

SWAPS

Moduly systému

CHECK-MATES

Vedúci tímu: Ing. Marián Šimko, PhD.

Členovia tímu: Bc. Igor Jurík, Bc. Matej Liskovec, Bc. Jaroslav Loebel, Bc. Martin Tamajka,
Bc. Peter Truchan, Bc. Ľubomír Vnenk

Akad. rok: 2014/2015

Obsah

Modul 1 - Párovací engine	2
Analýza.....	2
Návrh.....	3
Implementácia	3
Testovanie	3
Modul 2 - Vytváranie turnajov a propozícií, vyhľadanie turnaja a prihlásenie	4
Analýza.....	4
Návrh.....	5
Implementácia	7
Testovanie	8
Modul 3 - Manažment turnajov.....	9
Analýza.....	9
Návrh.....	9
Implementácia	10
Testovanie	11
Modul 4 - Profil hráča	12
Analýza.....	12
Návrh.....	12
Implementácia	13
Testovanie	13
Modul 5 - Automatický partiár.....	14
Analýza.....	14
Návrh.....	14
Implementácia	15
Testovanie	18

Modul 1 - Párovací engine

Analýza

Párovanie je základom celej aplikácie. Bez jeho implementácie by nebolo možné umožniť organizátorovi riadiť turnaje a to najmä s ohľadom na komplexnosť párovacieho algoritmu. Párovací algoritmus, aby ním bolo možné párovať aj hráčov na oficiálnych turnajoch v rámci FIDE, musí byť v súlade s jeho oficiálnou definíciou. V našom prípade sme sa rozhodli implementovať systém používaný na drvivej väčšine turnajov, tzv. Dutch systém (švajčiarsky systém). Jeho definícia je uvedená na adrese

<http://www.fide.com/fide/handbook.html?id=167&view=article>

V rámci tohto analytického používateľského príbehu bolo našim cieľom zanalyzovať algoritmus, a taktiež implementovať nástroj na získavanie informácií o hráčoch, ktoré budú použité v priebehu procesu párovania (jedná sa o národné ELO jednotlivých hráčov. FIDE ELO je totiž možné získať v prehľadnej forme priamo zo stránky FIDE).

V rámci tohto US bol analyzovaný švajčiarsky párovací systém definovaný štandardom FIDE. Výstupom bol diagram opisujúci tento systém. Naším cieľom je dosiahnuť implementáciu spĺňajúcu štandardy FIDE, nakoľko chceme dosiahnuť úspešné odporúčenie našej implementácie zo strany FIDE.

Boli zanalyzované stavy hráča a skupiny hráčov s rovnakým skóre dosiahnutého počas turnaja. Tieto charakteristiky majú výrazný vplyv na ďalší priebeh párovania. Analýza je zapísaná v prehľadnom dokumente na dátovom úložisku tímu. Vysvetlenie a základné pojmy je možné nájsť na adrese

<http://www.fide.com/fide/handbook.html?id=167&view=article>.

Na základe komunikácie s tajomníkom FIDE pre párovanie sme ďalej analyzovali aspekty Dutch systému ako takého. Najväčšou pomôckou bol pre nás dokument obsahujúci vysvetlenie častí Dutch systému na ukážkovom turnaji, ktorý nám poskytlo FIDE. Napriek tomu, že tento dokument bol mimoriadne nápomocný, ani v ňom sa nenachádzali opísané všetky možnosti, ktoré môžu nastať pri párovaní.

Návrh

V rámci TP boli navrhnuté dve verzie párovacieho enginu. Prvá z nich bola už naimplementovaná a nasadená v reálnom turnaji. Pri tejto verzii sa však ukázalo, že je slabo štruktúrovaná a pomerne ťažko otestovateľná. Z tohto dôvodu prišlo k návrhu druhej verzie, ktorá sa významne opiera o návrhové vzory. Diagram tried druhej verzie je uvedený v technickej dokumentácii.

Implementácia

Párovací engine je v rámci SWIPS implementovaný ako samostatná knižnica v jazyku Ruby. Pri implementácii sme využili postupy softvérového inžinierstva a skúsenosti z praxe tak, aby bola dosiahnutá efektivita a robustnosť, nakoľko v prípade veľkých turnajov s viacerými hráčmi je dôležité mať na pamäti aj obmedzenia hardvéru. Pri nesprávnej implementácii by sa mohlo stať, že hlavná pamäť by nemusela postačovať.

Okrem toho je nutné využívať inteligentné optimalizácie v situáciách, kedy nie je možné vyskúšať všetky možnosti tak, ako to definuje švajčiarsky systém. Za týmto účelom sme implementovali rôzne pomocné metódy, ktorých úlohou je najmä určiť, koľko možností je nutné preskočiť tak, aby existovala šanca, že nájdené párovanie bude validné.

Momentálne sa pracuje na implementácii druhej verzie párovacieho enginu podľa návrhu vytvoreného v tomto semestri.

Testovanie

Párovací engine je testovaný automatizovanými testami, konkrétne rspec. Je testovaný ako na úrovni jednotlivých metód, tak na úrovni integračnej. Ako sa však ukázalo, automatizované testy nedokážu zaručiť bezchybnosť riešenia. Počas testovania párovacieho enginu v rámci reálneho turnaja došlo k porušeniu biznis logiky, pričom implementačne šlo o triviálnu záležitosť. Táto skúsenosť nás však presvedčila o nutnosti testovania v reálnych situáciách.

Modul 2 - Vytváranie turnajov a propozícií, vyhľadanie turnaja a prihlásenie

Analýza

Základným krokom ku konaniu turnaja je jeho vytvorenie. Patria sem procesy ako vytvorenie propozícií k turnaju, uverejnenie propozícií a nastavenie turnaja v systéme, ktorým bude riadený. Chronologicky je vytvorenie turnaja prvý proces, ktorý trvá až po spustenie turnaja. Prvá vec, ktorú musí organizátor turnaja spraviť, aby oznámil turnaj, je vytvorenie propozícií turnaja. Propozície sú dokumenty poskytujúce základné informácie o turnaji. Analýzou niekoľko desiatok aktuálnych propozícií sme získali prehľad o súčasnom stave. Identifikovali sme nasledovné problémy súčasného stavu:

- *dlhý proces tvorby,*
- *žiadna jednotná forma,*
- *každý poskytuje informácie v inom poradí.*

Výsledkom je celková nekonzistentnosť vytvorených propozícií a často zmätení hráči, ktorí sa v nich zle orientujú. Zistili sme informácie, ktoré sa musia v propozíciách nachádzať a logicky sme ich usporiadali podľa dôležitosti. Rovnako sme preskúmali možnosti samotného vytvárania a uverejňovania propozícií. Organizátori si vytvárajú každý vlastné propozície (väčšinou pomocou nástroja MS Word). Sú prípady, keď pri usporiadaní väčších turnajov vznikajú jednorazové internetové stránky poskytujúce propozície. V súčasnosti sa na Slovensku propozície uverejňujú primárne na stránke chess.sk. Vytvorenie turnaja sa v súčasnosti dohráva hlavne v systéme Swiss-Manager. Po vyskúšaní systému a jeho pozorovaní v praxi, sme systém podrobne rozobrali. Identifikovali sme nutné vlastnosti turnaja, ktoré musia byť explicitne zadané organizátorom. Keďže našim cieľom je poskytnúť používateľovi lepší používateľský zážitok, než poskytujú súčasné riešenia, je potrebné nastavenia turnaja roztriediť do logických kategórií. Výsledkom analýzy bolo rozhodnutie vytvoriť štruktúrované propozície obsahujúce dôležité informácie. Následne využiť údaje z nich a tak používateľovi zjednodušiť vytvorenie základných nastavení turnaja. V turnaji obsiahnuť potrebné nastavenia, ktoré budú logicky rozdelené, aby bol produkt jednoduchý na používanie.

Z pohľadu hráča ktorý sa chce prihlásiť na turnaj je stav následovný: ak hľadá konkrétny turnaj, musí si manuálne prezrieť všetky propozície rad do radu, nakoľko nad propozíciami nie je v súčasnosti spravené rozumné vyhľadávanie. Registrácia na turnaj prebieha väčšinou prostredníctvom mailu, prípadne telefonátu. Záplava mailov od hráčov je zbytočná príťaž aj pre samotných organizátorov, nehovoriac o zdĺhavosti mailovej komunikácie (v niektorých prípadoch).

Kostrbatý model registrácie na turnaj sa negatívne prejavuje aj na samotnom turnaji. Prítomnosť hráčov pravidelne zdržuje začiatok turnaja, nakoľko organizátor musí prichádzajúcich hráčov ručne nahráť do softvéru na párovanie. Nahodenie hráčov do systému pred turnajom sa nevypláca, nakoľko sa bežne stáva, že hráč registrovaný mailom nepríde, prípadne príde niekto kto sa mailom neregistroval vôbec.

Všetky výsledky analýz sú uložené v materiáloch na tímovom úložisku.

Návrh

V prvej verzii modulu sme navrhli osobitné vytváranie klasických propozícií a osobitné vytváranie turnaja evidovaného v našom systéme. Tento návrh bol braný ako kompromis medzi klasickým oznamovaním turnaja cez papierové propozície a elektronickou verziou turnaja a vychádzal z predpokladu, že časť organizátorov si bude chcieť v našom systéme iba vytvoriť papierové propozície.

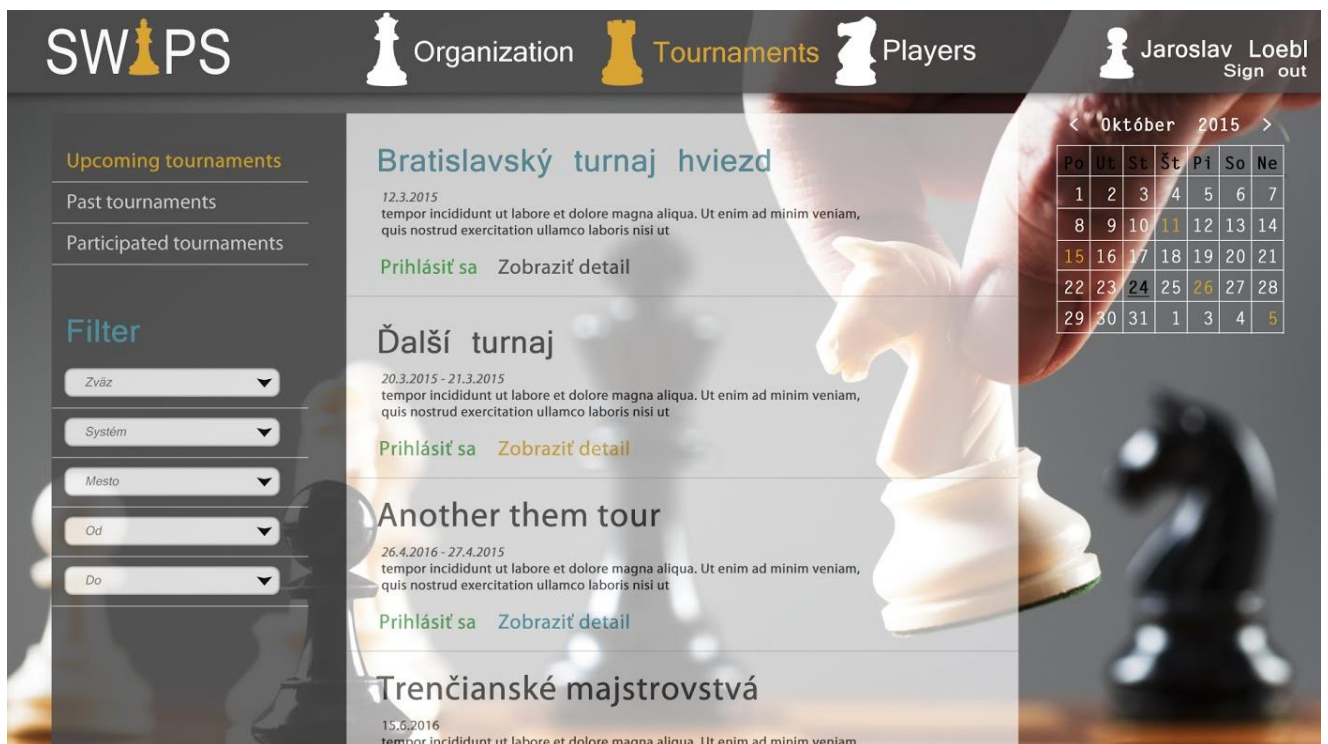
Neskôr sme tento prístup prehodnotili a rozhodli sme proces vytvárania zjednotiť. Navrhli sme jednotný formulár, v ktorom sa dajú vyplniť všetky informácie, ktoré sa do propozícií dávajú, pričom povinných polí sme sa snažili zachovať minimum. Informácie, ktoré ma organizátor možnosť zadať sme zoskupili do logicky súvisiacich celkov, aby bol zabezpečený plynulý prechod formulárom. Navigácia na ľavo od formulára slúži na lepšiu orientáciu vo formulári, ako aj na rýchly prechod do inej časti v prípade potreby.

Po vytvorení turnaja v našom systéme je následne možné si exportovať propozície v .pdf formáte. High fidelity návrh je možné vidieť na obrázku 1.



Obrázok 1 Formulár vytvárania turnaja

Za účelom vyhľadávania a registrácie na turnaj sme navrhli obrazovku so zoznamom všetkých nadchádzajúcich turnajov, ktoré je možné filtrovať na základe vopred daných kritérií, ako vidíme na obrázku 2. Turnaje sme pritom rozdelili do troch základných kategórií a to nadchádzajúce turnaje, minulé turnaje a turnaje na ktorých sa prihlásený hráč zúčastnil.



Obrázok 2 Zoznam turnajov

Chceli sme hráčom umožniť čo najjednoduchšiu registráciu na turnaj, preto sme už do prehľadu turnajov pridali okrem tlačidla na detail turnaja aj možnosť rovno sa na turnaj registrovať. V prípade, že má turnaj viacero kategórií, do ktorých je možné sa registrovať, tak sa zobrazí popup, v ktorom si hráč vyberie danú kategóriu.

Implementácia

Dátový model bolo nutné zjednodušiť. Zjednodušenie modelu bolo dosiahnuté odstránením prvkov generickejšieho vzhl'adom na druh hry z dátového modelu (napr. odstránenie hierarchie tournament - single_tournament - chess_single_tournament a jej nahradenie hierarchiou tournament - single_tournament). Bolo nutné rozšíriť dátový model o tabuľky modelujúce propozície. Tento model obsahuje množstvo redundantných informácií. Cieľom tejto redundancie je umožniť vytvorenie propozícií, bez nutnosti vytvoriť turnaj v systéme. Tiež chceme umožniť vytvorenie propozícií obsahujúcich informácie nekorešpondujúce s požiadavkami FIDE (ak ide napr. o lokálny exhibičný turnaj), na základe ktorých by nebolo možné automaticky vytvoriť v systéme turnaj. Na základe kvalitného dátového modelu sa podarilo jednoducho vytvoriť modely reprezentujúce súčasný stav viacerých propozícií. Každý model mal pre jednoduchú revitalizáciu databázy vytvorené vymyslené dáta, z ktorých sa dá kedykoľvek vytvoriť plnohodnotná databáza. Vytvorili sa prvotné formuláre, ktoré odzrkadľujú polia v návrhu, pričom sa dbalo na udržateľnosť a znovupoužiteľnosť, aby podobný formulár a jeho fragmenty boli použiteľné aj v iných pohľadoch. Podarilo sa vyrobiť validáciu na strane servera, pričom sa validuje pred pridaním do databázy na základe pravidiel, ktoré logicky vymedzujú korektnosť propozícií.

Vyhľadanie turnaja bolo implementované ako filter s nasledujúcimi položkami:

- meno turnaja
- štát
- mesto
- okruh v km, v ktorom sa od daného mesta má turnaj nachádzať
- dátum od
- dátum do

Za účelom vyhľadávania aj v okolí miest bolo do našej PostgreSQL pridané GIS rozšírenie. Polia meno turnaja, štát a mesto boli navyše obohatené o autocomplete. Kategórie turnajov

(minulé, nadchádzajúce, zúčastnené) sa zobrazujú v samostatných zoznamoch, akoby boli rozdelené do tabov.

Registrácia na turnaj bola implementovaná v súlade s návrhom, teda ak bola hráčovi zobrazená ponuka prihlásiť sa na danú kategóriu.

Testovanie

Testovanie prebieha na viacerých úrovniach. Na implementačnej vrstve je otestovaný hlavne model propozícií, ktorý je zodpovedný za spracovanie a uloženie objektu. Takisto sme otestovali metódy, ktoré sme dopĺňali do triedy ActiveRecord.

Kompletné testovanie prebehlo na frontend strane, kedy sme vyskúšali všetky možnosti vkladania, mazania a úpravy polí, pričom overenie správnej funkcionality prebehlo na úrovni databázy. Frontend je takisto otestovaný na autocomplete a predvypĺňanie formulára, čím je teda otestovaná kompletná funkcionality tohto formulára.

Keďže nám chýbajú prostriedky na testovanie PDF dokumentu, propozície sú otestované len na úrovni generovania pri rôznych podmienkach, nie na vizuálnej.

Na vyhľadávanie turnajov a registráciu boli taktiež vytvorené testy. Konkrétne sa testoval autocomplete jednotlivých fieldov, filtrovanie a registrácia na jeden turnaj.

Modul 3 - Manažment turnajov

Analýza

Po úspešnom vytvorení a zverejnení turnaja nasleduje ďalšia fáza - manažovanie. Pod manažovaním sa myslí ovládanie celkového procesu turnaja od fyzického príchodu hráčov až po vyhlásenie výsledkov. Preto ako prvotné bolo potrebné si osvojiť tieto turnajové procesy. Analyzovali sa súčasné riešenia, ktoré sa používajú v praxi.

Prvým krokom je prezencia hráčov, keďže hráči sa prihlasujú pomocou e-mailu, je potrebné ich pri zápise vložiť do párovacieho enginu. Následne sa spustí párovanie a hrá sa turnaj. Pred každým kolom je potrebné vyvesiť hráčske dvojice, aby každý hráč vedel kde a s kým hrá. Po odohraní partie hráči nahlasujú svoje výsledky. Tie slúžia ako vstup pre párovací engine na ďalšie kolo. Výsledky rovnako ako rozpis kôl sa vyvesuje na dostupné miesto vo formáte A4. Na záver turnaja sa vyhlásia finálne výsledky turnaja a uverejnia sa na internete.

Počas týchto procesov sme identifikovali niekoľko problémov, ktoré systém rieši. Prvým podstatným nedostatkom bol vzhľad, ktorý sa javí ako veľmi zastaralý. Ďalšou znepokojivou súčasťou je ťažká ovládateľnosť. Dokonca po rozhovoroch priamo s organizátormi ohľadom súčasných softvérov sme sa dozvedeli, že pre ovládanie je potrebné absolvovať školenie. Tým pádom je kompetentných ľudí na ovládanie len obmedzený počet. Priamo na turnaji sme boli svedkami toho, že program mal problémy s diakritikou a niekedy s párovaním pri malom počte hráčov.

Ďalším problémom je publikovanie výsledkov. Ak hráč odíde z turnaja pred jeho vyhodnotením, musí čakať aj niekoľko hodín pokiaľ nájde výsledky umiestnené na webe, čo je u hráčov nežiaduci stav. Je to spôsobené aj tým, že všetky výsledky sa zadávajú väčšinou na stránke chessresults.com. Po ich nahraní je problémom sa dostať k relevantným dátam, nakoľko sa na stránke zle orientuje. Nedosiadnuteľnosť výsledkov prináša aj iné problémové situácie. Ak je turnaj viacdňový, hráči sa nedozvedia s kým hrajú ďalší deň, ak sa nedostanú k informáciám a tým pádom sa nemôžu pripraviť na súpera.

Návrh

Nakoľko na základe zistených skutočností z analýzy sme sa predovšetkým sústredili na používateľsky jednoduchý a prívetivý dizajn. Základným návrhom bol obsah stránok, na ňom sa postavili prvé návrhy dizajnu. Najskôr sa vytvorili papierové prototypy, v ďalšej fáze

pokračovali medium fidelity návrhy a nakoniec high fidelity návrhy obrazoviek, ktoré sú na obr. X. Už dopredu sme sa sústredili na potreby budúcich používateľov.

Začali sme jednoduchou a rýchlou prezenciou hráčov. Ak sa hráči prihlásia na turnaj cez náš systém, systém ich automaticky vloží do párovacieho enginu. Organizátor potom len odstráni hráčov, ktorí neprišli, poprípade pomocou krátkeho formulára vloží nových účastníkov a môže spustiť párovanie. Párovanie zabezpečuje vyššie opísaný modul. Všetky výstupy párovacieho enginu ako výsledkové listiny či rozpisy sú hneď prístupné. Je možné ich napr. premietiť na plátno pre jednoduchý prístup. Hráči môžu reagovať na prípadné zle zadané výsledky. Víziou je aj zadávanie výsledkov pomocou tabletu či smartfónu. Pri zadávaní výsledkov pre párovanie ďalšieho kola je zaužívané pokročilé zadávanie cez klávesy, preto sme tento štandard zaručili aj my. Návrh počíta aj s real-time zadávaním výsledkov, aby sa každá zmena odoslala na server a hneď prejavila. Na koniec finálne výsledky sú po ukončení turnaja okamžite vyvesené na webovej stránke s jednoduchým prístupom pomocou filtra.

The screenshot shows the SWIPS website interface. At the top, there is a navigation bar with the SWIPS logo and menu items: Organizácia, Turnaje, Players, and Admin Odhlásiť. The main content area displays tournament results for 'Incredible Steel Pants' in the 'Small Wooden Table' category. The results are shown for the 2nd round on 2015-04-27. A table lists the results for 6 pairs of players, including their names, ELO ratings, colors, and scores. The table is structured as follows:

Export párovania				Export výsledkov			Výsledková listina			
No.	Hráč 1	ELO	Skóre	Color	Výsledok	No.	Hráč 2	ELO	Skóre	Color
7	lískovec, matej	1450		WHITE	1-0 0.5-0.5 - 0-1	4	timar, peter	1600		BLACK
8	siroky, jurej	900		WHITE	1-0 0.5-0.5 - 0-1	3	truchan, peter	1600		BLACK
5	timar, pavol	1600		WHITE	1-0 0.5-0.5 - 0-1	2	tamajka, martin	1650		BLACK
6	loebli, jaroslav	1500		WHITE	1-0 0.5-0.5 - 0-1	1	mercury, fredy	1800		BLACK

Obrázok 3 Obrazovka manažovania turnajov

Implementácia

Implementované obrazovky boli vyvinuté podľa high fidelity návrhov. Pre aplikačný vzhľad systému sme použili AJAX a jQuery. Teda používateľ má pocit, ako keby pracoval s klasickou desktopovou aplikáciou, na ktorú je zvyknutý. Na backendovej strane sa napojila funkcionalita modulu párovacieho enginu a manažovania turnajov. Implementovali sme

vhodné akcelerátory, aby sme používateľovi urýchlili prácu so systémom. Akcelerátory sú implementované pomocou klávesových skratiek. Pre export výsledkov bola využitý vhodné rozšírenie pre Ruby. V ňom bolo naimplementované hlavné rozloženie a pravidla potrebné pre generovanie prehľadných výsledkových listín.

Testovanie

Otestovaná je zatiaľ iba funkcionálnosť systému. Keďže na úrovni backendu tam zatiaľ zložitá logika hodná testovania nie je, testujeme iba na frontende. Testujeme pridávanie hráčov, ako už existujúcich tak nových. Testujeme schvaľovanie hráča na turnaj aj jeho zamietanie a celkovú správu hráčov. Testujeme aj zmenu nastavení.

Netestujeme zatiaľ korektné zobrazenie všetkých údajov a zadávanie výsledkov. Netestujeme ani exporty.

Kompletné manažovanie turnaja bolo otestované v ostrej prevádzke na nami organizovanom šachovom turnaji. Vyskytli sa pri ňom niektoré chyby z pohľadu biznis logiky, ktoré sme následne opravili.

Modul 4 - Profil hráča

Analýza

Analýza profilu hráčov sa začala výsledkami dotazníkov, ktoré nám odhalili potreby hráčov. Väčšina hráčov označila súčasné možnosti za nepostačujúce a bola toho názoru, že by privítala konkrétnejšie štatistiky. Tento fakt potvrdzuje aj to, že na Slovensku boli pred dvoma rokmi zverejnené prídavné štatistiky hráčov naprieč výsledkami vo všetkých slovenských ligách, ktoré boli veľmi žiadané, avšak aktuálne už nie sú k dispozícii. Analyzovali sme potreby hráčov a aj konkurenčné weby, ktoré ponúkajú niektoré štatistiky. Z analýzy vyplynuli dôležité aspekty, ktoré chceme hráčom ponúkať v prehľadnej forme.

Z analýzy okrem štatistík vyšli ďalšie dôležité výstupy. Sú to napríklad osobné údaje alebo už spomínané úspechy a záľuby. Dôležitou vlastnosťou profilu je aj spojenie hráčovho profilu v našom systéme s oficiálnym FIDE profilom.

Návrh

Na základe analýzy sa navrhlo niekoľko štatistických pohľadov. Pohľady boli v niekoľkých kategóriách. Buď sú personalizované priamo pre hráča alebo sú to celkové štatistiky na základe partií všetkých hráčov. Celkové pohľady by mali obsahovať štatistické vyhodnotenie úspešností jednotlivých otvorení šachových partií. Personalizované zahŕňajú vývoj používateľa v čase. Návrh počíta aj s prehľadom porovnania tohto vývoja s inými používateľmi, ktorých si používateľ definuje.

Ďalšou podstatnou zložkou návrhu je zobrazovanie dosiahnutých používateľových výsledkov a ocenení, nakoľko analýza ukázala, že práve dosiahnuté výsledky sú motivujúcou stránkou hráčov, na ktorú sú obzvlášť hrdí.

Na obrázku 4 sa nachádza návrh obrazovky.

SWPS

Organizácia
 Turnaje
 Hráči

Jozef Pajtáš
Odhlásenie

Osobné údaje

Turnaje & Ligy

História partíí

Štatistiky

Meno: **Jozef Pajtáš** FidelID: **XX123456**

Mesto: **KadeTade** ELO: **2225 (Fide)**

Klub: **ŠK Zajtra** Actual ELO: **2240 (Fide)**

Rok narodenia: **1985** Titul: **FM**

Pohlavie: **Muž** Schválený od: **31.12.2014**

APPROVED

Vývoj ELA

< Október 2015 >

Po	Ut	St	Št	Pi	So	Ne
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	3	4	5

Obrázok 4 HF návrh obrazovky profilu hráča

Implementácia

Hlavnou časťou implementácie bola implementácia grafov. Naprogramovali sme graf vývoja ratingu hráča tak, že sa v ňom môže porovnať s iným hráčom. Hráč si môže zobrazit' vývoj národného alebo medzinárodného ratingu. Ďalej je naprogramované zobrazenie všetkých odohratých partíí hráča a jeho súperí. Ako posledná vec sa naprogramovalo spojenie hráča s jeho profilom FIDE a zobrazenie osobných informácií.

Testovanie

Otestované boli najmä grafy vývoja ratingu, ktorý je najviac u hráčov žiadaný. Vybralo sa niekoľko hráčov, na ktorých bolo toto zobrazenie plne otestované. Taktiež boli napísané testy na spojenie hráča s jeho FIDE profilom a zobrazenie jeho partíí.

Modul 5 - Automatický partiár

Analýza

Zaznamenané partie predstavujú veľmi cenný zdroj pre každého hráča. Pomocou nich sa môžu pripravovať na prichádzajúce zápasy a po zápase analyzovať svoje chyby, prípadne taktiku súpera. V súčasnosti je zaznamenaných okolo 7 miliónov partií, bohužiaľ nie vo forme verejne dostupnej databázy. Reálne je však partií násobne viac. Problém spočíva v ich zaznamenávaní.

Momentálne sú dva možné spôsoby zaznamenávania partie. Tou bežnejšou je manuálne prepisovanie ťahov na papier, z ktorého potom organizátor, alebo iná povolaná osoba ručne prepisuje ťahy do elektronickej formy. Nevýhody tohto procesu sú jasné - ťahy sa zapisujú ručne a musia to robiť sami hráči. Čítanie rukopisu jednotlivých hráčov tiež môže byť niekedy problém.

Druhý možný spôsob zaznamenania partie je prostredníctvom elektronickej šachovnice. Tu zas môže byť problém dostupnosť a cena, ktorá sa pohybuje v rozmedzí 500 až 1000 eur, čo je pre organizátorov menších turnajov nemalá investícia.

Zaznamenané partie pritom pre nás predstavujú veľmi cenné dáta - na analýzach partií je postavená značná časť nášho biznis modelu. Inovácia vo forme cenovo dostupného partiára pomocou ktorého by sa mohli zaznamenávať partie po celom svete prakticky na akýchkoľvek turnajoch.

Návrh

Navrhli sme formu automatického partiára postavenú na obyčajnej webkamere a metódach a technikách počítačového videnia, ktoré poskytuje C++ knižnica OpenCV. Cieľom je voľne dostupný softvér, ktorý by prostredníctvom webkamery zaznamenával ťahy a rovno ich odosielať k nám na server. Kamera pritom nemusí byť zavesená nad šachovnicou, čo by sa v praxi dosť ťažko dosahovalo, ale stačí ak je na statíve vedľa šachovnice.

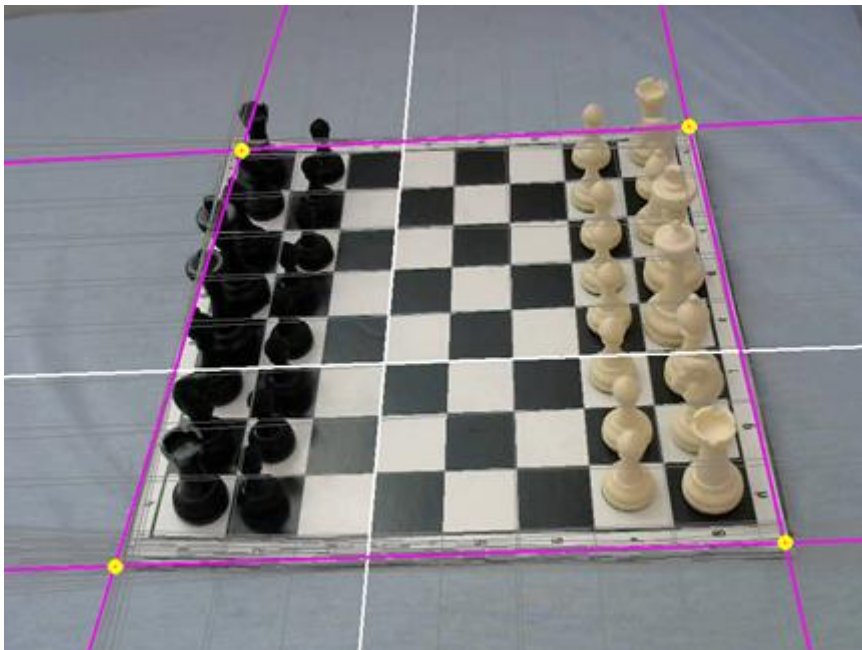
Problém zaznamenávania ťahov sa dá rozdeliť na viac podproblémov. Tým prvým je samotná detekcia šachovnice, ďalej je to detekcia pohybu nad šachovnicou (hlavne teda hráčovej ruky) a až potom samotné hľadanie pozície na ktorú sa figúrka v prípade ťahu presunula.

Možností ako jednotlivé problémy vyriešiť a mnoho. To, ktorá metóda je najlepšia je predmetom ďalšieho výskumu a testovania v reálnych podmienkach.

Implementácia

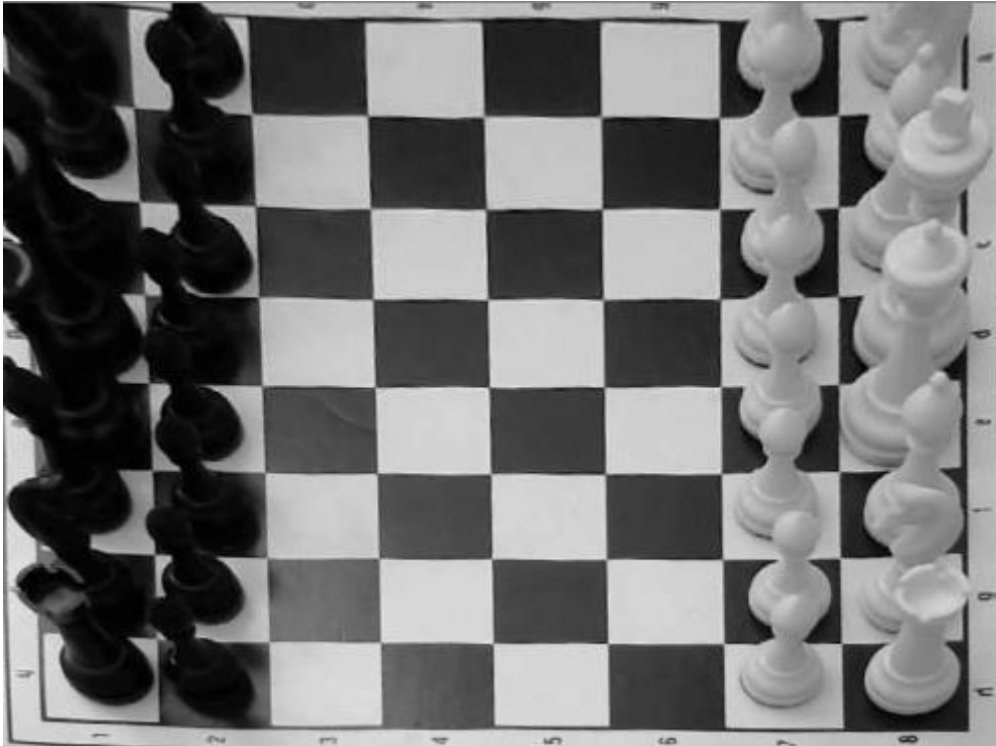
V súčasnosti máme implementované dva prototypy partiára, pričom oba k problému pristupujú vlastným spôsobom. Oba sú implementované v C++ ako desktopová aplikácia.

Prvá detekuje šachovnicu pomocou algoritmu detekcie čiar Hough Lines. Snaží sa identifikovať čiary, ktoré reprezentujú kraje šachovnice a na ich základe potom vypočítať body, ktoré reprezentujú rohy šachovnice, tak ako to vidíme na obrázku 5.



Obrázok 5 Zistenie rohov šachovnice

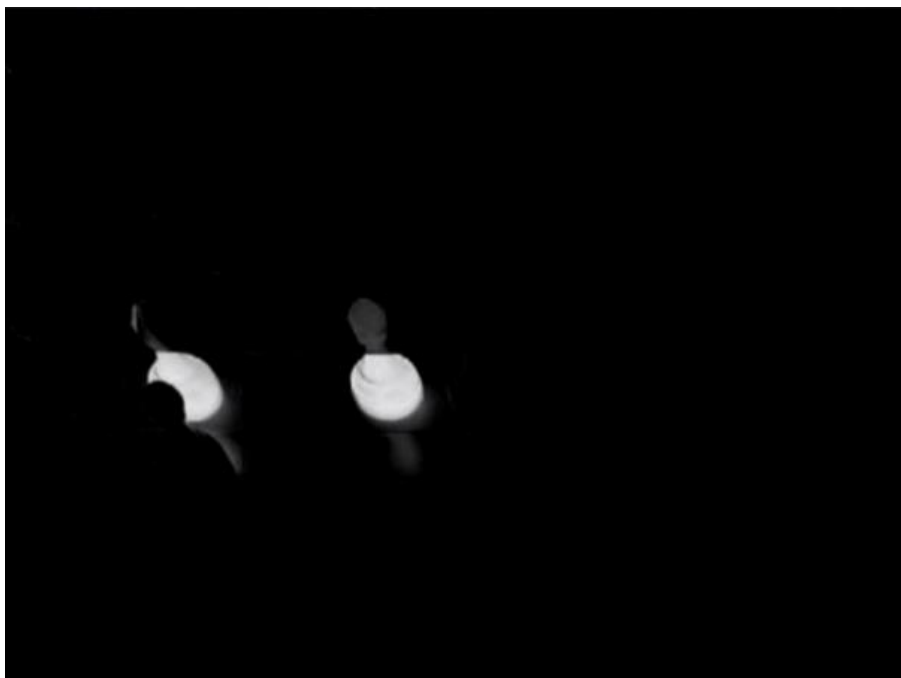
Vidíme, že rohy šachovnice sú vypočítané ako prieniky hraničných čiar (krajov šachovnice). Tieto body následne použijeme na získanie transformačnej matice, pomocou ktorej si šachovnicu premietneme do “vtáčej” perspektívy, tak ako to vidíme na obrázku 6.



Obrázok 6 Perspektívna transformácia šachovnice

Výhodou tohto postupu je, že na šachovnice, ju nepotrebuje mať čistú bez figúrok.

Na zisťovanie, či sa na šachovnici vykonáva nejaký pohyb sme použili MOG2 algoritmus. Pomocou neho vieme detegovať, kedy začal hráč ťahať figúrkou a kedy skončil. Následne spravíme rozdiel obrázka pred ťahom a po ňom, čo nám označí miesta, kde nastala zmena, teda odkiaľ kam sa presunula figúrka, ako to vidíme na obrázku 7.



Obrázok 7 Rozdiel dvoch snímok

Druhý prístup je založený na dvoch fázach:

- Detekcia šachovnice a identifikácia políček
- Detekcia ťahov

V prvej fáze je nutné mať kameru staticky namierenú na prázdnu šachovnicu. Tu sa detekuje šachovnica a jednotlivé políčka. Zdetekované políčka je možné vidieť na obrázku:



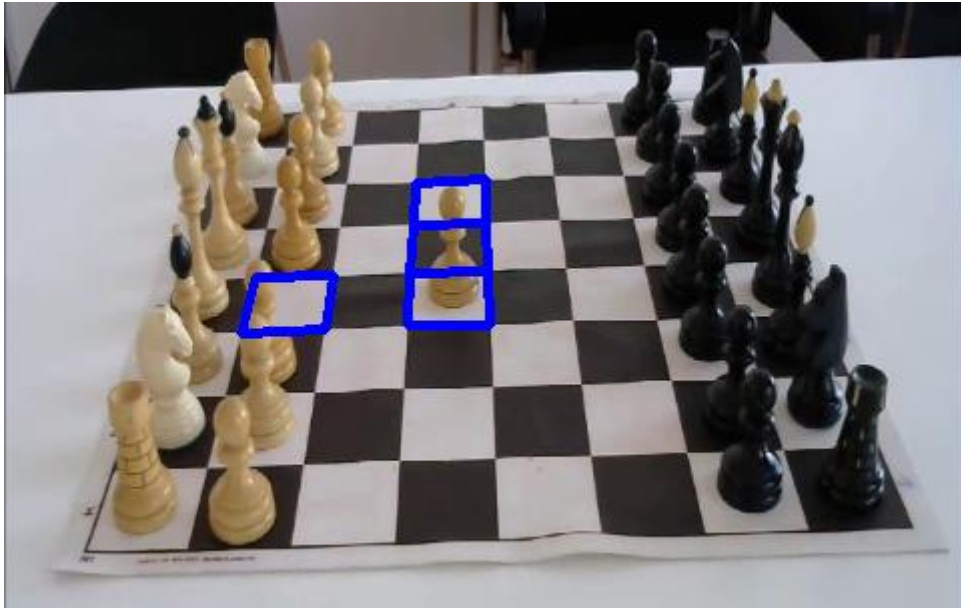
Obrázok 8 Detekcia políček šachovnice

Po detekcii šachovnice program vyzve používateľa, aby rozostavil figúrky. Po potvrdení sa spúšťa fáza dva. V druhej fáze je algoritmom Mixture of Gaussians (MOG) detekovaný pohyb. Ide pritom o pohyb šachistu pri presúvaní figúrok. Určitý čas po tom, ako bol zaznamenaný posledný pohyb na obrazovke, sa porovnáva stav obrazovky pred začiatkom pohybu a po jeho ukončení, podobne ako v prvom prototypu. Výsledkom je maska, ktorá modeluje, na ktorých miestach sa odohral pohyb.

Keďže poznáme pixely patriace jednotlivým políčkam, dokážeme pre každé políčko spočítať, koľko pixelov sa v ňom zmenilo, ako sa zmenila jeho priemerná intenzita a na koľkých jeho pixeloch zdetekoval MOG pohyb. Všetky hodnoty sú pritom prahované, aby sa zabránilo vplyvu prípadného šumu a zminimalizoval sa dopad tieňov na výsledok metódy. Na masku generovanú MOG sa taktiež aplikuje erózia s elipsoidným jadrom nasledovaná dilatáciou. Jej cieľom je odstránenie “falošných” pohybov spôsobených šumom a nekvalitou kamery.

Výsledok detekcie je možné vidieť na obrázku nižšie. Modrou farbou sú označené políčka, ktoré metóda vyhodnotila tak, že sa na nich odohral pohyb. Prírodzene, sú zdetekované aj

políčka, na ktorých figúrka nestojí, ale z pohľadu kamery ich napriek tomu figúrka pokrýva. Za políčka, na ktorých sa odohral pohyb, pritom berieme tie najnižšie v oboch stĺpcoch. Ak bol zdetegovaný pohyb vo viac ako dvoch stĺpcoch, berieme tie stĺpce, kde bol zdetegovaný pohyb na políčku najbližšie ku kamere. Každopádne do políčok, na ktorých sa odohral pohyb, musí byť zaradené jedno o ktorom vieme, že sa na ňom pred začiatkom pohybu nachádzala figúrka.



Obrázok 9 Detekcia zmeny políčka

Testovanie

Implementované prototypy boli zatiaľ testované iba na niekoľkých videách partií. Zatiaľ neprebehlo testovanie na reálnej partií. V reálnom scenári môže nastať viacero situácií, v ktorých majú prototypy zatiaľ problém a to je napríklad dlhšie držanie figúrky šachistom. Ruka sa po nejakej dobe začne javiť ako statická a v tom momente nás zradí detekcia pohybu - vyhlási sa, že ťah bol skončený, nakoľko sa na kamere nič nehýbe, ale v skutočnosti ťah ešte neprebehol. Ďalší podobný prípad je, keď súper odpovie príliš rýchlo a prototyp nestihne zareagovať na pohyb prvého hráča. Udejú sa dva ťahy, pričom prototyp medzi nimi nezaznamenal dlhšiu pauzu a teda nevyhodnotil prvý ťah.

Samostatnú skupinu problémov tvorí aj okolie šachovnice. Špeciálne prvý prototyp mal problém vtedy, keď sa okolo šachovnice nachádzalo príliš veľa vecí a nedokázal na základe výstupu Hough Lines algoritmu spoľahlivo určiť, ktoré čiary reprezentujú hrany šachovnice.

Druhý prototyp bol testovaný podobným spôsobom. Výsledkom bola veľmi presná detekcia pohybu. Problémom zostávajú čierne figúrky na čiernych políčkach, pričom presnosť sa znižuje so zvyšujúcou sa vzdialenosťou od kamery.