

Šah - Multimedijski sustavi

Matija Halavanja, Marin Hanić

4. lipnja 2022.

Sadržaj

1	Opis programa	1
2	Dizajn programa	1
2.1	sketch.js	2
2.2	Board	2
2.3	Piece	2
3	Poboljšanja	3

1 Opis programa

Za projekt iz kolegija Multimedijski sustavi odlučili smo napraviti šah u programskom jeziku Javascript koristeći p5.js. Odlučili smo se za p5.js umjesto Processinga jer smo htjeli u isto vrijeme istražiti taj library i napraviti projekt koji bi bio dostupan na webu. Šah je igra za dva igrača s pravilima koji su poznati većini ljudi pa ih sada nećemo ovdje objašnjavati. Figure se pomiču tako da kliknemo i povučemo na željeno polje. Ako figura po pravilima šaha ne može doći na to polje, vraća se na početno mjesto i isti igrač je i dalje na potezu.

2 Dizajn programa

Program se dijeli na dva dijela. Prvi dio je usko vezan uz p5.js te obuhvaća sve što korisnik aplikacije vidi i kako interagira s aplikacijom. Većina tog dijela aplikacije se nalazi unutar *sketch.js* datoteke. Drugi dio aplikacije je logički prikaz svih događaja koji su posljedica prikaza i interakcije korisnika s aplikacijom. Između ostalog, to znači da moramo na početku postaviti sve figure na početne pozicije, validirati je li određen potez moguć ili ne, registrirati kraj partije te registrirati dolazak pješaka na zadnji red ploče. Svaka figura je implementirana u zasebnoj klasi koja se nalazi u zasebnoj datoteci. Svaka klasa figure extenda generičku klasu *Piece* u kojoj se nalaze neke zajedničke funkcije i svojstva. Također, postoji posebna klasa *Board* koja predstavlja ploču i objedinjuje logiku pomicanja figura po šahovskim pravilima koja se za svaki potez moraju provjeravati. Takva klasa mora postojati radi hrpe šahovskih pravila npr. ako je kralj u šahu, igrač mora napraviti neki potez tako da mu ne bude šah nakon tog poteza. Radi takvih pravila trebamo stalno pratiti koja polja svaki igrač napada sa svojim figurama. Da bi to postigli, za svaku figuru moramo imati funkciju koja nam ovisno o trenutnom polju na kojem se nalazi i ostalim figurama na ploči, vraća na koja polja se ta figura može pomaknuti. Ta ista polja su i također polja koja ta figura napada. Jedina iznimka tomu je en passant koji je morao onda posebno biti implementiran. Svaka figura još implementira funkcije potrebne za njeno pravilno pomicanje i smještena je u vlastitoj datoteci i vlastitoj

klasi. Te klase nećemo posebno spominjati ovdje, nego ćemo se fokusirati na prethodno spomenute tri datoteke odnosno klase.

2.1 sketch.js

Datoteka sadrži varijable potrebne za pravilno iscertavanje ploče i figura te se brine o tome što se događa na pritisak tipke miša i njeno otpuštanje.

Funkcija *setup* kreira instancu klase *Board* i priprema canvas.

Funkcija *draw* iscertava ploču i figure na njoj, ovisno o tome je li figura spuštена ili podignuta.

Funkcije *mousePressed* i *mouseReleased* služe za prepoznavanje podizanja i spuštanja figure na polje. Moraju paziti da se miš nalazi unutar okvira ploče i odrediti o kojem polju se točno radi.

Funkcija *choosePiece* sadrži dio logike potreban kod promocije pijuna u figuru.

2.2 Board

Klasa *Board* je najveća klasa u projektu. To je zato što se u njoj provjeravaju sva pravila šaha pri pomicanju figura koja uključuju interoperabilnost figura. Također sadrži puno jednostavnih, pomoćnih funkcija od po svega par linija koje nećemo spominjati tu.

Funkcija *getMovingPiece* za dano polje vraća referencu na figuru koja se nalazi na tome polju.

Funkcija *showMovingPiece* se brine da se slika figure pokazuje ispravno pri pomicanju figure mišem.

Funkcija *startPieces* inicijalizira i vraća figure za danog igrača.

Funkcija *makeMove* pomiče odabranu figuru na dano polje. Iz ove se funkcije onda poziva većina logike u aplikaciji jer se pritom svakog poteza mora provjeriti da je potez legalan po pravilima šaha.

Funkcija *isGameOver* poziva pomoćne funkcije i provjerava je li igra gotova. Funkcija *isTherePossibleMove* se koristi kod provjere pata.

Funkcija *canPreventCheck* se koristi kod provjere mata jer ako igrač ne može nikako spriječiti šah, zapravo je u matu.

Funkcija *checkPawnToPieceChange* je prvi dio logike provjere je li pješak došao do zadnjeg reda za promociju.

Funkcija *isCausingSelfCheck* provjerava da pomicanjem dane figure igrač ne prouzročuje šah sam sebi. Primjer toga je kada je npr. skakač između svog kralja i protivničkog topa pa se ne smije pomaknuti.

Funkcija *setCheck* postavlja svojstvo *inCheck* objekta klase *King* na true ako je protivnik svojim zadnjim potezom dao šah.

Funkcija *show* se poziva u *sketch.js* u funkciji *draw*, a služi za grafički prikaz ploče i svih figura.

2.3 Piece

Ova klasa obuhvaća svojstva i funkcije koje svaka druga klasa koja predstavlja figuru extendira. Funkcija *setSquare* zadužena je za pravilni logički i grafički prikaz figure na dano polje pri njenom pomicanju. Funkcije *show* i *moveImage* se brinu za pravilan prikaz slika figura na ploči, ovisno o polju na kojem se nalaze ili poziciji miša. Funkcija *isPlayableSquare* provjerava je li figura može biti odigrana na dano polje, ovisno o tome je li polje prazno i kojem igraču pripada. Funkcija *checkCapture* provjerava je li pri pomicanju figure neka protivnička figura pojedena. Funkcija *isDoubleCheck* provjerava je li kralj danog igrača u duplom šahu. To je potrebno provjeravati jer u toj situaciji nije dovoljno pojesti figuru koja zadaje šah, nego jedini način da kralj ne bude u šahu je da se pomakne.

3 Poboljšanja

Neka moguća poboljšanja su:

- Uvođenje vremenske kontrole za svakog igrača po partiji
- Uvođenje izbora vremenske kontrole, npr 3 min po igraču plus 2 sekunde po potezu, 5 min po igraču
- Zapisivanje odigranih poteza
- Mogućnost lokalnog spremanja odigranih poteza u datoteku
- Mogućnost izbora izgleda figura i ploče
- Dodavanje opcije okretanja ploče tako da figure igrača koji je na potezu budu dolje radi lakšeg pregleda ploče