvorenie šachového servera, pomocou ktorého si cez internet alebo lokálnu sieť, ale i na jednom PC môžete zahrať šachy. Riešenie informačného systému napodobňuje v skutočnom svete hrané šachové partie medzi dvoma hráčmi. Podľa základných pravidiel v šachu je riešená kontrola správneho ťahu figúrok, analyzovanie špeciálnych ťahov ako branie mimochodom, rošáda, premena a analyzovanie šachov a matov na kráľa. Počas hrania hry je vedená štatistika ťahov a po dokončení partií na základe výsledku je počítaná výkonnosť hráčov a je vedená štatistika získaných ELO bodov.

Klíčová slova

Šach, server, PHP, MySQL, ELO, hodnocení, figúrka, mat, pat, databáze, ER diagram, král, dáma, veža, pešiak, jazdec, strelec, rošáda, branie mimochodom, ťah, notácia

Abstract

This work deals with the creating, and solving the problem of a chess server. It is about creating a chess server wherewith people can play chess through the Internet or a local network and also on a PC. In the real world the construction of the informational system resembles chess games between two players. According to the basic rules of chess the control of the right movements, the analysis of special movements like taking a figure and also rooking, projection and the analysis of checks and mates on the king are solved. During the game statistics of the movements are kept and after the end of the game the performance of the players are calculated by the virtue of record and statistics are kept about the gained ELO (rating) points.

Keywords

Chess, server, PHP, MySQL, ELO, rating, piece, checkmate, stalemate, database, ER diagram, king, queen, rook, pawn, knight, bishop, castling, en passant, move, notation

Citace

Bárcezi Viktor: Šachový server. Brno, 2008, bakalářská práce, FIT VUT v Brně.

Šachový server

Prehlásenie

Prehlasujem, že som túto bakalárskú prácu vypracoval samostatne pod vedením Ing. Ota Jiráka.

Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Viktor Bárcezi
14.05.2008

Pod'akovanie

Týmto by som chcel poďakovať môjmu vedúcemu Ing. Ota Jirákovi za odborné vedenie, cenné rady, konzultácie, pripomienky, ktoré mi prispeli k dokončeniu bakalárskej práce.

© Viktor Bárcezi, 2008.

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

Obsah.....	1
Úvod.....	3
1 Šach.....	4
1.1 Dejiny šachu.....	4
1.1.1 Výhra počítača nad majstrom sveta.....	4
1.2 Základné pravidlá hry šachu.....	5
1.2.1 Základné pravidlá.....	5
1.3 Rating - ELO systém.....	6
2 Návrh šachového serveru.....	9
2.1 Informačné Systémy.....	9
2.1.1 Kvalita informačného systému.....	10
2.1.2 Metódy získavanie informácií na požiadavky.....	10
2.1.3 Metodológia.....	11
2.1.4 Etapy životného cyklu.....	11
2.1.5 Agilná metodológia.....	13
2.2 Analýza požiadaviek.....	14
2.2.1 Užívatelia systému.....	14
2.2.2 Funkcionalita.....	15
2.2.3 Dáta.....	18
2.2.4 Rozhranie systému.....	18
2.3 Modely navrhovaného systému.....	19
2.3.1 Use case diagram.....	19
2.3.2 E-R diagram.....	22
2.4 Analýza možností riešenia.....	24
2.4.1 Ako testovať šachy kráľa.....	24
2.4.2 Možnosti ako zakončiť partiu.....	25
2.4.3 Notácia partí.....	26
2.4.4 Výpočet ELO hodnotenia - rating.....	27
3 Implementácia.....	30
3.1 Použité jazyky a technológie.....	30
3.2 Implementované funkcie systému.....	31
3.2.1 Inštalácia databázy a vytvorenie tabuliek.....	31
3.2.2 Grafické užívateľské rozhranie.....	31
3.2.3 Registrácia nových užívateľov.....	32

3.2.4	Sessions a prihlásenie	32
3.2.5	Posielanie správ	32
3.2.6	Vygenerovanie šachovnice	33
3.2.7	Komunikácia s databázou	34
3.2.8	Označenie figúrky na ťahu.....	34
3.2.9	Riešenie ťahov	35
3.2.10	Testovanie ohrozenia kráľa	41
3.2.11	Vrátenie postavenia	42
3.2.12	Zdielanie počítača.....	42
3.2.13	Vedenie notácie	43
3.2.14	Počítanie ELO bodov.....	43
4	Záver	45
	Literatura	46
	Seznam příloh	47

Úvod

Je úplne neuveriteľné, kam sa v súčasnosti dostáva ľudské poznanie vedy a techniky. Dobrou ukázkou toho je raketový nástup PC, internetu či mobilov v uplynulých rokoch.

Z histórie a rozprávania mojich rodičov spomeniem sálové počítače, kam zápočtové úlohy zadávali ešte na diernych štítkoch. Neskôr to bola éra osobných počítačov ako PMD, ZX Spektrum, Amigo, Comodore 64, Atari a pod. Prvý počítač, na ktorý som mal od roku 1992 pravidelný prístup bola SX386-ka, ktorá bežala na frekvencii 25 MHz, vybavená 8 MB RAM a 40 MB disku. Na tomto počítači som mal nainštalovaný prvý šachový program Fritz.

Dnešné zostavy prinášajú o 1000-násobne vyššie výkony za zlomok ceny svojich predchodcov. Ešte pred pár rokmi sa nám disková kapacita servera 1 terabajt zdala neskutočne veľká. Disková kapacita súčasných serverov to tiež niekoľkonásobne prevyšuje a výhliadky nanotechnológií prinášajú nové možnosti nielen v oblasti informačných technológií, ale pre ľubovoľnú oblasť života.

Víziou používateľov sú mobilné zariadenia disponujúce výkonom umožňujúcim nerušenú prácu, hry či sledovanie DVD video, ktoré možno jednoducho nosiť všade so sebou a bez obmedzenia sa pripojiť na internet. Všade okolo seba máme čoraz viac elektronických pomocníkov. Počítač v práci aj doma, mobilný telefón, internet, to všetko sú dnes veci, ktoré neodmysliteľne patria k životu väčšiny ľudí.

Práve rozšírenie internetu bola základom aj pre vytváranie šachových serverov. V súčasnosti existuje niekoľko desiatok šachových serverov, ktoré poskytujú „on-line“ hru, pri ktorej odpoveď od súpera príde v priebehu niekoľkých sekúnd. Naozaj obľúbené šachové servery okrem „samotnej“ hry poskytujú aj iné služby užívateľovi.

Témou mojej bakalárskej práce je šachový server. Mám pomerne jasnú predstavu, ako by asi mal vyzerat' takýto šachový server a aké služby by mal poskytovať. V mojej práci sa však budem zaoberať predovšetkým modelom systému pre podporu šachového herného serveru podporujúceho offline hru, počítajúceho ELO hráčov a vedúcu základné štatistiky.

Pre riešenie uvedenej problematiky som sa zoznámil so základnými pravidlami hry šachu, s technológiami pre vývoj webových aplikácií a spôsobom výpočtu ELO hráčov, ktoré sú tiež podrobne rozvedené v tejto práci.

1 Šach

Pre vytvorenie modelu systému pre podporu šachového herného serveru podporujúceho offline hru ako prvý krok je potrebné zoznámiť sa s pravidlami hry šachu. Pri tejto príležitosti považujem za potrebné v krátkosti spomenúť dejiny šachu a významné medzníky v jeho vývoji.

Spomeniem niektoré zaujímavé udalosti, ktoré sa odohrali počas jeho vývoja ako zostrojenie prvého šachového stroja, prvá výhra počítača nad majstrom sveta a históriu ELO systému.

1.1 Dejiny šachu

Prvá podoba šachu vznikla v severnej Indii okolo roku 500 pod názvom čatarunga. Bolo to pred asi poldruha tisícročím a možno aj skôr, keď dakto zhotovil magickú čierno-bielu štvorcovú dosku so šesťdesiatimi štyrmi políčkami a vyrezal z dreva alebo možno zo slonoviny dve armády figúrok. Táto múdra a spravodlivá hra si už vtedy vybojoval veľkú popularitu. Prv sa dostala do Perzie a odtiaľ okolo 11. storočia sa tromi cestami dostala na náš kontinent – perzskí kupci ju priniesli do Ruska, byzantskí do talianskych miest, maurskí dobyvatelia na Pyrenejský polostrov.

Tisícpäťstoročná história šachovej hry má svoje medzníky. Prvá európska šachová kniha pochádza z 13. storočia. Vydanie prvej tlačenej knihy Španiela Louia de Lucenu roku 1497 znamenala šachovú revolúciu zrazu v dvoch smeroch – zavedenie dnešných pravidiel o pohybe šachových kameňov a zrod šachovej literatúry, ktorú dnes už prezentujú desaťtisíce zväzkov. Šachová literatúra je základným a nenahraditeľným prostriedkom rozvíjania šachového umenia.

Druhým medzníkom je rok 1851, keď v Londýne usporiadali prvý medzinárodný šachový turnaj. Od vzniku šachových turnajov sa dovedy z úcty používané oslovenie „majster“ stáva titulom a neskôr pre najlepších sa kanonizoval titul „veľmajster“.

Ďalším historickým medzníkom je založenie Medzinárodnej šachovej federácie (Fédération Internacionale des Échess – FIDE) v Paríži roku 1924. Dnes má FIDE vyše sto členských krajín a je organizátorkou majstrovstiev sveta v súťažiach jednotlivcov i družstiev.

V roku 1867 na medzinárodnom šachovom turnaji v Paríži boli použité prvé šachové hodiny.

1.1.1 Výhra počítača nad majstrom sveta

Deep Blue bol šachový hrací super počítač vyrobený firmou IBM. Prvýkrát tento počítač vyhral partiu proti stávajúcemu majstrovi sveta rusovi Garri Kasparovovi dňa 10. februára 1996. Dvojzápas na šesť partií však ešte vyhral Kasparov v pomere 4:2, pričom 3 partie vyhral a 2 remizoval.

Počítač Deep Blue bol narýchlo upgradovaný a dvojzápas na šesť partií sa zopakoval v máji 1997. Deň 11. máj 1997 vošiel do histórie, super počítač IBM Deep Blue porazil majstra sveta rusa

Garri Kasparova v dvojzápase na šesť partii v pomere 3,5:2,5, pričom 2 partie vyhral a 3 partie remizoval.

Čo sa stalo za ten jeden rok, čomu môže ďakovať tento obrovský úspech počítača? Základný algoritmus počítača Deep Blue, algoritmus minimax sa za predchádzajúcich 20 až 25 rokov v podstate nezmenil, avšak čo sa zmenilo bola rýchlosť vykonaných úkonov. Počítač bol schopný analyzovať 200 miliónov pozícií za sekundu, čiže bol dvakrát rýchlejší ako verzia z roku 1996.

1.2 Základné pravidlá hry šachu

Šach je strategická hra medzi dvoma hráčmi, pri ktorej náhoda žiadnu, alebo prakticky žiadnu úlohu nehrá, ale o výsledku rozhodujú schopnosti hráčov. Cieľom hry je dať súperovi mat. Má v sebe prvky umenia, vedy a športu. Hrajú ju dvaja hráči podľa šachových pravidiel. Šach rozvíja logické myslenie a cvičí pamäť.

1.2.1 Základné pravidlá

Šach je hra, založená na ťahoch bieleho a čierneho po šachovnici. Šachovnica je súhrnný názov pre 64 štvorcových polí, zoskupených do štvorca 8×8, striedavo bielych (jednej farby) a čiernych (druhej farby), ľavé dolné pole je čierne. Skupina polí v jednom smere, navzájom sa dotýkajúcich stranami, sa nazýva frontála. Skupina polí v jednom smere, navzájom sa dotýkajúcich vrcholmi, sa nazýva diagonála.

Biely je súhrnný názov pre kamene biele (jednej farby), čierny je súhrnný názov pre kamene čierne (druhej farby), zároveň je to označenie šachistu, ktorý pohybuje kameňmi danej farby. Biely a čierny sú súper. Kameň je súhrnný názov pre kráľa, dámu, vežu, strelca, jazdca a pešiaka.

Na jednom poli môže stáť buď jeden kameň (obsadené pole), alebo nijaký kameň (prázdne pole). Rozmiestnenie kameňov na šachovnici sa nazýva postavenie. Základné postavenie na šachovnici je zrejmé z obrázku s číslom 1.



Obraz č. 1. - Základné postavenie na šachovnici [9]

Ťah je premiestnenie kameňa buď na prázdne pole (krok), alebo na pole obsadené súperovým kameňom (branie); zvláštne ťahy: rošáda, premena pešiaka, branie mimochodom.

Partia je sled ťahov, začínajúci sa základným postavením partie a končiaci sa výhrou bieleho, výhrou čierneho, alebo remízou. Voľba ťahu závisí iba od šachistu, ktorý ho vykonáva. Prvý ťah v partii vykonáva biely, po ťahu bieleho nasleduje ťah čierneho a po ťahu čierneho nasleduje ťah bieleho.

Každý hráč má k dispozícii nasledujúce figúrky:

- 1 kráľa
- 1 dámu
- 2 veže
- 2 strelcov
- 2 jazdcov (nesprávne kôň)
- 8 pešiakov (nesprávne sedliak)

Dáma a veža sú ťažké figúry, strelca a jazdec sú ľahké figúry. Šachovnica leží medzi oboma hráčmi, každý z hráčov má napravo zo svojej strany biele pole. Biely hráč má zo svojho pohľadu na druhom rade ôsmich pešiakov a na prvom rade zľava do prava: vežu, jazdca, strelca, dámu, kráľa, strelca, jazdca a vežu. Pritom každá strana postaví dámu na pole vlastnej farby.

Orientačná relatívna sila figúrok je nasledovná:

- pešiak 1
- ľahké figúry 3,5 (3)
- veža 5
- dáma 10 (9)

Z uvedených hodnôt vidno, že orientačne troch pešiakov možno vymeniť za jednu ľahkú figúru, vežu za dve ľahké figúry, dámu za dve veže. Skutočná sila figúrok prirodzene závisí od ich celkového rozloženia na šachovnici. V úvode partie má napríklad veža obmedzený manévrovací priestor a preto je výhodnejšie ju vymeniť za dve ľahké figúry, v koncovke jej sila relatívne narastá a takáto výmena môže byť výhodnejšia pre opačnú stranu.

Kráľ je v šachu najdôležitejšia figúrka, pretože cieľom súpera je zasadiť kráľovi šach-mat. To znamená ohroziť ho s jednou alebo viacerými figúrkami tak, aby nemohol uniknúť na iné neohrozené políčko.

1.3 Rating - ELO systém

Mám 1875. A tvoje ELO je aké? - to je typická otázka medzi dvoma šachovými hráčmi pri prvom stretnutí. Druhý hráč s ELO-m 1750 hneď viem, že jedna partia medzi nimi nebude pre neho

jednoduchou záležitosťou. Remíza by nebola zlým výsledkom a výhrou by mohol dosiahnuť významný úspech.

Čo je to ELO rating? Hodnota ELO ratingu má význam len v porovnaní s inými hráčmi. Garri Kasparov bol prvým šachistom, ktorý v júli 1989 ako prvý prelomil vysnenú hranicu 2800 ELO. Jeho výkon použitím výpočtu ELO systému je porovnateľný s výkonmi iných hráčov v minulosti ako aj súčasnosti. Systém výpočtu ELO ratingu sa na prvý pohľad zdá zložitým, ale sa zakladá na základoch matematickej štatistiky. Je jednoznačné, že dvaja hráči s rovnakým ratingom majú rovné šance na výhru vo vzájomnom zápase. Menej jednoznačné je to, že rovnaká hodnota rozdielu medzi ELO ratingmi dáva rovnaké šance na výhru. Hráč s 2400 ELO ratingom má rovnaké šance na výhru proti hráčovi s ELO ratingom 2200, ako hráč s ELO 1600 proti hráčovi s ELO 1400. Hodnota rozdielu ELO je v oboch prípadoch 200. Ešte by som k tomu dodal, že to platí v tom prípade, keď hráči odohrali dostatočný počet hodnotených partií a ich ELO rating sa „ustálil“, nakoľko len vtedy odzrkadľuje ich hráčske kvality.

V najširšom okruhu na výpočet ratingu sa používa ELO systém. Jeho názov je odvodený z mena jeho tvorca Élő E. Árpád, ktorý sa narodil v Maďarsku roku 1903, mal 10 rokov keď s rodičmi emigroval do USA. Od roku 1935 do roku 1965 bol profesorom fyziky a astronómie na Universite Marquette. Od roku 1935 do roku 1937 bol administrátorom Amerického šachového zväzu. Élő bol 9-násobným majstrom v šachu štátu Wisconsin. Élőm vypracovaný systém od roku 1960 prevzal USCF (Americký šachový zväz), a od roku 1970 tento systém uplatňuje aj FIDE. Élő zomrel v roku 1992. V knihe s názvom „Rating šachových hráčov v minulosti a súčasnosti“ Élő píše toto: „Len málo hráčov vie objektívne hodnotiť pozíciu pri šachovnici, a ešte menej je tých, ktorí svoje šachové schopnosti a rating objektívne hodnotia. Prevažná väčšina si myslí, že jednoducho boli vo forme, keď na jednom turnaji podali výkon vysoko nad svojimi schopnosťami a pritom zabúdajú nato, že ich súťažný výsledok rovnako mohla spôsobiť aj prípadná slabšia hra viacerých hráčov, ako ich vlastná fantastická hra. Je pravda v tom paradoxne, že každý šachový hráč si myslí o sebe, že je lepší od jemu rovnocenných hráčov“.

Výška ratingu hráčov rozdeľuje do tried. Triedy USCF sú nasledovné:

- 2400- Veľmajster
- 2200-2399 Majster
- 2000-2199 Pokročilý
- 1800-1999 Trieda A
- 1600-1799 Trieda B
- 1400-1599 Trieda C

Tento spôsob rozdelenia hráčov dáva možnosť, aby na turnajoch a súťažiach hrali hráči približne rovnakého ratingu. Hráč triedy A na turnaji triedy A môže hrať s rovnakými šancami na víťazstvo.

Ani jeden systém nie je dokonalý, má niekoľko chýb aj systém výpočtu Elo bodov. Prirodzeným javom je deflácia ratingu, čo spôsobuje príchod mladých, rýchlo sa vyvíjajúcich hráčov a odchod starých, stabilných hráčov zo systému. Môžu byť aj podvody a manipulácie, keď neserióznymi organizátormi organizáciám vedúcim ratingy hráčov zašlú podvodné správy z turnajov. Za tieto a podobné problémy sú zanedbateľné popri tom čo ELO systém pre šachový svet dal. Za uznanie bol Élő v Amerike prijatý medzi nesmrteľných šachu.

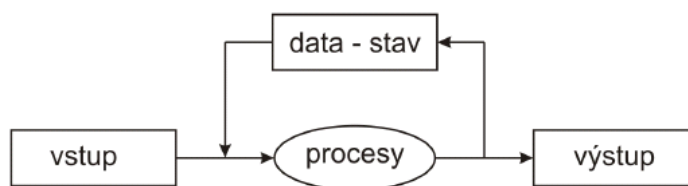
2 Návrh šachového serveru

V Nasledujúcej kapitole rozoberám analýzu a návrh systému šachového servera, ktorý by mal umožňovať zahrať šachovú hru medzi dvoma hráčmi a viesť základnú štatistiku o počte získaných a stratených bodov, popri tom by umožňoval sledovať výkony jednotlivých hráčov, a to v závislosti od výšky ich ELO bodov (výška ELO bodov vyjadruje výkonnosť jednotlivých hráčov v šachu, označenie ELO je odvodené od jeho vynálezca Árpáda Élőho).

Pred implementáciou je potrebné získať čím viac informácií od užívateľov a od zadávateľa projektu. Za týmto účelom som použil rôzne metódy a metodológie na získavanie týchto údajov. V tejto kapitole sa ešte zmienim o obecných vlastnostiach informačného systému, jeho životný cyklus vytvorenia. Niečo o agilných metodológiách a Rational Unified proces bude popísané tiež v tejto kapitole. Po zoznámení sa s metódami riešenia podrobnejšie pozriem na problematiku a navrhmem diagramy a možné riešenia..

2.1 Informačné Systémy

Informačné systémy (IS) sú otvorené systémy, ktoré modelujú reálny svet a používajú pre zber, udržiavanie, spracovávanie a poskytovanie informácií a dát. Informačný systém pracuje s nehmotnými zdrojmi, dátami, ktoré je potreba uložiť. Nato sú potrebné databázy kde sú ukladané dáta. Z dát po spracovaní užívateľom sa stanú informácie. Informačný systém má vstupnú a výstupnú časť, na vstupnej strane dáta skladajú na výstupnej strane dáta získavajú. Medzi vstupom a výstupom nad dátami pracujú rôzne procesy a algoritmy pod spoločným meno transakcie, ktoré prevádzkajú transformáciu dát. Typickou časťou informačného systému je spätná väzba (obr. č.2.), ktorá využíva uložený stav systému. Výstupy závisia na stavu systému.



Obraz č. 2. - ukážka spätnej väzby [1]

Aby informačný systém skutočne napodobňoval skutočný svet musí dodržiavať dve základné pravidlá.

Modelovacie prostredie, v ktorom dáta sú modelované v nejakej databáze a procesy v nejakom programovacom jazyku. Databáza pritom musí dodržiavať tiež niektoré pravidlá, musí byť perzistentná (v čase sa nemeniaca). Rôzne typy databáz napr.: relačný, objektový a sieťový.

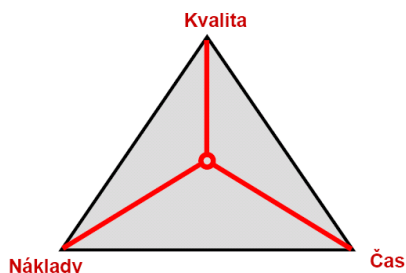
Pri modelovaní procesov zase musíme dávať pozor na konzistentnosť (súdržnosť) a na transakčné pracovanie.

Komunikačné prostredie, ktoré zaistí otvorenosť systému s okolím, v dnešnej dobe je to väčšinou počítačová sieť (lokálny, internet).

2.1.1 Kvalita informačného systému

Pri rozboru informačného systému je nutné predviesť predbežnú analýzu hodnoty produktu. Tento produkt často dostane meno projekt. Projekt je časovo ohraničená aktivita, ktorý musí smerovať k nejakým cieľom. Veľa projektov začína s nejakou myšlienkou, víziou, plánom s cieľom splniť určité alebo odvodené potreby užívateľov organizácie. Je to vlastne jedinečná neopakovateľná činnosť, ktorá je závislá od času a ceny a pritom musí splniť i určitú kvalitu (obr. č.3.).

Každý projekt musíme aj nejako riadiť a to vyžaduje špeciálne postupy, z tohto dôvodu zaviedli pojem životný cyklus IS.



Obraz č. 3. - Kvalita výrobku od závislosti času a nákladov [2]

Životný cyklus má obecnú marketingovú teóriu a môžeme rozdeliť na niekoľko fáz, a to podľa Sommervilla, ktorý doporučuje nasledujúce časti typického plánu projektu:

- **Inicializácia** - definovanie cieľa a hlavné obmedzenia (čas, náklady, akosť) projektu,
- **Organizácia projektu** - popis organizácie vývojového tímu a ich činnosť,
- **Analýza rizík** - identifikácia možných rizík a spôsob ako ich opatriť a riešiť,
- **Požiadavky na HW a SW zdroje** - špecifikácia HW a SW požiadaviek potrebné pre vývoj produktu,
- **Členenie práce** - definícia činností, míľnikov a dodávok,
- **Plán projektu** - definícia a rozdelenie času a zdrojov odpovedajúce činnostiam,
- **Monitorovací a reportovací mechanizmus** - určenie potrieb pre monitorovanie a údržbu systému.

2.1.2 Metódy získavanie informácií na požiadavky

Aby sme vytvorili systém, ktorý vyhovuje požiadavkám užívateľov, musíme od nich získať kvalitné informácie pri problémových častiach systému, preto je nevyhnutná komunikácia s užívateľom

systému. Tento prístup je však často problematický z pohľadu užívateľa, lebo On často nemá ucelenú predstavu o výslednom softwarovom systéme. Pritom pri formulácii požiadaviek často používajú prirodzený jazyk, ktorý môže spôsobiť nejednoznačnosť. Požiadavky môžu byť rozporné alebo sa môžu meniť. Napriek tomu existuje niekoľko metód na získavanie informácií. Tieto metódy sú nasledovné:

- interview (orientačný, štruktúrovaný),
- dotazníky,
- štúdium dokumentov,
- pozorovanie práce u zákazníka,
- priama účasť na prácach zákazníka,
- analýza existujúceho softwarového systému.

2.1.3 Metodológia

Aby proces vývoja softwaru bol čím viac efektívnejší a prediktovateľný, managery a analytici používajú rôzne metodológie. Tieto metodológie delíme na process-oriented a na people-oriented, ďalej na prediktívne a adaptívne metódy.

V procesne-orientovanej je človek chápaný ako prediktovateľný komponent. Tam sú pevne definované kroky, čo spôsobí, že procesy sú nadefinované tak, aby za všetkých okolností fungovali správne, kým ten druhý je orientovaný na ľudí, čiže každý proces prispôbuje na vývojový tím a procesy nevytvárajú znalosti, ale mali by uľahčiť prácu tímu.

V prediktívnej metóde je dlhší čas vyznačený na pripravovanie a plánovanie systému, pokúsi sa predísť zmenám, kým adaptívny práveže na tieto zmeny sa čím skôr pokúsi reagovať. Problémom je, že požiadavky u väčšiny projektov sa neustále menia, preto je niekedy ťažko rozhodnúť, ktorú metodológiu máme na mysli použiť. Často sa môže stať, že na začiatku projektu nemáme jasnú predstavu čo môže vyvolať otázku, je možné riadiť taký neprediktovateľný proces?

Za riešenie na riadenie takýchto softwarových projektov zaviedli interaktívny vývoj. Pritom každú iteráciu je možné rozdeliť na etapy. Tieto etapy sú rozobrané v nasledujúcej kapitole.

2.1.4 Etapy životného cyklu

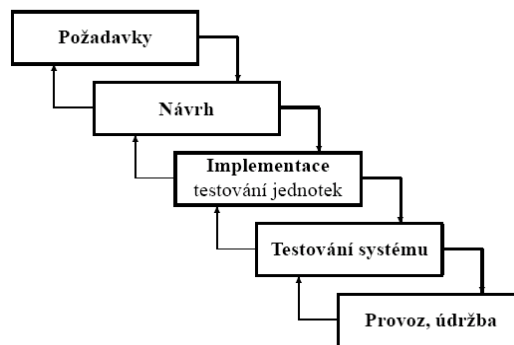
Jednotlivé etapy životného cyklu pri vývoji softwaru sú: analýza požiadaviek (8%), návrh (7%), implementácia (12%), testovanie (6%), prevádzka a údržba (67%) (do zátvorky som uviedol približný podiel práce ohľadom na celkovú prácu). Stručná charakteristika jednotlivých etáp:

- Analýza a špecifikácia požiadaviek je úvodnou fázou, ktorú podľa požiadaviek zadávateľa neformálne špecifikujeme a definujeme požiadavky na systém. Pokúsime sa získať čím jasnejšie informácie, čo od nás chcú. Pri tejto etape získame informácie neformálne, komunikáciou medzi profesionálom a zadávateľom.

- Architektonický návrh je nasledujúcim bodom, kde už formálnejšie spracujeme požiadavky a pozornosť je venovaná dekompenzácii systému. Celý systém rozoberieme na časti (podsystemy) definícia komunikačného rozhrania medzi podsystemami. Niekedy naplánuje aj postup nasadenia systému do prevádzky, prípadne zaškolenie užívateľov.
- Podrobný návrh sústreďuje na výber najvhodnejších algoritmov, stanovenie logických a fyzických štruktúr dát, ošetrovanie neočakávaných dát.
- Implementácia zahŕňa programovú realizáciu softwarových súčiastok, otestovanie už hotových častí podsystemu.
- Integrácia a testovanie systému, po vytvorení podsystemov tie sa spoja do celku a otestuje celý systém (tak sa odhalia chyby, ktoré nie je možné nájsť pri testovaní podsystemov).
- Akceptačné testovanie a inštalácia, systém prevádzame do prevádzky na strane užívateľa. Užívateľia majú možnosť v praxi vyskúšať výrobok.
- Prevádzka a údržba zahŕňa priebežné riešenia problémov, ktoré sa počas používania systému môžu objaviť, prípadne doplnenie systému o nové funkcie.

2.1.4.1 Iteratívny vývoj

Iteratívny vývoj je model životného cyklu systému, kde každá iterácia sa rozdeľuje na etapy. Tieto etapy sú vlastne súčasťou najzákladnejších modelov životného cyklu a toto je vodopádový model (pozri obrázok č. 4).



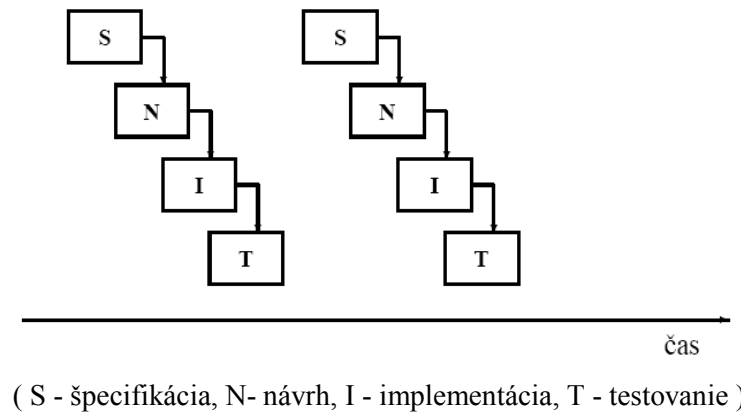
Obráz č. 4. - Vodopádový model [2]

Vodopádový model však má určité nedostatky, lebo sa nevie prispôbiť na nové požiadavky užívateľov a tak v prípade zmeny sa dostaneme na začiatok projektu a musíme znovu previesť analýzu a návrh. Jeho rozšírením sú modely iteratívny, inkrementálny a špirálový.

2.1.4.2 Iteratívny a inkrementálny model

V iteratívnom modeli každú iteráciu chápeme ako inštanciu vodopádového modelu. To pôsobí, že užívateľ po každej iterácii má spustiteľnú verziu softwaru a tak je ľahšie produkt prispôbiť jeho požiadavkám. Negatívnosť tohto modelu robí, že pri väčšej zmene môžeme štruktúru zdrojového kódu strašne zhoršiť.

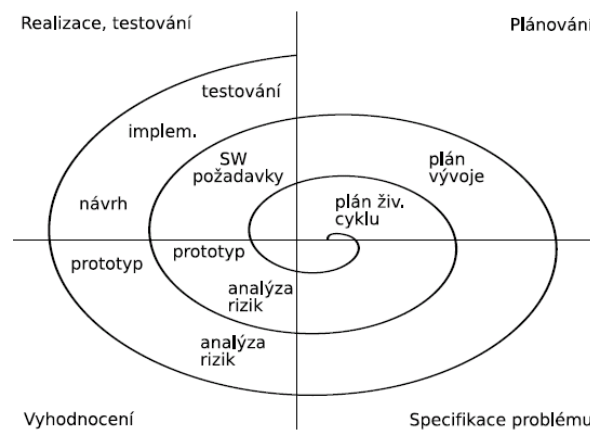
Inkrementálny model je veľmi blízko k interaktívnemu modelu. Tu totiž na základe špecifikácii celého systému sa stanovia ucelené časti a odovzdávajú sa užívateľovi postupne. Štruktúra v tomto prípade môže byť navrhnutá lepšie.



Obraz č. 5. - Iteratívny model [2]

2.1.4.3 Špirálový model

U špirálového modelu sa kladie dôraz na analýzu rizík, tu sa objaví pojem prototyp, čo je vlastne predbežná verzia výsledného produktu, každá predchádzajúca verzia sa zahodí a vytvorí sa nová. Nedostatkom je, že musíme v tomto modeli vyvíjanom systéme viac programovať, zas výhodou je, že spustiteľné verzie už od začiatku existujú a tak je odhalené viac chýb.



Obraz č. 6. - Špirálový model [2]

2.1.5 Agilná metodológia

Metodológia založená na klasických modeloch životného cyklu projektu musia dodržiavať rady pravidiel a preto často vývoj nie je efektívny a pôsobí predrazenie celého systému. Odpoveďou na tieto problémy boli agilné metodológie, ktoré kladli väčší dôraz na ľudský faktor pri vývoji výsledného produktu.

Sem patrí napríklad extrémne programovanie, kde vysoké nároky sú kladené na komunikáciu medzi členmi vývojári, manažéri a zákazníci. Charakterizuje ďalej snahu mať čo najrýchlejšie mať

najdôležitejšiu časť systému v prevádzke, čiže spätná väzba by mala byť rýchla a jednoduchá. Niekedy odvaha na zahodenie už hotového kódu a vyskúšať niečo nového.

2.1.5.1 Rational Unified Process

Rational unified process je koncepcia založená podľa výskumu firmy Rational, kde vývoj softvérového produktu je riadený iteračným spôsobom, po každej iterácii je dostupný spustiteľný kód, dôraz je kladený na vizualizáciu systému riadenie zmien a priebežná kontrola kvality. Základným vyjadrovacím prostriedkom je jazyk UML.

Pre zachytenie požiadaviek sa v RUP používajú diagramy prípadu užívania, je súčasťou modelovacieho jazyka UML. V diagrame prípadu užívania jednotlivé požiadavky sú znázornené ako účastníci a ku každému účastníkovi je priradený detail vybraného prípadu užívania. Jeden prípad užívania môžeme chápať ako funkciu, ktorú systém vykonáva v mene jednotlivých účastníkov alebo v ich prospech. Každý prípad užívania má svoj názov, jednoznačný identifikátor a špecifikáciu. Diagramy užívania znázorňujú:

- hranice navrhovaného systému,
- účastníkov analyzovaných prípadov užívania,
- analyzované prípady užívania.

V projekte z uvedených možností som vybral technológiu Rational Unified Process a UML diagramy, lebo tento prístup je vytvorený na základe poznatkov a skúseností viacerých programátorov. Pomocou agilných metodológií je možné jednoduchšie riadiť požiadavky od užívateľov, vyvíjať prípady užívania, modelovať údaje a v prípade zmeny je jednoduchšia aj správa.

2.2 Analýza požiadaviek

V tejto časti dokumentácie je popísané, čo všetko potrebuje zadávateľ projektu vo svojom informačnom systéme. Jednotlivé názory som pozbieral do niekoľkých bodov a charakterizoval som jeho vlastnosti podľa nasledujúcich bodov:

- popis požiadaviek - obsahuje samotný popis požiadavku, čo od nás zadávateľ potrebuje,
- priorita - priorita na vyriešení požiadavku (1 - najvyšší ... 5- najnižší),
- náročnosť - predpokladaná náročnosť na vyriešenie (1 - najvyšší ... 5- najnižší).

2.2.1 Užívatelia systému

Služby hracieho servera užívateľa môžu využívať až po prihlásení, čoho podmienkou je registrácia na stránke. Pre registráciu je potrebné poskytnúť základné osobné údaje (meno, heslo, email) prípadne užívateľ môže poskytnúť aj ďalšie údaje, napr. výšku ELO bodov. Keď užívateľ neposkytne svoju hodnotu ELO bodov, tak server užívateľovi automaticky priradí základnú hodnotu ELO bodov.

V prípade, že registrovaný užívateľ zabudol svoje heslo, bola by možnosť zaslania nového hesla na zadanú emailovú adresu. Užívateľia, ktorí využívajú služby šachového servera, by boli rozdelení do päť skupín.

2.2.1.1 Neregistrovaný užívateľ

Prvú skupinu by tvorili neregistrovaní užívateľia, ktorí by mali možnosť sledovať len rating registrovaných hráčov bez možnosti hrania šachových partií.

Priorita: 1 Náročnosť: 5

2.2.1.2 Registrovaný užívateľ

Všetci ostatní registrovaní užívateľia po prihlásení sa na server by mali možnosť sledovať rating registrovaných hráčov a zároveň by mohli pozvať druhého registrovaného užívateľa na odohranie on-line šachovej partie. Toto pozvanie druhá strana (pozyvaný) má možnosť odmietnuť alebo prijať. Po prijatí pozvania na hru je možnosť na odohranie partie.

Priorita: 1 Náročnosť: 4

2.2.1.3 Správca systému

Ďalšiu skupinu by tvorili systémoví užívateľia, ktorí by mali aktívny prístup k serveru v tom zmysle, že by mohli editovať informácie registrovaných hráčov. Jeho prvoradou úlohou je vyriešiť prípadné problémy pri registrácii, správa užívateľov a pridelenie prípadne zmena práv. Prípadne by mohol vylúčiť neaktívnych užívateľov.

Priorita: 2 Náročnosť: 4

2.2.1.4 Správca turnajov

Má za úlohu organizovať turnaje. Iba takýto užívateľ má právo založiť alebo editovať turnaje. Môže tam definovať určité obmedzenia, ktoré spôsobia, že len určití registrovaní užívateľia majú možnosť prihlásiť sa na turnaj. Také obmedzenie môže byť napríklad výška ELO bodov.

Priorita: 4 Náročnosť: 2

2.2.1.5 Rozhodca

Užívateľ s právami rozhodcu, by mohol ovplyvňovať priebeh partií. Napríklad keď nejaká partia sa ukončí, rozhodca môže takúto partiu analyzovať a zmeniť konečný výsledok.

Priorita: 3 Náročnosť: 2

2.2.2 Funkcionalita

Hlavným zmyslom zadania je prevádzkovanie hracieho servera. Server by mal umožňovať zahraničie šachu medzi dvoma registrovanými užívateľmi. Užívateľia medzi sebou by komunikovali pomocou

správ, takzvaných pozvánok. Po dokončení hry, podľa výsledku, ELO body by boli prepočítané. Jednotlivé požiadavky na funkcionality sú rozobrané v nasledujúcich bodoch.

2.2.2.1 Posielanie pozvánok

Po prihlásení užívateľ môže poslať vybranému hráčovi správu, že má záujem s ním zahrať partiu, druhá strana vidí, že dostal pozvánky od hráčov a podľa nálady môže ich prijať, alebo odmietnuť. Pomocou posielania správ sú informovaní užívatelia aj o iných udalostiach, ako požiadavka na remízu, vzdanie sa partie alebo vrátenie stavu v partií.

Priorita: 1 Náročnosť: 2

2.2.2.2 Hranie hry

Ak dvaja užívatelia úspešne založia hru, budú mať možnosť zahrať si šachovú partiu. Pri hraní jednotlivé ťahy sú urobené figúrkami. Správny pohyb figúrok musíme kontrolovať. V prípade o pokus nemožného ťahu sa ťah neprevedie. Tu musíme dávať zvláštny pozor na analýzu šachov a matov.

Priorita: 1 Náročnosť: 1

2.2.2.3 Počítanie ELO

Počítanie ELO by sa mala uskutočniť po dokončení šachovej partie. Na základe starého ELA a ELA protihráča sa vypočíta nové ELO hráča a tak zistený nový výsledok by nahradil starú hodnotu. Počítanie ELA vlastne po každej ukončenej hre sa znovu prepočíta a vymení.

Priorita: 2 Náročnosť: 1

2.2.2.4 Vedenie notácie

V reálnom svete len pri takých partií počítajú ELO body, kde sú zaznamenané na partiáry ťahy spravené v partiu. Preto je potreba aj v tomto projekte zobrazit' jednotlivé ťahy hráčov. Na boku obrazovky by sa mal objaviť formulár, ktorý by obsahoval postupnosť ťahov hráčov. Takáto notácia môže byť i cudzojazyčná.

Priorita: 3 Náročnosť: 2

2.2.2.5 Správny ťah s figúrkou

Každá figúrka na šachovnici má určitú farbu a typ. Podľa typu majú iné vlastnosti. Figúrky sú nasledovné: pešiak, veža, strelce, jazdec, dáma, kráľ. Ku každej figúrke budeme potrebovať nadefinovať funkciu, ktorá bude charakterizovať jej vlastnosti.

Pešiak je kameň, ktorý kráča len po frontále smerom dopredu o jedno pole, zo základného postavenia partie môže kráčať o jedno, alebo o dve polia dopredu. Berie o jedno pole po diagonále vo dvoch smeroch dopredu. Smerom dopredu sa rozumie smer, ktorým sa kameň približuje

k súperovi (vzťahovaný na základné postavenia partie). Zvláštne ťahy pešiaka sú premena pešiaka a branie mimochodom.

Jazdec je kameň, ktorý ťahá na druhé najbližšie pole opačnej farby v akomkoľvek smere bez ohľadu na to, ktoré z ostatných polí šachovnice sú prázdne a ktoré obsadené.

Strelec je kameň, ktorý ťahá po prázdnej diagonále v akomkoľvek smere o ľubovoľný počet polí, ak na konci prázdnej diagonály stojí kameň opačnej farby, strelc ho môže brať.

Veža je kameň, ktorá (s výnimkou rošády) ťahá po prázdnej frontále v akomkoľvek smere o ľubovoľný počet polí, ak na konci prázdnej frontály stojí kameň opačnej farby, veža ho môže brať.

Dáma je kameň, ktorý ťahá buď ako veža, alebo ako strelc.

Kráľ je kameň, ktorý po vlastnom ťahu, alebo po ťahu iného kameňa rovnakej farby nesmie byť v šachu (kráľovi). Šach (kráľovi) je postavenie, v ktorom na pole kráľa pôsobí súperov kameň. Kameň pôsobí na pole, ak na tomto poli môže brať súperov kameň, pôsobí však naň aj vtedy, keby toto branie nemohol uskutočniť pre porušenie pravidiel. Kráľ je kameň, ktorý (s výnimkou rošády) ťahá na najbližšie pole po frontále alebo diagonále v akomkoľvek smere.

Priorita: 1 Náročnosť: 1

2.2.2.6 Rošáda

Z hora uvedených pravidiel ťahov sú niektoré takzvané zvláštne ťahy. Jeden z nich je Rošáda.

Rošáda je súčasný ťah kráľa a veže tej istej farby z polí, na ktorých stoja v základnom postavení partie: kráľ ťahá o dve polia smerom k veži a veža, ku ktorej sa priblížil, postaví sa na pole medzi pôvodným a novým poľom kráľa; rošádu možno vykonať len vtedy, ak sú polia medzi kráľom a vežou prázdne, ak kráľ a veža od začiatku partie neťahali a ak na pôvodné a nové pole kráľa, ako aj na pole medzi nimi nepôsobí súperov kameň.

Priorita: 1 Náročnosť: 2

2.2.2.7 Branie mimochodom

Tiež jeden zo zvláštnych ťahov. Pešiak, ktorý môže brať súperovho pešiaka bezprostredne po jeho kroku zo základného postavenia partie o jedno pole, môže ho brať aj bezprostredne po jeho kroku o dve polia, a to práve tak, ako keby bol kročil iba o jedno pole. Takýto ťah sa nazýva branie mimochodom.

Priorita: 1 Náročnosť: 1

2.2.2.8 Premena pešiaka

Nepatrí až tak do zvláštnych ťahov, ide o takú situáciu, keď pešiak postúpi až na posledné pole, ide o jeho odstránenie zo šachovnice a nahradenie dámou, vežou, strelcom, alebo jazdcom rovnakej farby na tom istom poli. Takýto ťah sa nazýva premena pešiaka. Pešiak sa premení na jeden z uvedených kameňov bez ohľadu na druhy a počty ostatných kameňov na šachovnici.

Priorita: 2 Náročnosť: 1

2.2.2.9 Napadnutie kráľa - analýza ťachov

Pokiaľ hráč nejakým ťahom napadne súperovho kráľa, tzn. presunie sa tak, že by potenciálne nasledujúcim ťahom mohol kráľa brať, hovoríme, že súperovi dal ťach. Pokiaľ takáto situácia nastane, súper je povinný hrať tak, aby túto hrozbu odvrátil, tzn. jedným z nasledujúcich spôsobov.

Tiahnuť kráľom na pole, ktoré nie je napadnuté, alebo pred kráľa postaviť nejaký kameň tak, aby napadajúci kameň odvrátil. Tento spôsob nie je možné použiť v situácii, keď súper napadá kráľa svojím jazdcom, lebo jazdec sa môže pohybovať (alebo brať) aj cez bránice kamene. Ak predchádzajúce možnosti nevieme uplatniť existuje ešte možnosť napadajúci kameň vybrať.

Priorita: 1 Náročnosť: 1

2.2.2.10 Výhra - matovanie kráľa

Pokiaľ neexistuje žiadny taký ťah, ktorý by vyhovoval pravidlám, znamená to mat (tiež ťach-mat), tzn. koniec hry, víťazstvo hráča, ktorý takto úspešne súperovho kráľa napadol. Musíme dávať pozor aj na to, že hráč musí napadnutého kráľa ochrániť (alebo hra skončí), v hre teda nemôže nastať situácia, že by hráč súperovho kráľa zobral zo šachovnice.

Priorita: 1 Náročnosť: 1

2.2.3 Dáta

Pri uložení dát do databázy je požiadavka na minimálne zaťaženie siete. Do databázy budú vlastne ukladané informácie o užívateľoch, o aktuálnej pozícii figúrok a o ťahoch. O užívateľoch sú ukladané informácie: meno, priezvisko, email, ELO body a získané body za partie. Pri pozíciách figúrok bude potreba zaznamenať typ, farbu a pozíciu. Pozícia bude definovaná dvoma číslami. Jeden bude určovať riadok na šachovnici a druhý stĺpec na šachovnici.

Pri ťahov budeme potrebovať dve koordináty, a to počiatočný stav, kde stála figúrka a koordináty cieľového poľa.

2.2.4 Rozhranie systému

Komunikácia medzi užívateľom a systémom sa bude odohrávať cez užívateľské rozhranie. Na užívateľské rozhranie neboli kladené zvláštne požiadavky, preto som sa rozhodol zobrazíť jednotlivé položky v menu a na stránke čím jednoduchšie a prehľadnejšie. Väčšiu pozornosť pri

vytvorení rozhrania by mala dostať šachovnica, aby aj menej zdatný hráči sa vedeli na nej vyznať. Dobré by bolo, keby pohľad na šachovnicu vedel užívateľ sám prispôbiť.

2.2.4.1 Kompatibilita v prehliadačoch

Užívateľské rozhranie by nemalo spôsobiť problémy v rôznych internetových prehliadačoch, ako Internet Explorer, FireFox alebo Opera.

Priorita: 3 Náročnosť: 2

2.2.4.2 Vygenerovanie šachovnice

Pred tým, než začneme implementovať šachovú hru, musíme pochopiť a premyslieť si, ako vytvoríme šachovnicu a ako budeme na nej zobrazovať figúrky. Musíme určiť minimálne softwarové požiadavky na šachovú hru. Základné otázky, ktoré budeme musieť riešiť budú: Ako reprezentovať šachovnicu na stránke, aby v pamäti počítača zaberol čo najmenej miesta? Ako reprezentovať na šachovnici figúrky a stále uchovávať tento aktuálny stav? Vymyslieť správny algoritmus, ktorý analyzuje správnosť aktuálneho ťahu, v prípade nesprávnosti upozorňuje. Užívateľské grafické rozhranie na posun figúrok z jedného na druhé miesto. Ako podľa pravidiel šachovej hry vieme, šachovnica je tvorená zo 64 políčok rozdelených na 8 riadkov a 8 stĺpcov, ktoré sa skladajú striedavo z čiernej a bielej farby.

Priorita: 1 Náročnosť: 2

2.3 Modely navrhovaného systému

Ako už v predchádzajúcich kapitolách som uviedol v projekte som použil ako model navrhovaného systému modelovací jazyk UML (Unified Modeling Language). Pred samotnou implementáciou som vymyslel dva modely, USE CASE a E-R diagram, ktoré reprezentujú podstatu komplexného problému. Tieto diagramy umožňujú pochopiť podstatu problému aj spôsob jeho riešenia.

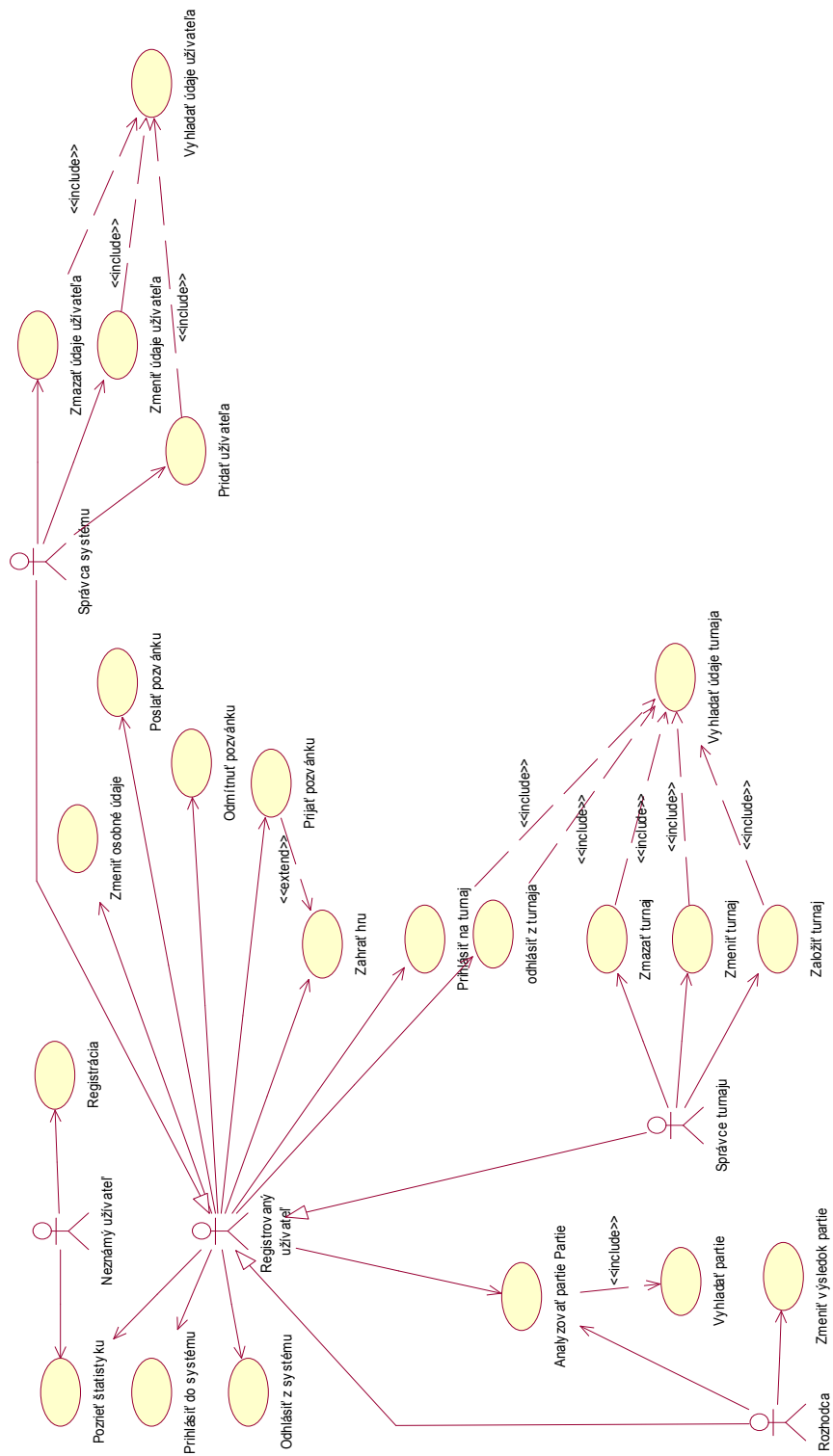
2.3.1 Use case diagram

Prvým navrhovaným diagramom je use case diagram (obr. č.7.). Ten definuje a modeluje systém z pohľadu užívateľov.

V mojom prípade, ako je to znázornené na obrázku, je tam päť aktérov, neznámy užívateľ registrovaný užívateľ, rozhodca správca turnajov a správca systému. Neznámy užívateľ má možnosť pozrieť len na štatistiku alebo zaregistrovať na stránku. Po zaregistrovaní sa stane registrovaným hráčom a má už možnosť prihlásiť sa a odhlásiť zo systému. Po prihlásení môže zmeniť niektoré osobné údaje v systéme, ale najdôležitejšie je, že má možnosť poslať pozvánku iným hráčom. Ak niekto takúto pozvánku prijme, môžu si zahrať šachovú partiu.

Správca zodpovedá za užívateľov systému, môže užívateľom priradiť práva prípadne aj zmazať ich zo systému. Správca turnajov zase zodpovedá za turnaje. On je vlastne ten človek, kto založí nový turnaj a edituje alebo pridá určité obmedzenia.

Rozhodca v systéme je ten človek, ktorý zodpovedá za dodržanie pravidiel v šachu, keď ho niekto vyžiada, môže preanalyzovať partiu a zmeniť výsledok ukončenej šachovej partie.



Obraz č. 7. - Use case diagram systému

2.3.2 E-R diagram

E-R diagram je grafická reprezentácia entít, vzťahov a obmedzení, ktoré tvoria daný návrh. Tak isto ako u iných grafických návrhových metodológií, poskytuje grafický súhrn návrhov. Je to veľmi užitočné pre návrhára, nielen pre validáciu správnosti návrhu, ale i pre diskusiu s kolegami a programátormi, ktorým je treba problém vysvetliť. Neexistujú obecné platné konvencie pre nakreslenie E-R diagramov a preto môžu existovať rozdiely a rôzne typy E-R diagramov.

E-R diagram obsahuje entity, atribúty, vzťahy:

- Entita – vec reálneho sveta (objekt) rozlíšiteľný od iných objektov
- Atribút – vlastnosť entity, ktorá nás v kontextu daného problému zaujíma
- Vzťah – asociácia medzi niekoľkými entitami

Pri vytvorení E-R diagramu je treba rozhodnúť, ktoré entity, vzťahy a obmedzenia použijeme pre modelovanie fyzického systému. U jednoduchého príkladu môže toto rozhodnutie byť jednoduché, ale pre praktické informačné systémy ide čiastočne i o umenie a skúsenosti.

V mojom E-R diagrame sú vytvorené nasledovne tabuľky užívateľ, profil, správy, hry, ťahy, figúrky, turnaje a zaregistroval.

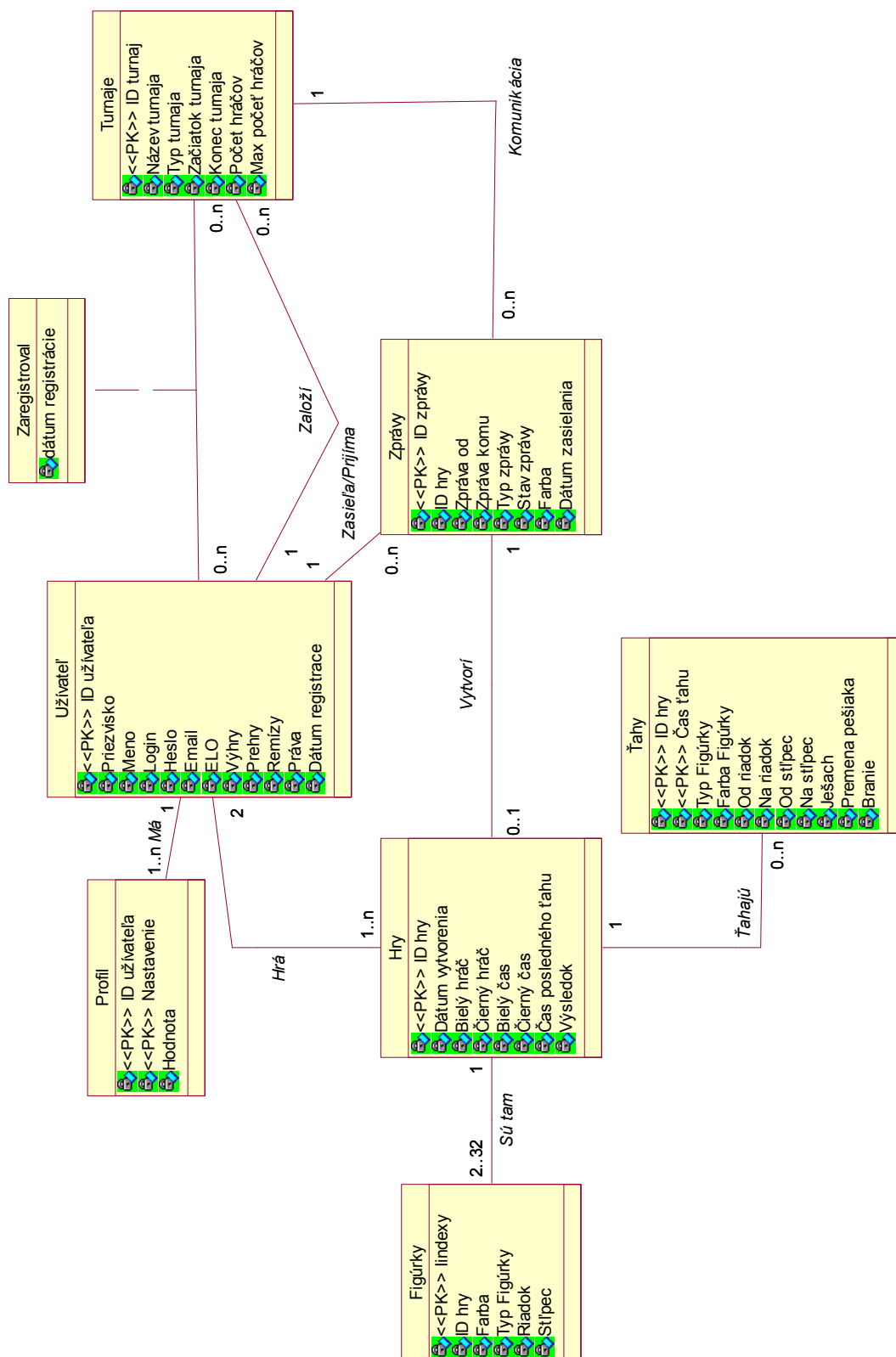
V tabuľke užívateľ sú ukladané základné informácie o užívateľovi ako meno, priezvisko, email, login, heslo, jeho získané body a ELO body. Na túto tabuľku sa napája tabuľka profil, ktorá obsahuje jednotlivé osobné nastavenia užívateľov, ako napríklad za koľko minút sa má stránka aktualizovať, jazyk notácie, alebo štýl zobrazenia šachových figúrok.

Komunikácia medzi užívateľom prebehne pomocou tabuľky správy. Jedna správa je viazaná ku hre medzi dvomi užívateľmi, je tam typ správy, stav správy, kedy bola poslaná a farba kto poslal správu. Správy vytvoria hru.

Tabuľka hry nesie v sebe hodnoty bieleho a čierneho hráča, uplynutý čas v hre, dátum vytvorenia hry, dátum posledného ťahu a výsledok. Ku každej hre patria figúrky počet figúrok v jednej hre môže byť maximálne 32 (všetky figúrky sú postavené) a minimálne 2 (len dve králi). U figúrok v tabuľke je definované, ku ktorej ID hre patria, akú farbu majú, o aký typ figúrky ide. Riadok a stĺpec jednoznačne určí pozíciu na šachovnici. Ťahy figúrok je zaznamenaný do tabuľky ťahy. V tabuľke ťahov je postupne vložený čas ťahu a ID hry, v ktorom bol urobený. Zaznamená ešte akou figúrkou a farbou, z ktorého zdrojového riadku a stĺpca idem na cieľový riadok a stĺpec. Pritom sú tam pomocné prvky na zaznamenanie šachov, branie mimochodom a premeny pešakov.

V tabuľke turnaje sú uložené dáta týkajúce turnajov, ako identifikačné číslo, názov, typ, začiatok a koniec turnaja, maximálny počet hráčov a počet už zaregistrovaných hráčov. Dáta o registrovaných hráčoch je uložený v tabuľke zaregistroval, kde registrácia je definovaná s dátumom uskutočnenia tejto udalosti.

Tento E-R diagram nájdete na obrázku č. 8.



Obraz č. 8. - ERD diragram šachového servera

2.4 Analýza možností riešenia

V tejto kapitole budem vlastne podrobnejšie pozerat' na možnosti riešenia projektu, pozrieme sa na možnosti ako zistiť šach na kráľa, ako je notácia partii a ako sa vypočíta nová hodnota ELO bodov, uvediem niekoľko možností a myšlienok, ako by koncové riešenie malo vypadat'.

2.4.1 Ako testovať šachy kráľa

Šach je vlastne vtedy, keď konkrétna figúrka opačnej farby má možnosť vstúpiť na to pole, kde stojí kráľ protihráča. Test na to, že konkrétna figúrka má takúto možnosť, musíme otestovať. Ako prvé musíme pristúpiť k tomuto problému, ako zistiť pole kráľa, kde stojí. Najjednoduchším spôsobom je jedným cyklom prejsť cez prvky pola a hľadať kód kráľa, hodnotu pola, ktorá zodpovedá hodnote kráľa. Keď sme zistili takúto hodnotu, tak pôjde o figúrku typu kráľa. Koordináty tohto pola zapamätám.

Potom s ďalším cyklom by som opäť prešiel cez polia šachovnice (cez každé pole musím prejsť aj cez prázdne i obsadené). V prípade keď nájdem figúrku a figúrka, ktorú som našiel je opačnej farby skontrolujem, či má možnosť na pole, kde stojí kráľ pristúpiť. Ak to je možné, tak by išlo o šach, ak nie je možné, tak nie je šach. Tento spôsob by však bol z hľadiska výpočtu veľmi náročný, lebo musím skontrolovať celú šachovnicu a pritom keď nájdem figúrku, musím skontrolovať jej farbu, a až potom možnosti jeho ťahov.

Ďalší nápad bol, že šachy nehľadať od zdroja, od figúrky ktorý napadol kráľa, ale práve od cieľa naspäť, od kráľa otestovať všetky možné body, odkiaľ by vedelo niečo kráľa ohroziť. To by fungovalo tak, že spresním miesto kráľa a od toho dookola začnem otestovať všetky susedné a susedné polia. Tak by som vlastne išiel po špirále dookola cez šachovnicu, a keď nájdem nejakú figúrku, otestujem jej farbu a potom jej možnosť pristúpiť na pole kráľa. Problém by bol, že opäť musím prejsť cez všetky polia šachovnice a ani tak by sa výpočty až tak nezlepšili ako v predchádzajúcom prípade, ale prístup, že začínam analyzovať šachy od kráľa už bola oveľa lepšia myšlienka.

Nasledujúci nápad bol, že najprv by som mohol nejako otestovať špeciálne ťahy koňa na šachy. To by som tiež začal analyzovať zo zdrojového pola, kde stojí kráľ a otestoval by som všetky okolité polia kam by bolo možné umiestniť koňa tak, aby vedel vstúpiť na pole kráľa. Ak berieme, že kráľ stojí niekde na strede šachovnice, tak môžeme rozlíšiť 8 možných stavov, kde kôň môže ohroziť kráľa. To sú nasledovné dva políčka dopredu a potom doľava alebo doprava, jedno políčko dopredu a potom doľava alebo doľava dva políčka. Ďalej odzrkadlene dole dva políčka a potom doľava alebo doprava jeden prípadne dole jedno políčko a potom tiež doľava alebo doprava dva políčka.

V prípade ostatných figúrok budeme mať trošku ťažšiu úlohu. Ako prvé najprv budeme musieť určiť smer, po ktorom budeme hľadať konkrétnu figúrku. Z pohľadu kráľa to bude 8 možných

smerov, a to po riadkoch doprava a doľava, po stĺpcoch hore a dole. Po diagonálach zase z ľavého horného rohu, z ľavého dolného rohu, z pravého horného rohu a z pravého dolného rohu. Na každé z možných ôsmich smerov postupom pozrieme na figúrky. Keď nájdeme nejakú figúrku, tak informácie o tejto figúrke uložíme do nejakého premenného. Uložíme tam vlastne farbu, typ, presné koordináty, kde sa nachádza a potrebujeme i diaľku (v prípade pešiakov s diaľkami väčším ako jedna sa nebudeme zaoberať, preto musíme ukladať diaľku). Tu budeme musieť dávať pozor na okraji šachovnice, totiž aj tu sa môže stať, že hľadané pole už neexistuje (je mimo šachovnice). Takto ukladané figúrky z ukladaného poľa postupne vyberáme a pozeráme na jeho dáta. Ak farba je opačná a má možný ťah na pole kráľa, pôjde o šach.

Táto posledná metóda je najnáročnejšia, ale z pohľadu výpočtov by mala najmenej obťažovať počítač. Preto som pre riešenie problému vybral túto metódu.

2.4.2 Možnosti ako zakončiť partiu

Partiu vyhráva ten hráč, ktorý dá súperovi mat alebo súper prehlási, že sa vzdáva. Tím partia okamžite skončí. Po skončení partie keď nie je inak stanovený ten hráč, ktorý vyhrá hru získa jeden bod (1), za prehru nezíska žiadny bod (0). Existuje i taká situácia keď hra je nerozhodná je remíza v tom prípade obidvaja dostanú pol bodu (1/2). Analýza hore uvedených možností je rozobraná v nasledujúcich podkapitolách.

2.4.2.1 Nerozhodná hra - patové situácie

Partia je nerozhodná, ak kráľ hráča, ktorý je na ťahu nie je v šachu a hráč nemôže previesť žiadny prípustný ťah. Na takúto situáciu hovoríme, že ide o pat. Ďalšia patová situácia je, keď vznikne taká pozícia, kde ani jednému z kráľov sa nedá dať šach-mat, tomu hovoríme, že ide mŕtvu pozíciu. Taká mŕtva pozícia je napríklad ak na koniec partie jednému z hráčov zostane iba jeden strelec alebo jeden jazdec. V niektorých prípadoch ani s dvoma jazdcami sa nedá vyhrať partiu.

Analyzovať však také pozície je veľmi ťažké, jednoduchšia je v takom prípade využiť pravidlo, podľa ktorého ak obaja hráči previedli posledných 50 po sebe idúcich ťahov a to bez toho, že by vybrali nejaký kameň alebo ťahali s pešiakom, je hra nerozhodná. To by sa dalo kontrolovať tak, že zaznamenáme jednotlivé ťahy a každému ťahu priradíme postupne čísla od 1 až do 50, ak však ide o branie figúrku alebo pohyb s pešiakom tento počítadlo nulujeme a počítanie začína zase od 1. Ak dosiahneme číslo 50, tak hra skončí a zaznamená obidvom hráčom polovina bodu.

Počas hrania hry je aj taká možnosť, že súper sa dohodnú počas hry o remízu. To by som v mojom systéme by vyriešil tak, že ten kto ponúkne remízu pošle namiesto pozvánky správu o tejto úvahe. Túto správu súper dostane a môže byť prijať alebo odmietnuť. Keď súper odmietne, tak hra pokračuje, keď prijme, tak hra je skončená remízou.

Ďalšia možnosť na situáciu remízy je, keď taká istá pozícia sa vyskytne najmenej trikrát počas hry. Tento jav by som detektoval tiež tak, že informácie o ťahoch budú ukladané a porovnané. Aj tu

by platilo pravidlo, keď ide o posun s pešiakom alebo vyberieme figúrku analýza na remízu začína opäť od začiatku. To, že ten istý stav sa opakuje je vlastne detekovaný, tak že tá istá figúrka spravila ten istý ťah opakovane trikrát, kým súperova figúrka tiež trikrát.

2.4.2.2 Výhra - ako na matové situácie

O mate môžeme rozprávať vtedy, ak kráľ súpera má šach a nemôže sa pohnúť s kráľom tak, aby z tohto šachu vystúpil, ostatné figúrky jeho i súperove v tom bránia, a figúrku, ktorá ohrozuje kráľa nie je možné zo šachovnice vybrať alebo nie je možné pokryť šach inou figúrkou. Takže analyzujeme tieto tri prípady.

Za prvé kráľ dostane šach, vtedy má, vlastne bude mať šach. Najprv sa budeme zaoberať tým, či kráľ, ktorý má šach, môže pohnúť na také susedné pole, kde nie je ohrozený prípadne obsadený iným kameňom. Pri analýze šachov môžeme využívať metódu, ktorú som už popísal v kapitole ako testovať šachy kráľa. Keď nenájdem také miesto kde by kráľ mohol postúpiť bez toho aby mal šach postúpime na ďalší bod.

Druhou možnosťou je, keď ide o figúrku, ktorou sme dali šach a je možné odstrániť z šachovnice, môžeme ju vybrať, tiež nejde v takom prípade o mat. Kameň, ktorú môžeme vybrať bez toho aby potom ohrozili kráľa nie je matom. Ak ani branie nie je možný, tak ideme na nasledujúci tretí bod.

Tretou možnosťou zostane blokovať cestu figúrky a tak vlastne blokovať šach na kráľa. Jazdca blokovať nemôžeme, ale ostatné figúrky áno, keď budeme testovať, táto možnosť vlastne analyzuje, či je možné umiestniť figúrku medzi kráľom a atakujúcou figúrkou.

Zaujímavý je prípad, ak kráľ je ohrozený dvomi súperovými figúrkami, to znamená že nie je postačujúce skontrolovať len jednu ale musíme všetky možné polia otestovať na súperove figúrky.

Ak z hore uvedených možností ani jeden nie je možné splniť ide o matovú situáciu a hra sa ukončí výhrou hráča, ktorý spravil posledný ťah.

2.4.3 Notácia partíí

Federácia FIDE uznáva iba také turnaje a zápasy, kde partie sú zaznamenané. Na zaznamenanie partie však potrebujeme určité ustanovenia. FIDE preto doporučuje používať jednotné šachové notácie (taká notácia je používaná i v literatúre a časopisoch). Je to systém algebraický, kde každý kameň okrem pešiaka (pešiak počas zaznamenania nemá žiadne zvláštne označenie, vo väčšine prípadov ich označenie chýba) je označený väčšinou prvým písmenom svojho mena a to veľkými písmenami. Tu však musíme upozorniť, že u rôznych národov toto písmeno sa môže líšiť, preto niektoré publikácie dávajú prednosť grafickým označeniam, rôzne obrázky kameňov. Cudzojazyčná notácia je pre niektoré jazyky je uvedená v tabuľke č.1.

Slovenská notácia	Anglická notácia	Nemecká notácia	Španielska notácia	Francúzka notácia
K - Kráľ	K - King	K - König	R - Rey	R - Roi
D - Dáma	Q - Queen	D - Dame	D - dama (reina)	D - Dame
V - Veža	R - Rook	T - Turm	T - torre	T - Tour
S - Strelec	B - Bishop	L - Läufer	A - alfil	F - Fou
J - Jazdec	N - Knight	S - Springer	C - caballo	C - Cavalier
P - Pešiak	P - Pawn	B - Bauer	P - peón	P - Pion

Tabuľka č. 1. - Cudzojazyčné notácie

Aby sme vedeli zaznamenať presne ťahy na šachovnici osem stĺpcov zľava doprava pre bieleho hráča, sprava doľava pre čierneho hráča je označená písmenami od *a* do *h*. Kým osem riadkov zospodu nahor zo strany bieleho hráča a z hora dole zo strany čierneho hráča je očíslované od *1* do *8*.

V základnom stave sú teda biele kamene postavené na 1. a 2. riadku a čierne kamene na 7. a 8. riadku. Vieme, že šachovnicu tvorí 64 políčok, z toho a z hore uvedených postavení môžeme každé pole na šachovnici jednoznačne adresovať pomocou kombinácie písmena a čísla. Vlastne každý ťah je jednoznačne označený začiatočným písmenom názvu príslušného kameňa a názvom cieľového poľa, na ktoré figúrkou ťaháme. Pešiáci so svojim písmenom neoznačujú sa poznáme ich podľa toho, že písmeno tam chýba.

Ak ide o branie figúrky tento jav je zaznamenaný tak, že medzi prvé písmeno kameňa a cieľové pole vloží symbol "x". V prípade pešiakov je symbol "x" vložený medzi zdrojovým stĺpcom a cieľovým.

Špeciálne možnosti ťahov tiež musíme nejako zaznamenať, u rošády je to značené 0-0 (malá rošáda) alebo 0-0-0 (veľká rošáda). Pri premene pešiaka na konci adresy ťahu ešte pripisuje písmeno figúrky, na ktoré bola premena uskutočnená. Pri braní mimochodom je zase na konci skratka e.p čo znamená, že ide o ťah typu en passant.

2.4.4 Výpočet ELO hodnotenia - rating

Teda ako sa počíta ELO rating? Už v roku 1959 USCF rating systém svojvoľne určilo jednu hornú hranicu 2000, čo zodpovedalo úrovni silných registrovaných hráčov, a zoradilo ich do tried po 200 bodov. Élő ponechal toto tradičné rozdelenie. Z tabuľky č.2. je možno vyčítať hodnoty pre pravdepodobný výsledok jednej partie (P-stĺpec), pričom za podklad je zobrať rozdiel ELO

bodov. (D-stĺpec). Napríklad dvaja hráči s rovnakým ratingom (D=0) rovnako 50 % (P=0,5) šanci majú proti sebe. Hráč ktorý má rating vyšší o 100 bodov (D=102 je najbližšia hodnota v tabuľke) má šance 64 % (P=0,64) na víťazstvo proti slabšiemu hráčovi.

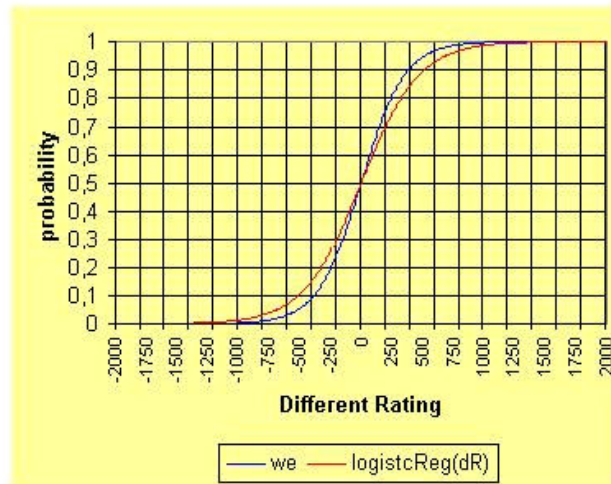
Očakávaný výsledok je vypočítaný na základe nasledujúceho vzorca [11]:

$$W_e = \frac{1}{10^{\frac{\Delta R}{400}}} + 1$$

- **W_e** - Očakávaný výsledok
- **ΔR** - Rozdiel medzi ELA hráčov

P	D	P	D	P	D	P	D	P	D	P	D
1.0		.83	273	.66	117	.49	-7	.32	-133	.15	-296
.99	677	.82	262	.65	110	.48	-14	.31	-141	.14	-309
.98	589	.81	251	.64	102	.47	-21	.30	-149	.13	-322
.97	538	.80	240	.63	95	.46	-29	.29	-158	.12	-336
.96	501	.79	230	.62	87	.45	-36	.28	-166	.11	-351
.95	470	.78	220	.61	80	.44	-43	.27	-175	.10	-366
.94	444	.77	211	.60	72	.43	-50	.26	-184	.09	-383
.93	422	.76	202	.59	65	.42	-57	.25	-193	.08	-401
.92	401	.75	193	.58	57	.41	-65	.24	-202	.07	-422
.91	383	.74	184	.57	50	.40	-72	.23	-211	.06	-444
.90	366	.73	175	.56	43	.39	-80	.22	-220	.05	-470
.89	351	.72	166	.55	36	.38	-87	.21	-230	.04	-501
.88	336	.71	158	.54	29	.37	-95	.20	-240	.03	-538
.87	322	.70	149	.53	21	.36	-102	.19	-251	.02	-589
.86	309	.69	141	.52	14	.35	-110	.18	-262	.01	-677
.85	296	.68	133	.51	7	.34	-117	.17	-273	.00	
.84	284	.67	125	.50	0	.33	-125	.16	-284		

Tabuľka č. 2. - Tabuľka pravdepodobnosti výhry [10]



Obraz č. 9. - Grafikon pravdepodobnosti výhry [12]

Predpokladajme, že si hral na jednom turnaji a tvoje výsledky sú +3-2=1 (t.j. 3 výhry, 2 prehry, 1 remíza) a priemerný rating tvojich súperov bol 1500. Tvoj výsledok proti nim bol 3,5-2,5=1, čo zodpovedá 58%-nému plneniu ($P=0,58$). Pre túto hodnotu v tabuľke zodpovedá $D=57$ bodový rozdiel, teda tvoj výkon v ELO bodoch je $1500+57=1557$. Ak priemerná hodnota súperových ELO bodov by bol 2000, potom tvoj výkon vyjadrený v ELO bodoch by bol 2057.

Tento spôsob sa používa na určenie počiatočného ratingu pre hráča, ktorý ešte nemá ELO rating. Čím viac partii tvorí základ výpočtu, tým bude presnejší počiatočný rating. Hráčom, ktorí už majú ELO body, zmena ELO bodov sa vypočíta podľa nasledovného vzorca:

$$R_n = R_o + K (W - W_e)$$

- **R_n** = Nové Elo
- **R_o** = Staré Elo
- **K** = „hodnota“ jednej partie, násobiteľ
- **W** = Výsledok, 1.0 za víťazstvo, 0.5 za remízu, 0.0 za prehru
- **W_e** = Na základe R_o (staré Elo) očakávaný výsledok

Ako náhle sa jeden turnaj skončí, nový rating hráčov sa vypočíta na základe starého ratingu a na turnaji dosiahnutom výsledku pomocou uvedeného vzorca.

Ako aj zo vzorca vidno zmenu ELO bodov dáva rozdiel na turnaji dosiahnutého výsledku (**W**) a očakávaného výsledku (**W_e**) upravený o určitý koeficient „**K**“. Toto číslo spravidla spadá medzi 10 a 40. Čím nižšia je jeho hodnota, tým väčšou váhou spadajú staršie turnajové výsledky hráča, teda staré ELO body, a nové ELO body sa len pomalšie budú meniť. Vyšší koeficient väčšou váhou berie do úvahy posledné turnaje a ELO body sa rýchlejšie budú meniť. Tak je logické, a tak je aj v skutočnosti, čím vyššie sú ELO body hráča, tým nižší je koeficient „**K**“ a naopak.

3 Implementácia

Táto časť sa bude zaoberať s implementáciou systému, tu bude popísané aké techniky a programové algoritmy som používal na riešenie problémov. V nasledujúcej kapitole sa budem zaoberať implementáciou jednotlivých funkcií. Zároveň vysvetlím jednotlivé funkcie systému a zmienim sa aj o implementačných prostriedkoch.

3.1 Použité jazyky a technológie

Na vytvorenie webového sídla existuje niekoľko možností ako ASP, Cold Fusion, Perl, Java, Python, PHP, Javascript. Tieto programy patria do komunity open source programov, teda sa netýka len voľne šíriteľného softwaru ale aj otvorenosti zdrojového kódu. Väčšinou hore uvedené skriptovací jazyky generujú HTML (Hyper Text Markup Language) kód stránky. Takto vygenerovanými stránkami výzor je definované pomocou technológie kaskádové štýly (CSS).

V súčasnosti sme svedkami, že web a databázové informačné systémy sa stávajú jedným uceleným systémom, v ktorom web technológia v spojení s prehliadačmi slúži ako strana klientov a www server spolu s databázovým serverom ako zdroj dát a informácií. Na zadávanie dopytov do databázy sa používa jazyk SQL. Aj v mojej úlohe je potreba pristupovať k dátam z databázy, preto aktívne bude využívať SQL dotazy. Databázová časť je implementovaná na databázovom serveru MySQL [8], preto lebo je voľne šíriteľný.

Pri implementácii som použil skriptovací jazyk PHP [8], pretože umožňuje ľahko a rýchlo vytvoriť www stránky, ktoré sa generujú dynamicky, napríklad z databázy či súboru. PHP je mocným nástrojom na tvorbu moderného webu dneška, umožňuje naprogramovať široké portfólio internetových aplikácií - a to ľahko a rýchlo, nakoľko sa veľmi podobá na jazyk C. Ďalším pozitívnym vlastnostiam patrí, že veľmi dobre spolupracuje s databázovými servermi MySQL.

Aby ľahšie ovládala aplikácia na strane klienta som vybral skriptovací programovací jazyk JavaScript. Jazyk je používaný najmä pri tvorbe webových stránok. Výhodou tohto skriptovacieho jazyka je, že všetky udalosti sa odohrávajú na strane klienta, zo strany servera nezaťažuje. Pre správne fungovanie treba mať povolený tento skriptovací jazyk v prehliadači. JavaScript je jazyk jednoduchý, ale zároveň dosť účinný. JavaScript podá webovým stránkam dynamickosť pomocou rôznych animácií a farebných efektov. Pretože v mojej práci je potreba pridať jednotlivým figúrkam dynamickosť, aktívne ho budem využívať.

Pri definovaní vzhľadu stránky som použil kaskádové štýly, známe tiež pod skratkou CSS (z anglického Cascading class Sheets) je jazykom umožňujúcim účinné formátovanie stránok popísaných v jazykoch HTML a XHTML. Hlavným zmyslom je umožniť návrhárom oddeliť vzhľad dokumentu od jeho štruktúry a obsahu.

3.2 Implementované funkcie systému

V tejto kapitole popíšem jednotlivé naimplementované funkcie informačného serveru šach server.

3.2.1 Inštalácia databázy a vytvorenie tabuliek

V prvom rade podľa ERD diagramu sa napísal SQL skript na vytvorenie tabuliek v MySQL databáze. Tento SQL skript robí to, že tabuľka s určitým názvom sa najprv zmaže, potom vytvorí novú s tým istým názvom a tam nadefinované atribúty a vzťahy sa nahrajú do databázy. Tento SQL skript je spustený pomocou PHP skriptu installdb.php, v ktorom pred vytvorením máme možnosť nadefinovať názov vytvorenej databázy, a server, kam bude databáza rozmiestnená nainštalovaná. Na prístup k viacerým serverom je kontrolované loginom a heslom, tieto údaje musíme tiež zadať. Tieto informačné údaje potom uložia do súboru dbnastav.php, kde bude použitý ako prostriedok na pripojenie k databáze. Ak niektoré údaje zmenia napríklad heslo alebo adresu databázy, práve v súbore dbnastav.php budeme potrebovať editovať.

3.2.2 Grafické užívateľské rozhranie

Keď niektorý z užívateľov navštívi stránku, zobrazí sa mu úvodná stránka. Na úvodnej stránke na ľavej strane obrazovky sa nachádza menu systému, okno na prihlásenie a okno o novinkách. V menu v základnom stavu neznámy užívateľ sa môže pozrieť na štatistiku registrovaných hráčov, prípadne môže zaregistrovať (registrácia nových hráčov je popísaná nižšie v kapitole 2.2.3 Registrácia nových užívateľov). Po prihlásení sa objavia ďalšie položky v menu podľa práva užívateľa. V sekcii pozvánky sú evidované prijaté a odoslané pozvánky správy užívateľov. Ak užívateľ má založenú partiu, v sekcii moje partie sú zaradené jeho partie, ktoré ešte nie sú dokončené. Užívateľ s právom naviac ako rozhodca má naviac položku analýza, kde má možnosť vyhľadať už ukončené partie a analyzovať ich.

Pod položkami menu je umiestnené okno na prihlásenie. Toto okno slúži na prihlásenie sa do systému. Na prihlásenie je potrebné zadať užívateľské meno (login) a heslo. Po úspešnom prihlásení sa namiesto prihlasovacieho okna objavia informácie o prihlásenom užívateľovi a jeho údaje a práva. Je to i okno, pomocou ktorého je možné odhlásiť sa zo systému.

Po vybraní niektorého zo založených hrách načíta sa šachovnica s aktuálnym postavením. Šachovnica je vygenerovaná php skriptom, jeho podrobný popis bude nižšie. Pod šachovnicou sú umiestnené ďalšie tlačítka, tieto tlačítka sú Aktualizuj, Pozvánky, Vrátiť stav, Remíza a Prehra. S tlačítkom Aktualizuj prevádzam aktualizáciu stránky, to znamená, že opäť načítam informácie či niekto neurobil nový ťah. Tlačítko Pozvánky zobrazí stránku s pozvánkami a správami. Vrátiť stav slúži na vrátenie posledného ťahu spraveného užívateľom. Tlačítka Remíza a Prehra slúžia na ukončenie partie.

3.2.3 Registrácia nových užívateľov

Neznámy užívateľ má možnosť zaregistrovať na stránku. Formulár registrácie sa objaví po kliknutí na položku registrácia v menu. Pri registrácii je vyžadovaný užívateľ, aby zadal svoje údaje ako meno, priezvisko, emailovú adresu, login (ten slúži ako identifikácia na prístup do systému), musí zvoliť heslo a aj potvrdiť ho. Potvrdiť heslo preto je potrebné, aby som vedel skontrolovať či ho správne zadal a nepreklepol niektorý znak na klávesnici.

V poslednej sekcii registrácie môže zadať svoj aktuálny počet ELO bodov. V prípade, že tu nezadá žiadne číslo, automaticky sa uloží základná hodnota 1350, alebo v prípade keď nadefinuje príliš veľkú hodnotu, tak mu uloží 2650 ako hodnotu ELO bodov.

Ďalej je tam možnosť definovať rôzne osobné nastavenia podľa požiadaviek užívateľa, ako jazyk notácie pri zaznamenaní ťahov. Štýl zobrazovaných figúrok a čas, za ktorý sa má stránka znovu aktualizovať.

Po kliknutí na tlačítko *Ulož informácie* sa odosielajú, v prípade ak užívateľ nezadal niektorú z požadovaných informácií, systém sa mu hneď ohlási. Napríklad je riešené aj testovanie emailových adries pomocou regulárnych výrazov. Ak všetky dáta sú v poriadku tak sa ukladajú do databázy a užívateľ sa automaticky prihlási.

3.2.4 Sessions a prihlásenie

V systéme budem využívať relácie, čiže sessions. Pomocou relácie totiž môžem presne identifikovať užívateľa a jeho pohyb na serveri. Keď sa pristúpime na nejaký server alebo stránku server overí našu identitu a podľa predchádzajúcich našich nastaveniach zobrazí informácie. Session pracuje podľa nasledujúceho princípu. Keď užívateľ pristupuje na stránku dostane nejaké jednoznačné číslo takzvané SID (Session ID). S týmto číslom bude potom užívateľ identifikovaný. V mojom prípade tiež dostanú po prihlásení do systému také identifikačné číslo. Po prihlásení sa do systému načítajú práva z databázy a podľa toho vygenerujú položky menu popísané už v predchádzajúcich kapitolách.

3.2.5 Posielanie správ

Po prihlásení a klikaní na položku pozvánky je možné poselať ostatným registrovaným užívateľom pozvánku na novú hru. Zo zoznamu je možné vyberať hráča, s ktorým chcem zahrať partiu, v lište zobrazia loginy užívateľov. Po vybraní hráča je možné zvoliť farbu figúrky a to buď čiernu alebo bielu, keď vyberiem tretiu možnosť tak na farbe figúrok nezáleží, náhodným generátorom sa vygenerujem číslo a podľa toho je určená farba, ktorou budem hrať. Poslanú správu keď si rozmyslím mám možnosť zrušiť.

Druhá strana, čiže protihráč na stránke pozvánky bude oznámený o prijatí správy a objaví sa mu možnosť buď prijať alebo odmietnuť. V prípade odmietnutia nič zvláštne sa nenastane, len v databáze k správe sa nastaví hodnota odmietnuté a ten užívateľ, ktorý ho poslal, objaví sa mu oznámenie

na stránke, že poslaná pozvánka bola odmietnutá. V prípade prijatia pozvánky sa vytvorí nová hra s novým identifikačným číslom. Do databázy uloží základný stav figúrok a objaví v sekcii moje Partie položka, kde po klikaní je možné zahrať hru.

3.2.6 Vygenerovanie šachovnice

Teraz sa pozrime na možnosti vytvorenia šachovej tabule. Je niekoľko možností ako reprezentovať šachovnicu, ale najlogickejšia cesta je použitie niekoľkých prvkov pola. Prečo zobrazovať šachovnicu ako pole? Jedna z mnoho použiteľných vlastností pola je možnosť prechádzať v cykle a opakovane previesť procesy na jednotlivých prvkoch pola. Existuje viac spôsobov ako toto dosiahnuť v závislosti od indexovania pola. Najjednoduchším je však možnosť prechádzať pole pomocou cyklu *for()*. Tento cyklus pracuje správne, ak indexy pola začínajú nulou a indexy sú priradené sekvenčne. To znamená, že každé políčko na šachovnici dostane jeden index od nuly do 63. Tieto indexy sa dajú i vypočítať, ak poznáme číslo riadku a číslo stĺpca, tieto tiež sú očíslované od nuly do sedmičky. Vzorec na vypočítanie indexov je: číslo riadku * 8 + číslo stĺpca = index pola.

Ja však namiesto indexovania prvkov používam koordináty, čo je číslo riadku a číslo stĺpca, potom z toho odvodím miesto figúrky. Počas renderovania tabule na určité pozície pola je možné rozmiestniť premenné, figúrky. Napríklad pri ťahaní figúrkou je vlastne určené, že z ktorého políčka (napr.: A1) na ktoré políčko (napr.: A3) bolo posunutie hodnoty prvku pola spravené. Ak budeme používať pole na ukladanie takýchto informácií s jednoduchým cyklom môžem prejsť cez tieto prvky pola a môžeme určiť aktuálny stav na šachovnici. Ideálne by bolo používať na reprezentovanie figúrky ako bity, určité hodnoty by mali jednotlivé položky pola, prípadne pri vizualizácii obrázky jednotlivých figúrok, ktorý by boli načítané podľa hodnoty pola.

Šachovnica je tvorená zo 64 políčok zoskupených do štvorca 8 x 8, čo by znamenalo, že potrebujeme 64 prvkové pole, u ktorého sú dva cykly, prvý cyklus na vygenerovanie 8 riadkov a druhý na 8 stĺpcov. Do jednotlivých prvkov pola by boli ukladané dáta v bitoch. Ak pole znesie v sebe nulový bit, znamenalo by, že políčko na šachovnici je prázdne, nestojí tam žiadna figúrka, ak je tam určitá hodnota, znamenalo by, že stojí tam figúrka.

Figúrky podľa farby môžeme rozdeliť na biele a na čierne, kým podľa typu na pešiaka, jazdca, strelca, vežu, dámu a na kráľa. Každý typ figúrky má iné vlastnosti vzhľadom na ich ťahy a možnosti, preto musíme ich nejako odlíšiť od seba. Keď už jednotlivé hodnoty v prvkoch polí zobrazíme ako hodnotu, na každú figúrku vytvoríme bitové vzory podobnej dialky. Každý typ figúrky dostane na nej charakteristické bitové označenie. Farbu figúrky bude označovať najvyšší bit vo vzorku, to znamená, že keď najvyšší bit je nula, tak budeme hovoriť o bielej figúrke, ak je to jednotka, tak budeme brať, že ide o čierne figúrky. Je to výhodné z toho pohľadu, že ak nadefinujeme farebnú masku samými jednotkami, iba v najvyššom bite bude nula, ktorú s bitovým operátorom OR logicky vynásobíme

s bitovým vzorom získame naspäť typ figúrky. Tak je možné jednoducho odstrániť farbu figúrky.

Bytové vzorky budú vypadáť nasledovne:

- Pešiak 0000 0001
- Jazdec 0000 0010
- Strelec 0000 0100
- Veža 0000 1000
- Dáma 0001 0000
- Kráľ 0010 0000
- Maska Farby 0111 1111

Vizualizácia jednotlivých figúrok je riešený obrázkami. Ku každému vzorovému kódu je priradený k jemu zodpovedajúci obrázok v type .GIF a je načítaný na požadované miesto. Ďalej pri vizualizácii šachovnice je možnosť zvoliť perspektívny pohľad zo strany bieleho alebo čierneho hráča. To je riešené tak, že meníme niektoré parametre cyklu *for()* tak, aby riadky šachovnice načítal opačne. Grafické podanie šachovnice zmena čiernych a bielych políčok a okraje sú definované pomocou CSS štýlov.

V mojom prípade je šachovnica vygenerovaná pomocou PHP skriptu i pomocou Javascriptu, je to vlastne riešené tak, že parametre šachovnice sa medzi dvoma jazykmi súčasne vymenia. Tak môžem spojiť prácu programovacích jazykov PHP a Javascript.

3.2.7 Komunikácia s databázou

Komunikácia s databázou je veľmi dôležitá lebo tam sú natrvalo uložené informácie o jednotlivých hrách. Každá hra má identifikačné číslo a k tomuto číslu sú priradené jednotlivé figúrky. Ako som sa už predtým zmienil, figúrka v databáze je jednoznačne určená jej farbou, typom a koordinátami kde je rozmiestnená. Tieto koordináty určia riadok a stĺpec, kde figúrka stojí, keď je uskutočnený nejaký ťah v databáze sa aktualizujú práve tieto hodnoty. V databáze sú pritom evidované i ťahy od začiatku hry až do konca. Nové ťahy s figúrkami sa postupne ukladajú do tabuľky, kedy bol ťah prevedený, z ktorého zdrojového pola na ktoré cieľové pole. Komunikácia s databázou je pomocou PHP skriptu, v súbore *aktualizacedb.php* sú funkcie, ktoré robia zmeny v dátach uložené v databáze, po každom ťahu sa zavolá požadovaná funkcia, ktorá aktualizuje informácie na nové hodnoty.

3.2.8 Označenie figúrky na ťahu

Po načítaní hry na obrazovke sa objaví šachovnica s figúrkami. Podľa toho, ktorý hráč je na ťahu, je možné na figúrky kliknúť. Pri kliknutí na figúrky sú zavolané javascriptové funkcie. Ak vyberiem takúto figúrku, ktorou hrám ja, tak na šachovnici sa objaví okolo neho označenie žltou farbou. Táto udalosť je urobená tak, že namiesto pôvodného obrázku sa načíta nový obrázok už s označením. Keď ešte raz klikneme na tú istú označenú figúrku, označenie zmizne a načíta sa pôvodný obrázok. Takto je vlastne riešená udalosť, keď sme si rozmysleli a chceme urobiť ťah s inou figúrkou.

Teraz opäť zoberieme stav, keď je už určitá figúrka vybraná a označená, keď teraz klikneme na figúrku tej istej farby označenie sa zmení na novú figúrku a pôvodná zmizne, tak je vlastne riešená udalosť, keď namiesto pôvodnej figúrky vyberiem inú.

Keď je už vybraná, označená figúrka a klikneme na prázdne alebo opačným hráčom obsadené pole, tak parametre riadok a stĺpec cieľového pola uložia a začína analýza, že vybraná figúrka na zvolené políčko má možnosť pristúpiť.

3.2.9 Riešenie ťahov

V tejto časti dokumentu budem analyzovať možnosti ťahov jednotlivých figúrok, pretože každá figúrka má iné vlastnosti, rozoberiem na časti podľa typu figúrok. Nasledujúce figúrky budú analyzované: pešiak, jazdec, strelec, veža, dáma a kráľ.

Ešte na začiatku uvediem, že očíslovanie na pôvodnej šachovnici je od 1 do 8 pri stĺpcoch a od A do H pri riadkoch. Ja však pri implementácii z dôvodu jednozdušenia výpočtov namiesto číslovania od 1 do 8 používam číslovanie od 0 do 7 a namiesto písmen som tiež použil číslovanie od 0 do 7. To znamená, že keď v skutočnosti mám na mysli pole A3, v analýze bude myslené na pole s koordinátami 2;0 (druhý riadok, nultý stĺpec). Od teraz teda budem používať hore uvedené pravidlá.

Pretože analýza ťahov je veľmi náročná a zbytočne nevyužíval zdroje php severu všetky tieto analýzy pobežia na strane klienta, takže na implementovanie som použil javascript, ktorý má výhodu práve v tom, že pracuje na klientskej časti.

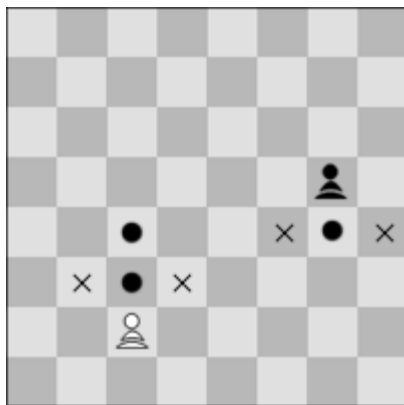
3.2.9.1 Ťahy pešiaka

Keď matematicky chceme vyjadriť možné ťahy pešiaka, tak musíme brať najprv pôvodné koordináty čiže riadok a stĺpec, kde stál pred ťahom a nové koordináty, kam ho chceme posunúť. Najjednoduchšia je, keď ideme len o jedno políčko dopredu, vtedy totiž len porovnáme či číslo stĺpca pôvodného miesta a číslo stĺpca nového miesta zhoduje, tak je vlastne obmedzený, aby sa pohyboval iba po stĺpcoch zvisle. Ťah o jeden riadok dopredu je vlastne odvodený od toho, že rozdiel medzi novým riadkom a starým riadkom musí byť práve jedna a pritom nemôžeme zabudnúť ani na to, že dozadu taký ťah je nezmysel a to potrebujeme ešte kontrolovať. Táto kontrola je odvedená od farby figúrky.

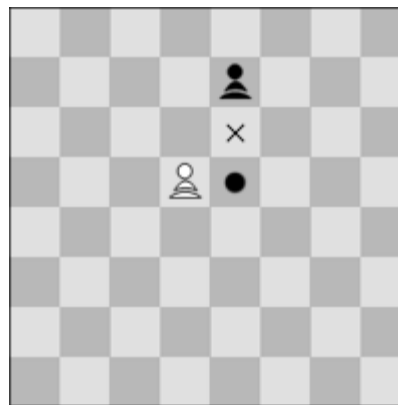
Zo základného postavenia je možné spraviť ťah aj o dve políčka dopredu, to od predchádzajúceho prípadu sa bude líšiť iba v tom, že rozdiel medzi číslom riadkov bude dva. Ďalej by sme nemali zabudnúť na to, že nejako musíme skontrolovať aj to, že aktuálna pozícia bielej a čiernej figúrky je v základnom stave, čiže riadok číslo 1 a v prípade čiernej figúrky je to 6 riadok.

Ak súperov kameň je na uhlopriečke susediaceho pola, môžeme ho vybrať. Pri tomto pokuse pôvodné číslo stĺpca sa zmení na susedné číslo stĺpca pola. To vlastne znamená, že absolútna hodnota nového stĺpca mínus starého stĺpca by mala rovnať jednotke. Číslo riadku podobne ako pri štandardnom ťahu sa povýši v závislosti od farby pešiaka. Potom musíme ešte otestovať,

že na vybranom mieste už stojí nejaký kameň a tento kameň má opačnú farbu. To urobím tak, že sa pozrieme na hodnotu pola. Ak hodnota bude niešť hodnotu nuly, tak ide o prázdne políčko a výber nie je výberom, tak sa nemôže ani uskutočniť. V prípade ak je tam nejaká hodnota, tak skontrolujem či ide o figúrky inej farby. Ak áno, tak ide o branie figúrky.



Obraz č. 10. - Ťahy pešiaka [9]



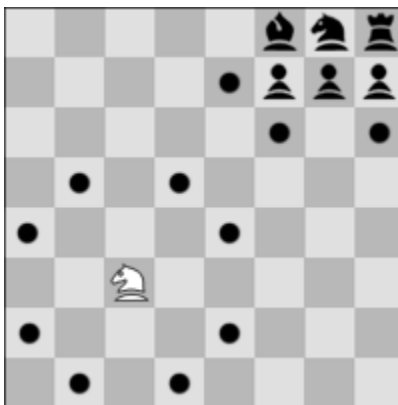
Obraz č. 11. - En passant [9]

Zvláštnym ťahom je branie mimochodom (en passant). Pokiaľ pešiak ohrozuje pole, ktoré súperov pešiak preskočil tým, že z úvodnej pozície postúpil o dve polia, potom ho môže tento pešiak vybrať, akoby súper postúpil len o jedno pole. Tento ťah sa smie previesť len bezprostredne potom, čo súper týmto pešiakom ťahal. Takže to matematicky bude vypadáť nasledovne, pešiak s ktorým chceme vybrať druhého pešiaka musí stáť v prípade bielej figúrky na piatom riadku a v prípade čiernej figúrky na štvrtom riadku. Vedľa neho by mal stáť pešiak opačnej farby (u bielej figúrky tiež na piatom riadku, u čiernej figúrky tiež na štvrtom riadku) a hodnota poľa na šiestom riadku (v prípade bielej) alebo treťom riadku (v prípade čiernej) by mala byť prázdna. A musíme kontrolovať nejako aj to, že výber sa uskutoční hneď po ťahu, čo získame z histórie ťahov. Pod históriou ťahov je myslené postupnosť jednotlivých ťahov ukladaná do databázy.

Pešiak, ktorý postúpil na posledné pole šachovnice sa premení na inú figúrku. Pôsobnosť premeneného kameňa je okamžitá, tzn. môže napríklad dať šach či sa zúčastniť matovania. (Premenený kameň sa typicky berie z hromady vyhodенých kameňov, ale toto nie je obmedzujúca podmienka, jeden hráč môže mať v hre kľudne deväť dám súčasne.) Premena sa odohráva nasledovne, ak v systéme je zistené, že išiel o ťah pešiaka a ten postúpil na siedmu, resp. nultú radu v hornej časti stránky nad šachovnicou objaví dialógový formulár, kde užívateľ má možnosť vybrať dámu, vežu, strelca a jazdca. Po kliknutí na ok vybraná figúrka sa objaví na cieľovom poli. Tento jav v postupnosti ťahov zaznamená tak, že sa objaví v položke premena, že na aký typ figúrky sa premena uskutočnila. Túto udalosť potrebujeme zaznamenať preto, lebo pri vracaní postavenia môže robiť nejasnosti.

3.2.9.2 Ľahy jazdca

Jazdec sa pohybuje skokmi v tvare písmena L (dve polia rovno a jedno nabok, resp. jedno rovne a dve nabok). MÔže preskakovať kamene svoje aj súperove, to znamená, že nie je potreba dávať pozor na testovanie, či pri ťahu stojí nejaká figúrka v ceste. Jediné čo potrebujeme kontrolovať je miesto kam idem, či je prázdne (je tam hodnota nula), alebo stojí tam figúrka opačnej farby a je možné ho vybrať.



Obraz č. 12. - Ľahy jazdca [9]

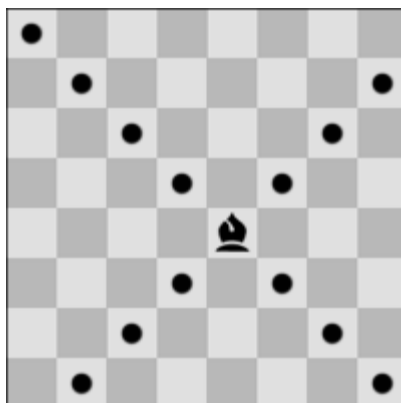
Ak matematicky pokúsime vyjadriť zo základnej pozície, berme koordináty x_1 a y_1 na cieľovú pozíciu s koordinátami x_2 a y_2 môžeme dostať nasledujúcim spôsobom. Najprv dve políčka dopredu to znamená, že postúpim dve riadky takže x_2 mínus x_1 musí mať vzdialenosť práve dva alebo mínus dva, prípadne ak jedno políčko dopredu tento rozdiel je práve jedna alebo mínus jedna. Potom v prvom prípade, keď sme išli už dve dopredu musíme i doľava alebo doprava jedno pole, takže $y_2 - y_1$ musí byť práve jedna alebo mínus jedna a ak berieme druhý prípad, keď sme len išli o jedno pole dopredu, tak tento rozdiel musí byť rovný práve dvojke alebo mínus dvojke. Ak pri výpočte berieme ich absolútne hodnoty krátíme výpočty lebo musíme počítať s ich kladnými hodnotami. Keď hore uvedené pravidlá splnia jazdec robí ťah typu L.

Aby naozaj vedel pristúpiť na požadované pole potrebujeme jeho hodnotu otestovať. Keď hodnota pola nesie nulu ide o prázdne pole, ak je tam definovaná hodnota skontrolujeme farbu hodnoty. Ak táto kontrola vráti farbu opačnú, s akým hráčom ja, išlo by o branie a môžem ťahať s figúrkou uskutočniť.

3.2.9.3 Ľahy strelca

Strelec sa pohybuje po diagonálach. Z daného pola sa môže posunúť pozdĺž jednej z dvoch diagonál, na ktorých stojí. Strelec nesmie pri svojom pohybe preskočiť žiadny kameň, ktorý by stál v ceste. Z pohybu po diagonálach vyplýva, že žiadnym ťahom nemožno zmeniť farbu pola, na ktorom strelec stojí. Takže strelec, ktorý na začiatku hry stojí na čiernom poli, sa bude až do konca partie môcť dostať len na čierne polia. Podľa toho sa tiež strelci daného hráča môžu (napr. pri rozbere partie)

označovať ako čierne či biely. Často sa ale používajú menej mäťúce pojmy strelce čiernopolný a bielopolný.

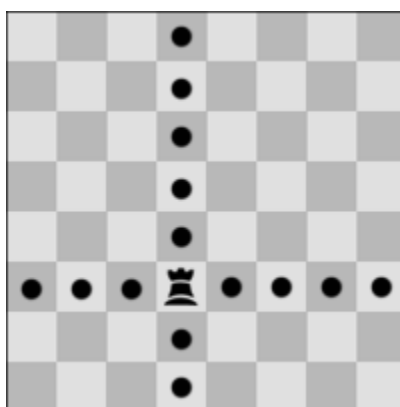


Obraz č. 13. - Ťahy strelca [9]

Keď sa pokúsime o matematické vyjadrenie ťahu strelca ide vlastne o to, že rozdiel pôvodného riadku a cieľového riadku prípadne rozdiel pôvodného stĺpca a cieľového stĺpca sa musí rovnať. Vzorcom to popísané je nasledovne, ak pôvodné pole je $x_1;y_1$ a cieľové pole je $x_2;y_2$, tak $x_2 - x_1 == y_1 - y_2$. Keď predtým popísaný vzorec je platný, ťah môžeme uskutočniť, ale pozor musíme dávať pozor na to, či políčka na diagonále medzi cieľovým a pôvodným sú prázdne, nie je obsadené nejakou figúrkou. Pri tomto rozbere rozdelíme možnosti na dva prípady, keď cieľový riadok y_2 je väčší ako y_1 , v tomto prípade zase rozdelíme výpočet na ľavú a pravú stranu. V prípade keď ľavá strana x_2 je väčšia ako x_1 , cyklusom prejdeme cez ľavú hornú diagonálu a každé pole otestujeme, či je prázdne (pole má hodnotu nula), keď nebude pole prázdne, tak ťah nebude dovolený. Tento test prázdnych polí musíme na všetky štyri možnosti podľa pôvodného a cieľového miesta určiť.

3.2.9.4 Ťahy veže

Veža sa pohybuje po riadkoch a stĺpcoch. Z daného pola sa môže presunúť na také pole, ktoré je buď v rovnakom riadku, alebo v rovnakom stĺpci. Nesmie však preskočiť žiadny kameň, ktorý by stál v ceste.

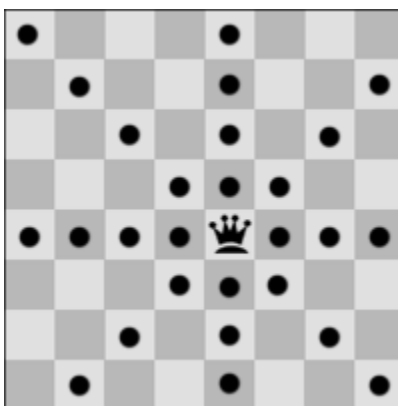


Obraz č. 14. - Ťahy veže [9]

Takže, keď sa pohybuje po riadkoch musíme vlastne porovnať, že cieľová koordináta riadku y_2 (zase berieme, že $x_1;y_1$ je pôvodné a $x_2;y_2$ je cieľové pole) rovná sa pôvodnej koordináte y_1 , ak tento vzťah nesplní, určite nie je pohyb po riadkoch. Zase, keď sa pohybuje po stĺpcoch, berieme číslo pôvodného stĺpca x_1 sa musí rovnať číslu stĺpca cieľového stĺpca x_2 . Ak predchádzajúce pravidlá sa uplatnia musíme ešte jedným cyklom prejsť cez políčka až k cieľovým, či nestojí v ceste nejaká figúrka a blokuje jeho pohyb. Ak tento cyklus nájde pole kde je hodnota odlišná od nuly znamená, že pohyb veže je blokový a nemôže na cieľové pole postúpiť.

3.2.9.5 Ťahy dámy

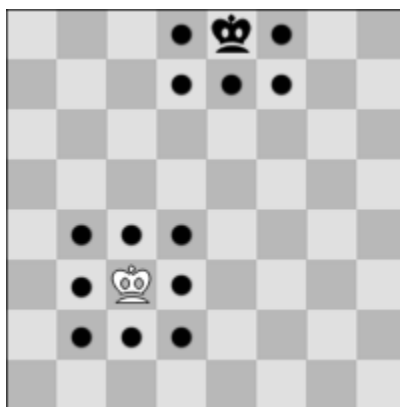
Dáma sa pohybuje po stĺpcoch (riadkoch) alebo diagonálach. To jej dáva vysokú mobilitu, pre ktorú je dáma najcennejším kameňom. Pri svojom pohybu nesmie preskočiť žiadny kameň, ktorý by stál v jej ceste. V sebe nosí vlastne vlastnosti veže a strelca. Preto pri implementácii môžeme ich algoritmy využívať. Takže, keď analyzujem jeho možnosti na jeho ťahy zavolám už len predtým definované algoritmy. Ak jeden z nich vráti *povoleno* je jeho ťah povolený.



Obraz č. 15. - Ťahy dámy [9]

3.2.9.6 Ťahy kráľa

Kráľ sa žiadnym svojím ťahom nesmie dostať na ohrozené pole (na pole, na ktoré by sa v nasledujúcom ťahu mohol presunúť súperov kameň a tým kráľa brať). Tento prípad je podrobnejšie rozobratý v kapitole testovanie šachov. Kráľ sa pohybuje o jedno pole v ľubovoľnom smere, tzn. na ľubovoľné zo susediacich polí (vrátane polí susediacich po diagonále). To matematicky vyjadrené znamená, že keď opäť berieme pôvodné koordináty ako $x_1;y_1$ a cieľové koordináty ako $x_2;y_2$. Rozdiel medzi $x_2 - x_1$ a $y_2 - y_1$ musí byť buď nula, buď jedna alebo buď mínus jedna (len koordináty nula nie sú povolené lebo by nešlo a ťah). Výpočty môžeme aj v tomto prípade zjednodušiť tým, že pri výpočtoch budeme počítat' s absolútnymi hodnotami.



Obraz č. 16. - Ťahy kráľa [9]

Druhým spôsobom ťahu kráľa je tzv. rošáda, pri ktorej sa pohybuje súčasne kráľom a vežou rovnakej farby v situácii, keď oba stoja na svojich pôvodných pozíciách a medzi nimi je voľné miesto. Kráľ sa týmto ťahom posunie o dve polia smerom k veži a veža ho „preskočí“ na pole, ktoré kráľ práve prešiel. Celý tento postup sa považuje za jediný ťah kráľom. Rošádu nie je možné previesť v prípade, že kráľom alebo vežou, ktorá sa má rošádu zúčastniť, už bolo v tejto hre ťahané (ani pokiaľ sa vrátili na pôvodné miesta). Pokiaľ ale v danej partii tiahla len jedna z veží a kráľ sa doposiaľ nepohol, je možné (za splnenia ostatných podmienok) naďalej previesť rošádu za účasti druhej veže. Rošádou však nie je možné uniknúť zo šachu (nie je možné ju previesť, ak je kráľ ohrozovaný), ani kráľ nesmie prejsť či skončiť na ohrozenom poli. Naproti tomu, ak je napadnutá veža, rošáde to nebráni.

Pokúsme hore popísané prípady teraz rozobrať a analyzovať matematickým prístupom. Ako prvá podmienka musí platiť, že riadok kráľa pôvodného miesta musí byť to isté ako riadok cieľového poľa. Ďalšou podmienkou je, že kráľ aj jeden z veží musia stáť na presne zadaných miestach a majú konkrétne koordináty. V prípade bielych veží sú to koordináty 0;0 a 0;7 v prípade čiernych veží sú to koordináty 7;0 a 7;7. Keď je rošáda, kráľ sa vlastne pohybuje o dve políčka, čo znamená, že pôvodne koordináty stĺpca mínus koordináty cieľového pola bude buď dva alebo mínus dva. Pri testovaní musíme nejako otestovať, že kráľom alebo vežou sme sa ešte neposunuli. Tieto informácie môžeme napríklad získať z partiára z uložených ťahov v databáze. Ak sa v postupnosti ťahov objaví ako posunutá figúrka kráľ alebo jeden z veží, rošáda už nebude možná.

Nemôžeme zabudnúť ani na to, že počas posunutia kráľa a veže políčka medzi týmito figúrkami musia byť prázdne. Ak tento vzťah sa nespĺňa, aspoň jeden z políčok bude mať inú hodnotu ako nula, znamenalo by, že tam stojí nejaká figúrka a tak sa už rošáda nemôže uskutočniť.

Pri rošáde sa pohybujeme s kráľom čo spôsobí, že musíme dávať pozor na šachy. Ak kráľ má šach, hneď môže byť príčina nemožnosti uskutočniť tento ťah. Na tieto šachy musíme dávať pozor aj v prípade posunutia kráľa. Cez pole, po ktorých sa kráľ pohybuje, musíme otestovať na šachy a tiež aj cieľové pole je potrebné na tieto šachy otestovať. Testovaním šachu zaoberá nasledujúca kapitola.

3.2.10 Testovanie ohrozenia kráľa

Pri testovaní ohrozenia kráľa musíme rozlíšiť dve situácie, prvý z nich je štandardný jednoduchý šach, keď kráľ je iba jednoducho ohrozený. Druhá možnosť je, keď kráľ nemá žiadnu možnosť na vyhnutie zo šachu, tomuto stavu sa hovorí, že dostal mat. Analýza takého stavu je veľmi náročný ohľadom na výpočty i logickosti.

3.2.10.1 Testovanie na šachy

Matematický postup na testovanie šachov by vypadal nasledovne: berme, že kráľ stojí na pozícií $x_0; y_0$, bod od toho kde môžeme rozmiestniť jazdca bude $(x_0 + 2 ; y_0 + 1)$, $(x_0 + 2 ; y_0 - 1)$, $(x_0 + 1 ; y_0 - 2)$, $(x_0 + 1 ; y_0 - 2)$, $(x_0 - 2 ; y_0 + 1)$, $(x_0 - 2 ; y_0 - 1)$, $(x_0 - 1 ; y_0 + 2)$, $(x_0 - 1 ; y_0 - 2)$. Zaujímavý prípad nastane, ak kráľa umiestnime na kraj šachovnice. Totiž v takom prípade niektoré z hore uvedených koordinát nemá zmysel. V takom prípade tieto nemožné situácie musíme obmedziť. To urobíme tak, že v takomto prípade tieto odpadajúce polia nepočítame, obmedzíme výpočet na možné polia. Tak vlastne všetky možnosti dať šach s jazdcom ukryli.

V prípade ostatných figúrok postup vyzerá nasledovne, je tam cyklus ktorý pobeží osem možných smerov okolo kráľa. Smery som očísloval nasledovne:

0	1	2
7	-- Kráľ --	3
6	5	4

Vytvoril som objekt, do ktorého ak nájde figúrku, postupne uložíme informácie o tejto figúrke a diaľku od kráľa, kde sa našla. Z tohto naplneného pola postupne začnem analyzovať figúrky. Najprv sa pozriem či sa našla nejaká figúrka a na to či farba nájdenej figúrky je opačný akou hrá hráč. Potom som analýzu šachov rozdelil na dve väčšie podskupiny. Prvá skupina kontroluje iba stĺpce a riadky, v tomto prípade môžu ohroziť dáma, veža a kráľ. Kráľ môže ohroziť iba v prípade, keď diaľka je jedna. Druhá skupina po diagonálach prevedie kontrolu, tu ohrozenie na kráľa môže spôsobiť dáma, strelec a keď diaľka je jedno pole tak aj kráľ a pešiak. Čierny pešiak môže ohroziť iba zo smeroch 0 a 2, biely pešiak len zo smeroch 6 a 4. Napríklad keď beriem bieleho pešiaka a čierneho kráľa umiestnim na pozíciu $x_0; y_0$ tak pozície pešiaka, z ktorého môžu ohroziť kráľa budú na koordinátach $(x_0 + 1 ; y_0 - 1)$ alebo $(x_0 - 1 ; y_0 - 1)$. Keď v poli sú uložené takéto figúrky, ktoré zodpovedajú hore uvedeným pravidlám, tak ide o šach na kráľa.

3.2.10.2 Testovanie na maty

Ak kráľ dostane šach, hneď po tom je aj otestované, či nie je náhodou matová situácia a kráľ už nemá možnosť kam uniknúť. Postup bude niečo podobné ako pri šachoch. Najprv aj v tomto prípade som zistil pozíciu kráľa, kde sa nachádza. Tieto koordináty som uložil a odstránil som kráľa

zo šachovnice. Potom rozmiestním kráľa na novú pozíciu. Preverím, či na okolité polia je možné pristúpiť a ak áno, nebude tam šach na kráľa. Výpočet koordinátov okolitých polí je nasledovný:

$$\begin{array}{ccc} 1;-1 & 1;0 & 1;1 \\ 0;1 & \text{KRÁĽ} & 0;1 \\ -1;1 & 1;0 & -1;1 \end{array}$$

Napríklad keď kráľ by stál na poli s koordinátami 2;4 a postúpi na vedľajšie pole s označením 1;-1, tak jeho nové koordináty budú 3;3 (2+1;4-1). Aj tu musíme dávať pozor na okraji šachovnice, je tam funkcia, ktorá sa pozrie na to, či je platným polom na šachovnici, ktorú sa práve pokúšame analyzovať. Pri tomto testovaní môžem aktívne využívať funkcie už nadefinované pri analýze šachov.

Ak z hore uvedených polí ani na jeden nemôže kráľ pristúpiť, pozrieme sa na to, či figúrku by sa nedalo blokovat'. Podobne ako pri šachoch na možné smery zistím figúrku, ktorý dal šach a má prístup na pole kráľa (také figúrky môžu byť maximálne dve), preto nemôžem cyklus hneď ukončiť, musím pokračovať v hľadaní, aj keď som už našiel prvú. Koordináty atakujúcej figúrky uloží do pola. V poli je vlastne ukladaný jeho riadok a stĺpec odkiaľ ohrozuje kráľa. V prípade jazdcov šach nemôžem blokovat', tak to preskočím. U ostatných figúrok od koordinátu kráľa na koordináty ohrozujúcej figúrky pošlem lúč a každé pole, cez ktoré tento lúč prejde, skontrolujem či nejaká figúrka má možnosť pristúpiť, aj v tomto prípade môžem využiť funkcie definované pri analýze šachov. Ak je takáto figúrka, ktorú môžem rozmiestniť na pole, tak ide o blokovanie matu.

Poslednou možnosťou je branie ohrozujúcej figúrky zo šachovnice. Je tam skontrolované, či figúrka ktorá dala šach je v bezpečí na uloženej pozícii v poli ohrozujúcich figúrok. Ak môžeme vybrať túto figúrku pravdepodobne nie je mat.

3.2.11 Vrátanie postavenia

V niektorých situáciách sa zide možnosť, keď niektorý ťah v partii môžeme vrátiť, aj ja v mojom projekte som implementoval takú možnosť, táto funkcia robí to, že ťahy v partii zoradí podľa času a posledný ťah v zápase zmaže zo systému a tak uloží predchádzajúce postavenie. Pred tým, než sa tento jav prevedie, druhý hráč by dostal správu o tomto jave a môže súhlasiť alebo odmietnuť túto požiadavku.

3.2.12 Zdielanie počítača

Môže nastať i taká situácia, keď dvaja hráči chcú zahrať partiu na jednom počítači, v takom prípade môžu zahrať partiu striedavo, môžu sa vymieňať pred počítačom. Keď zvolia túto možnosť hry, systém požiada hráča o jeho heslo, po zadaní hesla sa hra načíta, ale nie je obmedzené ťahanie figúrok na jednu farbu. Hráči kludne môžu zahrať so svojimi i súperovi figúrkami.

3.2.13 Vedenie notácie

Vedenie notácie je riešené tak, že na kraji šachovnice po každom spravenom ťahu vygeneruje nový prvok v jednej tabuľke. Tieto prvky sú generované na základe, s ktorou figúrkou bol ťah urobený a aké má jazykové nastavenia užívateľ. Údaje k vygenerovaniu sú získané z databázy, kde sú ťahy ukladané.

Funkcia notácie robí to, že v jednom cykle prejde cez zatiaľ urobené ťahy (je počítané koľko ťahov bolo zatiaľ urobených) a výpis sa strieda medzi ťahmi čiernym a bielym hráčom. Označenie figúrky je odvodené na základe jeho mena uložený do databázy, je tam prepínač, ktorý podľa zistených dát prehodí meno na potrebné označenie. Po zistení figúrky musíme zistiť i názov, adresu pola, z ktorého kam sa posunul. Toto je riešené tak, že už koordináty jednotlivých ťahov v databáze je definovaný v hodnote riadku a stĺpca, len toto musíme prekonvertovať na požadované číslo a písmeno.

Počas výpisu sa môžu objaviť špeciálne ťahy, také je branie keď namiesto pomlčky je označenie "x". Branie je detektované na základe toho, že v databáze hodnota branie je nastavená. Ak táto hodnota nesie v sebe premenné, tak ide o branie.

Rošáda tiež má špeciálnu notáciu, ten je detektovaný na základe, že bol posunutý kráľ a namiesto jedného políčka posunul dve políčka (na základe koordinátov je to vypočítané). Ak zistím takýto jav, tak ide o malú alebo veľkú rošádu a v partiáry to poznamenám ako 0-0 alebo 0-0-0.

3.2.14 Počítanie ELO bodov

Po každej zakončenej hre sa počet ELO bodov u hráčov znovu prepočíta a do databázy zapíše nový počet získaných bodov. Starú hodnotu ELO obidvoch hráčov zistím z databázy, načítam ho. Výsledok hry jednoducho vieme určiť podľa zakončenia partie, bude to buď 1 (jedna), 0.5 (polovina) alebo 0 (nula). Podľa zistených hodnôt vypočítam rozdiel v ELO bodoch ($\$diff$), toto číslo potrebujeme na zistenie predpokladaného výsledku. Tento očakávaný výsledok je potom u mňa vypočítaný nasledovne. Potrebujem absolútnu hodnotu rozdielu ELO bodov ($\$absdiff$). Potom ak tento rozdiel je väčší ako 735 vrátim ako očakávaný výsledok jednotku, keď je menší ako -735 tak nulu a v ostatných prípadoch použil som nasledujúcu rovnicu:

$$0.5 + (1.4217 * 0.001 * \$diff) - (2.4336 * 0.0000001 * \$diff * \$absdiff) - (2.5140 * 0.000000001 * \$diff * \$absdiff * \$absdiff) + (1.9910 * 0.0000000000001 * \$diff * \$absdiff * \$absdiff * \$absdiff)$$

Táto rovnica vráti očakávaný výsledok na základe rozdielu ELO bodov definované v tabuľke č. 2.

V mojom prípade K-faktor je určený podľa počtu partií a hodnoty ELO bodov. Počet partií je zistený na základe počtov výhry, prehry a remízy, tieto sú počítané. Ak hráč hral menej ako 30 partií a má ELO menej ako 2400 má K-faktor hodnotu 25, keď odohral už 30 partií ale ELO body ešte má

pod hranicou 2400 tak K-faktor je definovaný hodnotou 15 a keď už odohral 30 partií a prekročil hranicu a má viac ako 2400 ELO bodov tak hodnota K-faktoru je už len 10.

Keď všetky tie informácie sú už vypočítané, podľa vzorca $R_n = R_o + K (W - W_e)$ (je definované v návrhu) vypočítame nové ELO a zapíšeme to do databázy namiesto starej hodnoty. Zmeny sa hneď prejavajú v štatistike hráčov.

4 Záver

Témou mojej bakalárskej práce je šachový server. Pre túto tému som sa rozhodol preto, lebo ma zaujíma šach, ja sám som registrovaným šachistom a rád hrám túto hru. Okrem toho veľakrát navštevujem také www stránky, ktoré na internete poskytujú on-line hru šachu. V mojej bakalárskej práci som sa pokúsil vytvoriť podobný funkčný systém šachového servera, na ktorý podľa mojich podmienok a požiadaviek by som mohol pozvať mojich šachových priateľov.

Týmto návrhom a následnou implementáciou boli splnené všetky požiadavky kladené na tento systém a výsledné riešenie je schopné poskytnúť hranie šachu na reálnej počítačovej sieti. Dáva dobrý základ na postupný rozvoj a rozšírenie o ďalšie funkcie. Teoretické postupy a analýzy prevedené v rámci tejto práce sú implementačne využiteľné aj v iných oblastiach dnes dostupných vývojových technológií.

Táto bakalárska práca ma veľmi zaujala a poskytla veľký pokrok v oblasti vytvárania informačných systémov. Okrem samotných programátorských znalostí poskytla veľa zaujímavých informácií aj z iných oblastí ľudskej činnosti. Dala mi základné vedomosti a kroky do náročnejších aplikácií, medzi ktoré proces vývoja softwarových produktov nepochybne patrí.

Literatura

- [1] Hruška, T.: *Informačné systémy*. Študijní opora. FIT VUT Brno, 2006.
- [2] Křena, B., Kočí, R.: *Úvod do softwarového inženýrství*. Študijní opora. FIT VUT Brno, 2006.
- [3] Křena, B., Kočí, R.: *Agilní metodologie*. Prednáška. FIT VUT Brno, 2007.
- [4] Zendulka, J., Bartík, V., Květoňová, Š.: *Analýza a návrh informačních systémů*. Študijní opora. FIT VUT Brno, 2006.
- [5] Sunek, B.: *Pravidla šachu FIDE* [online]. 2005 [cit. 2008-04-05]. Dostupný z WWW: <http://www.sweb.cz/chesskarvina/soubory/pravidla_fide.pdf>.
- [6] Lhoták, J.: *Pracujeme se session v PHP* [online]. 2001 , 1. 3. 2001 [cit. 2008-03-31]. Dostupný z WWW: <<http://interval.cz/clanky/pracujeme-se-session-v-php/>>.
- [7] Hanák, D.: *Kempelen Farkas* [online]. 2004 [cit. 2004-11-30]. Dostupný z WWW: <http://hu.wikipedia.org/wiki/Kempelen_Farkas>.
- [8] Ullman, L.: *PHP a MySQL*. Brno: Computer Press, 2004, 536 s. ISBN 8025100634
- [9] FIDE.: *Pravidla šachu FIDE* [online]. 2001 [cit. 2008-04-30]. Dostupný z WWW: <<http://www.chess.cz/web/informace/legislativa/fide/fide.html>>.
- [10] Weeks, M.: *Calculate Your Chess Rating* [online]. 2008 [cit. 2008-05-05]. Dostupný z WWW: <http://chess.about.com/od/organizations/ss/aa04104_3.htm>.
- [11] U. S. Chess Federation.: *Chess ratings* [online]. 2006 [cit. 2008-05-06]. Dostupný z WWW: <http://www.sizes.com/sports/chess_ratings.htm>.
- [12] *Sport betting with statistical methodes* [online]. 2008 [cit. 2008-05-06]. Dostupný z WWW: <<http://www.sport-prognose.de/ELO.html>>.
- [13] Wikipedia.: *Elo rating system* [online]. 2008 [cit. 2008-05-06]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Elo_rating>.
- [14] Sikora, P.: *Pravidla šachu* [online]. 2008 [cit. 2008-05-07]. Dostupný z WWW: <http://www.so-tj-sokol-mosty.wz.cz/Oddil/Pravidla/pravidla1_1.html>.

Seznam příloh

Příloha 1. CD/DVD