

Tytuł pracy magisterskiej

Stanisław Zawadzki

Wrocław, 27 maja 2011

Spis treści

1	Wstęp	3
1.1	Motywacja	3
1.2	Słownik terminów szachowych	3
2	Terminologia szachowa	5
2.1	Szachownica i figury	5
2.2	Zasady poruszania się figur	5
2.2.1	Pionki	5
2.2.2	Lekkie figury	7
2.2.3	Ciężkie figury	7
2.2.4	Król	8
2.3	Posunięcia niezwykłe	8
2.3.1	Bicie w przelocie	8
2.3.2	Roszada	9
2.3.3	Promocja	9
2.4	Początek, przebieg i koniec rozgrywki	9
2.4.1	Rozpoczęcie rozgrywki	10
2.4.2	Różne sposoby zakończenia partii	10
2.5	Dodatkowe informacje o partiach szachowych	11
2.5.1	Czas rozgrywki	11
2.5.2	Zapis rozgrywki	11
2.6	PGN i FEN - sposoby zapamiętywania partii	12
2.7	Oznaczenia figur szachowych	13
2.8	Szachownice elektroniczne	13
2.9	Silniki szachowe	14
2.10	Algorytmy stosowane w silnikach szachowych	16
2.11	Universal Chess Interface i xBoard	17

3	Ewaluacja dotychczasowych rozwiązań	19
3.1	Ankiety na temat istniejących rozwiązań	19
3.2	Kafejki szachowe	22
3.3	Programy szachowo-bazodanowe	23
3.4	Chessbomb - wady i zalety serwisu	26
3.5	Cechy wspólne dotychczasowych rozwiązań oraz ich ocena	27
4	Projektowanie własnego interfejsu	29
4.1	Rodzaje użytkowników	29
4.2	Przypadki i scenariusze użycia oraz papierowy interfejs	30
4.2.1	Scenariusze użycia	30
4.2.2	Przypadki użycia	31
4.2.3	Papierowy prototyp	31
4.2.4	Wnioski z testów na papierowym prototypie oraz ich wpływ na kształt interfejsu	32
4.3	Wygląd interfejsu	32
4.4	Dobór kolorów interfejsu	32
4.5	Rodzaje figur i szachownic	32
5	Implementacja programu	32
5.1	Technologia	32
5.1.1	Ruby on Rails	33
5.1.2	JRuby on Rails	33
5.1.3	Dodatkowe biblioteki	34
5.1.4	Programowanie zwinne	34
5.2	Struktura programu	35
5.2.1	Funkcjonalności	35
5.2.2	Struktura bazy danych	36
5.3	Pierwsza iteracja	38
5.4	Ewaluacja z użytkownikami po pierwszej iteracji	38
5.5	Druga iteracja	39
5.6	Ewaluacja po drugiej iteracji	40
5.7	Fragmenty najciekawsze algorytmicznie	40
6	Podsumowanie	40
7	Bibliografia	40
	Bibliografia40	

1 Wstęp

1.1 Motywacja

W ciągu dwóch ostatnich dekad stworzonych zostało mnóstwo programów i witryn internetowych związanych z szachami. Realizują one różne funkcjonalności, implementują dziesiątki różnych interfejsów graficznych, umożliwiają niemal wszystkie czynności, jakie człowiek może wykonać z rozgrywkami na 64 polach. Większości z nich łączy jednak jedna wspólna cecha - skupiają się na zagadnieniu programistycznym, które w przypadku szachów jest często fascynujące, zapominając jednak o opcjach pozwalających użytkownikom na sprawne posługiwanie się ich narzędziami. Motywacją do mojej pracy było stworzenie wygodnego dla użytkowników serwisu internetowego pozwalającego na łączenie wielu funkcji związanych z szachami. Głównymi odbiorcami programu będą kibice oglądający przez Internet turnieje szachowe, ale ma być również użyteczny dla osób szukających informacji o turniejach szachowych, a także dla zainteresowanych profesjonalnymi analizami partii. Nie zamierzam zaprzestawać prac nad programem - w dalszej przyszłości chciałbym rozszerzyć spektrum zastosowań strony o możliwość bezpośredniej gry przez internet i kalendarz do wyszukiwania turniejów szachowych. Rozważam także ideę napisania nowej biblioteki do obsługi partii szachowych.

Zanim podjąłem decyzję na temat funkcjonalności interfejsu przeprowadziłem badania obejmujące kilkudziesięciu szachistów oraz kilka już istniejących programów szachowych lub stron WWW o tematyce szachowej. Uwzględniając uwagi oraz mając we względzie swoje prywatne potrzeby zaprojektowałem i wykonałem system do przechowywania w bazie danych, przeglądania i komentowania partii szachowych.

1.2 Słownik terminów szachowych

W trakcie tej pracy będzie przewijać się sporo terminów dotyczących szachów i programów szachowych. W tym punkcie wyjaśnię znaczenie owych słów, a także podam listę wyrażen do nich synonimicznych.

- **Szachy** - gra planszowa rozgrywana na szachownicach przy użyciu figur szachowych; Synonim: królewska gra;
- **Szachownica** - plansza, na której rozgrywa się partię szachów. Podzielona na 64 pola - 32 białe i 32 czarne.
- **Figura szachowa** - Jedna z sześciu możliwych typów figur (pionek, goniec, skoczek, wieża, hetman, król). Zawodnicy przemieszczają je podczas swoich posunięć Synonim: bierka szachowa;figura
- **Posunięcie** - Zmiana położenia figury szachowej wykonana przy zachowaniu zasad poruszania się tych figur Synonim: ruch, półruch

- **Szach** - Zaatakowanie króla przeciwnika, zmuszające go do natychmiastowej reakcji
Synonim: atak na króla
- **Mat** - Szach, po którym nie można nie utracić swojego króla. Kończy natychmiast partię jako wygrana zawodnika, który wykonał posunięcie matujące
- **Wygrana** - Aby wygrać partię należy zamatować przeciwnika lub skłonić go do dobrowolnego poddania partii. Zawodnik wygrywający otrzymuje jeden punkt, jego przeciwnik nie zyskuje żadnych punktów.
- **Remis** - Uznanie partii za nierozstrzygniętą. Skutkuje podziałem punktów między obu zawodników (oba otrzymują wówczas $\frac{1}{2}$ punktu).
Synonim: podział punktu
- **Propozycja remisu** - Dobrowolna oferta składana w dowolnym momencie gry przez jednego przeciwnika drugiemu. Jej zaakceptowanie oznacza podział punktu, jej odrzucenie nie wpływa na przebieg rozgrywki
- **Reklamowanie remisu** - Istnieją sytuacje, w których można wnioskować o odgórne przyznanie remisu. Jeśli pozycja spełnia opisane w przepisach kryteria, wówczas sędzia może orzec podział punktu. W przeciwnym razie partii toczy się dalej.
- **Czas rozgrywki** - Choć nie jest to wymagane przepisami, w praktyce partie mają swój limit czasu. Partia trwa wówczas $2X$, gdzie X oznacza czas dla jednego zawodnika. Czas zawodnikowi płynie wówczas gdy jest kolej na jego posunięcie. Do odmierzania czasu korzysta się z zegarów szachowych. W momencie jak zawodnikowi kończy się czas do namysłu przegrywa on partię.
Synonim: tempo partii
- **Zegar szachowy** - Urządzenie składające się z dwóch czasomierzy i przełącznika między nimi. Początkowo czas na obu licznikach jest ten sam, jednak upływa on różnie u obu zawodników - w zależności od czasu spędzonego na namyśle nad swoimi posunięciami.
- **Czas dodawany** - W niektórych przypadkach wybierane jest tempo gry, które modyfikuje wartość czasu na zegarze po wykonaniu posunięcia. Jest to zazwyczaj wartość pomiędzy trzema a trzydziestoma sekundami. Dodaje się ona automatycznie po przełączeniu dźwigni zegara.
- **Zawodnik grający białymi** - Gracz rozpoczynający grę.
Synonimy: Pierwszy zawodnik, gracz białymi, białe
- **Zawodnik grający czarnymi** - Gracz wykonujący posunięcia jako drugi.
Synonimy: Drugi zawodnik, gracz czarnymi, czarne
- **Silnik szachowy** - Program oceniający pozycję analizujący możliwe przebiegi partii.
Synonimy: Program szachowy, silnik, silnik analityczny

- **Zapis partii** - Zapisanie ciągu wszystkich wykonanych posunięć wraz z opisem zawodników i zawodów
- **Szachownica elektroniczna** - Specjalna szachownica pozwalająca na automatyczne zczytywanie przebiegu partii

2 Terminologia szachowa

Aby móc rozmawiać o szachowych interfejsach trzeba posiadać podstawową wiedzę na temat zasad królewskiej gry. W bieżącym rozdziale postaram się opisać każdą możliwą sytuację w partii szachowej, ale także opiszę specyfikę sprzętu i oprogramowania, które będą kluczowe dla mojej aplikacji. Rozpocznę od opisu ruchu figur, później omówię posunięcia nadzyczajne a na końcu opiszę wszystkie inne aspekty, które są kluczowe w przypadku implementowania interfejsu szachowego. Na końcu zostaną opisane szachownice elektroniczne oraz silniki szachowe, które to umożliwią komputerowi znajdowanie potencjalnie najsilniejszych posunięć w pozycji.

2.1 Szachownica i figury

Choć zasady gry w szachy są wielu osobom znane, na potrzeby tej pracy chciałbym przytoczyć wszystkie informacje, które mają znaczenie dla interfejsów szachowych. Plansza, na której odbywa się rozgrywka jest nazywana szachownicą. Jest to obszar podzielony na 64 pola (8x8) numerowane w wierszach liczbami (1-8), w kolumnach zaś literami (a-h). W terminologii szachowej zarówno na wiersze, jak i na kolumny mówi się "linie". Patrząc z perspektywy gracza grającego białymi w lewym dolnym rogu jest pole a1, patrząc z perspektywy jego przeciwnika jest to pole h8. Pola są dwóch rodzajów - białe i czarne. Pola jednego koloru sąsiadują ze sobą po skosie i **pole a1 jest zawsze czarne**. Fizycznie nie ma wymagania, by używać dokładnie tych kolorów.¹ i w przypadku programów szachowych standardem jest uznanie każdego koloru poza białym jako czarny.

2.2 Zasady poruszania się figur

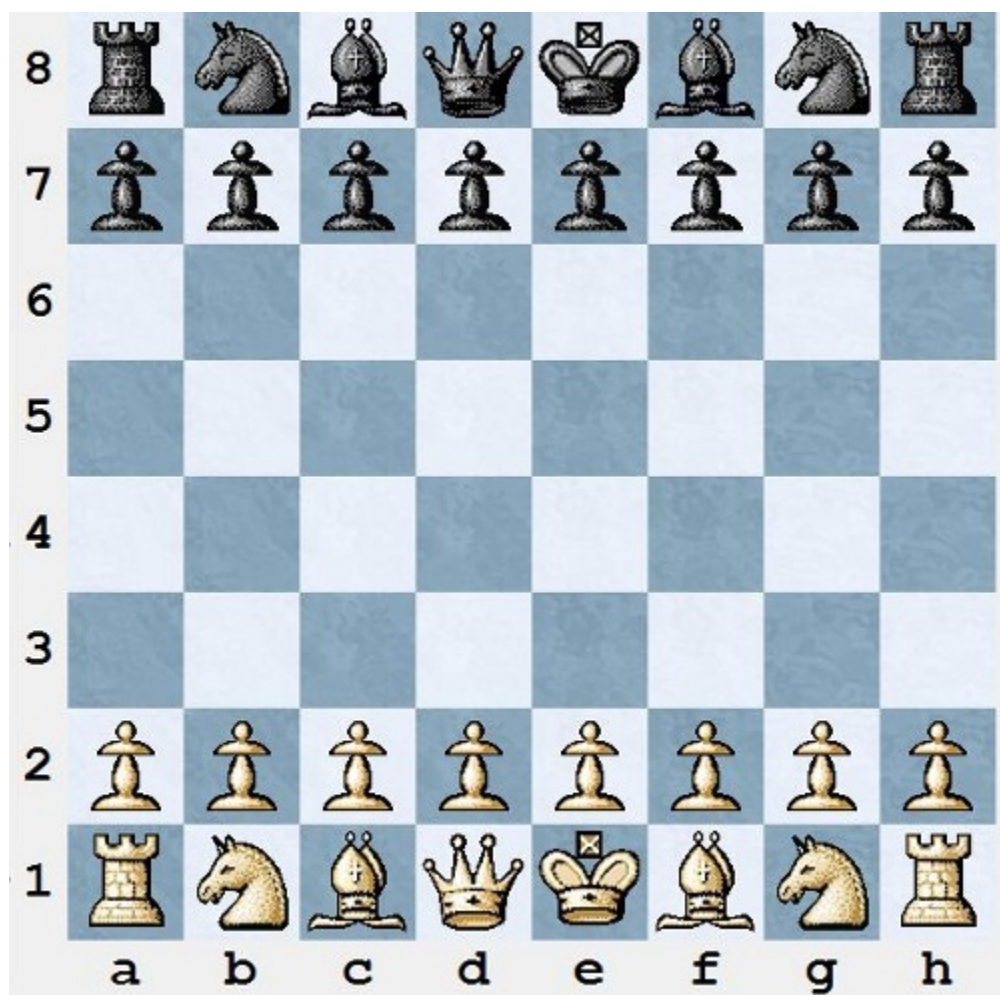
2.2.1 Pionki

W przypadku pionków sytuacja jest dość prosta. Te najliczniej reprezentowane figury mogą się przesuwać wyłącznie do przodu. Kiedy przypada kolej danego zawodnika może on wykonać pionkiem jedno z trzech możliwych posunięć:

1. **Przesunąć pionka o jedno pole do przodu** (t.j. wzdłuż kolumny)

Jest to podstawowe posunięcie możliwe do wykonania pionkiem. Można je wykonać wtedy, gdy pole na które zmierza pionek nie jest zajęte przez inną figurę (zarówno swoją, jak i cudzą).

¹W rzeczywistości czarne pola są najczęściej ciemnobrązowe, białe zaś mają kolor jasnożółty.



Rysunek 1: Pozycja początkowa

2. Przesunąć pionka o dwa pola do przodu

Posunięcie to można wykonać tylko wtedy, gdy pionek zajmuje początkową pozycję. Jak w poprzednim przykładzie pole, na które zmierza pionek, jak i pole pośrednie nie mogą być zajęte przez żadną figurę.

3. Zbić figurę przeciwnika poruszając się po skosie do przodu

Pionki jako jedyne szachowe figury biją w inny sposób niż się poruszają. W zwyczajnej sytuacji pionki poruszają się wzdłuż kolumn. Jedynym sposobem, aby pionek przeszedł na sąsiednią linię jest zabicie innej figury. Pionki biją "po skosie", czyli jeśli pionek białych stoi na polu e3, to może zbić figurę, która stoi na d4 lub f4. W przypadku gdyby na e3 stał pionek czarnych, mógłby on zbić figurę na polu d2 lub f2. W wyniku zbiccia stawiamy pionka na polu dotychczas zajmowanym przez daną figurę, tamtą zaś usuwamy z szachownicy

Dodatkowym i bardzo rzadkim ruchem jest tzw. „bicie w przelocie”, o którym napiszę w dziale "Posunięcia niezwykłe"

2.2.2 Lekkie figury

W nomenklaturze szachowej mianem „figury lekkiej” nazywamy gońce i skoczki. W tym punkcie zajmiemy się zdefiniowaniem sposobu poruszania się przez te dwie figury. Gонец, zwany czasem błędnie lauffrem, porusza się zawsze po skosie. Wynika z tego, że jeśli początkowo figura ta jest ustawiona na polu białego koloru, to do samego końca rozgrywki będzie się poruszała wyłącznie po polach tego właśnie typu. Gонец nie może „przeskoczyć nad przeszkodą”. Stąd wynika, że może poruszyć się maksymalnie tak daleko, jak dużo ma wolnego miejsca po diagonalu². Gонец może zbić pierwszą figurę, która staje na jego drodze.

Skoczek jest jedyną figurą, która może przeskoczyć nad innymi figurami. Ruchy skoczka jest najłatwiej porównać do litery „L”. Może on wykonać posunięcie w każdą stronę, pod warunkiem iż będzie to ruch opisany jako "dwa pola do przodu i jedno w bok". W przypadku skoczka zwrot „do przodu” oznacza dowolny z czterech kierunków. Bicie skoczkiem polega na tym, że skoczek może zabrać figurę z tego pola, na które uda mu się wskoczyć.

2.2.3 Ciężkie figury

Podobnie jak w przypadku lekkich figur termin „ciężkie figury” określa dwa rodzaje szachowych figur: wieżę i hetmana. Figury ciężkie w zdecydowanej większości przypadków są silniejsze od pozostałych figur.

Posunięcia wieży są podobne do posunięć gońca, z tą różnicą, że porusza się ona wzdłuż pionowych i poziomych linii. Tak jak w przypadku gońca może ona bić pierwszą figurę, która wystąpi na jej drodze. Innym ruchem, który może ona wykonać, jest roszada opisana

²W nomenklaturze szachowej każda ukośna linia mająca przynajmniej trzy pola jest diagonalą

w działach posunięć niezwykłych.

Jak porusza się hetman chyba nie muszę opisywać z jednego bardzo prostego powodu. Problem ośmiu hetmanów jest na tyle znanym zagadnieniem algorytmicznych, że każdy absolwent studiów informatycznych musiał się z nim wielokrotnie zapoznać. Skrótem opisując - hetman jest hybrydą wieży i gońca - łączy wszystkie cechy obu tych figur (poza możliwością roszady)

2.2.4 Król

Król jest, w zależności od punktu widzenia, albo najsłabszą, albo najsilniejszą figurą. Porusza się on w bardzo prosty sposób - może przejść na każde z okolicznych pól. Jedynym dodatkowym posunięciem w jego arsenale jest wzmiankowana wcześniej i opisana w następnym podrozdziale roszada. Jest jednak bardzo ważna cecha, która odróżnia króla od każdej innej figury. Monarcha nie może zostać umieszczony na polu atakowanym przez chociaż jedną figurę rywala. Tak samo trzeba zawsze coś przedsięwziąć, jeśli przeciwnik wykonując ruch zaatakował naszego króla. W tym przypadku trzeba wykonać jeden z trzech manewrów

- odejść królem na nieatakowane pole;
- zbić atakującą figurę;
- przesłonić obszar działania atakującej figury.

Król jest jedyną figurą z naszego arsenału, której nie możemy stracić. W momencie gdy monarcha zostaje tak osaczony, że nie może uniknąć zagłady, następuje koniec partii. Więcej o zakończeniu partii szachowej będzie napisane w następnych rozdziałach.

2.3 Posunięcia niezwykłe

W poprzednim punkcie przedstawiłem wszystkie zwyczajne sposoby na wykonanie posunięcia. Teraz nadszedł czas na opisanie przypadków posunięć nadzwyczajnych. Niektóre z nich zdarzają się bardzo rzadko, niektóre z nich są obecne w prawie każdej partii szachowej.

2.3.1 Bicie w przelocie

Bicie w przelocie (ang. *en passant*) jest najrzadszym szachowym posunięciem. Występuje ono tylko w jednym momencie. Warunkiem koniecznym i wystarczającym, aby można było zagrać ten ruch jest fakt, iż pionek przeciwnika ruszył się o dwa pola do przodu, zaś nasz pionek atakuje pośrednie pole między początkowym o końcowym ustawieniem pionka rywala. Aby dokonać zbitcia przesuwamy pionka na owo pole pośrednie i zdejmujemy pionka rywala (jest to jedyny przypadek w grze, że bije się figurę z pola, na które się nie postawiło swojej figury). Ważną sprawą jest to, że ten ruch można wykonać tylko bezpośrednio po ruchu pionem o dwa pola. Później takie bicie jest niezgodne z przepisami.

2.3.2 Roszada

Roszada, w przeciwieństwie do bicia w przelocie, choć jest posunięciem niezwykłym, zdarza się bardzo często. Jest ona jedynym szachowym posunięciem, które wprawie w ruch więcej niż jedną figurę. Polega ona na jednoczesnym przesunięciu króla i wieży po pierwsze linii (odpowiednio dla czarnych figur - ostatniej). Muszą być jednak spełnione następujące warunki:

1. Wszystkie pola między królem a wieżą muszą być puste;
2. Zarówno król, jak i wieża nie mogły wcześniej wykonać ani jednego posunięcia;
3. Król nie jest atakowany przed wykonaniem roszady, nie będzie zaatakowany po wykonaniu roszady i nie przekroczy zaatakowanego pola w trakcie jej wykonywania.

Oczywiście drugi z tych punktów determinuje, że roszadę jeden zawodnik może wykonać maksymalnie raz w czasie całej partii.³

Ponieważ są dwie wieże na szachownicy to rozróżniamy dwa typy roszad.

- roszada krótka (występująca częściej w partiach szachowych) - król przechodzi z pola e1 na pole g1, wieża przesuwa się z h1 na f1
- roszada długa - król przechodzi z pola e1 na pole c1, wieża przesuwa się z pola a1 na pole d1 (zauważmy, że w przypadku tej roszady pole b1/b8 **może** być atakowane).

2.3.3 Promocja

Promocji dokonuje się z pomocą przesunięcia pionka aż do końcowej linii szachownicy (ósmej w przypadku białego pionka, pierwszej w przypadku czarnego). W momencie gdy pionek osiąga skraj szachownicy **musi** on zostać zmieniony na hetmana, wieżę, gońca lub skoczka **tego samego koloru!** Dozwolone jest posiadanie więcej figur, niż określa to stan początkowy (czyli można mieć na szachownicy nawet dziewięć hetmanów). Promocji można dokonać każdym pionkiem i cała czynność, czyli postawienie pionka na ostatniej linii oraz zamienienie go na figurę trwa dokładnie jeden ruch.

2.4 Początek, przebieg i koniec rozgrywki

W poprzednim punkcie opiszę warunki początkowe i końcowe rozgrywki szachowej. Zwłaszcza zakończenie partii jest ważną kwestią, gdyż istnieje duża liczba sytuacji, w których partię uznaje się za zakończoną.

³Choć w praktyce turniejowej zdarzały się przypadki, iż zawodnicy w stresie zapomnieli o tym, że roszada miała lub miejsce lub król wykonał już posunięcie i w wyniku tego partia zaczęła się toczyć w sposób nieprawidłowy

2.4.1 Rozpoczęcie rozgrywki

Na początku figury są rozstawione w sposób przedstawiony na diagramie nr XXX. Po zwyczajowym przywitaniu się z przeciwnikiem następuje uruchomienie zegara szachowego i białe wykonują swój pierwszy ruch. Od tej chwili obaj zawodnicy wykonują na zmianę po jednym posunięciu, korzystając z opisanych we wcześniejszym rozdziale zasad. W momencie gdy zawodnik zaatakuje króla rywala, powinien powiedzieć "szach", aby ów wiedział, że ma obowiązek zaradzić temu problemowi. Posunięcia są wykonywane na zmianę aż do momentu, aż w jakiś sposób nadejdzie koniec partii - w przypadku gdy w pewnym momencie nastąpił nieprawidłowy ruch powraca się do ostatniej prawidłowej pozycji. Sposobów, aby zakończyć partię jest naprawdę wiele, każda z nich jest szczegółowo opisana w przepisach Międzynarodowej Federacji Szachowej (FIDE).

2.4.2 Różne sposoby zakończenia partii

Partiastandardowo może zakończyć się na dwa sposoby - remisem lub zwycięstwem jednej ze stron. W szczególnych przypadkach dozwolone są jednak inne wyniki, takie jak walkower (oznaczany + - jeśli walkowera oddały czarne i w -+ w przeciwnym przypadku), obustronna porażka (oznaczana 0-0, przyznawana w przypadku skrajnego złamania zasad fair-play) oraz wynik $\frac{1}{2} - 0$ (lub $0 - \frac{1}{2}$), który może się zdarzyć w wyjątkowych sytuacjach (jednemu z zawodników pozostał tylko król, drugi wciąż posiada materiał matujący, ale przekracza czas przeznaczony do namysłu). Trzeba jednak pamiętać, że wyniki niestandardowe zdają się w znikomym ułamku liczby partii (najczęściej z nich widuje się walkowera). Relatywnie najbardziej skomplikowane jest ustalenie, że partia skończyła się remisem. Zakończenie partii określają przepisy FIDE:

ARTYKUŁ 5. PARTIA ZAKOŃCZONA

⁴ **5.1. (a)** Partię wygrywa zawodnik, który zamatował króla przeciwnika. Mat natychmiast kończy partię pod warunkiem, że pozycja matowa powstała w wyniku prawidłowego posunięcia.

(b) Partię wygrywa zawodnik, którego przeciwnik oświadczył, że poddaje się. Poddanie natychmiast kończy partię. **5.2 (a)** Partia kończy się remisem, gdy zawodnik będący na posunięciu nie może wykonać żadnego prawidłowego ruchu, a jego król nie jest szachowany. Taką pozycję określa się słowem „pat”. Pat natychmiast kończy partię, pod warunkiem, że pozycja patowa powstała w wyniku prawidłowego posunięcia.

(b) Partia kończy się remisem, jeśli powstaje pozycja, w której żaden z zawodników nie może zamatować króla za pomocą jakiegokolwiek serii prawidłowych posunięć. Pojawienie się tzw. „martwej pozycji” też natychmiast kończy partię, pod warunkiem, że powstała ona w wyniku prawidłowego posunięcia. (Patrz art. 9.6)

(c) Partia kończy się remisem w wyniku uzgodnienia pomiędzy obydwoma zawodnikami w trakcie partii. Zgoda na remis natychmiast kończy partię. (Patrz art. 9.1).

⁴Przepisy gry od 01.07.2009 wg Światowej Federacji Szachów

(d) Partia może zakończyć się remisem, jeżeli identyczna pozycja ma pojawić się na szachownicy przynajmniej po raz trzeci lub już pojawiła się przynajmniej trzykrotnie. (Patrz art. 9.2).

(e) Partia może zakończyć się remisem, jeżeli obaj zawodnicy przynajmniej w ostatnich 50 posunięciach nie wykonali żadnego ruchu pionkiem, ani nie pobili żadnej bierki. (Patrz art. 9.3).

Trzeba zawsze pamiętać o każdym z wymienionych sposobów i posiadać narzędzia pozwalające sprawdzić, czy strony mogą zgodzić się na podział punktu. W ostatnich latach spora część zawodów szachowych dodała zasadę zakazu proponowania remisu przed zagranieniem 30 posunięć. Jest to jednak wciąż jedynie wewnętrzny przepis i organizatorzy turnieju nie mają obowiązku się do niego stosować - oficjalnie wystarczy, że zawodnicy zagrają dwa półruchy⁵ Nie licząc sytuacji zupełnie losowych⁶ partie mogą kończyć się wyłącznie w sposób określone w wyżej przytoczonych przepisach.

2.5 Dodatkowe informacje o partiach szachowych

Pisząc o partiach szachowych należy przytoczyć jeszcze kilka terminów, które dotyczą każdej rozgrywki. Nie mają one bezpośredniego związku z szachami i można towarzysko rozgrywać partię bez tych elementów, jednak w profesjonalnych turniejach są one najczęściej wymagane przepisami.

2.5.1 Czas rozgrywki

W zdecydowanej większości przypadków partie szachowe, nawet te towarzyskie, grane są z limitem czasu. Może to być zaledwie minuta dla zawodnika na partię, a może to być kilka godzin. W ostatnich latach coraz bardziej popularna jest opcja dodawania czasu po wykonaniu każdego posunięcia. Zawodnicy mogą zużyć swój czas w zależności od potrzeb na analizowanie swoich posunięć. W momencie gdy wykonają ruch przełączają upływ czasu na konto rywala. W przypadku, jeśli zabraknie czasu do namysłu, partia jest uznana za przegraną zawodnika, który przekroczył limit. Historycznie czas odmierzano na zegarach mechanicznych, jednak coraz bardziej się od tej praktyki odchodzi na rzecz elektronicznego mierzenia upływu czasu.

2.5.2 Zapis rozgrywki

W partiach granych dłuższym tempem (minimum godzina dla jednego zawodnika) istnieje obowiązek prowadzenia zapisu partii. Służy on zapamiętywaniu stanu partii po każdym wykonanym posunięciu, aby w każdej chwili można było prześledzić historię rozgrywki. Ogólnoświatową konwencją jest stosowanie tzw. skróconej notacji algebraicznej. Na począt-

⁵Wcześniej wg definicji partia nie jest jeszcze uznana za rozpoczętą

⁶Zdarzały się nawet przypadki śmierci przy szachownicy

ku zapisywana jest pierwsza litera figury⁷, później zaś pole, na które ona zmierza. Jeśli jest to nierozstrzygające dopisuje się jeszcze informację, która z figur⁸ W świecie rzeczywistym odbywa się to z użyciem papierowych blankietów, w świecie programów komputerowych przechowuje się wiadomości o partii z użyciem plików o określonej strukturze.

2.6 PGN i FEN - sposoby zapamiętywania partii

Od kiedy tylko informatyka włączyła się komercyjnie w świat szachów powstało pytanie, jak skutecznie zapisywać i odczytywać pojedyncze rozgrywki szachowe. W tym celu powstał format **PGN (Portable Game Notation)**⁹, w którym w prosty sposób można zapisać wszystkie kluczowe informacje o partii szachowej.

PGN jest formatem tekstowym, czytelnym dla bardziej doświadczonego szachisty. Nawet nie dysponując żadnym programem, który graficznie przedstawiłby daną rozgrywkę, bez najmniejszych problemów można otworzyć ten plik w edytorze tekstowym i przeczytać nagłówki i posunięcia. Format ten sprawdza się dobrze w przypadku pojedynczych partii lub zbiorów nie przekraczających kilkadziesiąt tysięcy rozgrywek, jednak w przypadku wielkich baz (kilka milionów partii) staje się bardzo nieefektywny z uwagi na zerową kompresję i brak wsparcia dla algorytmów wyszukiwania.¹⁰ Plik PGN ma bardzo prostą strukturę - składa się z zapisów partii oddzielonych znakami nowej linii. Zapis każdej partii składa się z dwóch części: nagłówka i treści partii. W nagłówku podawane są informacje na temat warunków grania partii (miejsce, data, turniej czy runda turnieju), zaś w treści partii wypisane są posunięcia a także możliwe warianty i komentarze słowne. Dane w nagłówku przedstawione są w postaci [klucz "wartość"], przy czym obecność części kluczy nie jest wymagana. Posunięcia i warianty podawane są w skróconej lub rozszerzonej notacji algebraicznej. W samych plikach nie ma żadnej kontroli poprawności danych, zadanie to jest przerzucone na programistów korzystających z owych plików.

FEN (Forsyth-Edwards Notation) służy nie do zapisu partii, ale do zapamiętania konkretnej sytuacji na szachownicy. Czasem bowiem mamy do czynienia z sytuacją, kiedy nie potrzebujemy wcześniejszego przebiegu rozgrywki i interesuje nas tylko konkretna pozycja. Początkowo wydaje się to trywialnym zadaniem, jednak po głębszym namyśle odkrywamy problemy - jak zapisać, czy któraś ze stron może jeszcze zrobić roszadę (nie wynika to z samego ustawienia, bo król mógł używając przynajmniej dwóch posunięć raz jeszcze znaleźć się na pozycji początkowej), czy też możliwość bicia w przelocie? Do tego trzeba zadbać o zwięzłość tego zapisu. Notacja FEN opisuje linią po linii, każde pole na szachownicy, zliczając pola puste.

Przykładowo - pozycja z pięcioma figurami, białym królem na polu g5, hetmanem na polu e6 oraz czarnymi figurami na polach g7(król), f7 (wieża) i h7 (pionek) wygląda następująco "8/5rkp/4Q3/6K1/8/8/8/8 w - - 0 1"

⁷W niektórych językach, jak n.p. w angielskim jest to nie zawsze możliwe. Wówczas stosuje się inne oznaczenie

⁸Na przykład Sde2 - skoczek z linii 'd' idzie na pole e2

⁹dokumentację można znaleźć m.in. pod adresem <http://pgn.freeservers.com/Standard.txt>

¹⁰Wówczas stosowane są inne, komercyjne formaty.

Znak „/” oznacza nową linię poziomą szachownicy, wielkie litery opisują białe figury, małe litery opisują figury czarne. Litera 'w' oznacza, że grę rozpoczynają białe. Myślniki świadczą o braku możliwości zrobienia roszady. W ten sposób, korzystając z zaledwie kilkunastu znaków opisaliśmy całą pozycję. Z notacji FEN często korzysta się w plikach PGN, gdyż dzięki niej można rozpocząć zapis partii od dowolnej pozycji, a nie tylko od początkowej.

2.7 Oznaczenia figur szachowych

W formacie PGN należy rozstrzygnąć w jaki sposób oznaczamy figury. W szachach nie ma jednolitego międzynarodowego nazewnictwa, każdy kraj stosuje nazwy w swoim języku. Jedynym standardem jest omijanie przedrostka, w przypadku gdy dochodzi do posunięcia pionkiem. Każdy profesjonalny szachista potrafi korzystać z nazw figur w języku angielskim¹¹: król-K(king), hetman-Q(queen), wieża-R(rook), goniec-B(bishop), skoczek-N(knight). Nie należy jednak zakładać, że każdy użytkownik będzie posiadał tę wiedzę. Ciekawym rozwiązaniem jest korzystanie z piktogramów, które nie tylko już są pewnym standardem¹², ale są także graficznie zbliżone do rzeczywistych figur.

2.8 Szachownice elektroniczne

Jeśli myślimy o transmisjach partii szachowych na żywo musimy także zrozumieć sposób, w jaki posunięcia szachowe są wysyłane do sieci. W tym celu korzysta się ze specjalnego sprzętu zwanego szachownicami elektronicznymi¹³. Z wyglądu przypominają one zwykłe szachownice, jednak podłączone są one przewodowo¹⁴ do komputera, który przy pomocy programu autorstwa firmy DGT odczytuje sygnały z szachownic i interpretuje je jako posunięcia. Wówczas można już te partie zapisywać do plików i wysyłać na serwery do odczytu przez programy prowadzące transmisje.

Podczas prowadzenia transmisji z użyciem szachownic elektronicznych zdarzają się błędy, które można podzielić na kilka kategorii:

- Błędy fizyczne

Na salach turniejowych (wyłączywszy nieliczne superturnieje) nie ma możliwości, aby kable łączące szachownice elektroniczne z komputerem poprowadzić pod podłogą lub w inny sposób uniemożliwiający przypadkowe ich zerwanie. Z tego powodu trzeba brać pod uwagę, iż któryś z kibiców lub zawodników¹⁵ spowoduje zerwanie kabla z transmisji. Niesprawne lub niepoprawnie przekazujące dane zegary także są częstą przyczyną błędów w przypadku oglądania partii on-line;

¹¹Nie należy jednak tego mylić z tzw. notacją angielską, która jest obecnie już nieużywanym sposobem zapisywania posunięć

¹²Istnieją choćby w domyślnych czcionkach systemów biurowych jak Microsoft Word czy OpenOffice

¹³Aktualnie rynek szachownic elektronicznych jest zmonopolizowany przez holenderską firmę DGT znaną także z produkcji najlepszych zegarów elektronicznych

¹⁴Przez gniazdo COM

¹⁵W czasie namysłu rywala wielu szachistów wstaje od swojej partii i spaceruje po sali gry

- Błędy szachistów

Doświadczeni szachiści bardzo rzadko popełniają błąd polegający na wykonaniu nieprawidłowego posunięcia¹⁶. Mimo to, zwłaszcza w przypadku ograniczonego czasu do namysłu, zdarzają się sytuacje, w której zawodnik wykona nielegalne posunięcie¹⁷. Spora część programów transmitujących partie nie bierze pod uwagę takich sytuacji. Powoduje to, że dalsza część partii jest odrzucana przez aplikację jako nieprawidłowa. Skutkuje to nie tylko zawieszeniem transmisji, ale także nie przekazuje żadnego komunikatu widzom, dlaczego stan szachownicy się nie zmienia;

- Błędy sędziego/obsługi technicznej zawodów

Kolejną osobą, która może popełnić błąd jest sędzia lub specjalista od obsługi technicznej zawodów. Mając dostęp do panelu administrowania szachownicami może wpisać źle dane partii, może także wpisać zły wynik. Może oczywiście również źle skonfigurować sieć, przez co pliki nie będą prawidłowo umieszczane na serwerze.

- Błąd „późniejszej analizy”

Jest to bardzo częsty błąd, który musi być starannie usuwany przez sędziego. Bierze się on z zachowania szachistów, którzy po zakończonej partii często przesuwiają figury, aby sprawdzić potencjalny rozwój sytuacji. Komputer cały czas zapisuje zmiany na szachownicy jeśli nie otrzyma informacji, że obaj zawodnicy uzgodnili już słownie wynik partii. Co prawda teoretycznie unika się tego problemu poprzez specyficzne ustawienie królów zaraz po zakończeniu partii¹⁸, ale w ferworze walki szachiści bardzo często o tym zapominają. Analizy po zakończeniu partii są bardzo często algorytmicznie nierozróżnialne od samej partii, zatem sędzia musi ręcznie, na podstawie papierowych zapisów partii, zweryfikować jej zakończenie. Kolejny problem tego samego rodzaju wynika, jeśli w końcowej pozycji postawienie króla na jednym z wyróżnionych pól jest możliwe. Wówczas ustawienie króla w celu wysłania informacji o wyniku zostanie potraktowane jako zwyczajne posunięcie królem.

2.9 Silniki szachowe

Silniki szachowe są programami, które jako dane wejściowe otrzymują pozycję, jako rezultat zwracają zaś ocenę pozycji wraz z propozycjami najlepszych posunięć. Nie należy jednak podchodzić do nich w sposób jaki podchodzi się do zwyczajnych programów - ich wynik w zasadzie nigdy nie jest rozstrzygający i pewny, a ich działanie może być pod względem praktycznym nieskończone. Bierze się to z olbrzymiej ilości możliwych kontynuacji z danej

¹⁶Amatorzy dla odmiany rzadko grają na szachownicach elektronicznych, gdyż jest to drogi sprzęt i z tego powodu jest używany niemal wyłącznie na prestiżowych zawodach

¹⁷Najczęściej jest to wykonanie nieprawidłowej roszady, choć zdarzają się też inne możliwości jak przypadkowe wykonanie ruchu złą figurą

¹⁸Biały król na e4, czarny na d5 w przypadku wygranej białych, biały król na e4, czarny król na e5 w przypadku remisu oraz biały król na d4, czarny na e5 w przypadku wygranej czarnych

pozycji. Mimo iż szachownica posiada zaledwie 64 pola, a każda z figur posiada ograniczone pole działania przestrzeń możliwych sekwencji ruchów jest olbrzymia.

Liczba wszystkich możliwych partii szachowych jest skończona. Wynika to jednoznacznie z trzech obserwacji:

1. Ilość figur na szachownicy może się jedynie zwiększać¹⁹
2. Każdy pionek ma skończoną możliwą liczbę posunięć - może wykonać maksymalnie sześć ruchów
3. W przypadku zagrania 50 posunięć bez bicia lub posunięcia pionkiem jest ogłaszany remis.

Z punktów 1. i 2. wynika, iż ilość bić i posunięć pionkiem jest ograniczona do $30(zbić) + (2 \cdot 8 \cdot 6)$ (możliwe posunięcia wszystkich pionków), czyli łącznie możemy mieć maksymalnie 126 podobnego typu posunięć, co w połączeniu z punktem trzecim daje nam maksymalnie $50 \cdot 126 = 6300$ posunięć w jednej partii szachowej, co udowadnia, że szachy są grą skończoną.

Mimo udowodnienia faktu, że każda partia szachowa musi się kiedyś zakończyć, trzeba być świadomym jak olbrzymia przestrzeń możliwych kontynuacji. Shannon szacuje się, że możliwych pozycji jest około 2^{43} [2], zaś Tromp udowodnił górną granicę na liczbę możliwych partii na 2^{155} [3]. Jest to oczywiście poza zasięgiem zdolności obliczeniowej komputerów. Nawet jeśli podejdziesz się do sprawy bardziej praktycznie, eliminując za pomocą heurystyk jawnie błędne kontynuacje, to ilość możliwości jest wyzwaniem nawet dla współczesnych procesorów. Mimo iż komputery w chwili obecnej bez problemu pokonują najlepszych szachistów świata²⁰, to nie należy zwieść się iluzji o ich wszechpotędze. Programy osiągają przewagę nad ludźmi dlatego, że nie popełniają błędów i cały czas grają na równym, bardzo wysokim poziomie.²¹ Cały czas jednak istnieją wyzwania, z którymi poradził sobie umysł ludzki, a maszyna wciąż nie potrafi.

Historia silników szachowych sięga lat 50-tych ubiegłego wieku, jednak dopiero w latach 80-tych maszyny zaczęły prezentować poziom, który pozwalał im na nawiązanie walki z profesjonalnymi szachistami. Lata dziewięćdziesiąte były czasem walki o supremację między ludźmi i maszynami, zaś pierwsza dekada obecnego stulecia doprowadziła do sytuacji, że człowiek jest już bez szans w pojedynku z maszyną. Prawda jest jednak taka, że podstawowe zasady programowania silników szachowych pozostały podobne do tych, które rozwijali uczeni kilkadziesiąt lat temu.

Istnieją dwa podstawowe typy filozofii tworzenia silników szachowych. Pierwsze działają korzystając z algorytmów *brute-force*, drugie zaś dokonują wstępnej selekcji możliwych

¹⁹Promocja może zwiększyć jakość figur, jednak nie zwiększy ich ilości

²⁰Dekadę temu Garri Kasparow toczył ciężkie boje z superkomputerem Deep Blue, w chwili obecnej pokonałby go zapewne komercyjny program uruchomiony na zwykłym komputerze osobistym

²¹Szachy są grą błędów. Najlepsi szachiści świata także popełniają wiele błędów - mniej jednak niż ich konkurenci

posunąć. Co może dziwić, przez wiele lat programy z pierwszej z tych grup osiągały lepsze wyniki. Dlaczego tak się działo? Odpowiedź jest prosta - bardzo trudno dokonać poprawnej selekcji posunąć. Nawet jeśli w 99% przypadków program dokona słusznej eliminacji, wciąż w wielu momentach partii nie będzie on rozpatrywał najsilniejszego posunięcia²². W takim przypadku zysk z eliminacji, czyli możliwość spojrzenia dalej w przyszłość, jest niwelowany przez ominięcie kluczowych wariantów. Drugi sposób działa w sposób przypominający ludzkie myślenie, jednak przez lata doprowadzało to do sytuacji, że programy korzystające z heurystyk selekcji posunąć były mniej konsekwentne niż programy pierwszego rodzaju, zaś wciąż zbyt słabo naśladowujące umysł profesjonalnego szachisty. Jednak koniec końców programistom udało się doprowadzić do sukcesu drugiej grupy programów.

2.10 Algorytmy stosowane w silnikach szachowych

Silniki szachowe przedstawiają partię szachową jako drzewo. Cytując profesora Andrzeja Kisielewicza[4] *"Początkowym wierzchołkiem drzewa (korzeniem) jest stan początkowy gry (wraz ze wskazaniem, na kogo przypada ruch; przypomnijmy, że gracz zaczynający nazywa się MAX). Następnikami korzenia są wszystkie stany, jakie mogą powstać po wykonaniu ruchu przez MAX-a (w szachach jest ich 20). W każdym z tych stanów ruch przypada z kolei na MIN-a i następniki wyznaczone są przez wszystkie możliwe ruchy MIN-a. Proces ten kontynuujemy, wyznaczając dla każdego stanu następniki, którymi są wszystkie stany, jakie mogą powstać z danego stanu po wykonaniu ruchu przez gracza, na którego przypada kolej ruchu"*.

Obliczenia działają w sposób typowy dla algorytmów MIN-MAX-owych. Na początku obliczane są wartości w liściach (w przypadku szachów 1,0,-1 jeśli zakładamy, że możemy przedstawić pełne drzewo szachów) Zawodnik o nazwie MAX stara się maksymalizować wartość w wierzchołku, podczas gdy MIN dąży do minimalizacji tej wartości. W ten sposób przesuwamy się od liści aż do korzenia, który jest końcową wartością działania algorytmu. Jest to jednak teoretyczny przykład, gdyż nie jesteśmy w stanie stworzyć tak wielkiego drzewa.

[4] *"Ta sytuacja nie zraziła badaczy AI (sztuczna inteligencja przyp. aut.), a raczej zdołowała, pokazując, że tu jest właśnie miejsce na wprowadzenie sztucznej inteligencji. Komputer(program) powinien być zdolny do oceny aktualnej sytuacji, do stwierdzenia czy przewagę (lepsze widoki na zwycięstwo) mają białe, czy czarne. Następnie mogłby zanalizować różne sytuacje kilka lub kilkanaście ruchów naprzód i wybrać najbardziej obiecującą"*

Silnik szachowe działają właśnie na zasadzie podobnej opisanej w powyższym akapicie. Twórcy rezygnują ze stworzenia pełnego drzewa szachów, budują jednak strukturę na kilkanaście pólruchów naprzód i dając użytkownikowi częściowe wyniki obliczane za pomocą algorytmu MIN-MAX, gdzie wartości liścia wylicza się specjalnymi funkcjami ewaluacyjnymi. Funkcje te uruchamiane są dla każdej pozycji końcowej i z użyciem różnych kryteriów (tworzonych bardzo często przez szachistów) liniowo wyliczają wartość pozycji. Jest to

²²Selekcja nie tylko odbywa się w korzeniu obliczeń, ale w większości jego potomnych węzłów również

wartość orientacyjna, gdyż stworzenie dobrej funkcji ewaluacyjnej jest wyjątkowo trudnym zadaniem. Warto zwrócić uwagę, że algorytm tworząc liść od korzenia od razu wylicza wartość funkcji dla późniejszych węzłów, gdyż każdy wierzchołek kiedyś był liściem.

Aby uzyskać dobre praktyczne wyniki trzeba jeszcze bardziej poluzować założenia. W sytuacji przedstawionej wyżej komputer jest w stanie doliczyć maksymalnie do kilkunastu posunięć wprzód. Są pozycje, w której to jest niewielka wartość. W tym przypadku może pomóc metoda nazwana "odcinaniem alfa-beta" (alpha-beta pruning). Wykorzystuje ona fakt, że jesteśmy w stanie dynamicznie podczas tworzenia drzewa wykazać, że dana gałąź nie rokuje szans na wygenerowanie rozwiązania. Wówczas algorytm MIN-MAX może uciąć tę gałąź, zmniejszając liczbę węzłów do przeszukania. Jest to poważne usprawnienie standardowego algorytmu MIN-MAX, tym bardziej, że w pozycji szachowej bardzo często jest wiele ruchów, które można niemal automatycznie odrzucić. Dzięki odcinaniu możemy nie interesować się nie rokującymi nadziei na wynik gałęziami i wykorzystać zasoby do przeliczania tych szansownych kontynuacji.

W ostatnich latach silniki szachowe zostały wzbogacone o dodatkowe elementy. Z jednej strony dodano do nich programy zawierające wiedzę dotyczącą debiutów i końcówek, z drugiej w samych funkcjach oceniających dodano podprogramy. Służą one do pokonywania słabości algorytmu MIN-MAX - problemu dalekiego horyzontu oraz rozpoznawania twierdź. Problem horyzontu następuje wówczas, jak pozycja ma jasną i stabilną ocenę, jednak powód tej oceny uwidoczni się dopiero za kilkanaście ruchów. Problem twierdź był natomiast zgoła odwrotny - komputer nie potrafił zauważyć, że pomimo bycia stroną silniejszą nie potrafi on wzmocnić pozycję tak, żeby osiągnąć zwycięstwo. Doprowadzało to do kuriozalnych sytuacji, kiedy program nakazywał chodzić w kółko nie mogąc przeprowadzić konstruktywnego planu, lecz cały czas uznawał swoją pozycję za wygraną. Obecne programy szachowe już radzą sobie z tym zadaniem.

2.11 Universal Chess Interface i xBoard

Aby programy mogły korzystać z silników szachowych konieczne było ustalenie wspólnego interfejsu. W chwili obecnej standardem jest stworzony w 2000 roku przez Rudofa Hubera i Stefana Meyear-Kahlana, autorów silnika Shredder. Różni się ona od starszego rozwiązania, xBoard, lepszym wsparciem dla dodatkowych programów szachowych - książek debiutowych oraz tabeli końcówek.

Aby silnik spełniał implementował UCI musi wspierać komendy wydawane z interfejsu do silnika oraz z silnika do interfejsu.²³ W przypadku komend przekazywanych z interfejsu (graficznego lub tekstowego) są to w większości polecenia dotyczące analiz - zmień ustawienie, ustaw pozycję, rozpocznij i zakończ analizę. W drugą stronę silnik zwraca do interfejsu najlepsze posunięcie oraz informacje o pozycji. W chwili obecnej zdecydowana większość nowopowstających silników wspiera właśnie standard UCI.

XBoard jest starszym rozwiązaniem, które starciło w ostatnich latach na popularności, jednak jest wciąż rozwijane - także dlatego, że wspiera także gry podobne do szachów

²³Specyfikacja UCI dostępna jest pod adresem <http://wbec-ridderkerk.nl/html/UCIProtocol.html>

jak xiangqi (szachy chińskie), shogi (szachy japońskie) czy makruk (szachy tajskie). Część współczesnych silników wspiera nie tylko UCI, ale także xBoarda.

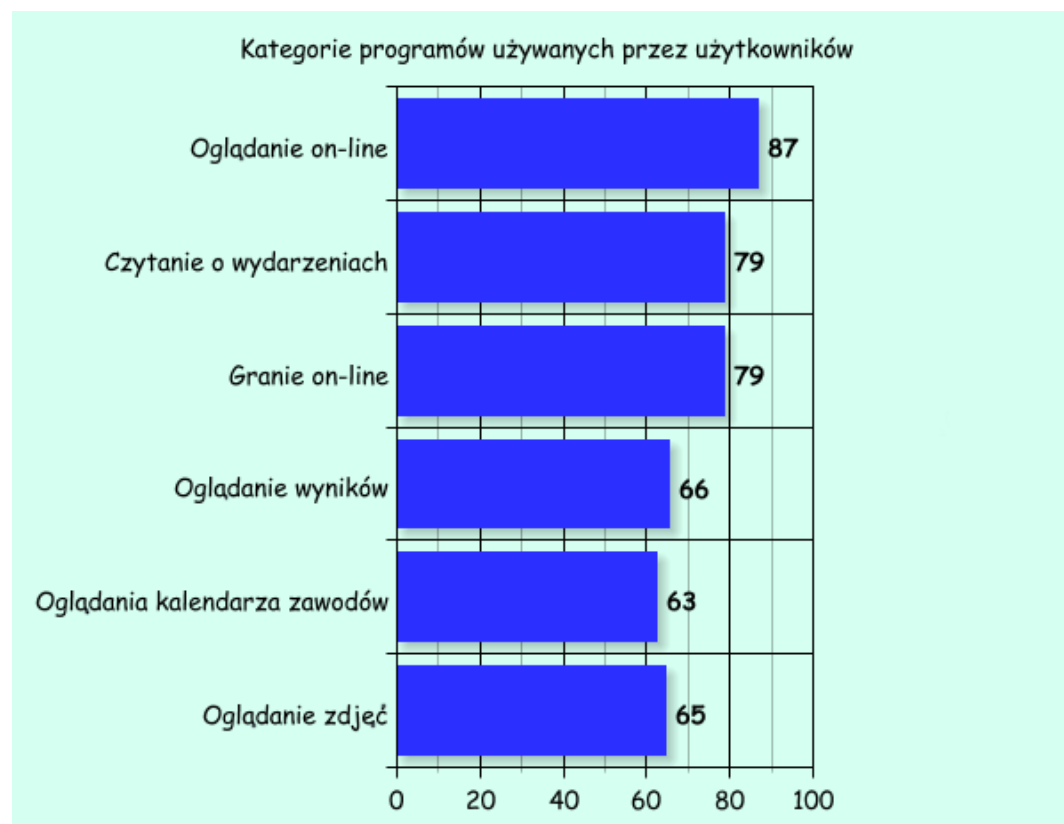
3 Ewaluacja dotychczasowych rozwiązań

Programy służące do przeglądania partii szachowych istnieją już od dziesięcioleci. Profesjonalni szachiści wraz ze sporą liczbą amatorów korzystają z całej gamy obecnych na rynku produktów. Niemal każdy szachista, który wyszukuje w Internecie informacje o szachach korzysta także z programów realizujących inne funkcje.

3.1 Ankiety na temat istniejących rozwiązań

Ocenienie już istniejących rozwiązań jest kluczowym aspektem tworzenia nowego oprogramowania. Przy dziesiątkach istniejących programów szachowych niemal każdy miłośnik królewskiej gry korzystał z kilku z nich. Umieściłem na stronie internetowej sondę, w której wzięło udział 139 użytkowników.

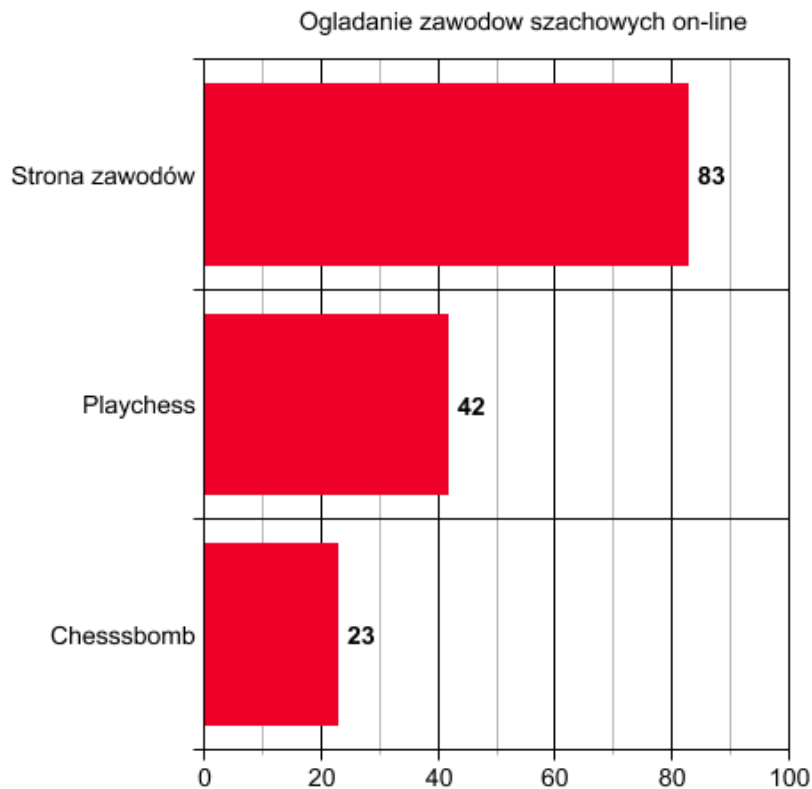
Pierwsze pytanie dotyczyło czynności szachowych wykonywanych przez dane osoby na Internecie.



Z wyników tego pytania można wnioskować, że większość szachistów-użytkowników Internetu korzysta z szerokiego spektrum funkcjonalności szachowych. Prawie wszyscy korzystają z możliwości, jaką dają im transmisje internetowe na żywo. Bardzo duży odsetek

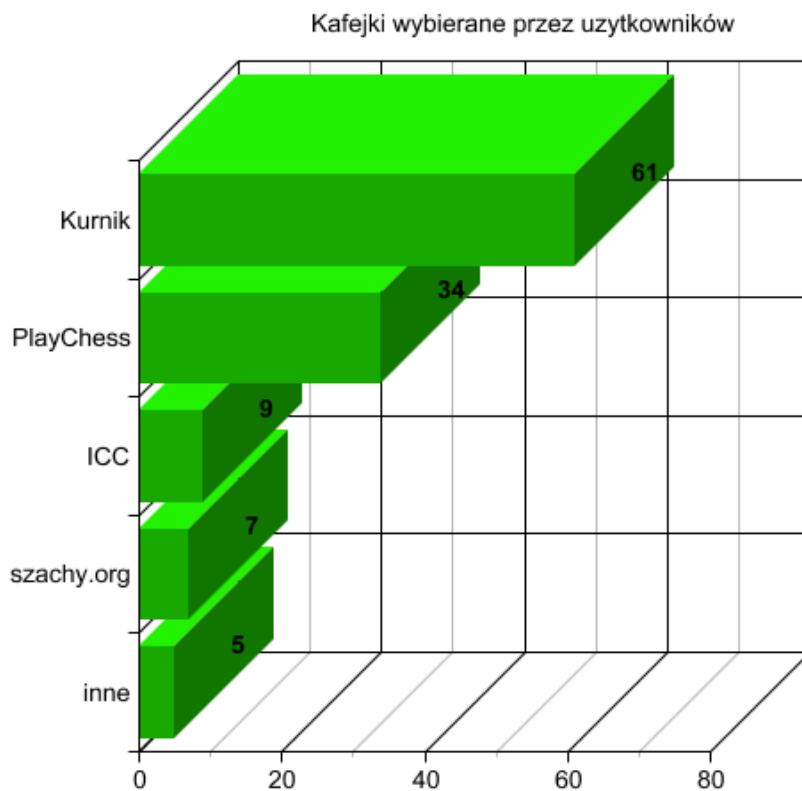
osób dodatkowo czyta informacje o turniejach oraz uczestniczy w rozgrywkach internetowych.

Kolejne moje pytanie dotyczyło informacji na temat sposobu oglądania partii on-line.

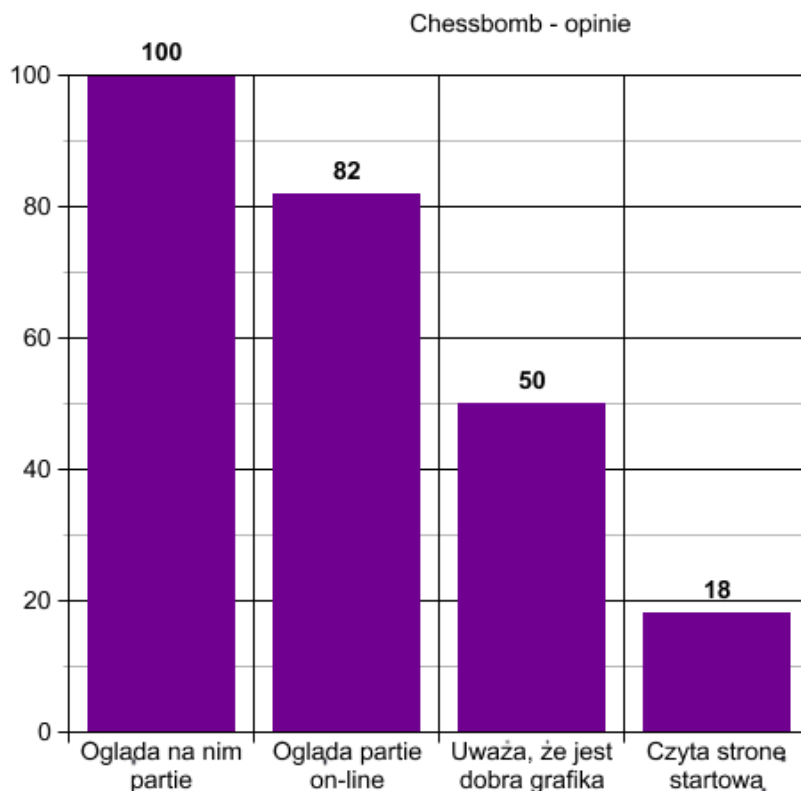


Widać, że tutaj również większość użytkowników postępuje podobnie. Zdecydowana większość widzów śledzi partie bezpośrednio na stronie organizatora, zaś niemal połowa poprzez kafejkę internetową Playchess. Zaledwie $\frac{1}{4}$ badanych korzysta z najpopularniejszego serwisu wyświetlającego partie z różnych turniejów. Można z tego wysunąć dwa wnioski. Pierwszy, że użytkownicy nie są przyzwyczajeni do żadnej witryny pokazującej szachy oraz to, że są dość elastyczni i korzystają z więcej jak jednego sposobu na oglądanie partii.

Kolejne pytanie dotyczyło kafejek szachowych, na których polscy internauci rozgrywają swoje partie



Widać, że Polacy wybierają polski serwer kurnik.pl. Wyprzedza on komercyjne serwisy Playchess czy ICC ale to w pewnym sensie nie dziwi - kurnik jest darmowy, zaś gra na dwóch wyżej wymienionych serwerach kosztuje około 100zł w skali roku.



3.2 Kafejki szachowe

Od samego początku funkcjonowania Internetu ludzie uznali, że ciekawym sposobem na wykorzystanie tej technologii jest rozgrywanie partii szachowych na odległość. Już w latach 80-tych ubiegłego wieku powstały pierwsze technologie i protokoły pozwalające na rozgrywki poprzez sieć. Od początku obecnego wieku jest to rozrywka coraz popularniejsza i skupiająca ludzi z całego świata. Różne kafejki internetowe oferują użytkownikom możliwość gry w szachy²⁴ zarówno z ludzkimi rywalami, jak i przeciwko komputerom szachowym. Użytkownicy posiadają do wyboru opcje wyszukiwania przeciwnika wg ustalonych przez siebie kryteriów. Preferencje graczy są różne - niektórzy chcą zagrać w ciągu godziny jedną partię, inni w tym czasie chcą ich skończyć kilkanaście. Cechą wspólną dla każdego użytkownika jest jednak praktyczne podejście do gry - muszą mieć możliwość sprawnego rozgrywania partii szachowej, podczas której rzadko przełączają się między oknami przeglądarki czy też odchodzą od komputera.

Badania dotyczące kafejek szachowych przeprowadziłem na podstawie najpopularniejszego w Polsce serwisu na platformie kurnik.pl. Mimo iż profesjonalni szachiści omijają ten serwis z daleka, tysiące mniej zaawansowanych graczy spędza na nim godziny swojego czasu. W ramach badań przeprowadziłem ankietę, w której wzięło udział kilkudziesięciu zawodników korzystających z kafejki. Pierwszym wnioskiem z wyników ankiet jest fakt, iż

²⁴A czasem także w gry podobne do szachów

szachiści niezbyt przejmują się szatą graficzną interfejsu. Dochodzi tutaj do paradoksu, że szachiści korzystający z tej aplikacji w dużych ilościach uznają ją za estetyczną, natomiast każda osoba z zewnątrz świata szachowego wyraźnie krytykowała wygląd. Jest to dość ważna obserwacja, gdyż można z niej wyciągnąć wnioski przy projektowaniu grafiki serwisu. Nie oznacza to, że sprawę wyglądu można całkowicie pominąć, jednak nie należy kłaść na nią najwyższego priorytetu. Ważną kwestią, o która anietowani zostali zapytani, był sposób oznaczenia ostatniego posunięcia. Większość osób uznała, że sposób w jaki na kafejce kurnik jest zadowolający, jednak istnieje także spora liczba osób, które nie było zadowolone z prezentowanego rozwiązania. Inne serwisy rozwiązują ten problem w rozmaite sposoby. Na serwisie ICC korzystającym z programu BlitzIn ostatnie posunięcie oznaczone jest czerwonymi obwódkami wokół dwóch pól²⁵. Podobny system jest używany w serwisie Chessbomb, o którym będę pisał w dalszym rozdziale. On także używa obwódek wokół pól, jedynie zaznacza je kolorem niebieskim. Kafejka Kurnik funkcjonuje w podobny sposób, jednak zamiast obwódek zmienia kolor całego pola na nieco ciemniejszy. Zupełnie inną filozofie prezentuje serwis PlayChess, który ostatnie posunięcie zaznacza za pomocą kolorowej strzałki prowadzącej od pola początkowego do końcowego.

Kwestia zapisu partii nie jest aż tak istotna, jak mogłoby się to wstępnie wydawać. Użytkownicy po skończonej partii rzadko zastanawiają się, gdzie popełnili błąd, częściej zaś siadają od razu do następnej rozgrywki. Wśród ponad sześćdziesięciu uczestników ankiety tylko jedna osoba zwróciła uwagę na to, że w podstawowym wyglądzie okna nie ma zapisu. Pozostałe kafejki udostępniają zapis w podstawowym widoku, jednak najwyraźniej rozwiązanie przyjęte w kurniku spotyka się z aprobatą graczy.

Jako główną zaletę kafejki kurnik przedstawiana często była duża liczba osób dostępna do rozgrywki. Wynika z tego, że wskazanym jest, aby użytkownicy także mieli kontakt z innymi osobami korzystającymi ze strony. W przypadku kafejek szachowych, gdzie główną ideą jest rozgrywanie partii z innymi użytkownikami jest to bardziej oczywiste niż na pozostałych serwisach, jednak kontakt z innymi szachistami wydaje się wskazany dla każdej witryny internetowej.

Wgląd do świata kafejek szachowych jest istotny, nawet jeśli nie są one podobne do aplikacji służących do oglądania partii na żywo. Mają inne cele, inne priorytety i inne sposoby przedstawiania rozgrywki. Kafejka musi zapewnić użytkownikowi wygodę w znalezieniu dopasowanego przeciwnika, musi działać w czasie rzeczywistym i dostarczać bardziej zabawę niż edukację. W kafejce szachowej dużo ważniejsze jest przedstawienie czasu i ostatniego posunięcia, mniej ważny jest zaś dostęp do zapisu posunięć. Mimo to sposoby oznaczania szachowni i figur są wspólne dla obu zagadnień i dlatego wiedza z rozwiązań występujących w kafejkach szachowych można wykorzystać podczas tworzenia innych serwisów.

3.3 Programy szachowo-bazodanowe

Profesjonalni szachiści bardzo często korzystają z programów pomagających im zarówno w treningu, jak i wyszukiwaniu informacji o przeciwnikach. Jest to bardzo dobra grupa

²⁵Startowym i końcowym polem figury która wykonała ostatnie posunięcie

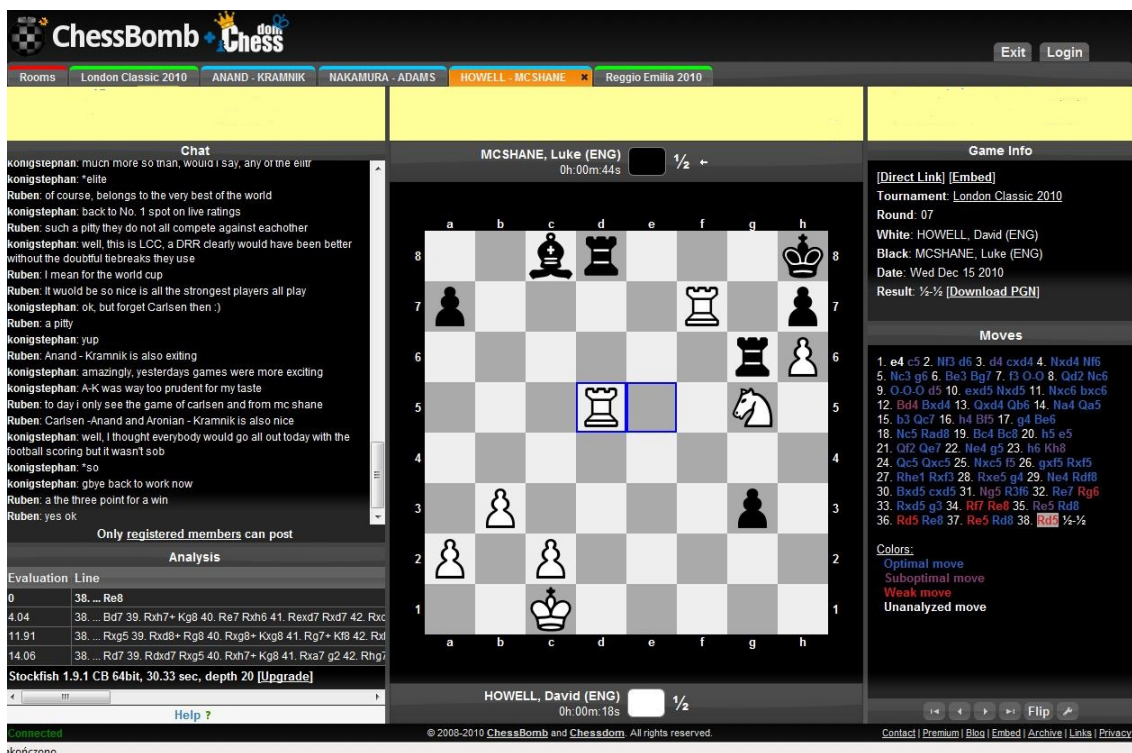
użytkowników do badań, gdyż najczęściej używają oni wybranych produktów w dłuższym okresie trwania swojej sportowej kariery. Ostatecznie przekłada się to na tysiące godzin spędzonych nad aplikacją, której zalety i wady poznają od podszewki. W przypadku tych programów sprawa się ma podobnie jak z kafejkami szachowymi - realizują one inne cele niż aplikacje służące relacji na żywo, jednak część rozwiązań może być wykrozystana także w przeglądarce partii. Najważniejszą cechą tych programów jest wspieranie wyszukiwania partii po kluczowych danych (czyli czynność zupełnie niespotykaną zarówno w kafejkach szachowych, jak i w programach do przeglądania partii). Dodatkowo konieczną funkcją jest zaimplementowanie interfejsu do obsługi silników szachowych²⁶.

Obliczenia przeprowadzane przez silniki szachowe są bardzo kosztowne dla zasobów komputera - zarówno pod względem zużycia procesora, jak i pamięci. Z tego powodu każdy program szachowy włącza silnik dopiero na wyraźne żądanie użytkownika. Doświadczony szachista najlepiej wie, w którym momencie partii należy włączyć program analizujący.²⁷ Programy po włączeniu pokazują kilka najlepszych według nich opcji dla danej pozycji²⁸ posortowanych względem oceny końcowej wyliczonej algorytmem MIN-MAX. Parametry uruchomienia silnika mogą podlegać modyfikacjom. Można modyfikować wielkość tablicy haszującej²⁹, można systemowo przydzielić procesowi wyższy priorytet. Wiadome jest, że z czasem to drzewo będzie miało tak wiele liści, iż później obliczenie każdego nowego półposunięcia będzie trwało mnóstwo czasu. Istnieją jednak sytuacje, gdy warto pozostawić dane obliczenia nawet na kilka-kilkanaście godzin. Drużyny narodowe niektórych krajów, aby móc wzmocnić siłę obliczeń zainwestowały w kilkunastordzeniowe komputery przenośne, tylko po to, aby uzyskać przewagę nad konkurentami.³⁰

W przypadku programów szachowych kluczowe znaczenie ma zapis partii. Jako że służą one do nauki, przebieg dotychczasowych ruchów jest niemal zawsze kluczowy dla edukacji. Przy przeglądaniu rozgrywki obok szachownicy zawsze wyraźnie widać zapis dotychczasowych ruchów. Więcej, zazwyczaj jest on tak ułożony, by nie trzeba było przewijać aby zobaczyć całą partię. Jest on zwyczajowo interaktywny - kliknięcie kursorem na posunięcie powinno automatycznie przenieść ekran partii do momentu, w którym ruch został wykonany.



Rysunek 2: Interfejs graficzny dla silnika Rybka4



Rysunek 3: Serwis ChessBomb - widok na szachownicę

3.4 Chessbomb - wady i zalety serwisu

Serwis Chessbomb powstał i został udostępniony użytkownikom na początku 2009 roku. Służy on do przeglądania on-line partii szachowych **wraz z wbudowanym modulem oceniającym za pomocą silnika szachowego aktualną pozycję**. Owa funkcjonalność jest nowością - wcześniejsze przeglądarki on-line nie oferowały natychmiastowej analizy pozycji, co zmuszało użytkowników do korzystania ze swoich lokalnie zainstalowanych programów. Trzeba od razu podkreślić, że wcześniejsze rozwiązanie posiadało jedną przewagę nad sposobem udostępnionym przez ChessBomb. Użytkownik posiada możliwość poświęcenia na wyliczenie oceny dla danej pozycji całą moc swojego procesora i tyle czasu, ile on uzna za konieczne. Dlatego analizy dokonywane lokalnie są nie tylko wiarygodniejsze, ale także czasem udowadniają, że ocena po stronie serwera była błędna. Jeśli tworzymy aplikację, która ocenia pozycję po stronie serwera musimy się z tym pogodzić - nasze analizy i tak będą statystycznie dużo lepsze niż „ludzkie przemyślenia”. Choć w czysto teoretycznym sensie nie będą one idealne, w praktyce mają duże zastosowanie. Jakże są zatem zalety uruchamiania silnika szachowego po stronie serwera?

- Obliczenia dostępne są zazwyczaj szybciej, niż w przypadku samodzielnego obsługiwanego programu grającego (zwłaszcza jeśli śledzi się więcej jak jedną partię). Wypatrywanie nowych posunięć, wprowadzanie nowych posunięć, włączanie i wyłączanie silnika - każda z tych operacji jest nie tylko irytująca dla użytkownika, ale także i czasochłonna.
- Dużo łatwiej jest zarządzać obserwowaniem analiz, nie trzeba przełączać okienek systemowych - wszystko mamy pod jednym widokiem
- Unika się błędów związanych ze złym wprowadzeniem posunięć (każdy krok pomiędzy wykonaniem posunięcia przez szachistę a wyświetleniem być może na drugim krańcu świata analizy tej pozycji jest zautomatyzowany)
- Można korzystać z analiz, nawet jeśli nie posiada się programu grającego³¹

Jako pierwsze rozwiązanie tego typu ChessBomb nie jest pozbawiony wad, jednak testy przeprowadzone z użytkownikami wykazały, iż większą część czynności wykonuje się na

²⁶Każdy szachista ma swój ulubiony silnik. Niektórzy używają różnych programów do oceny innych aspektów gry

²⁷W naszej aplikacji nie będziemy mieli takiego luksusu - wszystko będzie się odbywało automatycznie

²⁸Czasem zdarza się tak, że kilka ruchów jest równorzędnie dobrych. W innych przypadkach może się okazać, że różne posunięcia mogą doprowadzić do tej samej pozycji, a co za tym idzie oba warianty będą jednakowo ocenione

²⁹Aczkolwiek jest to bardzo nieefektywny sposób. Wg Vasika Rajlicha, twórcy programu Rybka, podwojenie wielkości tablicy haszującej powoduje wzrost siły gry na poziomie 5 punktów ELO (obecnie komputery osiągają ranking rzędu 3200)

³⁰Jeszcze dalej posunął się Bułgar, Veselin Topałow. Przygotowując się do meczu o Mistrzostwo Świata korzystał z najszybszego komputera w swoim kraju udostępnionego przez sofijski bank

³¹Silniki szachowe są bardzo często bezpłatne. Inaczej jest z programami implementującymi interfejs do tych silników

ChessBomb don't Chess

Rooms | London Classic 2010 | ANAND - KRAMNIK | NAKAMURA - ADAMS | HOWELL - MCSHANE | **Reggio Emilia 2010** | Exit | Login

Room Info

Standings

#	Name	ELO	Pts	S-B
1	Gashimov Vugar	2733	6	26.5
	Vallejo Pons Francisco	2698	6	23.75
3	Movsesian Sergei	2721	5	21.75
	Navara David	2708	5	19.25
	Ivanchuk Vassily	2764	5	19
6	Caruana Fabiano	2709	4.5	19.5
	Onischuk Alexander	2683	4.5	19.25
8	Morozevich Alexander	2700	4	17.25
9	Short Nigel	2680	3.5	13.75

Chat

pregnantour: when will tournament start?

Bacchante: lol it just writes when it will start in the first line above your post and you ask when its gonna start.)

siamesedream: Vugar won tournament?

Zebiakob80: when next game to start?

JKnight: Never

JKnight: It's over

Zebiakob80: thanks !!!

km_nikoleav: so vugar won it, right?

cafe4stream: yes, in spite of the tie with Vallejo Pons Francisco

siamesedream: 'Gashimov won the title on tiebreaks over Vallejo Pons'

siamesedream: http://2.bp.blogspot.com/_Q0ITIPVTG40/TSZ_BdW4I0U/AAAAAAAAABk6o/GTmWnXyLW/s1600/Players%252BStaff.JPG

Bacchante: lol @ chucky

Only registered members can post

[Help ?](#)

Games

Round	White	ELO (W)	Black	ELO (B)	Result
▼ 09					
	Godena Michele		Caruana Fabiano		½-½
	Ivanchuk Vassily		Morozevich Alexander		1-0
	Movsesian Sergei		Vallejo Pons Francisco		½-½
	Navara David		Short Nigel		1-0
	Onischuk Alexander		Gashimov Vugar		½-½
► 08					
► 07					
▼ 06					
	Caruana Fabiano		Vallejo Pons Francisco		½-½
	Gashimov Vugar		Ivanchuk Vassily		1-0
	Morozevich Alexander		Onischuk Alexander		0-1
	Navara David		Godena Michele		1-0
	Short Nigel		Movsesian Sergei		0-1
► 05					
▼ 04					
	Gashimov Vugar		Godena Michele		1-0
	Ivanchuk Vassily		Movsesian Sergei		½-½
	Morozevich Alexander		Caruana Fabiano		½-½
	Navara David		Onischuk Alexander		½-½
	Short Nigel		Vallejo Pons Francisco		0-1
► 03					
► 02					
► 01					

© 2008-2010 ChessBomb and Chessdom. All rights reserved. [Contact](#) | [Premium](#) | [Blog](#) | [Embed](#) | [Archive](#) | [Links](#) | [Privacy](#)

Rysunek 4: Serwis ChessBomb - widok na listę partii

nim w sposób intuicyjny i nie przysparzający problemów. Serwis posiada dość niewielką liczbę funkcjonalności, jednak te, które są dostarczane, są relatywnie łatwe w użyciu nawet dla niedoświadczonego użytkownika. Takie wnioski wyniknęły zarówno z internetowych ankiet, jak i bezpośrednich testów z użytkownikami.

Wygląd interfejsu programu ChessBomb często uznawany jest za nieciekawý, tym bardziej że twórcy nie udostępnili żadnego sposobu jego zmiany. Za bardzo wygodny w użytkowaniu uznany jest system zakładek, za pomocą których można bez najmniejszych problemów przełączać się między oknami poszczególnych partii.

3.5 Cechy wspólne dotychczasowych rozwiązań oraz ich ocena

Choć każdy z przedstawionych dotychczas programów zajmował się szachami, to każdy z nich skupiał się na innym aspekcie rozgrywki. Wyniknęło z tego, że rozwiązania te nie mają wielu wspólnych elementów

1. **Obsługa i generowanie plików PGN** Zdecydowana większość programów szachowych potrafi zarządzać plikami PGN. Choć programy bazodanowe najczęściej operują na swoich własnych strukturach, to zawsze dodatkowo obsługują format PGN.

Jeszcze częściej zjawisko to widać w przypadku przeglądarek graficznych plików - w większości przypadków PGN to jedyny obsługiwany format plików. W przypadku kafejek szachowych sytuacja jest odwrotna - lepsze serwisy generują pliki PGN na podstawie rozegranych partii.

2. **Umieszczanie szachownicy w środku ekranu** Niezależnie od rozmaitych celów aplikacji, szachownica przedstawiająca rozgrywkę zawsze pokazuje się w centrum ekranu. Czy jest to kafejka szachowa, czy jest to baza partii czy witryna do oglądania szachów na żywo - szachownica zajmuje zawsze centralną pozycję na stronie. Dodatkowo warto zauważyć, że w przypadku gdy sama wielkość szachownicy nie jest modyfikowalna, to szachownica zajmuje między $\frac{1}{3}$ a $\frac{2}{3}$ szerokości okna (chyba że ekran jest wyjątkowo nieproporcjonalny - trzeba pamiętać, że szachownica zawsze jest kwadratem)
3. **Wyświetlanie białych figur w kolorze białym oraz czarnych w pozostałych kolorach** Choć istnieje wiele interfejsów graficznych przedstawiających partię szachową, kolory wybierane do oznaczania figur w każdym rozwiązaniu są standardowe. Białe figury najczęściej przedstawiane są w różnych odcieniach bieli, dla czarnych figur jest zaś większa dowolność. Mogą być to klasycznie czarne, mogą być ciemnobrązowe, ale zdarzają się także ciemnoniebieskie i ciemnozielone.
4. **Przemieszczanie figur z użyciem klawiatury** W dzisiejszych czasach coraz bardziej odchodzi się od korzystania z klawiatury w celach nawigacji. Programy szachowe stanowią jednak wyjątek. Standardem jest, że można przeglądać posunięcia korzystając z klawiaturowych strzałek. Jest to podyktowane wygodą - kiedy partia szachowa składa się kilkudziesięciu pól ruchów nikomu nie chce się tyle razy naciskać na ikonkę przewijania. Klawisz strzałki zaś można przytrzymać, co wyraźnie oszczędza czas użytkownika.
5. **Przedstawianie czasu** Podobnie sprawa się ma z przedstawianiem czasu. To dotyczy zarówno transmisji na żywo (gdzie podaje się czas, jaki pozostał zawodnikom), jak i partii rozgrywanych między internautami na kafejkach szachowych (gdzie pokazywany jest czas użytkowników). Wszędzie korzysta się ze standardu pokazywania, jak wiele czasu zostało zawodnikom. Jest to naturalne podejście (znow - takie samo jest w przypadku fizycznych zegarów szachowych) i każdy interfejs realizuje go w podobny sposób³². W tym przypadku można rozróżnić transmisje i grę na kafejkach, gdyż w transmisjach samo opóźnienie może być kilkudziesięciosekundowe³³, w kafejkach internetowych zaś nawet półsekundowe opóźnienie może być kluczowe³⁴. Dlatego też w kafejkach często podaje się czas z dokładnością do dziesiątych części sekundy, zaś przy transmisjach dokładność tego stopnia jest absolutnie niepotrzebna.

³²Nigdy nie podaje się czasu zużytego, gdyż w przypadku zakończenia partii nie ma on wpływu na żaden aspekt wyniku

³³Z powodów podanych w rozdziale dotyczących szachownic elektronicznych

³⁴Wiele osób, mnie włączywszy, grywa partie, w których na całość przeznaczona jest zaledwie jedna minuta

4 Projektowanie własnego interfejsu

Nadszedł czas, aby przejść od etapu oceniania gotowych rozwiązań do zaproponowania własnych. Moim założeniem jest, iż nie będę starał się wymyślić czegoś absolutnie nowatorskiego, a postaram się ulepszyć rozwiązania stosowane w serwisie ChessBomb. Zamierzam korzystać, podobnie jak ów serwis, z systemu zakładek reprezentujących turniej albo konkretną partię. W odróżnieniu od stosowanych w ChessBombie rozwiązań pozwolę użytkownikom na utworzenie kilku partii w jednej łączonej zakładce. Do uzupełnienia w zależności od tego, jak się uda

4.1 Rodzaje użytkowników

Nie sposób myśleć o programie jako o stuprocentowo automatycznej witrynie. Niektóre czynności z konieczności będą musiały być wykonywane przez administratora strony. Oczywiście należy dążyć do zminimalizowania nakładu pracy ludzkiej poprzez maksymalne ułatwienie tych czynności, jednak najprostsze operacje będą musiały być wykonywane przez administratora strony. Ogólnie planuję podzielić użytkowników na następujące kategorie:

- Administratorzy strony
- Moderatorzy treści
- Zarejestrowani użytkownicy strony
- Niezarejestrowani użytkownicy strony

Jest to dość typowy podział wśród witryn internetowych i będzie miał zastosowanie także przy moim projekcie. Zadanie administratora będą w zasadzie jedno - dodawanie turniejów. Aby móc to uczynić będzie udostępnione dla niego specjalne okno, w którym definiować będzie się wszystkie dane dotyczące danych zawodów. Oczywiście administrator będzie miał wszystkie uprawnienia użytkowników pozostałych kategorii.

Moderator strony będzie miał możliwość zarządzania użytkownikami witryny - będzie mógł on usuwać ich wpisy w oknie rozmów, będzie mógł także blokować czy też kasować konto użytkownika. Nie będą oni mieć dostępu do żadnych kluczowych dla wyświetlania witryny funkcji, będą jednak pełnić ważną rolę w momencie rozwinięcia systemu komentowania na bieżąco trwających partii.

Zarejestrowany użytkownik strony będzie miał możliwość korzystania z głównych funkcji strony - oglądania i komentowania partii. Odróżnia się go od niezalogowanego użytkownika, który nie ma możliwości wypowiadania się na stronie. Zalogowany użytkownik będzie także posiadał opcje edycji swoich danych osobowych, a także wysyłania wiadomości do innych użytkowników. Poprzez specjalny panel będzie mógł także skontaktować się z administratorami i moderatorami strony.

4.2 Przypadki i scenariusze użycia oraz papierowy interfejs

4.2.1 Scenariusze użycia

Scenariusze użycia są przydatnymi narzędziami podczas projektowania interfejsów. Pomagają one wyobrazić sobie kontekst, w jakim będzie używana aplikacja. Dzięki temu można lepiej zrealizować funkcje, tak aby przykładowym użytkownikom dobrze się z nich korzystało.

Przykładowe scenariusze użycia

Scenariusz #1

John Doe siedzi w swoim pokoju o godzinie 8:30 rano w niedzielę. Normalnie jeszcze by spał, ale jest relacja z Bardzo Ważnych Zawodów. Uruchomił stronę www, która to uruchomiła witrynę, na której prowadzona jest relacja na żywo. Na stronie głównej zauważył informację, że tego dnia nie będzie komentarza ludzkiego, jedynie automatyczny. John Doe uruchamia witrynę, która wyświetla mu listę wszystkich turniejów dostarczanych przez stronę internetową. Lista jest posortowana chronologicznie, więc wybiera z niej Bardzo Ważne Zawody. Pokazuje mu się lista aktualnie granych partii. Lista jest posortowana względem szachownic, jednak John wybiera opcję zmiany na sortowanie alfabetyczne. Wybiera z listy partię swojego ulubieńca, Seana Smitha, który gra z Andriejem Kałmukowem. W wyniku wybrania otwiera się nowa zakładka z partią Smith-Kałmukow. John Doe widzi, iż zostało wykonanych już 15 posunięć i białe zużyły 20 minut, czarne zaś 10. Z przebiegu partii wybiera pierwsze posunięcie i wybierając opcję pokaż następny ruch przechodzi do końca partii. Widzi ocenę pozycji po najlepszym ruchu Smitha i dwukrotnie klikając w wariant wyświetla go w postaci graficznej.

Scenariusz #2

Michael Collins siada wieczorem do komputera, aby zobaczyć wyniki z dzisiejszej rundy Bardzo Ważnego Turnieju. Uruchamia stronę internetową, po czym wchodzi do opcji logowania. Wpisuje swoje dane i bezproblemowo loguje się na serwer. Witryna wyświetla mu listę turniejów, po czym Michael wybiera Bardzo Ważny Turniej. Wyświetliła mu się lista partii z wpisanymi do niej wynikami. Michael otwiera zakładki z partiami z sześciu pierwszych szachownic i przegląda każdą z nich osobno. Po przejrzaniu partii zamyka zakładkę. Po przejrzaniu wszystkich partii pozostaje w zakładce turnieju i w „shoutboxie” pisze komentarz na temat rundy. Po wykonaniu tej czynności wylogowuje się i zamyka przeglądarkę.

Scenariusz #3

Jack Wilshire przychodzi do komputera około godziny 12. Nie włącza niczego, gdyż zostawił przeglądarkę włączoną poprzedniej nocy. Widzi, że może kliknąć na pole oznaczające nową rundę i wyświetli mu się lista partii. Niektóre partie się zakończyły i są wpisane ich wyniki. Wchodzi na partię z pierwszej szachownicy i widzi, że poza posunięciami jest w zapisie partii komentarz słowny. Jack odchodzi od komputera i wraca po kwadransie z gorącą kawą. Widzi natychmiast, że zostało zagranych kilka ruchów, bo mimo iż nie dotknął ani

myszki, ani klawiatury, pozycja się zmieniała. Po kolejnych pięciu minutach widzi wpisany wynik partii – zakończyła się remisem.

4.2.2 Przypadki użycia

Przypadki użycia ukazują interakcję między użytkownikiem a systemem, pokazując ścieżki za pomocą których użytkownik może osiągnąć wyznaczone cele i sposób, w jaki system realizuje wszystkie kroki pośrednie. Są one szczególnie przydatne podczas implementacji i późniejszej ewaluacji interfejsu - można porównać przypadki użycia z fantycznym sposobem dochodzenia do celów w naszym programie.

4.2.3 Papierowy prototyp

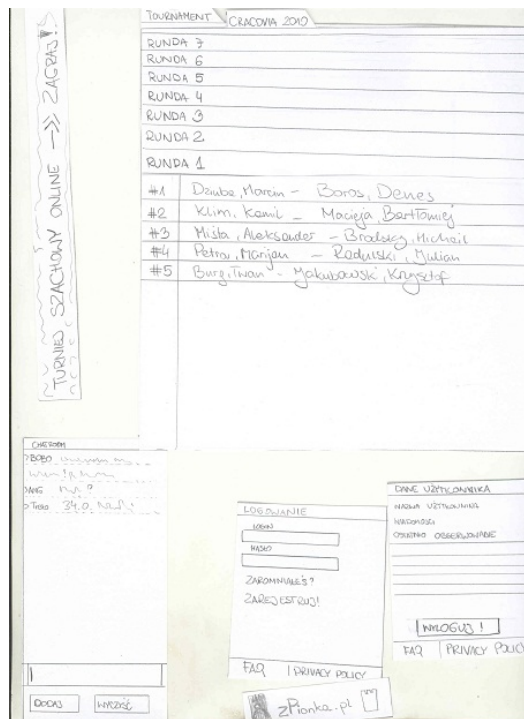
Papierowy prototyp, jak sama nazwa wskazuje, jest szablonem wyglądu interfejsu stworzonym za pomocą przyrządów biurowych. Mimo iż wydaje się to nieco nieprofesjonalnym podejściem, pozwala on na szybkie i mało kosztowne przetestowanie użyteczności interfejsu. Tworząc taki schemat twórca zupełnie odcina się od sposobu w jaki tworzy interfejs i skupia się na samych funkcjonalnościach. W wyniku temu można szybko znaleźć największe problemy dotyczące użyteczności programu.

Papierowy prototyp pokazuje wszystkie najważniejsze elementy interfejsu, pozwalając testowanej osobie zobaczyć, jaki układ informacji będzie prezentowany na stronie. Uczestnicy testów dostali listę zadań do wykonania, podczas gdy obserwoałem sposób i mierzyłem czas jaki zajmuje im dojście do określonego celu.

Lista i kolejność zadań była następująca:

1. Zalogować się na stronie, a później się wylogować.
2. Zalogować się na stronie, znaleźć listę turniejów, wybrać z niej najstarszy turniej i napisać komentarz na chatroomie.
3. Pozostając w tym turnieju znaleźć partię z pierwszej rundy graną na pierwszej szachownicy
4. Przewinąć partię do początku, a następnie ruch po ruchu ją przejrzeć

Uczestnicy testów byli zachęceni do wygłaszania swoich uwag i propozycji, a każde badanie było nagrywane, by można było odtworzyć przebieg badania. Nie było także ustalonego limitu czasu na wykonanie danej czynności. Jako grupę testową użyłem szachistów w wieku 15-40 lat - docelową grupę użytkowników programu. Byli wśród nich zarówno doświadczeni użytkownicy interfejsów szachowych, jak iniedoświadczeni szachiści



Rysunek 5: Papierowy prototyp interfejsu

4.2.4 Wnioski z testów na papierowym prototypie oraz ich wpływ na kształt interfejsu

4.3 Wygląd interfejsu

Papierowy prototyp nie testował ustaleń dotyczących niektórych aspektów grafiki przyszłej aplikacji. Powód tego jest prosty - kolory ze skali RGB czasem nie tak łatwo reprezentować za pomocą fizycznych przyrządów. I choć najlepiej pozwolić użytkownikom na wybór preferencji graficznych, trzeba wybrać kolory choćby dla dobra pierwszego wrażenia przyszłego użytkownika.

4.4 Dobór kolorów interfejsu

4.5 Rodzaje figur i szachownic

5 Implementacja programu

5.1 Technologia

Program zdecydowałem się pisać w coraz modniejszej w ostatnich latach technologii Ruby on Rails. Choć do korzystania w pełni z możliwości udostępnianych przez biblioteki musiałem także korzystać z języków Java i JavaScript oraz ze znaczników HTML, to rdzeń

aplikacji jest napisany w obiektywnym języku Ruby.

O wyborze technologii, w której implementowałem aplikację zadecydowały dwie rzeczy. Pierwszy argument był programistyczny: Ruby on Rails jest technologią zwinną (agile) wspierającą wzorzec MVC, który bardzo dobrze pracuje w przypadku serwisów webowych. Drugi argument był zupełnie inny: chciałem poprzez pisanie pracy magisterskiej nauczyć się nowej technologii. Wcześniej nie miałem do czynienia ani z językiem Ruby, ani z frameworkiem Rails.

5.1.1 Ruby on Rails

“Ruby on Rails jest przełomem w dziedzinie programowania aplikacji internetowych. Potężne aplikacje, których tworzenie do tej pory zabierało tygodnie czy miesiące, są teraz tworzone dosłownie w kilka dni.” -Tim O'Reilly, Założyciel O'Reilly Media³⁵

O produkcie często najlepiej świadczy renoma ich klientów. Duże międzynarodowe firmy nie korzystają ze złych rozwiązań ani w biznesie, ani w procesie tworzenia aplikacji. O sile frameworka Ruby on Rails świadczą między innymi tak potężne serwisy internetowe jak Twitter czy Groupon. Język Ruby powstał w roku 1995, stworzony przez Yukihiro “Matza” Matsumoto. Jest w pełni obiektywnym i typowanym dynamicznie językiem bazującym na wielu językach, takich jak CLU, Eiffel, Lisp, Perl, Python czy Smalltalk Ruby cechuje się dużą zwiezłością kodu, przy posiadaniu wciąż czytelnej składni. Co ciekawe, w Ruby można modyfikować każdą klasę, włączając w to nawet klasę string. Jedną z głównych zalet języka jest jednak framework Ruby on Rails, który jest świetnym narzędziem do tworzenia aplikacji webowych

Ruby on Rails powstał jako niezależny projekt stworzony głównie przez duńskiego programistę Davida Heinemeiera Hanssona. Ruby on Rails jest narzędziem do szybkiego tworzenia aplikacji webowych **opartych na architekturze Model-View-Controller** przy zachowaniu zasady DRY³⁶ oraz reguły Convention over Configuration³⁷. Jedną z większych zalet Ruby on Rails jest fakt, że posiada bardzo użyteczny moduł ActiveRecord oparty na ORM³⁸, który pozwala na tworzenie modeli w architekturze MVC niezależnych od faktycznej bazy danych. Kolejną zaletą tego frameworku jest dostępność świeżo rozwijanych i użytecznych bibliotek. Jest też, co warto podkreślić, bardzo prosta w nauce - sam nie znałem tej technologii w momencie rozpoczęcia pisanie pracy magisterskiej.

5.1.2 JRuby on Rails

Choć Ruby jest już bardzo rozwiniętym językiem, czasami przydaje się jednak możliwość wykorzystania kodu napisanego w innych językach programowania. Takim rozwiązaniem

³⁵O'Reilly Media to uznana amerykańska kompania medialna

³⁶Don't Repeat Yourself - unikaj powtarzania kodu

³⁷Minimalna początkowa konfiguracja zastąpiona gotowymi wzorcami

³⁸Object-Relational mapping

jest JRuby, który pozwala na uruchomienie kodu Ruby na JVM (wirtualna maszyna Javy) oraz na korzystanie z klas napisanych w Javie. Podobnie jest samo Ruby on Rails został on stworzony przez grupę programistów, którzy chcieli stworzyć jeszcze bardziej użyteczne narzędzie do tworzenia stron webowych. Dzięki bibliotekom JRuby można umieszczać strony napisane we frameworku Rails na urządzeniach przenośnych. Problemem z JRuby jest jedynie lekkie opóźnienie do klasycznego Ruby, najnowsze funkcje z frameworka Rails nie zawsze od razu działają pod JRuby.

5.1.3 Dodatkowe biblioteki

W programie wykorzystywałem gotowe biblioteki. Główną z nich jest **Chesspresso**³⁹, bardzo obszerna biblioteka napisana w Javie na licencji LGPL, służąca do parsowania i interpretowania plików szachowych o rozszerzeniu pgn, a także interpretowania i tworzenia notacji FEN. Implementuje ona większość przydatnych funkcji potrzebnych do zarządzania partiami szachowymi, jej jedynym mankamentem jest nieistniejąca dokumentacja, która wymaga od programisty przynajmniej niewielkiego zagłębienia się w kod biblioteki.

Kolejną biblioteką jest nieznacznie zmodyfikowana na potrzeby pracy JavaScriptowa **jChess**⁴⁰ oparta na znanej bibliotece jQuery, która wizualnie prezentuje partie szachową a także umożliwia nawigację po niej - wykonywanie posunięć zarówno do przodu, jak i do tyłu. Jest to funkcjonalność, która jest także zawarta w Chesspresso, jednak z uwagi na funkcjonowanie tej biblioteki w środowisku Javy problemem byłoby włączenie chesspressowej przeglądarki graficznej do webowej zawartości serwisu.

Kolejną biblioteką jest **paperclip**⁴¹ - plugin napisany w Rubym służący do wygodnego importowania plików na serwer, który można łatwo zintegrować z wymienionym przed chwilą jQuery.

5.1.4 Programowanie zwinne

We wstępie do rozdziału wprowadziłem termin **programowanie zwinne** (Agile software development). Co kryje się pod tym terminem? Pod nazwą "metodyki zwinne" kryje się grupa metodyk opartych na programowaniu iteracyjnym. Różni się ona od standardowych metod tworzenia oprogramowania dużo mniejszą formalnością i większą reakcją na potrzeby użytkownika. Jej zaletami są szybkość tworzenia działającego kodu i szybka reakcja na następujące zmiany, jej słabością jest jednak niemożność w zastosowaniu w przypadku dużych projektów programistycznych, gdzie kontakt między poszczególnymi programistami jest utrudniony a brak dokumentacji technicznej jest już realnym problemem.

"Rails skupia się na ludziach i interakcjach między nimi. Nie znajdziesz tu rozbudowanych pakietów narzędziowych, skomplikowanej konfiguracji, złożonych procesów. Są tylko małe grupki programistów, ich ulubione edytory, trochę kodu w Rubym. Rezultatem jest większa czytelność relacji; wyniki pracy programistów są natychmiast widoczne dla klienta. **Istotą**

³⁹<http://www.chesspresso.org/>

⁴⁰<https://github.com/bmarini/jchess>

⁴¹<https://github.com/thoughtbot/paperclip>

tego pomysłu jest interaktywność” [1]

Programowanie zwinne szczególnie nadaje się w przypadku stron internetowych, gdzie od samego początku pracy można współpracować z klientem, który bez posiadania specjalistycznej wiedzy może wyrazić swoją opinię na temat wyniku prac. Można wtedy także co każdą iterację wprowadzać modyfikacje do przyszłego planu tworzenia aplikacji, uwzględniając zarówno aspekty, które wyszły na jaw w trakcie dotychczasowego pisania kodu, jak i zdanie i opinie klienta aplikacji. Taka metodyka jest też bardzo dobra dla współpracy z klientem pod względem psychologicznym - regularnie możliwość do wzglądu w coraz bardziej **działający** produkt daje mu poczucie pewności co do przyszłości aplikacji.

W Ruby on Rails na każdym kroku natykamy się na elementy zwinne. Tabele bazy danych są tworzone za pomocą migracji, za ich pomocą także można je prosto modyfikować. W ten sposób zamiast jednego wielkiego pliku z definicjami dziesiątek tabeli mamy kilkadziesiąt plików, dzięki którym możemy prześledzić wstecz historię rozrastania się bazy. Implikuje to w oczywisty sposób fakt, że można tworzyć jedynie niezbędne w danej chwili tabele, zaś późniejsze struktury i ich powiązania można zostawić na moment, w którym rzeczywiście będą potrzebne. Konwencje nazewnictwa frameworka także wspierają projektowanie zwinne, gdyż w możliwie dużym stopniu niektóre funkcje i komendy przypominają język naturalny, co ułatwia zrozumienie kod bez konieczności przeglądania dokumentacji.

5.2 Struktura programu

Ruby on Rails odgórnie narzuca architekturę MVC i choć oczywiście istnieją sposoby, aby korzystać z innych architektur, całe wsparcie frameworka najskuteczniej działa dla tego właśnie schematu. Ruby on Rails domyślnie interpretuje, że dla każdego modelu istnieje jeden kontroler, natomiast widoki są ściśle powiązane z metodami kontrolera dzieląc takie same nazwy. Pliki modeli i kontrolerów są pisane w Ruby, natomiast pliki widoków w specjalnych plikach `html.erb`, które są połączeniem pliku `html` z kodem Ruby umieszczanym między odpowiednimi znacznikami. Przy żądaniu wyświetlenia pliku `html` najpierw uruchamiany jest kontroler, który komunikując się z modelem zwraca do widoku potrzebne wartości.

5.2.1 Funkcjonalności

Jak zdecydowana większość programów sieciowych głównym odbiorcą aplikacji będzie zwykły użytkownik, wydzielona część z nich będzie mogła jednak wykonywać nieznacznie więcej czynności. Uprawnienia w aplikacji działają hierarchicznie, czyli administrator może zrobić wszystko to samo co zwykły użytkownik, ale będzie miał dodatkowe funkcje dostępne

Użytkownik

1. Założenie konta
2. Zalogowanie się na konto

3. Wylogowywanie się z konta
4. Edycja danych konta
5. Wysyłanie i odbieranie prywatnych wiadomości od innych użytkowników
6. Komentowanie artykułów
7. Usuwanie własnych komentarzy
8. Przeglądanie listy turniejów, rund, partii
9. Przeglądanie partii zawodników
10. Dodawanie partii do listy obserwowanych oraz usuwanie z tej listy
11. Przeglądanie pojedynczej partii w oknie graficznym
12. Przeglądanie zapisu partii i nawigacja za jego pomocą do odpowiednich miejsc w rozgrywce
13. Przeglądanie odpowiednich wariantów dla obecnej pozycji
14. Dodawanie wiadomości dotyczących turniejów

Moderator

1. Dodawanie nowych partii, rund i turniejów
2. Usuwanie komentarzy innych użytkowników
3. Dodawania i usuwanie informacji z panelu komentatorskiego turnieju
4. Dodawanie i usuwanie artykułów

Administrator

1. Usuwanie innych użytkowników
2. Zmiana uprawnień innych użytkowników

5.2.2 Struktura bazy danych

Podstawowymi tabelami, w których będzie przechowywana największa liczba informacji będą te zawierające dane dotyczące różnych turniejów. Wyróżniam wśród nich cztery najważniejsze tabele - Turniej, Runda, Partia i Pozycja, gdzie na każdym kroku nie tylko występuje relacja "jeden do wielu", ale także te proporcje będą w miarę stałe. Stosunki między liczbą poszczególnych krotek można oszacować następującą równością $\text{Turniej} = 9 * \text{Runda} = 10 * 9 = 90$ $\text{Partia} = 10 * 9 * 40 = 3600$ Pozycja . Ogólny schemat bazy danych przedstawia następujący schemat. Większość pozostałych tabel służy do zarządzania użytkownikami, ich statusem, wysłanymi wiadomościami, komentarzami oraz innymi funkcjonalnościami dostępnymi dla użytkownika.



Rysunek 6: Wygląd po pierwszej iteracji

5.3 Pierwsza iteracja

W ramach pierwszej iteracji zostały zaimplementowane bazowe klasy dla całej aplikacji - moduł opisujący użytkownika oraz wspierający funkcję uwierzytelniania na stronie wraz z obsługą sesji a także grupa modułów opisujących partię szachową (klasy Tournament, Round, Game, Position, Player). Najważniejszą funkcjonalnością, która została dodana w tej iteracji był komponent wyświetlający partię szachową w formie graficznej, lecz równie ważne było zaprogramowanie niewidocznych dla użytkownika funkcji parsujących pliki z partiami (w formacie .pgn) i dodającym informacje do bazy danych. Metody przekształcające plik tekstowy w hierarchiczną strukturę bazy danych zostały stworzone przy pomocy biblioteki Chesspresso. Udostępnia ona funkcję, które wyciągają z pliku zarówno informacje opisowe partii, jak i odczytujące i sprawdzające poprawność posunięć. Biblioteka ta, mimo wszystkich swoich zalet, miała także jedną wadę - jako iż została napisana w Javie, nie było łatwego sposobu umieszczenia domyślnej przeglądarki chespresso do kodu html. Chcąc uniknąć umieszczania apletów javy zdecydowałem się na wprowadzenie kolejnej biblioteki - jChess. Przeglądarka graficzna partii po pierwszej iteracji była niemal identyczna jak domyślna szachownica stworzona przez komponent jChess. Wyświetlała ona partie, pozwalając za pomocą buttonów przesuwając partię do przodu i do tyłu. Zaimplementowane zostały możliwości wyświetlania informacji o turniejach, wynikach końcowych oraz rezultatach w poszczególnych rundach. Powstały także strony umożliwiające podgląd wyników poszczególnych zawodników.

Jako dodatek do pierwszej iteracji zostały dodane klasy i funkcje pozwalające użytkownikom na wysyłanie między sobą wiadomości prywatnych, pisanie artykułów oraz dodawanie do nich komentarzy. Powstały też okna, w których użytkownicy mogą wyświetlać na żywo swoje opinie dotyczące całych turniejów, ale i poszczególnych partii.

5.4 Ewalucja z użytkownikami po pierwszej iteracji

Po zakończeniu pierwszej iteracji nastąpiła faza ewaluacji ówczesnego stanu aplikacji. Badania przeprowadzałem głównie na szachistach, jednak brała w nich udział także osoba bez doświadczenia z korzystania z programów szachowych. Stosowałem zasadę, iż błędy były naprawiane w czasie między badaniami (o ile arbitralnie uznałem, że uwaga jest zasadna). Dzięki tej zasadzie pozostałe osoby odkrywały nowe błędy, których naprawienie ulepszało jakość program. W wyniku badań zostało zauważonych kilkanaście błędów oraz zasugerowanych kilka alternatywnych rozwiązań. Najważniejsze z sugestii dotyczyły:

- Korzystanie ze strzałek na klawiaturze w celu przesuwania figur
- W oknie pokazywania partii szachowej brakowało linków do turniejów, do którego należała partia i do rundy, w której była grana.
- Umieszczenie w zazwyczaj pustym miejscu na prawym panelu informacji o najnowszych artykułach

- Ułatwienie użytkownikowi dostępu do szybkich linków przenoszących ich między powiązаныmi stronami
- Utworzenie stron, na których użytkownik może znaleźć albo szachistów, albo innych użytkowników
- W liście rund i turniejów wyróżnienie aktualnie przeglądanych
- Wyświetlanie niektórych elementów interfejsu (głównie linków i napisów). Szczególnie często uwaga zwracana była na kontrast między kolorem napisów a kolorem tła.
- Błędy w działaniu strony: nie działające funkcje, błędnie wyświetlane dane.

Na podstawie wyników ewaluacji z użytkownikami powstała lista elementów, które konieczne trzeba naprawić, a także lista zmian, których wprowadzenie było opcjonalne. Wnioski z pierwszej listy zostały w pełni wykorzystane i błędy na niej zawarte zostały usunięte z aplikacji. W przypadku drugiej listy zostały wprowadzone wszystkie poprawki spełniające dwie cechy: brak kolizji z moją wizją programu oraz możliwość wprowadzenia bez gruntownego modyfikowania struktury programu.

5.5 Druga iteracja

Po wprowadzeniu zmian, które wyniknęły z ewaluacji z użytkownikiem nastąpił etap planowania kolejnego etapu pracy. Podstawową funkcją, jaka została zaplanowana na ten etap pracy było podłączenie działającego silnika szachowego do wygenerowania analiz pozycji. Sam silnik nie został samodzielnie zaimplementowany - wykorzystałem dostępny na licencji freeware o nazwie XXX.⁴² Wyniki tych analiz miały się ukazywać przy przeglądaniu partii i dynamicznie zmieniać treść w zależności od wykonywanych posunięć. Dodatkową funkcjonalnością, na której rozwinięcie się zdecydowałem było ułatwienie umieszczania okien pozwalających na pokazywanie transmisji partii z zewnętrznych źródeł danych. Z dodatkowych opcji uwzględniona została opcja dodawania komentarzy do poszczególnych pozycji partii.

⁴²Informacje o silniku YYY

5.6 Ewalucja po drugiej iteracji

5.7 Fragmenty najciekawsze algorytmicznie

6 Podsumowanie

7 Bibliografia

Literatura

- [1] Dave Thomas, *Agile. Programowanie w Rails.*, Wydawnictwo Helion 2008, Tłumaczenie: Krzysztof Szafranek
- [2] Claude E. Shannon *Programming a Computer for Playing Chess*¹, Philosophical Magazine, 1950
- [3] John Tromp *Number of chess diagrams and positions*, John's Chess Playground, <http://homepages.cwi.nl/~tromp/>, 2010
- [4] prof. Andrzej Kisielewicz *Sztuczna inteligencja i logika*, Wydawnictwo Naukowo-Techniczne, 2011