

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 851

Igre na grafovima

Filip Čelepirović

Zagreb, prosinac 2024.

Zagreb, 10. ožujka 2023.

ZAVRŠNI ZADATAK br. 851

Pristupnik: **Filip Čelepirović (0036529608)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentorica: izv. prof. dr. sc. Anamari Nakić

Zadatak: **Igre na grafovima**

Opis zadatka:

Teorija grafova je grana matematike s brojnim primjenama u inženjerstvu. Matematički pojam grafa često se primjenjuje kao apstraktni model za rješavanje problema iz ljudske djelatnosti. Kombinatorne igre, tema ovog rada, predstavljaju matematičke modele za stratešku interakciju među racionalnim dionicima. U ovom radu proučit će se progresivne kombinatorne igre. Predstavit će se apstraktna analiza pobjedničkih strategija opisanih igara u okviru teorije grafova. Proučit će se Grundyjeva funkcija te će se predstaviti osnovni rezultati. Obradena analitička tehnika iz teorije grafova primijenit će se za analizu i odabir pobjedničkih strategija niza modifikacija igre Nim. Izradit će se interaktivna web aplikacija koja implementira jednu verziju igre Nim.

Rok za predaju rada: 9. lipnja 2023.

Posebna zahvala mojim roditeljima na podršci tijekom cijelog školovanja. Također, želim zahvaliti mentorici prof. Nakić na potpori i pomoći u izradi ovog završnog rada.

SADRŽAJ

Popis slika	v
Popis tablica	vi
1. Uvod	1
2. Osnovni pojmovi	2
2.1. Uvod i definicija	2
2.2. Primjeri kombinatornih igara	3
3. Analiza kombinatornih igara	6
3.1. Analiza igre oduzimanja	6
3.2. Analiza igre Nim	7
4. Tehničke značajke razvijene aplikacije	14
4.1. Struktura aplikacije	14
4.2. Funkcionalnosti aplikacije	15
5. Upute za korištenje razvijene aplikacije	23
5.1. Pristupanje igri	23
5.2. Prikaz igre	26
6. Zaključak	28
Literatura	29

POPIS SLIKA

2.1. Igra Nim.	4
3.1. Prikaz igre sa dvije hrpe.	8
3.2. Prikaz pozicije s jednim žetonom na svakoj hrpi.	8
3.3. Prikaz mogućih poteza iz pozicije (1,1,2).	9
3.4. Stablo svih mogućih pozicija u igri Nim s tri hrpe.	10
4.1. Prikaz strukture koda aplikacije.	14
4.2. Prikaz koda funkcije createServer().	16
4.3. Prikaz koda funkcije showFrontPage().	16
4.4. Prikaz tablice top lista iz funkcije showRanks().	17
4.5. Prikaz koda funkcija showRules(), showGameDiv() i resetGameDiv().	17
4.6. Prikaz koda funkcije playGame().	18
4.7. Prikaz koda funkcije login().	18
4.8. Prikaz koda funkcije logout().	19
4.9. Prikaz koda za stvaranje ploče za igru.	19
4.10. Prikaz koda za inicijalizaciju nove igre.	20
4.11. Prikaz određivanja novog poteza računala.	21
4.12. Prikaz funkcije xor().	21
4.13. Prikaz funkcije za uklanjanje loptice iz igre.	22
5.1. Stanje programa nakon pokretanja.	24
5.2. Stanje programa nakon pritiska na tipku top lista.	24
5.3. Forma za prijavu/registraciju korisnika.	25
5.4. Prikaz pravila igre.	25
5.5. Prikaz forme za pokretanje nove igre.	26
5.6. Stanje programa nakon odabira tipke za početak nove igre.	27
5.7. Stanje programa nakon pobjede igrača.	27
5.8. Stanje programa nakon poraza igrača.	27

POPIS TABLICA

3.1. Prikaz P-pozicija i N-pozicija u igri oduzimanja (1-11)	6
3.2. Prikaz P-pozicija i N-pozicija u igri oduzimanja(12-21)	7

1. Uvod

Teorija grafova vrlo je važna grana matematike koja se primjenjuje u mnogim poljima znanosti, kao što su računalna znanost, inženjerstvo, ali i u društvenim znanostima. Koncept grafa je apstraktni model koji se koristi za rješavanje različitih problema u ljudskom djelovanju. S druge strane, kombinatorne igre predstavljaju matematičke modele za stratešku interakciju između racionalnih dionika.

Ovaj rad proučava progresivne kombinatorne igre i predstavlja apstraktnu analizu njihovih pobjedničkih strategija koristeći teoriju grafova. Konkretno, u fokusu je Grundyjeva funkcija, koja se koristi za određivanje ishoda kombinatornih igara.

Analiza pobjedničkih strategija ključna je jer pomaže u razumijevanju procesa donošenja odluka i pruža uvid u međusobnu stratešku komunikaciju. Korištenje teorije grafova u ovoj analizi omogućuje nam modeliranje složenih sustava i njihovu strukturu i detaljnu analizu.

Nadalje, primijenjena je razrađena analitička tehnika iz teorije grafova za analizu i odabir optimalnih pobjedničkih strategija za niz modifikacija igre Nim. Nim je klasična kombinatorna igra koja je opsežno proučavana u literaturi i ima brojne primjene u računalnim znanostima i inženjerstvu.

Kako bismo analizu kombinatornih igara učinili što pristupačnijom i kao rezultat cjelokupne teorijske analize, izrađena je interaktivna web aplikacija koja implementira jednu verziju igre Nim. To omogućuje korisnicima da eksperimentiraju s igrom i bolje usavrše razumijevanje koncepta igre o kojima se govori u ovom radu.

Sveukupno, cilj je ovog rada pružiti sveobuhvatnu analizu kombinatornih igara i njihovih pobjedničkih strategija korištenjem teorije grafova.

2. Osnovni pojmovi

2.1. Uvod i definicija

Kombinatorne igre vrsta su igara u kojima se igrači bore za kontrolu nad skupom objekata, poput figura na šahovnici ili polja na ploči. Ove igre zovu se kombinatorne jer se igrači bore za kontrolu nad kombinacijom objekata i svojom strategijom igrači koriste svoje znanje i vještinu za kontrolu igre i ukupnu pobjedu.

Za svaku kombinatornu igru vrijede sljedeći uvjeti[3]:

- Dva igrača: Kombinatorne igre igraju se između dva igrača, koji mogu biti označeni različitim bojama, znakovima ili samo kao prvi i drugi igrač.
- Nema elemenata slučajnosti: U kombinatornim igrama, nema faktora slučajnosti koji bi utjecali na ishod igre. To znači da potezi koje igrači čine nisu određeni kockanjem, srećom ili nasumičnim događajima, već samo njihovom strategijom.
- Potpuna informacija: Svaki igrač u kombinatornoj igri ima potpuni uvid u sve informacije o trenutnom stanju igre. Nema skrivanja idućeg poteza, svaki igrač zna sve što je javno dostupno.
- Naizmjenična igra: Kombinatorne igre igraju se naizmjenično između dva igrača. Svaki igrač izvodi poteze jedan za drugim, izmjenjujući se dok igra ne završi.
- Postoji pobjednik: U kombinatornim igrama, jedan igrač uvijek mora pobijediti, a drugi izgubiti. Nema neodlučenog rezultata kao što je remi u šahu.
- Uvjet pobjede: Da bi igrač pobijedio, potrebno je doći do krajnje pozicije igre koja je unaprijed definirana. Ova krajnja pozicija može biti, na primjer, osvajanje protivničkog kralja u šahu ili postizanje zadane razine bodova u igri kao što je Go.

Također, moramo uvesti još jedan pojam, a to je nepristrana igra[1].

Definicija 1. *Igra je nepristrana ako i samo ako oba igrača imaju iste mogućnosti poteza za svaku trenutnu poziciju u igri.*

U svim nepristranim kombinatornim igrama postoje dvije vrste pozicija: *P-pozicije* (engl. previous) i *N-pozicije* (engl. next). P-pozicije su pozicije u kojima prethodni igrač ima pobjedničku strategiju, a sukladno tome, N-pozicije su pozicije u kojima sljedeći igrač ima pobjedničku strategiju. Postupak za određivanje vrste pozicije kojoj pripada određena pozicija započinje identificiranjem završnih, terminalnih pozicija kao P-pozicija. Zatim se identificiraju pozicije koje vode do P-pozicija i označavaju se kao N-pozicije. Nakon toga, traže se pozicije iz kojih se dolazi samo do N-pozicija i takve pozicije se označavaju kao P-pozicije, a to se sve ponavlja dok se ne odrede i identificiraju sve moguće pozicije u igri kao P-pozicije ili N-pozicije.

Ova podjela na P-pozicije i N-pozicije korisna je pri razvoju pobjedničke strategije u igri jer je cilj svakog igrača dovesti igru u P-pozicije, a to su pobjedničke pozicije. Također, početna pozicija određuje koji igrač ima pobjedničku strategiju. U idućem poglavlju analizirat ćemo P-pozicije i N-pozicije na primjerima igre 21 i igre Nim.

2.2. Primjeri kombinatornih igara

Postoje brojne popularne kombinatorne igre, a ovdje ćemo nabrojati samo neke od njih:

Igra oduzimanja

Vjerojatno najjednostavniji mogući primjer jedne nepristrane kombinatorne igre je *igra oduzimanja*. U igri oduzimanja dva igrača miču jedan, dva ili tri žetona sa hrpe koja se sastoji od 21 žetona. Gubitnik je onaj igrač koji ukloni zadnji žeton. Postoji mnogo mogućih varijacija ove igre, od broja dozvoljenih žetona koji se mogu ukloniti u jednom potezu, do ukupnog broja žetona na hrpi. U nastavku analize pokazat ćemo kako varijacije utječu na rezultat igre i strategiju igrača.

Nim

Igra Nim također je jedna jednostavna igra za dva igrača u kojoj se, za razliku od igre oduzimanja, koristi više hrpa žetona. Svaka hrpa žetona može imati proizvoljan broj žetona, a broj hrpa može biti različit od igre do igre. Na početku igre, igrači se dogovaraju o broju hrpa i broju žetona u svakoj hrpi.

Tijekom igre, igrači naizmjenice uzimaju žetone s ploče za igru. Svaki igrač mora uzeti žetone samo iz jedne hrpe, ali može uzeti bilo koji broj žetona iz te hrpe. Nakon što igrač uzme žetone, on ih uklanja s ploče za igru.

Cilj igre je ostati s barem jednim žetonom na ploči, tako da protivnik ne može uzeti zadnji žeton. Uzimanje zadnjeg žetona predstavlja gubitak igre, a pobjednik je onaj koji uspije natjerati protivnika da uzme zadnji žeton.



Slika 2.1: Igra Nim.

Hex

Igra Hex igra je za dva igrača koja se igra na šesterokutnoj ploči koja se sastoji od šestokutnih polja. Jedan igrač igra s bijelim figurama, a drugi s crnim figurama. Igrači igraju naizmjenično, stavljajući svoje figure na polja ploče.

Cilj igre je stvoriti poveznicu između dvije suprotne strane šesterokuta koju čine igračeve figure. Bijeli igrač pokušava stvoriti poveznicu između suprotne bijele strane ploče, dok crni igrač pokušava stvoriti poveznicu između suprotne crne strane ploče. Pobjeđuje igrač koji prvi uspije stvoriti poveznicu od jedne do druge strane ploče.

Figure se mogu postavljati na bilo koje slobodno polje na ploči. Igrači nastoje blokirati protivnika i povezati svoje figure kako bi stvorili poveznicu. Igrači također pokušavaju onemogućiti stvaranje poveznice za protivnika.

Go

Igra Go igra je za dva igrača koja se igra na ploči veličine 19x19 polja, ali može se igrati i na manjim pločama. Jedan igrač igra crnim figurama, dok drugi igrač igra bijelim figurama. Igrači igraju naizmjenično, stavljajući svoje figure na prazna polja ploče.

Cilj igre je osvojiti što više teritorija na ploči postavljanjem svojih figura na prazna polja i stvaranjem zatvorenih prostora oko tih figura. Figura se naziva kamenom, a igrači ih postavljaju na presjecišta linija koje čine ploču.

Kamenovi se mogu postavljati na bilo koje prazno polje, ali ne smiju se postavljati na polja koja su već okupirana od strane protivnika. Kada jedan igrač okruži prazna polja kamenovima, taj teritorij smatra se osvojenim.

Osim osvajanja teritorija, igrači također mogu zarobiti protivničke kamenove. Kamen se zarobi kada se okruži na sve strane praznim poljima ili poljima s kamenovima iste boje. Zarobljeni kamen se uklanja s ploče.

Igra Go je za razliku od ostalih navedenih igara vrlo složena i strateški izazovna igra u kojoj igrači pokušavaju predvidjeti poteze protivnika i stvoriti najbolju strategiju za osvajanje teritorija.

3. Analiza kombinatornih igara

U ovom poglavlju prikazat ćemo analizu kombinatornih igara i matematičku teoriju koja se nalazi u pozadini.

3.1. Analiza igre oduzimanja

Analizu igre oduzimanja započet ćemo od kraja prema početku. Ako je na hrpi ostao samo jedan, dva ili tri žetona igrač koji je idući na redu pobjeđuje uzimanjem svih preostalih žetona pa su zbog toga te pozicije N-pozicije.

Sada pretpostavimo da je ostalo četiri žetona. Ako igrač koji miče sljedeći uzme jedan ili dva žetona, protivnik mu može uzvratiti na način da ostavi hrpu sa jednim ili dva žetona, čime on dolazi u poziciju da gubi u igri. Međutim, ako igrač koji miče sljedeći uzme sva tri, ostavlja jedan žeton na hrpi. Tada njegov protivnik mora uzeti preostali žeton, a igrač koji je bio na potezu prije njega pobjeđuje pa je pozicija s 4 žetona P-pozicija.

Ako je na hrpi 5, 6 ili 7 žetona, trenutni igrač može uzeti dovoljno žetona da dovede protivnika u stanje igre u P-poziciju s 4 žetona. To znači da ako igrač uzme x žetona, njegov protivnik će morati uzeti $(4 - x)$ žetona, što će ga dovesti u poziciju s 4 žetona, gdje će izgubiti. Zbog toga su pozicije s 5, 6 ili 7 žetona na hrpi N-pozicije.

Kada se na hrpi nalazi 8 žetona, igrač koji je tada na potezu ne može ostaviti protivnika u P-poziciji s 4 žetona, što znači da je stanje igre s 8 žetona na hrpi klasificirano kao P-pozicija.

Slično kao u navedenim primjerima, možemo izvesti prikaz P i N pozicija za svako stanje igre pa tako dolazimo do prikaza svih P i N pozicija u igri oduzimanja.

Tablica 3.1: Prikaz P-pozicija i N-pozicija u igri oduzimanja (1-11)

0	1	2	3	4	5	6	7	8	9	10	11
P	N	N	N	P	N	N	N	P	N	N	N

Tablica 3.2: Prikaz P-pozicija i N-pozicija u igri oduzimanja(12-21)

12	13	14	15	16	17	18	19	20	21
P	N	N	N	P	N	N	N	P	N

Kao zaključak ove analize, možemo vrlo jednostavno definirati ova dva pravila:

1. Pozicija s koje je moguće pomaknuti se na P-poziciju je N-pozicija.
2. Ako je trenutna pozicija ona s koje se ne može doći do P-pozicije (to jest, svi su mogući potezi na N-pozicijama) ili ako nema više mogućih poteza, onda je to P-pozicija.

Lako je zaključiti da je svaka pozicija s n žetona, gdje je $n = 4k$, P-pozicija, a sve ostale pozicije ($n = 4k + 1$, $n = 4k + 2$, $n = 4k + 3$) su N-pozicije. Logičan dokaz toga jest činjenica da postoji potez od svakog broja koji nije djeljiv s 4 do onog koji je djeljiv s 4 (uzimajući jedan žeton u poziciji $4k + 1$, dva žetona u poziciji $4k + 2$ ili tri žetona u poziciji $4k + 3$).

Kada oba igrača primjete da je jasno od početka tko će biti pobjednik i da nakon prvog poteza samo izmjenjuju poteze, gdje jedan od njih uzima bilo koji broj žetona, a drugi odgovara tako da uzme ostatak od 4 igra postaje nezanimljiva, zbog čega ćemo u daljnjoj analizi nadograditi ovu igru tako što ćemo odvojiti žetone na više hrpa u igri Nim.

Analiza obrnutog oblika ove igre (misère) nije nimalo zahtjevnija od klasičnog oblika. U misère formi prisiljava se protivnika da mu na kraju ostane jedan žeton u posljednjem potezu, umjesto nijednog. Stoga su P-pozicije $n = 4k + 1$, a N-pozicije $n = 4k$, $n = 4k + 2$ i $n = 4k + 3$, a igrač koji započinje prvi je gubitnik.

3.2. Analiza igre Nim

N, P pozicije u igri Nim

Slično kao i u igri oduzimanja, pronalaženje najbolje strategije za pobjedu započinje od završne, terminalne pozicije. To je pozicija u kojoj nema više žetona na ploči za igru. Ova pozicija može se ostvariti samo iz pozicije s jednom hrpom na kojoj se nalazi bilo koji broj žetona pa je pozicija s jednom hrpom N-pozicija.

Igra s dvije hrpe nije nimalo kompliciranija. Prvi igrač bira hoće li ukloniti jedan žeton sa lijeve hrpe, jedan žeton sa desne hrpe ili oba žetona s desne hrpe. Ako ukloni

samo jedan žeton s desne hrpe na kojoj se nalaze dva žetona protivnik će morati ukloniti jedan od preostalih žetona pa će prvi igrač pobijediti. U suprotnom, ukloni li prvi igrač oba žetona ili žeton s lijeve hrpe omogućit će protivniku pobjedu.



Slika 3.1: Prikaz igre sa dvije hrpe.

Sada razmotrimo malo zahtjevniji primjer, kada se u igri nalaze tri hrpe sa po jednim, dva i tri žetona. Pristup pronalaženju najboljih poteza ostaje isti. Uzmimo primjer najjednostavnije pozicije s tri hrpe, $(1,1,1)$, kada se na svakoj od tri hrpe nalazi po jedan žeton. Vrlo je jednostavno zaključiti da u toj poziciji pobjeđuje prvi igrač jer ne postoji niti jedan potez koji bi omogućio drugom igraču da ostvari pobjedu. Zbog toga je pozicija $(1,1,1)$ N-pozicija.



Slika 3.2: Prikaz pozicije s jednim žetonom na svakoj hrpi.

Sljedeća pozicija koju bismo analizirali je pozicija $(1,1,2)$. Postoje tri moguća poteza koje igrač može odigrati iz ove pozicije: $(1,1)$, $(1,2)$ ili $(1,1,1)$. Pozicija $(1,2)$ je također N-pozicija jer igrač koji je tada na potezu može odigrati potez koji će poziciju pretvoriti u $(1,1)$, što je N-pozicija. Pozicija $(1,1,1)$ je također N-pozicija, kao što smo već vidjeli. Ostaje nam razmotriti poziciju $(1,1)$, koja predstavlja P-poziciju jer se radi o igri s dvije hrpe jednake veličine. S obzirom da postoji potez koji vodi iz pozicije $(1,1,2)$ u P-poziciju $(1,1)$, poziciju $(1,1,2)$ također možemo proglasiti N-pozicijom.

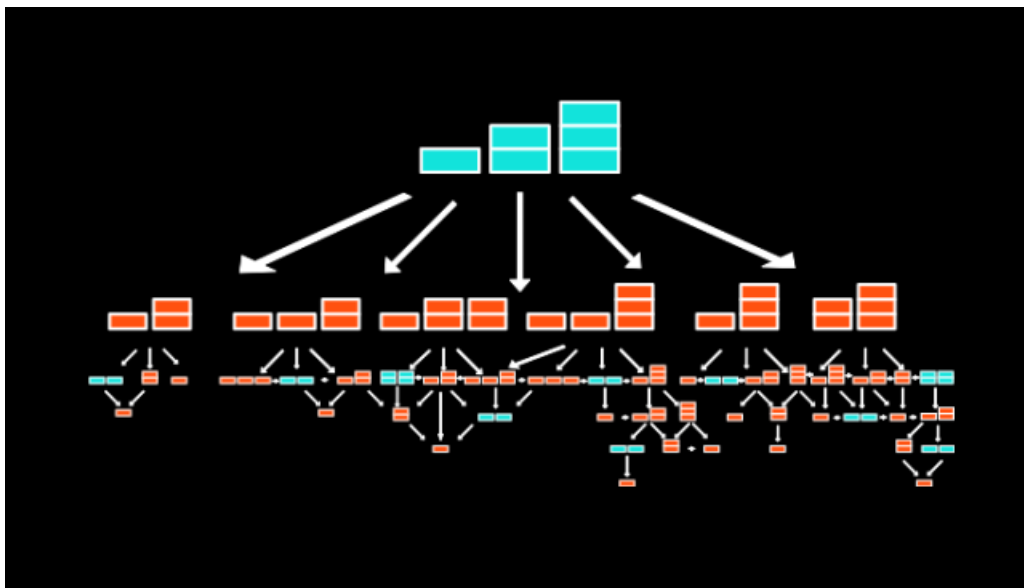


Slika 3.3: Prikaz mogućih poteza iz pozicije (1,1,2).

Sada ćemo analizirati poziciju (1,2,2). Svi mogući potezi iz ove pozicije su (1,1,2), (1,2) i (2,2). Pozicija (1,1,2) i pozicija (1,2) su N-pozicije, dok je pozicija (2,2) P-pozicija jer se radi o igri s dvije hrpe iste veličine. S obzirom da su svi mogući potezi iz pozicije (1,2,2) N-potezi, poziciju (1,2,2) također možemo proglasiti N-pozicijom.

Za analizu pozicije (1,2,3), postoji šest mogućih poteza. Pozicija (1,2) se identifikira kao N-pozicija jer se može reducirati na (1,1) i (1,2) koje su obje N-pozicije. Pozicija (1,1,2) također je N-pozicija jer se može reducirati na (1,1) i (2), obje N-pozicije. Pozicija (1,2,2) je N-pozicija jer se može reducirati na (1,2) i (2), obje N-pozicije. Na isti način reduciramo pozicije (1,1,3) na (1,3) i (1,1,1), (1,3) na (1,2) i (1), (2,3) na (1,2) i (1,1).

Kada se sve pozicije analiziraju, može se napraviti stablo svih mogućih pozicija, koje pokazuje da se oba gore navedena pravila mogu primjeniti na igru Nim s tri hrpe: svaka P-pozicija ima samo djecu pozicije koje su N-pozicije, a svaka N-pozicija ima barem jedno dijete koje je P-pozicija. Na slici je svaka P-pozicija prikazana plavom bojom, a N-pozicija crvenom bojom.



Slika 3.4: Stablo svih mogućih pozicija u igri Nim s tri hrpe.

Ako se ova analiza proširi na igru s više hrpa, proces pronalaženja P i N-pozicija postaje složeniji, ali osnovni principi ostaju nepromjenjeni.

Nim vrijednost

Svaku P i N-poziciju u igri Nim moguće je dobiti analizom u prethodnom poglavlju, no postavlja se pitanje praktičnosti analize igara s više hrpa, a samim time i većim veličinama hrpa. Zbog toga moramo uvesti bolje načine analize pozicija u igri Nim.

Jedan od novih i boljih načina analiza pozicija u igri Nim predstavili su matematičari Roland P. Sprague i Patrick M. Grundy.[2] Oni koriste binarne brojeve za opisivanje svakog stanja igre Nim, pri čemu se dekadski broj žetona na svakoj hrpi prebacuje u binarni oblik. Ukupna vrijednost pozicije nije samo vrijednost jedne hrpe, nego zbroj svih binarnih vrijednosti.

Definicija Nim zbrajanja glasi:

Definicija 2. $(x_n x_{n-1} \dots x_1 x_0) \oplus (y_n y_{n-1} \dots y_1 y_0) = (z_n z_{n-1} \dots z_1 z_0)$, gdje je svaki $z = (x + y) \bmod 2$.

Pomoću ove definicije možemo za svaku poziciju odrediti njezin Nim zbroj. No kako odrediti je li ta pozicija P-pozicija ili N-pozicija? Uzmimo ponovno za primjer igru Nim sa tri hrpe iz gornjeg primjera. Pretvorbom dekadске vrijednosti broja žetona na svakoj hrpi u binarni oblik dobivamo vrijednosti 1, 10 i 11. Nim zbroj tih binarnih vrijednosti daje rezultat 0.

$$1 = (1)_2$$

$$2 = (10)_2$$

$$3 = (11)_2$$

$$\oplus = (00)_2$$

Iz gornjeg primjera već znamo da je ta pozicija P-pozicija. Uzmimo sada kao primjer poziciju (1,2,2), za koju već znamo da je N-pozicija. Pretvorbom dekadskih vrijednosti u binarni oblik i Nim zbrajanjem dobivamo vrijednost 100.

$$1 = (1)_2$$

$$2 = (10)_2$$

$$2 = (10)_2$$

$$\oplus = (01)_2$$

Na temelju navedenih primjera možemo zaključiti da su sve P-pozicije u igri Nim vrijednosti jednake 0, dok su sve N-pozicije u igri Nim vrijednosti različite od 0. To se može izraziti kroz sljedeći korolar.

Korolar 1. [6] *Pozicija (x_1, \dots, x_n) , $n \in \mathbb{N}$ je P-pozicija ako i samo ako $x_1 \oplus \dots \oplus x_n = 0$*

Funkcionalnosti ove dvije obrađene metode mogu se činiti sličnima, no analiza Nim vrijednosti ima veliki utjecaj tijekom igranja igre jer pruža brzu i jednostavnu metodu pronalaženja najboljeg poteza iz bilo koje pozicije u igri. Dok N-pozicije i P-pozicije pokazuju samo karakter trenutne pozicije bez metode određivanja kako odabrati najbolji mogući potez, pomoću Nim vrijednosti lako se može odrediti najbolja opcija za kretanje kroz igru.

Na primjer, zamislimo igru Nim sa tri hrpe koja ima više od tri žetona na svakoj hrpi. Uzmimo za primjer da su količine žetona na svakoj hrpi 25, 21 i 10. Pomoću N-pozicija i P-pozicija teško ćemo odrediti koji potez odigrati, no pomoću Nim vrijednosti to možemo vrlo lako učiniti. Pretvorbom u binarni oblik dobivamo vrijednosti 11001, 10101 i 1010, a Nim zbrajanjem dobivamo vrijednost 6, što je N-pozicija.

$$25 = (11001)_2$$

$$21 = (10101)_2$$

$$10 = (1010)_2$$

$$\oplus = (100)_2$$

Najbolji izbor za igrača na potezu bi trebao biti svaki potez koji ima vrijednost Nim zbroja 0, što može lako napraviti ako pogleda zapis zbroja. Na primjer može uzeti dva žetona sa hrpe koja ima 21 žeton, što odmah mijenja Nim vrijednost u 0. Taj potez ostavlja protivnika u gubitničkoj poziciji i omogućava prvom igraču da dođe do pobjedničke pozicije.

$$25 = (11001)_2$$

$$19 = (10011)_2$$

$$10 = (1010)_2$$

$$\oplus = (000)_2$$

Sprague-Grundyjeva funkcija

Općenitiji način za analizu nepristrane igre je korištenje Sprague-Grundyjeve funkcije (S-G funkcija)[4] [5]. Prvo moramo upotrijebiti neke nove zapise koji općenito opisuju nepristrane igre kako bi se kasnije mogla definirati sama S-G funkcija. Ovaj novi zapis bit će predstavljen parom (X, F) , gdje X označava sve pozicije u igri, a F je funkcija koja definira sve moguće pozicije $x \in X$. Zbog činjenice da se svi potezi uvijek izvode iz jedne pozicije igre na skup pozicija, funkcija F nam daje podskup $F(x)$ od X na svaku poziciju $x \in X$. Drugim riječima, ova funkcija dodjeljuje sve dostupne pozicije za bilo koju odabranu poziciju u igri.

Formalna definicija Sprague-Grundyjeve funkcije slijedi u nastavku:

Definicija 3. [2] *Sprague-Grundyjeva funkcija grafa (X, F) je funkcija g , definirana na skupu X , koja uzima nenegativne cjelobrojne vrijednosti, pri čemu vrijedi:*

$$g(x) = \min (n \geq 0 : n \neq g(y), y \text{ in } F(x))$$

Sama definicija na prvi pogled izgleda poprilično zbunjujuća zbog korištenja rekurzije u definiciji. Međutim, algoritam postaje vrlo jednostavan za razumijevanje i primjenu kada se počne graditi od završne, terminalne pozicije. Sve terminalne pozicije imaju vrijednost 0, a skup $F(x)$ je prazan. Vrijednost g od terminalnog položaja

tada se dodjeljuje kao broj jednak minimalnoj vrijednosti skupa $n \in \{0, 1, 2, \dots\}$, što je 0. Uzimajući u obzir poziciju čiji je jedini sljedbenik terminalna pozicija, vrijednost funkcije postaje $g(x) = 1$.

Iz ovih uvjeta slijedi da je pozicija x P-pozicija ako je vrijednost Sprague-Grundyjeve funkcije 0, inače su sve ostale pozicije N-pozicije:

- Ako je x terminalna pozicija, tada je $g(x) = 0$.
- Za poziciju x za koju je $g(x) = 0$, svi su sljedbenici y od x takvi da je $g(y) \neq 0$.
- Za poziciju x za koju je $g(x) \neq 0$, postoji barem jedan sljedbenik y za koji je $g(y) = 0$.

Iz tih uvjeta možemo izvesti iduću propoziciju:

Propozicija 1. *Ako je vrijednost funkcije $g(x)$ jednaka 0 za određeni vrh $x \in X$, tada se vrh x označava kao P-vrh na grafu.*

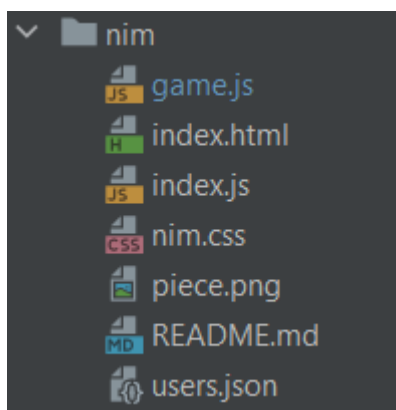
Iz tih uvjeta lako možemo vidjeti da postoji sličnost u funkcionalnosti između analize pomoću S-G funkcije i analize P-pozicija i N-pozicija. Procjena prema N-pozicijama i P-pozicijama označava sva stanja koja mogu dosegnuti P-poziciju iz N-pozicije, dok S-G funkcija svemu tome dodjeljuje broj različit od 0. S druge strane, P-pozicija je definirana kao ona čiji su jedini sljedbenici N-pozicije, što se izražava S-G vrijednošću 0 i kao ona pozicija sa sljedbenicima S-G vrijednosti različitih od 0.

Kako bismo analizu kombinatornih igara učinili što pristupačnijom i kao rezultat cjelokupne teorijske analize, u idućim poglavljima analizirat ćemo izrađenu interaktivnu web aplikaciju koja implementira jednu verziju igre Nim. To će omogućiti korisnicima da eksperimentiraju s igrom i bolje usavrše razumijevanje koncepata igre o kojima se govori u ovom radu.

4. Tehničke značajke razvijene aplikacije

4.1. Struktura aplikacije

Ova aplikacija napravljena je pomoću HTML-a, CSS-a i JavaScript-a, a sastoji se od sljedećih dijelova: jedne HTML datoteke, jedne CSS datoteke, dvije JavaScript datoteke, jedne json datoteke i jedne slike koja prikazuje lopticu na ploči za igru. Za izradu sučelja korišten je HTML u datoteci `index.html`, a za stil korisničkog sučelja korišten je CSS u datoteci `nim.css`. Glavna logika programa napisana je pomoću jezika JavaScript, u datotekama `game.js` i `index.js`. Baza podataka je datoteka `users.json` u kojoj su pohranjeni podaci o korisnicima i o njihovim igrama.



Slika 4.1: Prikaz strukture koda aplikacije.

Datoteka `index.js` sastoji se od tri funkcije: `processPostRequest()`, `checkCredentials()` i `createServer()`, koje služe za provjeru korisničkih podataka, obradu POST zahtjeva i kreiranje servera za pokretanje aplikacije.

Datoteka `game.js` prikazuje cjelokupnu logiku igre, a sastoji se od 15 funkcija gdje svaka funkcija ima svoju svrhu u samom programu. Funkcije koje se koriste u ovoj datoteci su:

- showFrontPage() – prikaz početne stranice
- showGameForm() - prikaz stranice s postavkama igre
- showRanks() – prikaz top liste najboljih igrača
- showRules() – prikaz pravila igre
- showGameDiv() – prikaz stranice sa igrom
- resetGameDiv() – uklanja stranicu s igrom
- playGame() – inicijalizacija nove igre
- restartGame() – ponovno pokretanje nove igre
- login() – prijava korisnika u svoj profil
- logout() – izlazak korisnika iz svog profila
- leaveGame() – izlazak iz igre
- Board() – prikaz ploče za igru
- NimGame() – stvaranje nove igre
- Piece() – stvaranje loptice na ploči za igru
- PC() – određivanje poteza računala

4.2. Funkcionalnosti aplikacije

Funkcija checkCredentials() kao parametre prima korisničko ime i lozinku korisnika. Na početku pomoću funkcije createHash kreira hash lozinke korisnika. Ako to korisničko ime ne postoji u datoteci users.json, onda ga zajedno s tim hashom ubacuje u datoteku users.json, inače javlja grešku da je upisana lozinka pogrešna.

Funkcija processPostRequest() kao parametre prima varijable zahtjeva i odgovora. Ona obrađuje POST zahtjeve ovisno o rezultatu poziva funkcije checkCredentials(). Ako je rezultat poziva funkcije jednak 2, vraća status 500, što označava grešku, a ako je rezultat poziva funkcije jednak 1 vraća status 400 s porukom da je korisnik registriran sa drugom lozinkom. U protivnom nema greške i vraća status 200. Na kraju datoteke pomoću funkcije createServer kreira se server na linku <https://nim-projekt.onrender.com> pomoću kojega se pokreće cijeli program.

```

http.createServer( requestListener: function(req : InstanceType<IncomingMessage> , res :
    switch(req.method){
        case "POST":
            processPostRequest(req, res);
            break;
        default:
            res.writeHead(501, headers);
            res.end();
            break;
    }
}).listen( port: 3000);

```

Slika 4.2: Prikaz koda funkcije `createServer()`.

Na početku datoteke `game.js` definiraju se sve varijable potrebne za pisanje programskog koda datoteke. Funkcija `showFrontPage()` prikazuje početnu stranicu i miče sve ostale elemente. Funkcija `showGameForm()` također miče sve ostale elemente, ali i ako igra još traje prikazuje ploču za igru i loptice. Također miče i formu u kojoj korisnik upisuje postavke igre. Ako korisnik nije ulogiran prikazuje formu za logiranje, a u suprotnom prikazuje formu za prikaz postavki igre. Ako upisana lozinka nije valjana prikazuje upozorenje o pogrešnoj lozinki, a isto to čini i za krivi upis veličine stupaca za igru.

```

function showFrontPage(){
    for(let i=0; i<divsArray.length; i++)
        document.getElementById(divsArray[i]).style.display = "none";
    homePageDiv.style.display = "block";
}

```

Slika 4.3: Prikaz koda funkcije `showFrontPage()`.

Funkcija `showRanks()` prikazuje top listu najboljih igrača. U tablici top lista nalaze se: korisničko ime igrača, ukupan broj njegovih igara, broj pobjeda, broj poraza i omjer njegovih pobjeda i poraza, a svi ti podaci upisuju se u `users.json` datoteku pomoću koje se ti podaci prikazuju korisniku. Tablica je sortirana prema omjeru pobjeda i poraza tako što je igrač sa najboljim omjerom na vrhu tablice. Funkcija `showRules()` prikazuje pravila igre nakon što korisnik klikne gumb za prikaz pravila igre, a s druge strane funkcije `showGameDiv()` i `resetGameDiv()` prikazuju i miču ploču s lopticama.

```

for(let i=0; i<localStorage.length; i++){
    let jsonUsername = localStorage.key(i);
    let json = JSON.parse(localStorage.getItem(localStorage.key(i)));
    finalText +=
        "<tr>" +
        "<td>" + jsonUsername + "</td>" +
        "<td>" + json.games + "</td>" +
        "<td>" + json.victories + "</td>" +
        "<td>" + (json.games - json.victories) + "</td>" +
        "<td>" + ((json.victories / json.games)*100).toFixed( fractionDigits: 2) + "%" + "</td>";
    finalText += "</tr>";
}

```

Slika 4.4: Prikaz tablice top lista iz funkcije showRanks().

```

function showRules(){
    for(let i=0; i<divsArray.length; i++)
        document.getElementById(divsArray[i]).style.display = "none";
    rulesDiv.style.display = "block";
}

function showGameDiv(){
    for(let i=0; i<divsArray.length; i++)
        document.getElementById(divsArray[i]).style.display = "none";
    gameDiv.style.display = "block";
}

function resetGameDiv(){
    while (gameDiv.firstChild)
        gameDiv.removeChild(gameDiv.firstChild);
}

```

Slika 4.5: Prikaz koda funkcija showRules(), showGameDiv() i resetGameDiv().

Funkcija playGame() pomoću odabranih vrijednosti korisnika o veličini polja i igraču s prvim potezom inicijalizira novu igru, a prije toga provjerava je li upisana vrijednost veličine polja ispravna. Funkcija restartGame() samo ponovno poziva prethodno navedenu funkciju. Pomoću funkcije login() POST zahtjevom šalje se korisničko ime i lozinka igrača u datoteku users.json gdje se vodi evidencija o svim njegovim igrama, što se prikazuje u tablici igrača. Također, korisničko ime prikazuje se u gornjem desnom dijelu ekrana kako bi igrač mogao znati da je ulogiran u aplikaciju. Ako se igrač odluči izaći iz profila dok traje igra dobiva upozorenje da se to ne dozvoljava dok igra još traje, nego mu se to dozvoljava tek kada već započeta igra završi pa se prikazuje početna stranica. Također, kada igrač odluči izaći iz igre, ali ostati u svom

profilu prikazuje mu se poruka o porazu i nudi se pokretanje nove igre.

```
function playGame(){
    let firstPlayer = playFirstForm.elements["playFirstButton"].value;
    let boardSize = boardSizeForm.elements["boardSizeInput"].value;

    if(boardSize%1!==0.0 || boardSize<=1 || boardSize>=7){
        showGameForm( goodPassword: true, goodBoardSize: false);
        return;
    }

    mainGame = new NimGame(firstPlayer, boardSize);
    mainGame.initiateGame();
}
```

Slika 4.6: Prikaz koda funkcije playGame().

```
function login(){
    username = userInput.value;
    password = passwordInput.value;
    let js_obj = {"nick": username, "pass": password};
    let xhr = new XMLHttpRequest();
    xhr.open( method: "POST", url: url+"register", async: true);

    xhr.onreadystatechange = function() {
        if(this.status === 200){
            showLoginDiv.style.display = "block";
            showLoginText.innerHTML = "Bok, " + username + "!";
            loginFlag=true;
            if(localStorage[username] == null)
                localStorage[username] = JSON.stringify( value: {"victories": 0, "games": 0});
            showGameForm( goodPassword: true, goodBoardSize: true);
        } else if(this.status === 400)
            showGameForm( goodPassword: false, goodBoardSize: true);
    }
    xhr.send(JSON.stringify(js_obj));
}
```

Slika 4.7: Prikaz koda funkcije login().


```
function logout(){
    if(gameInProgress){
        showLoginText.innerHTML = "Igra još traje!";
        setTimeout(function(){ showLoginText.innerHTML = "Bok, " + username + "!"; }, 4000);
        return;
    }

    showLoginDiv.style.display = "none";
    loginFlag=false;
    showFrontPage();
}
```

Slika 4.8: Prikaz koda funkcije logout().

Ploča za igru izgrađena je od loptica koje su poredane uzlazno po stupcima tako što je u prvom stupcu jedna loptica, a svakim stupcem broj loptica se povećava za jednu, sve dok ne dođe do maksimalne razine koju je odredio korisnik u svojim početnim postavkama igre. Funkcija NimGame() inicijalizira novu igru tako što postavlja prvog igrača na potezu i kreira novu ploču za igru. Također, postavlja i poruku koji je igrač trenutno na potezu, koja se izmjenjuje svakim potezom, te oznaku da je igra u tijeku kako igrač ne bi mogao izaći iz svog profila tijekom igre.

```
this.createBoard = function(){
    this.boardDiv = Object.assign(document.createElement( tagName: 'div'), source: {id:"boardDiv"});
    this.boardDiv.className = "boardDiv";
    this.boardDiv.style.width = "" + (90 * this.xMax) + "px";
    gameDiv.appendChild(this.boardDiv);

    for(let i=0; i<this.yMax; i++){
        this.boardQuantityArray.push(i+1);
        let tempDiv = document.createElement( tagName: "div");
        tempDiv.id = "cellDiv" + i;
        tempDiv.className = "cellDiv";
        for(let j=i; j>=0; j--){
            let piece = new Piece(i, j);
            tempDiv.appendChild(piece.html);
        }
        this.boardDiv.appendChild(tempDiv);
    }
}
```

Slika 4.9: Prikaz koda za stvaranje ploče za igru.

```

this.initiateGame = function(){
    this.board = new Board(this.boardSize, this.boardSize);
    this.pc = new PC();
    this.moves = 0;

    gameInProgress = true;
    resetGameDiv();
    let messageH = document.createElement( tagName: "h1");
    messageH.id = "messageH1";

    document.getElementById( elementId: "gameDiv").appendChild(messageH);
    this.board.createBoard();
    showGameDiv();
    leaveGameDiv.style.display = "block";
    this.updateMessageDiv();

    let _this = this;
    if(this.firstPlayer==="pc")
        setTimeout(function() {_this.pc.move(); }, 1500);
}

```

Slika 4.10: Prikaz koda za inicijalizaciju nove igre.

Klikom na lopticu ona se miče iz igre, a ako je to zadnja loptica na ploči, označava se kraj igre sa odgovarajućom porukom o pobjedniku, inače se provjerava koji je igrač bio na potezu i dozvoljava se drugom igraču da je na redu. Kada je igra gotova, u datoteku users.json igraču se dodaje nova igra i upisuje se je li pobijedio ili izgubio igru pa se sukladno tome računa i novi omjer pobjeda u igri. Korisnik tada može odlučiti igrati s istim ili drugačijim postavkama, a može i izaći iz profila i prestati igrati. Odlukom o izlasku iz igre dok igra traje također se u datoteku users.json dodaje nova igra i upisuje se da je korisnik poražen.

Glavni algoritam ovog programa je određivanje kakav će potez odigrati računalo i koje će loptice i u kojem stupcu ukloniti. Naime, u igri Nim s dva igrača, ako imamo n stupaca loptica (gdje se može ukloniti bilo koji broj loptica iz jednog stupca u jednom potezu) gubitničke su pozicije one u kojima je xor veličine stupca nije jednak 0. Zbog toga, računalo uvijek bira poteze u kojima (ako je to moguće na temelju korisnikovih poteza) xor veličine stupca nije 0 kako bi natjeralo korisnika da izgubi u igri.

```

this.move = function(){
    for(let i=0; i<mainGame.board.boardQuantityArray.length; i++){
        for(let j=0; j<mainGame.board.boardQuantityArray[i]; j++){
            let oldValue = mainGame.board.boardQuantityArray[i];
            mainGame.board.boardQuantityArray[i] = j;
            if(this.xor() !== 0)
                mainGame.board.boardQuantityArray[i] = oldValue;
            else {
                mainGame.board.boardQuantityArray[i] = oldValue;
                mainGame.deletePiece(i, j);
                return;
            }
        }
    }
    let x = Math.floor(Math.random() * mainGame.board.boardQuantityArray.length);
    while(mainGame.board.boardQuantityArray[x]===0)
        x = Math.floor(Math.random() * mainGame.board.boardQuantityArray.length);
    mainGame.deletePiece(x, mainGame.board.boardQuantityArray[x]-1);
}

```

Slika 4.11: Prikaz određivanja novog poteza računala.

```

this.xor = function(){
    let value = 0;
    for(let i=0; i<mainGame.board.boardQuantityArray.length; i++)
        value ^= mainGame.board.boardQuantityArray[i];
    return value;
}

```

Slika 4.12: Prikaz funkcije xor().

Pomoću funkcije Piece() stvara se nova loptica na ploči za igru. Kada je korisnik prešao mišem preko određenog broj loptica pomoću funkcije onmouseover označene loptice postaju bijele kako bi korisnik znao koje bi loptice uklonio eventualnim potezom. S druge strane, ako korisnik odustane od poteza, boja loptica ponovno postaje kakva je bila na početku bez uklanjanja. Tek klikom na lopticu korisnik uklanja loptice iz igre i čini svoj potez u igri, a loptice se tada brišu iz igre.

```

this.html.onclick = function(){
    if(this.className!="pieceDeleted" && ((mainGame.moves%2===0 && mainGame.firstPlayer=="player") ||
        (mainGame.moves%2!==0 && mainGame.firstPlayer!="player"))){
        this.deletePiece();
    }
}

this.html.deletePiece = function(){
    let length = this.id.length;
    let temp = this.id.indexOf( "|" );
    let x = parseInt(this.id.slice(5, temp));
    let y = parseInt(this.id.slice(temp + 1, length));
    mainGame.deletePiece(x, y);
}

```

Slika 4.13: Prikaz funkcije za uklanjanje loptice iz igre.

5. Upute za korištenje razvijene aplikacije

5.1. Pristupanje igri

Nakon pokretanja programa na zaslonu korisnika prikazuje se grafičko korisničko sučelje (prikazano na slici 1.) na kojemu je prikazano: tipka za prikaz top liste najboljih igrača, tipka za igranje nove igre gdje se prikazuje forma za registraciju i logiranje u aplikaciju, tipka za prikaz pravila igre. Pritiskom na tipku za prikaz top liste najboljih igrača korisniku se prikazuje tablica (slika 2.) na kojoj se mogu vidjeti broj pobjeda, broj poraza, ukupan broj igara i omjer pobjeda i poraza svakog igrača.

Tablica je sortirana po omjeru pobjeda i poraza svakog igrača pa je tako na vrhu prikazan igrač sa najboljim omjerom. Kada korisnik pritisne tipku za pokretanje nove igre, otvara se mogućnost da se registrira kao novi korisnik ukoliko ne posjeduje račun s odgovarajućim korisničkim imenom i lozinkom. Ista se forma prikazuje (slika 3.) i za prijavu i za registraciju novog korisnika aplikacije jer program sam prepoznaje postoji li u bazi korisnik sa upisanim korisničkim imenom i lozinkom.

Ako ne postoji račun sa nekim korisničkim imenom, onda program uspješno registrira novog korisnika. U suprotnom ako već postoji račun sa nekim korisničkim imenom, onda program dozvoljava registraciju samo za jednom lozinkom, inače šalje upozorenje da je lozinka pogrešna. Pritiskom na tipku pravila korisnik dobiva kratki prikaz najvažnijih pravila ove igre, što je i prikazano na donjim slikama.



Slika 5.1: Stanje programa nakon pokretanja.

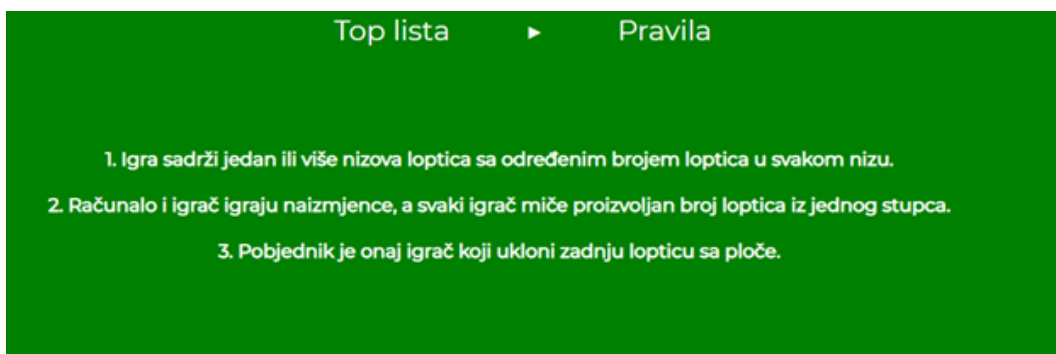
The screenshot shows the same green background as Slika 5.1. The 'Top lista' button is now highlighted in white. Below it, a table with 5 columns and 4 rows is displayed. The columns are labeled 'Igrač', 'Ukupno', 'Pobjeda', 'Poraz', and 'Omjer'. The rows contain data for three players: filip2, filip, and filip3.

Igrač	Ukupno	Pobjeda	Poraz	Omjer
filip2	14	4	10	28.57%
filip	82	19	63	23.17%
filip3	18	3	15	16.67%

Slika 5.2: Stanje programa nakon pritiska na tipku top lista.



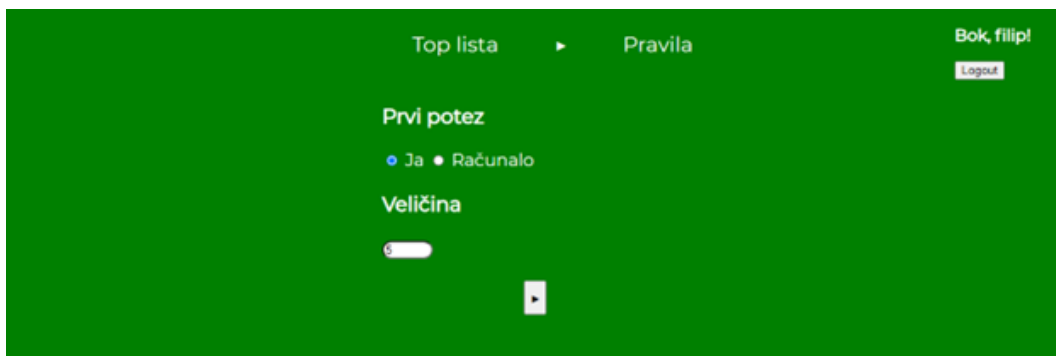
Slika 5.3: Forma za prijavu/registraciju korisnika.



1. Igra sadrži jedan ili više nizova loptica sa određenim brojem loptica u svakom nizu.
2. Računalo i igrač igraju naizmjenice, a svaki igrač miče proizvoljan broj loptica iz jednog stupca.
3. Pobjednik je onaj igrač koji ukloni zadnju lopticu sa ploče.

Slika 5.4: Prikaz pravila igre.

Nakon uspješnog ulaska u svoj profil korisniku se prikazuje forma na kojoj odlučuje hoće li on ili računalo biti prvi na potezu i koliko će stupaca loptica sadržavati igra. Na početku je zadano da je korisnik prvi na potezu i da se ploča sastoji od 5 stupaca loptica (stupci sadrže po 1,2,3,4 i 5 loptica). Broj stupaca može biti između 2 i 10, u suprotnom korisnik dobiva upozorenje da je neispravna veličina ploče. Kada je korisnik zadovoljan sa postavljenim početnim postavkama i kada su postavke ispravne korisnik može započeti novu igru pritiskom na donju tipku, a ako želi odustati u gornjem desnom dijelu može pritisnuti na tipku za izlaz.



Slika 5.5: Prikaz forme za pokretanje nove igre.

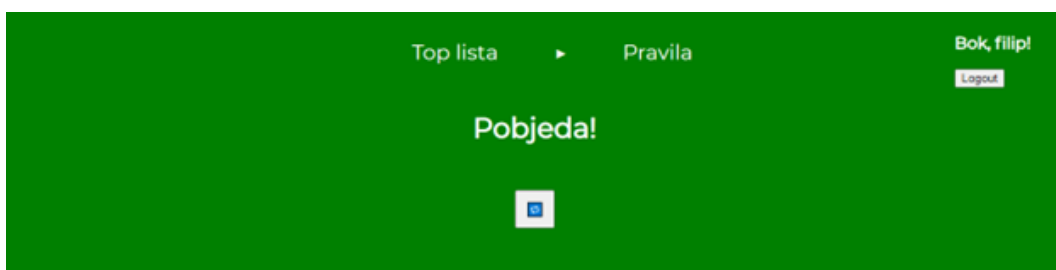
5.2. Prikaz igre

Nakon pokretanja nove igre na zaslonu korisnika prikazuje se grafičko korisničko sučelje (prikazano na slici 6.) na kojemu je prikazano koji je igrač trenutno na potezu i ispod toga trenutno stanje ploče za igru, odnosno broja loptica koji još nisu uklonjeni iz igre. Na slici je prikazana nova igra po početno zadanim postavkama na prvoj stranici igre. Prelaženjem miša preko loptica one postaju bijele i time korisnik vidi koje će loptice tim potezom ukloniti. Uklanjanjem miša od loptica one ponovno postaju sive, a klikom na loptice korisnik čini svoj potez u igri čime se oni više ne vide na ploči i time prepušta računalu da izvrši svoj sljedeći potez. Korisnik u jednom potezu može ukloniti loptice samo iz jednog stupca. Korisnik i računalu se međusobno izmjenjuju sve dok se zadnja loptica ne ukloni sa ploče.

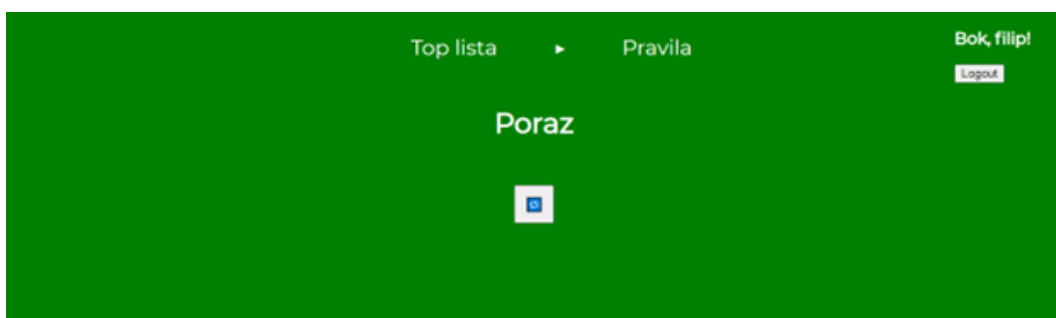
Cilj ove igre je natjerati protivničkog igrača da on ne uklanja zadnju lopticu sa ploče pa je u tom slučaju on gubitnik u igri. Nakon što ploča ostane bez ijedne loptice sukladno gore navedenim pravilima prikazuje se poruka da je korisnik pobijedio ili u drugom slučaju izgubio igru (prikazano na slici 7. i 8.) i ispod toga gumb za pokretanje nove igre. Gumb za povratak na početnu stranicu je također prikazan u gornjem desnom dijelu u slučaju da korisnik nakon igre želi odustati i izaći iz programa. U slučaju da korisnik pritisne tipku za izlaz iz programa dok je igra pokrenuta prikazuje se upozorenje da igra još traje, što znači da korisnik ako želi odustati od igre prvo mora pritisnuti donju tipku za izlaz kako bi završio igru, a tek onda izaći iz programa kako bi se pokrenuta igra uspješno završila.



Slika 5.6: Stanje programa nakon odabira tipke za početak nove igre.



Slika 5.7: Stanje programa nakon pobjede igrača.



Slika 5.8: Stanje programa nakon poraza igrača.

6. Zaključak

U ovom radu detaljno je proučena teorija i analiza kombinatornih igara. Teorija pruža alate i koncepte koji omogućuju modeliranje i analizu strateške interakcije u igrama. Kroz primjenu Grundyjeve funkcije, koja se temelji na teoriji grafova, možemo odrediti pobjedničke strategije u kombinatornim igrama.

Naglasak u izradi ovog rada je na progresivnim kombinatornim igrama, s posebnim fokusom na analizi niza modifikacija igre Nim. Primjena analitičke tehnike iz teorije grafova omogućila je dublje razumijevanje strukture i dinamike igre i identifikaciju optimalnih strategija za postizanje pobjede.

Uz teorijsku analizu, razvijena je i interaktivna web aplikacija koja implementira jednu verziju igre Nim. Ova aplikacija omogućuje korisnicima da igraju igru i istraže različite scenarije i strategije.

S druge strane, kroz matematičke formalizme i analitičke tehnike, možemo bolje razumjeti kompleksne igre i identificirati optimalne strategije. Ovi rezultati imaju široku primjenu u područjima kao što su računalna znanost, matematika, inženjerstvo i društvene znanosti.

U zaključku, ovaj rad pruža sveobuhvatnu analizu kombinatornih igara. Njegovi rezultati i primjena Grundyjeve funkcije ukazuju na mogućnosti optimizacije strategija u igri Nim i sličnim situacijama.

LITERATURA

- [1] E. R. Berlekamp, J. H. Conway, i Richard K. Guy. *Winning ways for your mathematical plays, 2nd edition*. Wellesey, SAD, 2001.
- [2] T.S Ferguson. *Game Theory*. UCLA, 2008.
- [3] W. Garner. *A Brief Introduction to Combinatorial Game Theory*. MathZoom Summer Camp, 2007.
- [4] P.M Grundy. *Mathematics and Games*. Eureka, 1939.
- [5] R.P. Sprague. *Über mathematische Kampfspiele*. Tohoku Mathematical Journal, 1973.
- [6] A. Tucker. *Applied Combinatorics, 6th edition*. John Wiley & Sons, Inc., 2012.

Igre na grafovima

Sažetak

Ovaj rad proučava progresivne kombinatorne igre i predstavlja osnove analize pobjedničkih strategija. Primjenom Grundyjeve funkcije istražuju se osnovni rezultati i tehnike za odabir pobjedničkih strategija u nizu modifikacija igre Nim. Također, izrađena je interaktivna web aplikacija koja implementira jednu verziju igre Nim. Rezultati potvrđuju važnost matematičkog pristupa u analizi kombinatornih igara i njenu primjenjivost u inženjerskim područjima.

Ključne riječi: teorija grafova, kombinatorne igre, pobjedničke strategije, Grundyjeva funkcija, igra Nim, web aplikacija

Graph games

Abstract

This thesis examines progressive combinatorial games and presents the fundamentals of analyzing winning strategies. Through the application of Grundy's function, basic results and techniques for selecting winning strategies in a series of modifications of the game Nim are explored. Additionally, an interactive web application has been developed to implement one version of the game Nim. The results confirm the importance of a mathematical approach in the analysis of combinatorial games and its applicability in engineering fields.

Keywords: graph theory, combinatorial games, winning strategies, Grundy function, Nim game, web application