

**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Ada Tikvan

IMPLEMENTACIJA IGRE S PREDIKTIVNIM AI PROTIVNIKOM –MISERE NIM

SEMINAR

UVOD U UMJETNU INTELIGENCIJU

Varaždin, 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Ada Tikvan

Matični broj: 0016158840

Studij: Informacijski i poslovni sustavi

**IMPLEMENTACIJA IGRE S PREDIKTIVNIM AI PROTIVNIKOM
–MISERE NIM**

SEMINAR

Mentor:

doc. dr. sc. Bogdan Okreša Đurić

Varaždin, siječanj 2025.

Izjava o izvornosti

Izjavljujem da je ovaj seminar izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autorica potvrdila prihvatanjem odredbi u sustavu FOI Radovi

Sažetak

Ovaj seminarski rad prikazuje implementaciju igre *Misere Nim* s prediktivnim AI protivnikom. Misere verzija igre upućuje da je cilj igre da pobjednik zapravo "izgubi" kako bi pobijedio u Misere varijanti. Nim je stara strategijska matematička igra u kojoj igrači imaju postavljene neke objekte na stolu (šibice, kockice), a cilj je igre da igrači tijekom igre uzimaju po 1 ili 2 predmeta s jedne ili više hrpa, ali proračunato jer da bi igrač pobijedio mora zadnji uzet 1 ili 2 predmeta sa stola. Opisana verzija predstavlja standardnu verziju igre, a implementirana je Misere verzija igre u kojoj je cilj pobjednika da ne uzme zadnji predmet jer bi tako izgubio u ovoj verziji. Za stvaranje AI protivnika korišten je minimax algoritam koji maksimizira poteze AI-ja kako bi minimizirao poteze igrača, odnosno kako bi AI sa svojim najboljim, maksimalnim potezima došao do pobjede. U Misere verziji, potezi za AI su zapravo minimizirani jer AI ne želi uzeti zadnji predmet već želi natjerati protivnika svojim potezima da to učini.

Ključne riječi: Nim, Misere, AI, minimax algoritam, najbolji potez, dubina rekurzije

Sadržaj

1. Uvod	1
2. Metode i tehnike rada	3
3. Kritički osvrt	4
4. Opis implementacije	5
5. Opis rada aplikacije	6
6. Zaključak	8
Popis literature	9
Popis slika	10

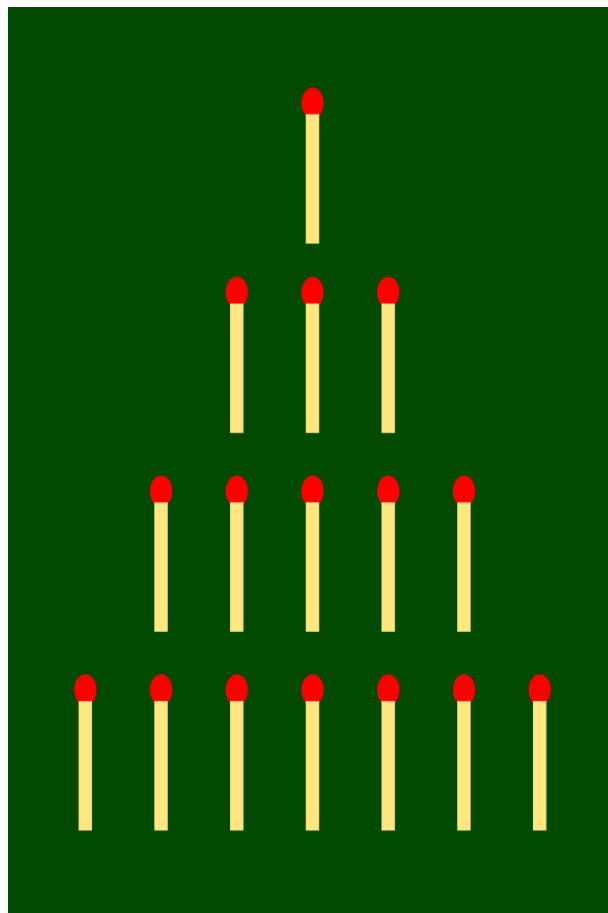
1. Uvod

Nim je stara matematička igra za koju se vjeruje da potječe iz Kine, a u 20. stoljeću je prvi put razvijena teorija igre kakvu mi danas poznajemo. [1]

Radi se o igri za 2 igrača u kojoj igrači imaju jednu ili više hrpa šibica (ili nekih drugih manjih predmeta), a u svakoj se hrpi nalazi određeni broj šibica. U istom trenutku igrač koji je na potezu može uzeti točno određeni broj šibica, najčešće su to 1 ili 2 šibice. Cilj igre za igrača je strategijama doći do posljednje šibice, izvući ju i tako ostvariti pobjedu. Navedena verzija je standardna verzija igre Nim, a obrađena je Misere verzija u kojoj je cilj igrača da dođe do pobjede tako da ne uzme posljednju šibicu, već da svojim strategijama natjera protivničkog igrača da to učini.

Kroz ovaj rad implementirana je i opisana Misere verzija Nim s prediktivnim AI protivnikom. Kako bi se implementirala ova igra da igrač može igrati protiv prediktivnog AI protivnika korišten je minimax algoritam. Minimax algoritam koristi se u igrama s AI protivnikom kako bi AI maksimizirao svoje poteze da dođe do pobjede, dok drugi igrač minimizira poteze protivničkog igrača kako bi ga natjerao da izgubi. U Misere verziji situacije je obrnuta, AI se trudi minimizirati svoje poteze kako ne bi uzeo posljednju šibicu i tako pobijedio dok se protivnik trudi maksimizirati pokrete drugog igrača kako bi ga natjerao da uzme posljednju šibicu i tako pobjedi. Algoritam funkcionira tako da predvidi sve moguće poteze oba igrača kao i potez protivnika te na kraju bira najoptimalniji potez kako bi osigurao najbolji mogći ishod.

Interes za obradu igre Nim pojavio se nakon odabira seminarske teme u kojoj je bio zadatak implementirati igru s AI protivnikom, nakon čega je započeto istraživanje igara za dva igrača uz uvjet da je jedan od igrača AI. Nim je odabran zbog svoje nepopularnosti trenutno pa se pojavila želja za upoznavanjem nove igre, principa igranja te na kraju kako je tako staru igru moguće implementirati danas kroz igranje s prediktivnim AI protivnikom.



Slika 1: Igra Nim. Izvor: <https://en.wikipedia.org/wiki/Nim>

2. Metode i tehnike rada

Minimax algoritam: algoritam koji maksimizira poteze jednog igrača dok drugi igrač minimizira poteze svojeg protivnika. Algoritam je implementiran tako da prolazi kroz sva moguća stanja igre te simulira najbolje moguće poteze s najoptimalnijim ishodom za igrača. Minimax je u ovom radu implementiran za Misere verziju igre gdje je cilj natjerati protivnika da uzme posljednju šibicu. [2]

Misere verzija igre: u ovoj implementaciji AI koristi modificiranu logiku za optimizaciju svojih poteza zbog odabira misere verzije igre trebalo je prilagoditi standardni minimax algoritam jer u ovoj verziji AI minimizira svoje poteze kako ne bi uzeo posljednju šibicu.

Interakcija s korisnikom: implementirana je interakcija s korisnikom kroz korisničko sučelje u kojem se može vidjeti trenutno stanje ploča s hrpama te omogućava korisniku da unese iz koje hrpe želi uzeti koliko šibica. Ograničenja su postavljena tako da korisnik može odabrati prvu ili drugu hrpu i minimalno jednu i maksimalno dvije šibice iz odabrane hrpe. Nakon korisnikovog unosa provjerava se ispravnost unosa.

Heuristika: metoda koja navodi algoritam da pronađe najoptimalnije izvore. Koristeći heuristiku u algoritmu otklanjamo naporno pretraživanje svih mogućih rješenja. U ovom je radu heuristika prisutna u minimax algoritmu gdje završno stanje igre teži da je pobjednik onaj igrač koji ne uzme posljednju šibicu, odnosno kroz minimax algoritam heuristika je provedena tako da AI minimizira svoje poteze kako ne bi posljednji povukao potez i tako izgubio. [3]

Optimizacija: pronalaženje najboljeg načina za rješavanje problema. U ovoj je igri optimizacija prisutna u minimax algoritmu gdje su najbolji potezi AI-ja da ne uzme posljednju šibicu koji ga navode do pobjede.[4]

3. Kritički osvrt

Cilj ovog projekta prije svega bilo je razumijevanje rada minimax algoritma. Kako bi se rad algoritma što bolje razumio, odabrana je strateška matematička igra Nim u kojoj dva igrača iz dvije hrpe mogu izvući jednu ili dvije šibice. U implementaciji je, radi prikaza rada algoritma, kao jedan igrač implementiran AI. AI, kroz minimax algoritam, u misere verziji igre, odabire najoptimalnije poteze kako bi došao do pobjede, u ovom slučaju najoptimalniji potezi su minimizirani jer AI ne želi povući zadnji potez jer to bi značilo da je izgubio već želi natjerati svog protivnika da to učini. Dakle, AI koristi heuristiku prilagođenu ovoj verziji igre kako bi od svih mogućih poteza, na temelju predviđanja završnog stanja, izabrao najoptimalniji potez koji bi ga mogao dovesti do pobjede. Heuristika može biti puno složenija od korištene, ali je za što bolje početničko razumijevanje odabrana baš ova implementacija s ne toliko kompliciranom heuristikom.

4. Opis implementacije

Funkcija *ploca(hrpe)* prikazuje trenutno stanje ploče, odnosno dvije hrpe gdje je u svakoj hrpi u početku nalazi šest šibica.

Funkcija *zavrsono(hrpe)* služi za provjeri je li igra gotova, odnosno jesu li obje hrpe prazne, return `sum(hrpe)==0`.

Funkcija *minimax(hrpe, dubina, ai)* implementira minimax algoritam koji simulira poteze za AI kako bi AI odabrao najoptimalniji potez za pobjedu. Ulazni parametri su hrpe, dubina i ai. Ako funkcija *zavrsono()* vrati da je igrač (`ai=False`) izgubio (posljednji povukao potez), vraća 1 što je povoljno za AI. Ako je AI (`ai=True`) zadnji povukao potez i tako "pobijedio" (što je loše u ovoj verziji), vraća -1.

Potezi za AI se minimiziraju jer se AI zapravo trudi izgubiti kako bi pobijedio, on prepoznaje mora natjerati igrača da posljednji povuče potez i tako planira poteze da ostavi opcije protivniku da završi igru pa se za AI biraju se potezi s najmanjom vrijednošću. S druge strane potezi protivnika se maksimiziraju kako bi pobjednik imao veće šanse da pobijedi i tako prisili AI na gubitak. Za ovu funkciju važan je i paramter dubina što označava dubinu rekurzije, odnosno broj nivoa na kojima funkcija poziva samu sebe i time povećava dubinu za jedan. [5]

Funkcija *aipotez(hrpe)* ova funkcija gleda koji je najbolji potez za AI koristeći minimax algoritam. Simulira sve poteze, a heuristika u implementiranom algoritmu govori funkciji da odabere najoptimalniji potez za AI te bira onaj potez koji će minimizirati gubitak za AI. Funkcija vraća iz koje hrpe je AI uzeo točan broj šibica.

Funkcija *igrac(hrpe)* funkcija koja omogućava korisnikov potez tako da unese broj hrpe i broj šibica koji želi uzeti iz odabrane hrpe. U ovoj funkciji postavljena su ograničenja da hrpa iz koje igrač uzima šibice nije već prazna i da igrač može uzeti u jednom trenutku jednu ili dvije šibice.

Funkcija *igra()* je glavna funkcija koja upravlja igrom. U funkciji se inicijalizira igra s dvije hrpe po 6 šibica, omogućava izmjenično igranje korisnika i umjetne inteligencije. Nakon svakog poteza prikazuje stanje hrpa i na kraju obavještava o pobjedi ili gubitku.

5. Opis rada aplikacije

Igra *Misere Nim* pokreće se funkcijom `igra()`. Na početku igre prikazano je početno stanje u kojem se nalaze dvije hrpe po 6 šibica. Prvi potez ima AI (`potezigrac=False`) koji koristi minimax algoritam kako bi izabrao trenutni najoptimalniji potez prema svim mogućim predviđenim potezima. Algoritam će u svakom koraku u kojem igra AI odabrati onaj potez koji minimizira šanse da AI izgubi, odnosno algoritam bira one poteze koji omogućiti da AI ne povuče zadnji potez. Nakon toga AI i korisnik naizmjenično povlače poteze. Korisnik prvo bira iz koje hrpe želi uzeti šibice (0 ili 1), nakon čega bira broj šibica koji će uzeti (1 ili 2) ovisno o tome koliko još ima šibica u odabranoj hrpi. Nakon svakog poteza broj šibica u odabranoj hrpi se smanjuje za odabrani broj i novo stanje se prikazuje na ažuriranoj ploči s hrpama. Igra završava kada su obje hrpe prazne, a gubitnik je onaj koji je uzeo posljednju šibicu. Ovisno o ishodu igre ispisuje se poruka da je igrač pobijedio jer je AI zadnji uzeo šibicu ili da je AI pobijedio jer je protivnik zadnji povukao potez.

```
[8]: igra()
Dobrodošli u igru Nim!

Trenutno stanje ploče:
Hrpa 0: | | | | |
Hrpa 1: | | | | |
AI je na potezu...
AI je uzeo 2 šibica iz hrpe 0.

Trenutno stanje ploče:
Hrpa 0: | | |
Hrpa 1: | | | | |
Na potezu ste...
Odaberite hrpu (0 ili 1): 0
Koliko šibica želite uzeti iz hrpe 0 (1 ili 2): 1

Trenutno stanje ploče:
Hrpa 0: | |
Hrpa 1: | | | | |
AI je na potezu...
AI je uzeo 2 šibica iz hrpe 0.

Trenutno stanje ploče:
Hrpa 0: |
Hrpa 1: | | | | |
Na potezu ste...
Odaberite hrpu (0 ili 1): 1
Koliko šibica želite uzeti iz hrpe 1 (1 ili 2): 1
```

Slika 2: Započeta igra. Osobna izrada

Trenutno stanje ploče:
Hrpa 0: |
Hrpa 1: | | | |
AI je na potezu...
AI je uzeo 2 šibica iz hrpe 1.

Trenutno stanje ploče:
Hrpa 0: |
Hrpa 1: | | |
Na potezu ste...
Odaberite hrpu (0 ili 1): 1
Koliko šibica želite uzeti iz hrpe 1 (1 ili 2): 1

Trenutno stanje ploče:
Hrpa 0: |
Hrpa 1: | |
AI je na potezu...
AI je uzeo 2 šibica iz hrpe 1.

Trenutno stanje ploče:
Hrpa 0: |
Hrpa 1:
Na potezu ste...
Odaberite hrpu (0 ili 1): 0
Koliko šibica želite uzeti iz hrpe 0 (1 ili 2): 1

Trenutno stanje ploče:
Hrpa 0:
Hrpa 1:
AI je pobijedio! Morali ste uzeti posljednju šibicu.

Slika 3: Kraj Igre. Osobna izrada

6. Zaključak

U implementaciji ove igre u središtu je bio razvoj minimax algoritma. Ovaj algoritam je jako zanimljiv jer omogućava da korisnik implementira prediktivni AI s kojim može odigrati igru. Ovaj se algoritam može primijeniti na daleko složenije igre, ali smatram da je, u današnjem svijetu naglog rasta korištenja umjetne inteligencije, ovo jedan od boljih načina da se netko i s malo znanja o istom upozna s aspektima rada umjetne inteligencije. Tijekom implementacije mi je posebno bilo zanimljivo što i neku staru igru poput Nima možemo u današnjici odigrati i na neke druge načine kao što je ovdje s umjetnom inteligencijom

Popis literature

- [1] Wikipedia. „Nim.” Dostupno na: <https://en.wikipedia.org/wiki/Nim>. (2022.), adresa: <https://en.wikipedia.org/wiki/Nim> (pogledano 23. 1. 2024.).
- [2] J. point. „Mini-Max Algorithm in Artificial Intelligence.” Dostupno na: <https://www.javatpoint.com/mini-max-algorithm-in-ai>. (), adresa: <https://www.javatpoint.com/mini-max-algorithm-in-ai> (pogledano 24. 1. 2025.).
- [3] K. Academy. „Using Heuristics.” Dostupno na: <https://www.khanacademy.org/computing/ap-computer-science-principles/algorithms-101/solving-hard-problems/a/using-heuristics>, Khan Academy. (2019.), adresa: <https://www.khanacademy.org/computing/ap-computer-science-principles/algorithms-101/solving-hard-problems/a/using-heuristics> (pogledano 24. 1. 2025.).
- [4] M. Gupta. „Min Max Algorithm in Artificial Intelligence.” Dostupno na: <https://www.appliedaicourse.com/blog/min-max-algorithm-in-artificial-intelligence/>. (), adresa: <https://www.appliedaicourse.com/blog/min-max-algorithm-in-artificial-intelligence/> (pogledano 23. 1. 2025.).
- [5] T. Sauce. „Learning Recursion: A Deep Dive into an Essential Computer Science Concept.” Dostupno na: <https://techsauce.medium.com/learning-recursion-a-deep-dive-into-an-essential-computer-science-concept-7f88c07d6427>, Medium. (2024.), adresa: <https://techsauce.medium.com/learning-recursion-a-deep-dive-into-an-essential-computer-science-concept-7f88c07d6427> (pogledano 24. 1. 2025.).

Popis slika

1.	Igra Nim. Izvor: https://en.wikipedia.org/wiki/Nim	2
2.	Započeta igra. Osobna izrada	6
3.	Kraj Igre. Osobna izrada	7

Korišteni alati:

- **Chat GPT** (<https://chat.openai.com>)
- **Jupyter Notebook**