Web aplikacija za evidenciju šahovske partije

Đuroković, Matej

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:804890

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Stručni studij

WEB APLIKACIJA ZA ANALIZU ŠAHOVSKE PARTIJE

Završni rad

Matej Đuroković

Osijek, 2017.

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 1
1.2. Link na stranicu 1
2. PRIMJENJENE TEHNOLOGIJE I ALATI 2
2.1. HTML 2
2.2. JavaScript i jQuery i Chessboard.js 2
3. REALIZACIJA APLIKACIJE 4
3.1. Stvaranje interaktivne ploče 4
3.2. FEN zapis šahovske partije 6
3.3. PGN zapis šahovske partije 7
3.4. Primjena dozvoljenih poteza i komentiranje poteza 8
3.5. Pomicanje kroz poteze 8
3.6. Komentiranje poteza 9
3.7. Spremanje i učitavanje analize 10
3.8. Učitavanje PGN i FEN 10
4. ZAKLJUČAK 12
LITERATURA 13
SAŽETAK 14
ABSTRACT 15
ŽIVOTOPIS 16
PRILOG A. INSERTACIJSKI KOD 17
PRILOG B. PLOČA SA POSTAVLJENIM DODATNIM FIGURAMA 17
PRILOG C. BRISANJE FIGURA, POSTAVLJANJE I OKRETANJE PLOČE 18
PRILOG D. PRIMJENA DOZVOLJENIH POTEZA 18
PRILOG E. FUNKCIJE FINDBYID I MOVECLICKED 19

1. UVOD

Cilj ovog završnog rada je napraviti web aplikaciju za analizu šahovskih partija. U aplikaciji bit će moguće odigrati šahovsku partiju, nakon čega ostaju dva načina zapisa (FEN i PGN) partije pomoću kojih se kasnije ta partija može analizirati, komentirati, pregledati ponovno te prikazati ostale moguće poteze. Šah se igra već stoljećima i može se reći da je to jedna od najpopularnijih igara na svijetu. Šah je strateška igra za dva igrača koja se igra na šahovskoj ploči od 64 (8x8) kvadratna polja na kojoj igrači pokušavaju protivnika dovesti u *mat* (kraj igre) poziciju. Igra se s 32 figure, svaki igrač dobije 16 figura, od toga 8 pješaka, po jedan par lovaca, skakača i topova te kralja i kraljicu. Jedan igrač dobiva bijele, dok drugi crne figure. U drugom poglavlju će biti opisana JavaScript, HTML, Bootstrap, jQuery te pomoćna JavaScript datoteka Chessboard.js koja će se koristiti za izradu šahovske ploče, određivanje dozvoljenih poteza, zapisivanje poteza, itd. Treće poglavlje opisivat će svrhu izrade aplikacije i njenu funkcionalnost. Četvrto poglavlje sadržavat će zaključak ovog rada.

1.1. Zadatak završnog rada

U ovom završnom radu potrebno je napraviti web aplikaciju koja omogućuje postavljanje i uređivanje šahovske pozicije. Potrebno je omogućiti uvoz i izvoz standardnih formata zapisa pozicije (FEN) i partije (PGN).

1.2. Link na stranicu

http://www.etfos.unios.hr/~mdurokovic/

2. PRIMIJENJENE TEHNOLOGIJE I ALATI

Web aplikacija je program kojemu se pristupa putem internet preglednika koristeći internet. Kod web aplikacija nije potrebno instaliranje aplikacije na računalo, dovoljno je otvoriti internet stranicu na kojoj je postavljena aplikacija koju želite koristiti putem internet preglednika i aplikacija je spremna za korištenje. Postoji više jezika koji se koriste za izradu web aplikacija, a u ovom radu susrećemo se sa skriptnim jezikom JavaScript, te njenom bibliotekom jQuery. Prednost web aplikacija je što rade bez obzira na operativni sustav koji je instaliran na računalu, programer ne mora raditi posebno sučelje za svaki operativni sustav. Nedostatci su što još uvijek nedovoljno usuglašeni standardi za *HTML* (eng. Hyper-Text Markup Language) , *CSS* (eng. Cascading Style Sheet) i *DOM* (eng. Document Object Model) od strane proizvođača internet preglednika, korisnik sam može u pregledniku napraviti promjene npr. fonta, veličine slova i dr., pa se narušava prikaz aplikacije, brzina rada ovisi o brzini interneta te problemi sa virusima[1]. U ovom radu radimo sa skriptnim jezikom JavaScript i njenom bibliotekom jQuery te dodatnom pomoćnom datotekom Chessboard.js.

2.1. HTML

Hiper-Text Markup Language je prezentacijski jezik za izradu web stranica. Prikaz HTML dokumenta omogućava internet preglednik[2]. HTML jezikom se ne može raditi nikakve operacije kao što su zbrajanje i oduzimanje. On služi samo za uređivanje teksta na stranici i prikaz texta kojim se kasnije može manipulirati pomoću JavaScript-a. To su tekstualne datoteke koje imaju ekstenziju .html

2.2. JavaScript i jQuery i Chessboard.js

JavaScript je programski (skriptni) jezik koji se izvršava u internet pregledniku[3]. Pomoću JavaScript-a se može mijenjati izgled naše web stranice, klikom na nekakav tekst može ga se obojiti, promijeniti font, promijeniti tekst itd. Isto tako pomoću JavaScript-a se može napraviti puno više kao što će se u ovom radu napraviti aplikacija za analizu šahovske partije.

JQuery je besplatna biblioteka JavaScripta, licencirana od strane MIT licenci[4]. Danas najpopularnija biblioteka koja se koristi, sa 65% instalacija u top 10 milijuna najprometnijih web stranica. Dizajnirana je kako bi se jednostavnije moglo izabirati *DOM* elemente, izrađivati animacije i proizvoditi web aplikacije.

Chessboard.js je dodatna JavaScript datoteka koja olakšava izradu šahovske ploče te u kojoj su upisana pravila igre šah. u ovom radu korištena je radi brže i jednostavnije izrade a licencirana je od strane *MIT* licence. Konfiguracija se postiže unošenjem HTML-a i JavaScript koda te njihovom kombinacijom se dobiju naredbe koje određuju što na šahovskoj ploči se može raditi (pomicanje figura samo dozvoljenim putanjama, zapis položaja FEN i PGN, dodavanje figura, itd.).

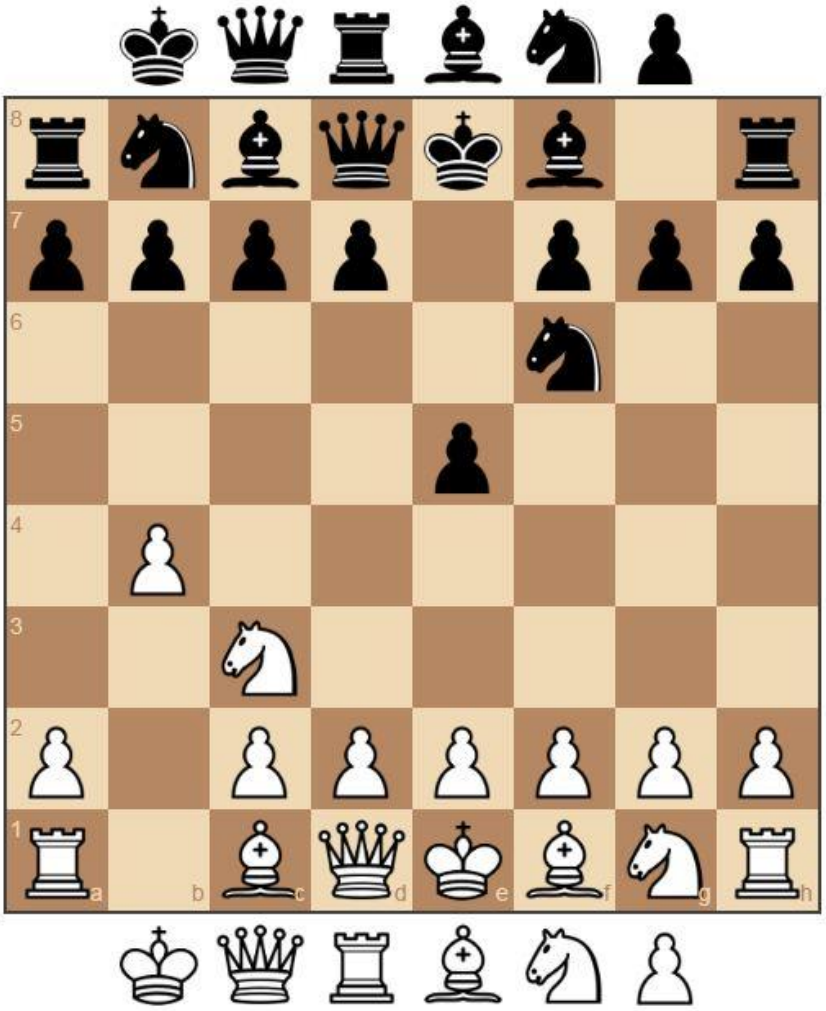
3. REALIZACIJA APLIKACIJE

3.1. Stvaranje interaktivne ploče

Stvaranje interaktivne ploče se dobiva pomoću insertacije Chessboard.js tako da priložimo u zaglavlje određene skripte te linkove sa skriptama. Insertacijski kod je prikazan PRILOGOM A, A na slici 3.1. prikazano je što se dobije tim kodom.

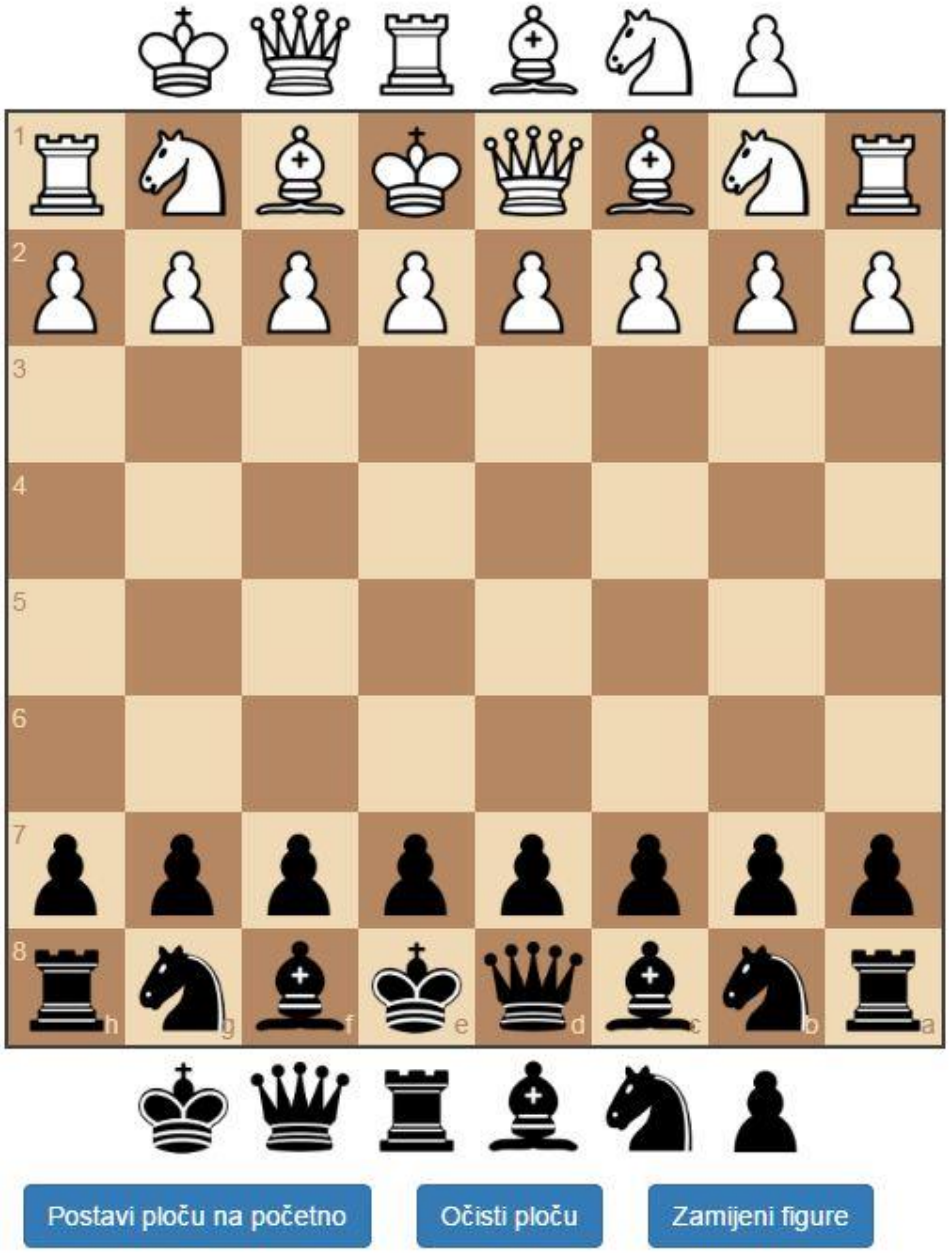
Daljnje crtanje ploče sa svim postavljenim figurama i mogućnošću pomicanja i dodavanja figura na svim pozicijama neovisno o pravilima dobiva se s jednostavnim HTML i JavaScript kodom, te Bootstrap kodom za uređivanje prikazano PRILOGOM B.

U skriptnom dijelu pod varijablu cfg spremamo naredbe *dragable* što dozvoljava povlačenje figura, *dropOffBoard: 'trash'* znači da ako povučemo figuru van ploče ista se briše, *position: 'start'* postavlja figure na ploču u početni sastav, te *sparePieces: true* postavlja dodatne figure uz ploču.



Slika 3.1. Ploča dobivena nakon odrade gornjeg navoda

Nakon stvaranja interaktivne ploče napravljena je mogućnost brisanja svih figura, slaganje na startnu poziciju i okretanje ploče tako da crne figure budu dolje. To je ostvareno pomoću jQuery biblioteke i jednostavnog HTML-a postavljanjem tri gumba. Kod je prikazan PRILOGOM C. Time smo dobili početnu funkciju aplikacije. Postavljanje ploče, pomicanje figura, okretanje ploče.



Slika 3.2. Primjer šahovske ploče sa zamjenjenim figurama

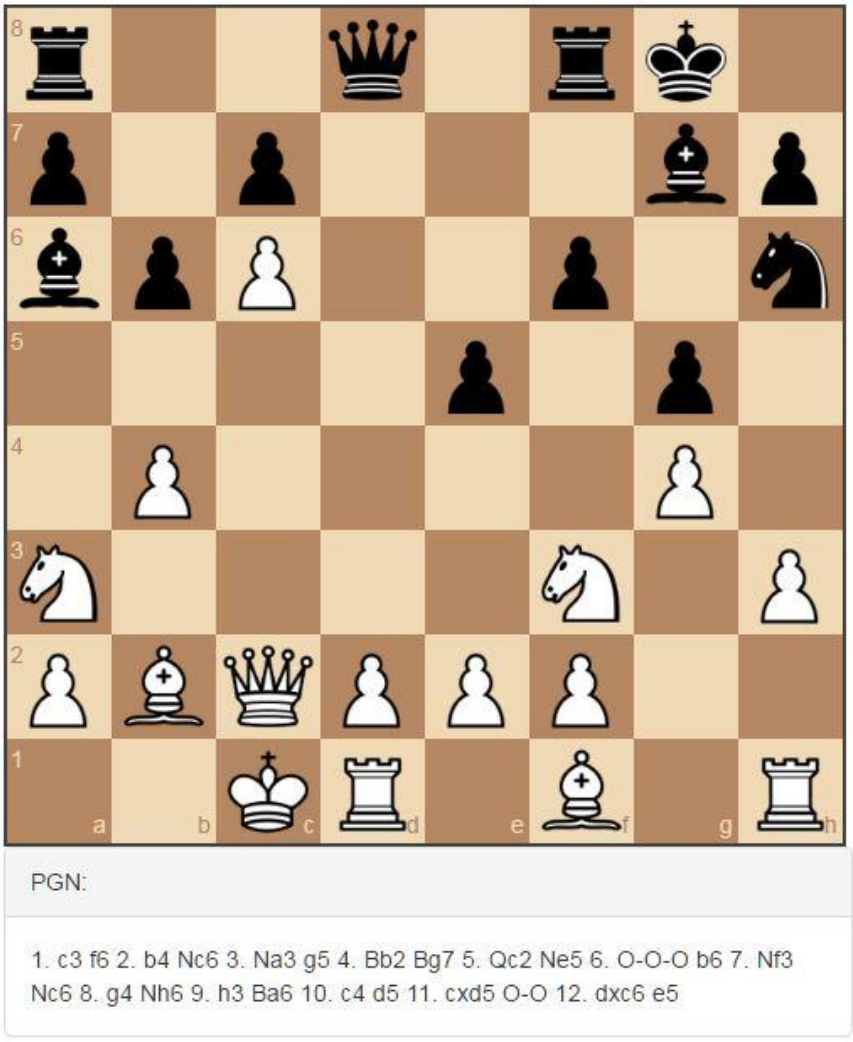
3.2. FEN zapis šahovske partije

FEN (Forsyth Edwards Notation) je kratka notacija kojom je moguće ispisati bilo koja pozicija na šahovskoj ploči. u 19. stoljeću FEN notacija je razvijena od strane Davida Forsytha, a Steven J. Edwards je nadogradio ovaj sustav zapisa za računalne programe kako bi mogli zapisati šahovsku partiju. FEN zapis se ispisuje red po red od 8. do 1. reda tako da je svaki red odvojen kosom crtom (/). FEN notacija prazne ploče: "8/8/8/8/8/8/8/8" gdje svaki broj predstavlja broj praznih polja unutar jednog reda, FEN notacija ploče sa svim figurama u početnom položaju: "rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR" gdje nam mala slova predstavljaju crne figure, a velika bijele figure. Svaka figura se zapisuje posebnim slovom, a počinje s početnim slovom engleskog naziva figure: p/P - crnu/bijeli pješak (eng. *pawn*), n/N - crni/bijeli skakač (eng. *knight*), b/B - crni/bijeli lovac (eng.*bishop*), r/R - crni/bijeli top (eng. *rook*), q/Q - crna/bijela kraljica (eng. *queen*), k/K - crni/bijeli kralj (eng. *king*). Brojevi označavaju koliko praznih polja slijedi[5].

Slika 3.3. Prikaz FEN notacije

3.3. PGN zapis šahovske partije

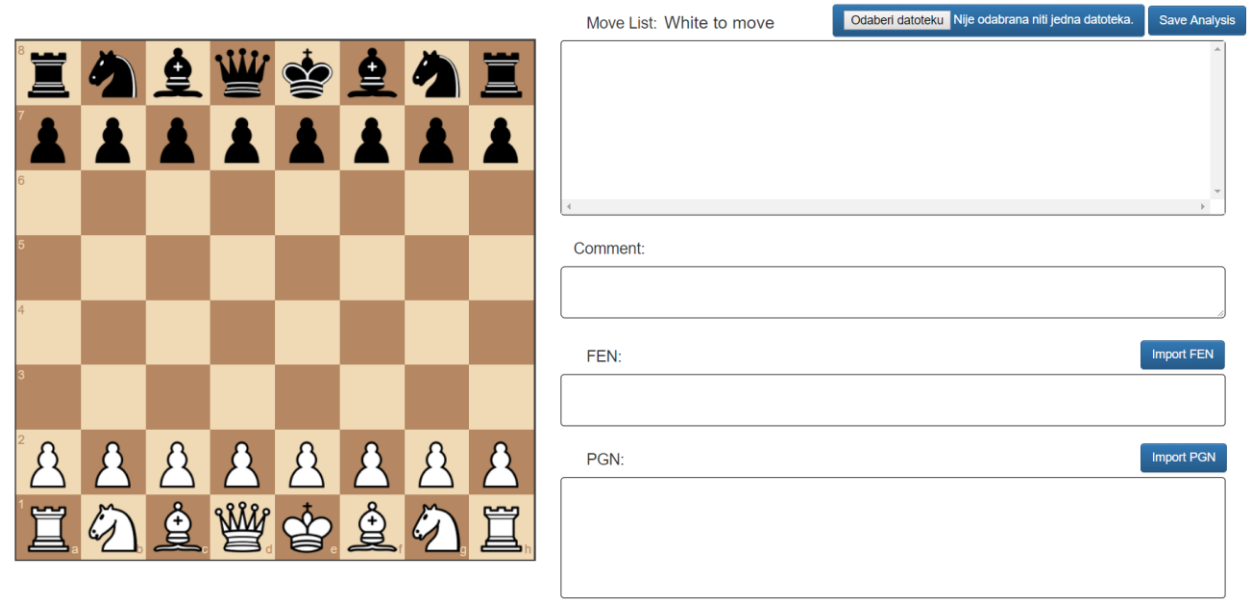
PGN (eng. *Portable Game Notation*) je razvijen od strane Steven J. Edwardsa 1993. godine. Omogućuje strukturalni zapis partije pomoću algebarske notacije. Prilikom bilježenja poteze označavamo rednim brojem. Najčešće se koristi skraćena notacija pri čemu se bilježi oznaka figure i završno polje, npr. Prva četiri poteza sa slike 3.4.: "1. c3 f6 2. b4 Nc6 3. Na3 g5 4. Bb2 Bg7". Ako dva topa, lovca ili skakača mogu doći na isto polje dodatnim znakom početnog slova figure se označava linija ili red. Npr. ako se bijeli skakači nalaze na "c3" i "g1" oba mogu doći na "e2" tada pišemo "Nce2" i "Nge2". Rokada je šahovski potez kako bi se napravila bolja zaštita kralja. Jedini potez u kojemu se pomiču istovremeno dvije figure a broji se kao jedan potez. Također jedino prilikom rokade kralj može prijeći dva polja. Malu rokadu zapisujemo znakom "0-0", a veliku rokadu "0-0-0". Promociju pješaka zapisujemo "c8=Q".



Slika 3.4. Prikaz PGN notacije

3.4. Primjena dozvoljenih poteza i komentiranje poteza

Primjena dozvoljenih poteza napravljena je pomoću stvaranja nove igre, gdje se gleda FEN i PGN zapis i status igre te prati tko je na potezu. Za praćenje statusa igre iskorištena je funkcija *updateStatus* koja prati koji je igrač na potezu, te u kojem je statusu igra (igra traje, mat, pat, šah) te istovremeno zapisuje PGN i FEN notaciju. Funkcije *onDragStart* i *onDrop* prate može li igrač podignuti ili spustiti figuru na određeno mjesto. Ako igrač koji pokušava podići figuru nije na potezu tada nije u mogućnosti podići figuru. Kada igrač koji je na potezu podigne figuru program provjerava gdje će igrač spustiti figuru i dozvoljava mu spustiti samo na određenim mjestima na kojima figura može biti spuštena zbog svog određenog kretanja. Ako igrač spusti figuru na polje koje nije u smjeru kretanja figure, figura se automatski vraća na poziciju s koje je podignuta. Kod je prikazan u PRILOGU D. Nakon dodanih komentara izgled kompletne aplikacije je prikazan slikom 3.5.

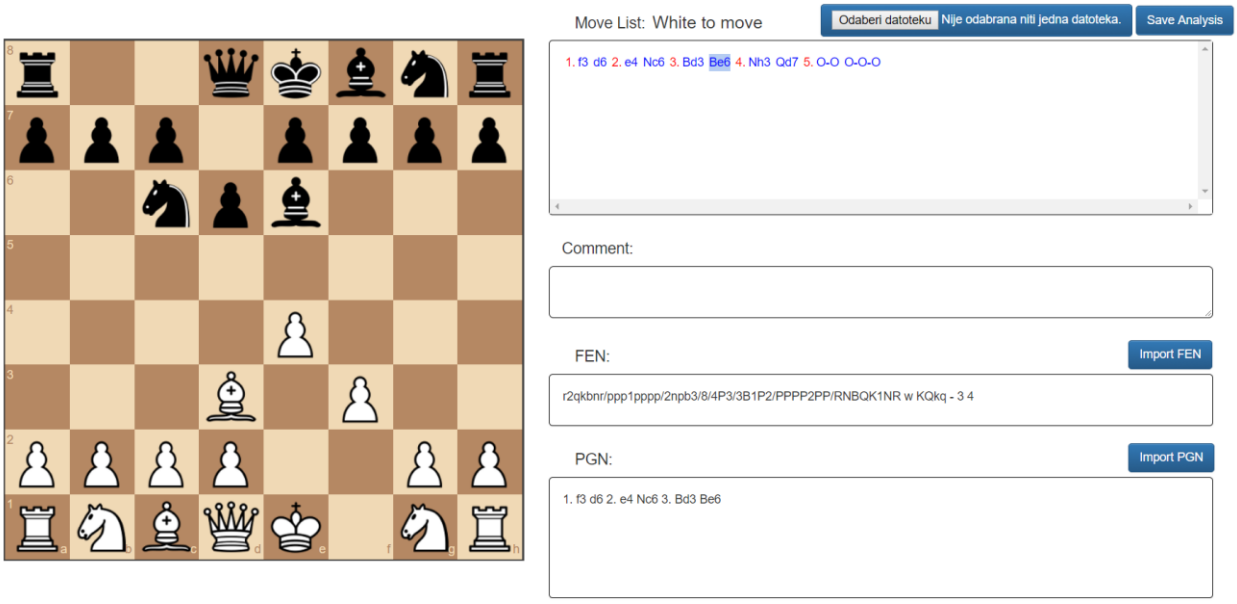


Slika 3.5. Krajnji izgled aplikacije

3.5. Pomicanje kroz poteze

Ovo je aplikacija za analizu šahovske partije i kako bi bilo moguće u potpunosti analizirati šahovsku partiju mora postojati mogućnost pomicanja kroz poteze. Problem pomicanja kroz poteze riješen je funkcijom *moveClicked.* Prilikom ispisivanja liste poteza svaki potez dobiva svoj *id. MoveClicked* poziva funkciju *FindById* koja prolazi kroz move listu i traži koji je *id*

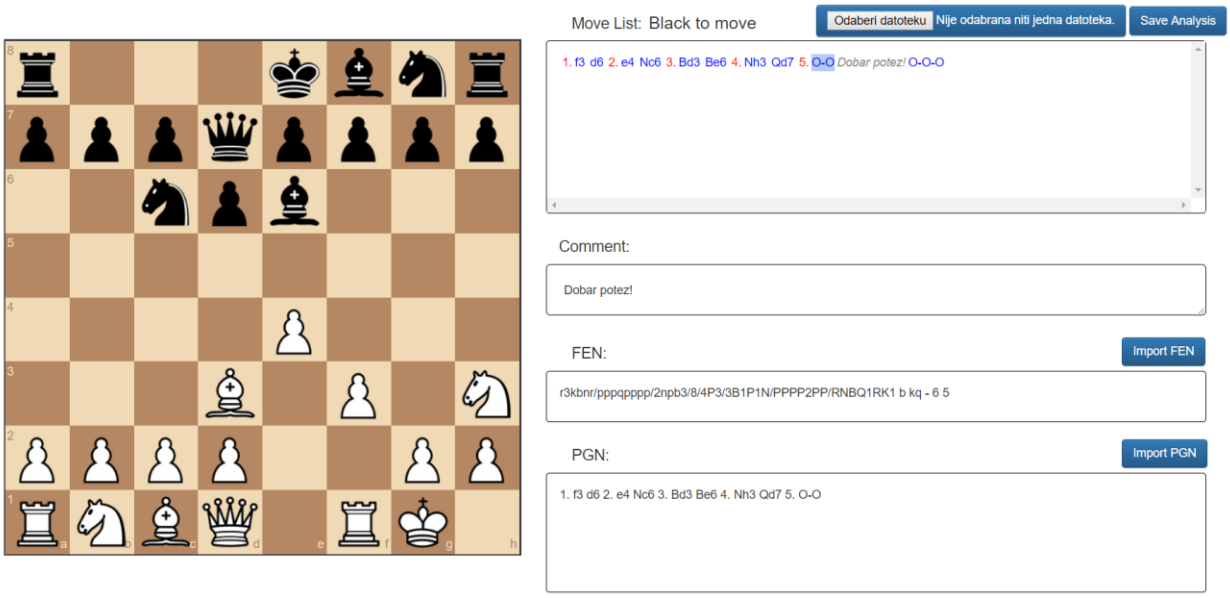
poteza koji je kliknut. Odrađuje se pomicanje figura na FEN zapis tog poteza. Osvježava se FEN zapis, PGN zapis i ploča s figurama te pozadina kliknutog poteza postaje plava. Na slici 3.6 prikazano je pomicanje kroz poteze. *MoveClicked* i *FindById* funkcije prikazane su PRILOGU E.



Slika 3.6. Pomicanje kroz poteze

3.6. Komentiranje poteza

Komentiranje poteza je omogućeno pomoću izrade *textarea* područja za upisivanje komentara i nastavlja se na pomicanje kroz poteze. Nakon pomaka na određeni potez i upisivanja komentara u za to predviđeni dio, na listi poteza se ispisuje komentar. Komentar se ispisuje automatski na listi poteza kako se upisuje u dio predviđen za komentiranje i ostaje zapamćen. Komentiranje je prikazano na slici 3.7.



Slika 3.7. Komentiranje poteza

3.7. Spremanje i učitavanje analize

Kako bismo mogli spremiti napravljenu analizu te ju kasnije učitati i nastaviti analizirati postvljena su dva gumba *Odaberi datoteku* i *Save analysis*. Klikom na gumb *Save analysis* pokreće se javaScript datoteka *FileSaver*. Od korisnika se traži da odabere mjesto na koje će spremiti napravljenu analizu.

Ako želimo učitati analizu, klikom na gumb *Odaberi datoteku* otvara se novi prozor koji traži od korisnika da odabere spremljenu analizu te odabirom željene analize ista se otvara i prikazuje se odražena analiza.

3.8. Učitavanje PGN i FEN

U aplikaciji su dodana dva gumba, *Improt FEN* i *Import PGN*. Klikom na *Import FEN* izlazi nam izbornik u kojemu možemo upisati svoj FEN koji želimo i ploča se postavlja na naš zadani FEN. Također klikom na *Import PGN* izlazi nam izbornik u kojemu možemo upisati svoj PGN i ploča se osvježava sa svim potezima koje smo upisali. Tu možemo komentirati svaki potez koji je odigran i kretati se kroz poteze. Izgled *Import FEN* i *Import PGN* prikazan je na slici 3.8.

Slika 3.8. *Import FEN* i *Improt PGN*

4. ZAKLJUČAK

U ovom završnom radu napravljena je aplikacija za analizu šahovskih partija koja sadrži sve potrebne alate za temeljitu analizu te . Interaktivna šahovska ploča sadrži sva potrebna šahovska pravila kako bi se mogla odigrati šahovska partija. Aplikacija sadržava mogućnost pomicanja kroz poteze, komentiranja odigranih poteza te FEN i PNG notaciju za zapis odigranih poteza u partiji. Aplikacija je izrađena u JavaScript skriptnom jeziku pa nije potrebna nikakva dodatna instalacija, dovoljno je imati internet preglednik. JavaScript je podržana u većini internet preglednika i aplikacija se može primijeniti *online* kako bi aplikacija mogla biti dostupna što većem broju korisnika.

LITERATURA

[1] Web aplikacije, prednosti i nedostatci, http://www.horizont.com.hr/web-aplikacije---prednosti-i-nedostatci-3-blog, lipanj 2016.

[2] w3schools, HTML, https://www.w3schools.com/html/, lipanj 2016.

[3] Wikipedija, JavaScript skriptni jezik korišten kod izrade završnog rada https://en.wikipedia.org/wiki/JavaScript, lipanj 2016.

[4] jQuery, Biblioteka JavaScripta, https://jquery.com/, lipanj 2016.

Wikipedija, Šahovska notacija, https://hr.wikipedia.org/wiki/%C5%A0ahovska_notacija, lipanj 2016.

[5] Šah, FEN notacija, http://sah.hr/forum/index.php?topic=67.0;wap2, lipanj 2016.

SAŽETAK

Naslov: Web aplikacija za analizu šahovske partije

U ovom završnom radu opisana je izrada web aplikacije za analizu šahovske partije koja je pisana u JavaScript skriptnom jeziku uz pomoć jQuery i Chessboard.js biblioteka. Aplikacija sadrži sva pravila igre šah dovoljne za igranje te dodatne mogućnosti za analizu odigrane šahovske partije kao što su zapisi FEN i PGN i mogućnost komentiranja poteza. Aplikacija je namijenjena korištenju na internetu kako bi bila pristupačna što većem broju korisnika te za nju nije potrebna instalacija. JavaScript ne zahtijeva puno, dovoljno je imati internet preglednik i otvoriti stranicu na koju je aplikacija postavljena.

Ključne riječi: web aplikacija, šah, analiza, JavaScript, FEN, PGN.

ABSTRACT

Title: Web application for chess game analysis

This final paper describes the development of web application for chess game analysis that is written in the JavaScript scripting language with help of jQuery and Chessboard.js library. The application contains all the rules of the game of chess enough to play and additional features for analysis of played game of chess, such as note of FEN and PGN notation and the capability of writing comments of moves. The application is intended for use on the internet in order to be accessible to a many users and installation is not required. JavaScript does not require a lot, it is enough to have a web browser and open the page on which the application is uploaded.

Key words: web application, chess, analysis, JavaScript, FEN, PGN.

ŽIVOTOPIS

Matej Đuroković rođen je 24. siječnja 1995. u Požegi gdje završava osnovnu školu i upisuje srednju Tehničku školu u Požegi te maturira 2013. Iste godine upisuje stručni studij na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija, smjer informatika. Na drugoj godini odrađuje stručnu praksu na kojoj izrađuje stranicu za jednostavnu izradu CV-a. Položio je sve ispite i sada se nalazi pred obranom svoga završnog rada.

_____

Matej Đuroković

PRILOG A. INSERTACIJSKI KOD

```
<head>
    <script src="js/chessboard-0.3.0.js"></script>
    <script src="js/chessboard-0.3.0.min.js"></script>
    <script src="js/jquery-1.12.4.min.js"></script>
    <script
src="http://www.willangles.com/projects/chessboard/js/chess.js"></scri
pt>
    <link rel="stylesheet" type="text/css" href="css/chessboard-
0.3.0.css">
    <link rel="stylesheet" type="text/css" href="css/chessboard-
0.3.0.min.css">
    <link rel="stylesheet"
href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min
.css">
</head>
```

PRILOG B. PLOČA SA POSTAVLJENIM DODATNIM FIGURAMA

```
<body>
<div class="container-fluid">
<div class="row">
<div class="col-md-4">
<div id="board" style="width: auto"></div>
</div>

<script>
var cfg = {
  draggable: true,
  dropOffBoard: 'trash',
  position: 'start',
  sparePieces: true};
var board = ChessBoard('board', cfg);
</script>
```

PRILOG C. BRISANJE FIGURA, POSTAVLJANJE I OKRETANJE PLOČE

jQuery dio:

```
$('#startPositionBtn').on('click', board.start);
$('#clearBoardBtn').on('click', board.clear);
$('#flipOrientationBtn').on('click', board.flip);
```

HTML dio:

```
<button class="btn btn-primary" style=" margin:10px;"
id="startPositionBtn">Postavi ploču na početno</button>
<button class="btn btn-primary" style=" margin:10px;"
id="clearBoardBtn">Očisti ploču</button>
<button class="btn btn-primary" style=" margin:10px;"
id="flipOrientationBtn">Zamijeni figure</button>
```

PRILOG D. PRIMJENA DOZVOLJENIH POTEZA

```
<script>
var board,
  game = new Chess(),
  statusEl = $('#status'),
  fenEl = $('#fen'),
  pgnEl = $('#pgn');
// Zabrani micanje figure ako je igra gotova
// Dozvoli pomicanje figure samo onom igracu koji je na potezu
var onDragStart = function(source, piece, position, orientation) {
  if (game.game_over() === true ||
      (game.turn() === 'w' && piece.search(/^b/) !== -1) ||
      (game.turn() === 'b' && piece.search(/^w/) !== -1)) {
    return false;
  }
};
var onDrop = function(source, target) {
  // Provjeri je li potez dozvoljen
  var move = game.move({
    from: source,
    to: target,
    promotion: 'q' // Uvijek promoviraj u kraljicu
  });
  // nedozvoljen potez
  if (move === null) return 'snapback';
```

```
  updateStatus();
};
// osvjezi pgn nakon spustene figure
// rosada en passant i promocija
var onSnapEnd = function() {
  board.position(game.fen());
};
var updateStatus = function() {
  var status = '';
  var moveColor = 'Bijeli';
  if (game.turn() === 'b') {
    moveColor = 'Crni';
  }
  // mat?
  if (game.in_checkmate() === true) {
    status = 'Igra je gotova, ' + moveColor + ' je u šahmatu.';
  }
  // pat?
  else if (game.in_draw() === true) {
    status = 'Igra je gotova, pat!';
  }
  // Ako igra i dalje traje
  else {
    status = moveColor + ' je na potezu';
    // Sah?
    if (game.in_check() === true) {
      status += ', ' + moveColor + ' je u šahu';
    }
  }
  statusEl.html(status);
  fenEl.html(game.fen());
  pgnEl.html(game.pgn());
};
var cfg = {
  draggable: true,
  position: 'start',
  onDragStart: onDragStart,
  onDrop: onDrop,
  onSnapEnd: onSnapEnd
};
board = ChessBoard('board', cfg);
updateStatus();
</script>
```

PRILOG E. FUNKCIJE FINDBYID I MOVECLICKED

```
function findById(id, array=gameHistory){
     var move;
```

```
	iterate(id, array)

	return move

	function iterate(id, array=gameHistory){
		for(var i=0; i<array.length; i++){
			if(array[i].id == id){
				if(array[i+1]){
					move =  [array[i], false, array, i];
				}else{
					move =  [array[i], true, array, i]
				}
			}
			else if(array[i].children.length != 0){

				for(var j=0; j<array[i].children.length; j++){
					iterate(id, array[i].children[j])
				}
			}
		}
	}

}


function moveClicked(id){

	//console.log(id)
	activeMove = findById(id, gameHistory)
	var move = activeMove[0]


	$("#commentBox").val(activeMove[0].comment)

	currentVariation = [];

	game.load(move.fen);
	board.position(move.fen);

	showFEN();
	showPGN();
	updateStatus();
	printMoves();


	return;
```