

# Razvoj agenta za igru Othello

Pristupnik: Ivan Milinović

Studij: Elektrotehnika i informacijska tehnologija i Računarstvo

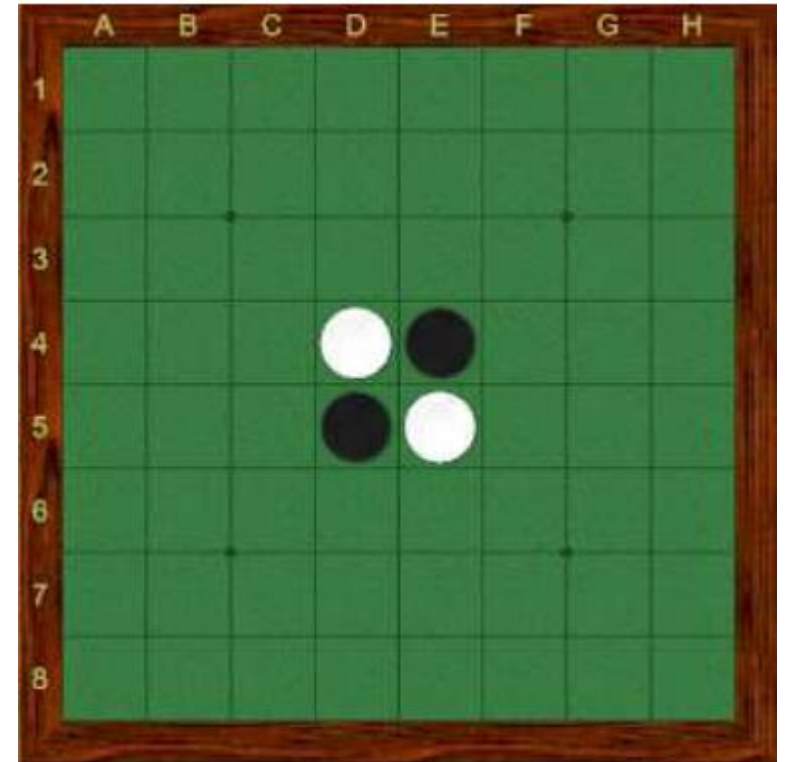
Modul: Računarstvo

Mentor: doc. dr. sc. Marko Đurasević

Akadska godina: 2022./2023.

# Igra Othello

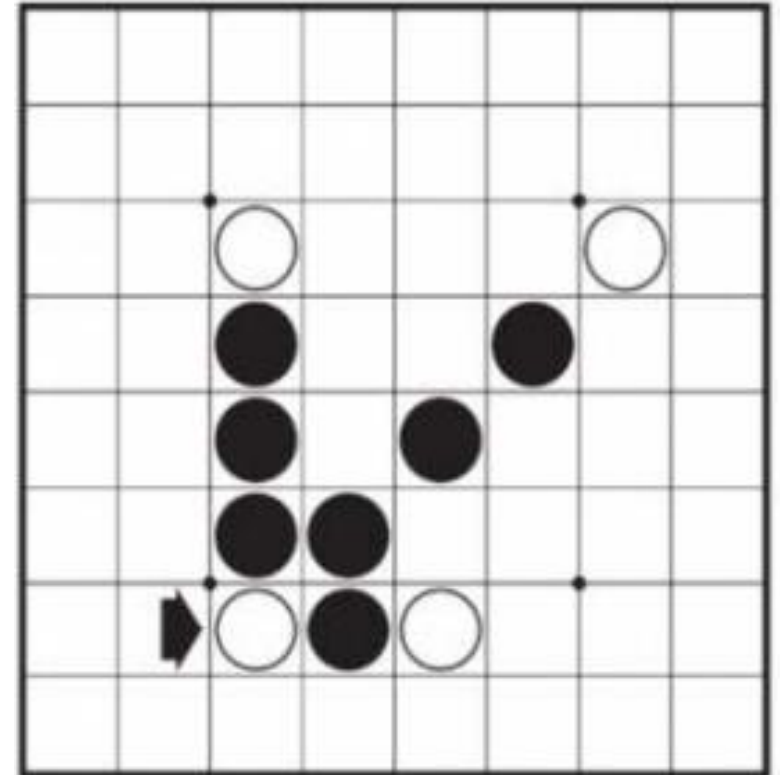
- Igra za 2 igrača koja se igra na 8x8 ploči
- Crni i bijeli igrač
- Žetoni koji su s jedne strane crni, a s druge bijeli
- Svaki igrač ima 32 žetone svoje boje
- Igrači naizmjenično igraju poteze
- Crni je prvi na potezu



*Slika 1. Početna konfiguracija ploče*

# Valjani potez

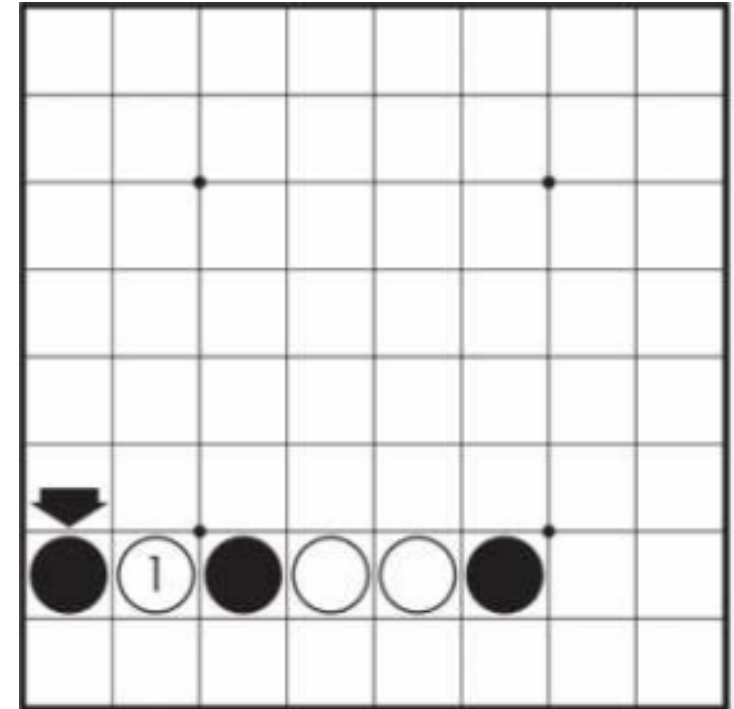
- Potez je valjan ako se njegovim odigravanjem zauzme (okrene) barem jedan protivnički žeton
- Ako se između novopostavljenog žetona i nekog drugog žetona iste boje (u bilo kojem od 8 smjerova) nalaze samo protivnički žetoni, svi oni bit će okrenuti.



Slika 2. Valjani potez

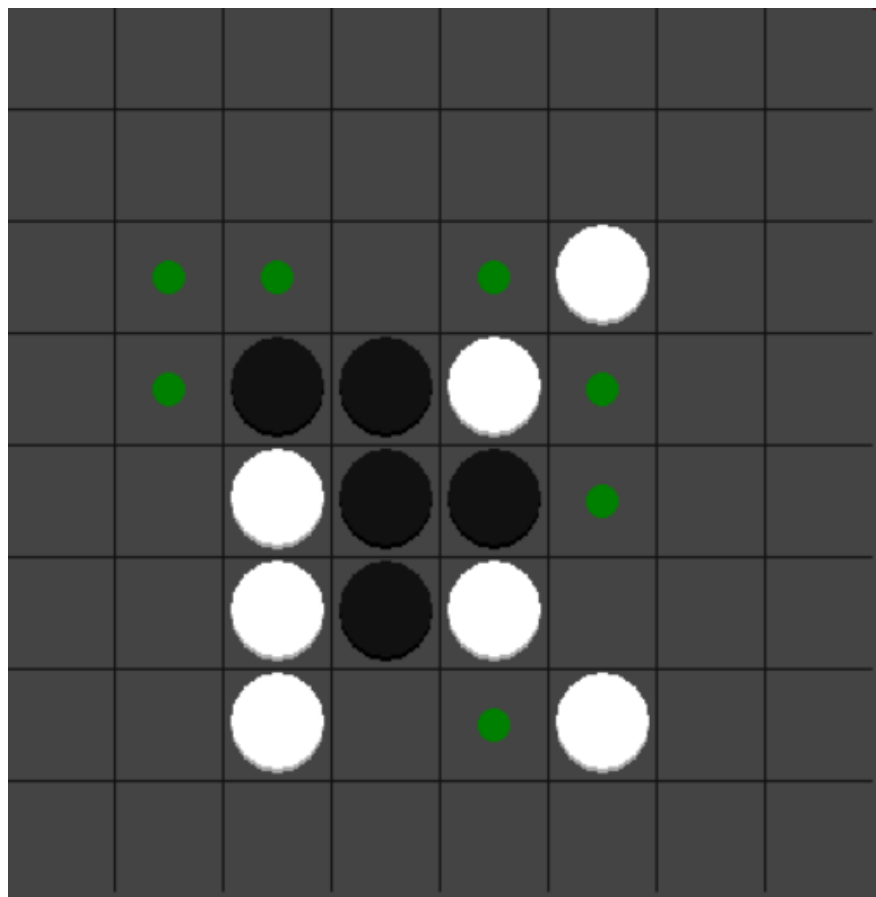
# Valjani potez

- Igrač na potezu mora odigrati valjani potez ako takav postoji
- Ako ne postoji valjani potez na red dolazi igračev protivnik
- Igra završava kad nijedan igrač ne može napraviti nijedan potez

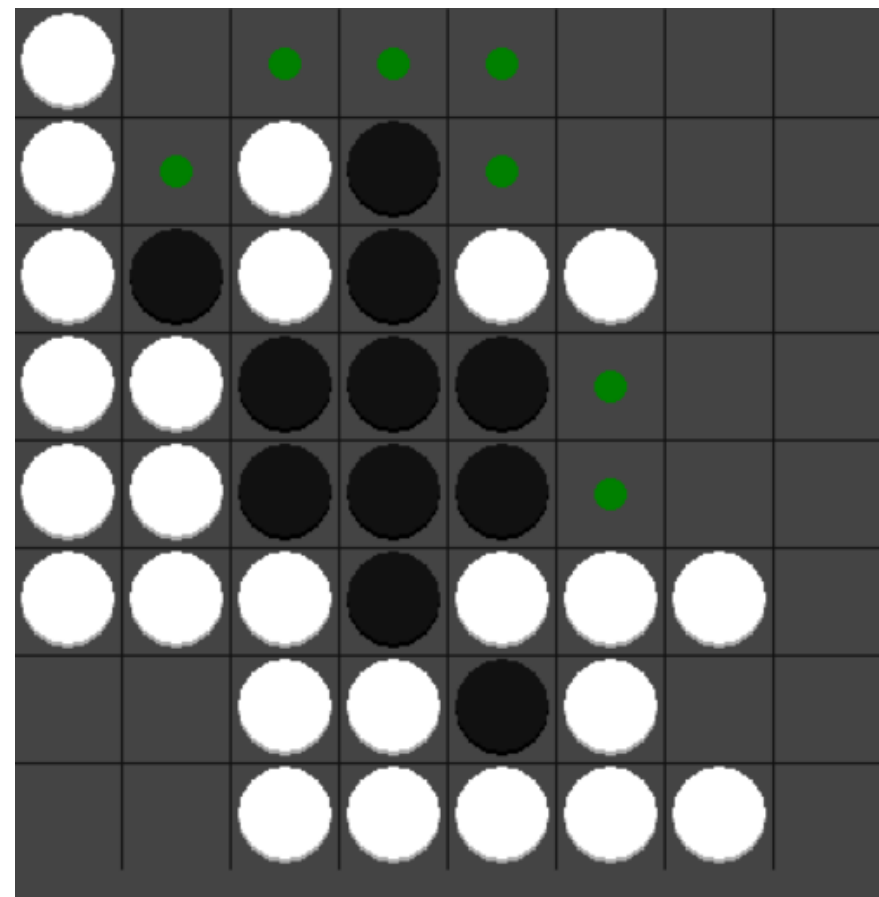


*Slika 3. Valjani potez*

# Primjer mogućih poteza

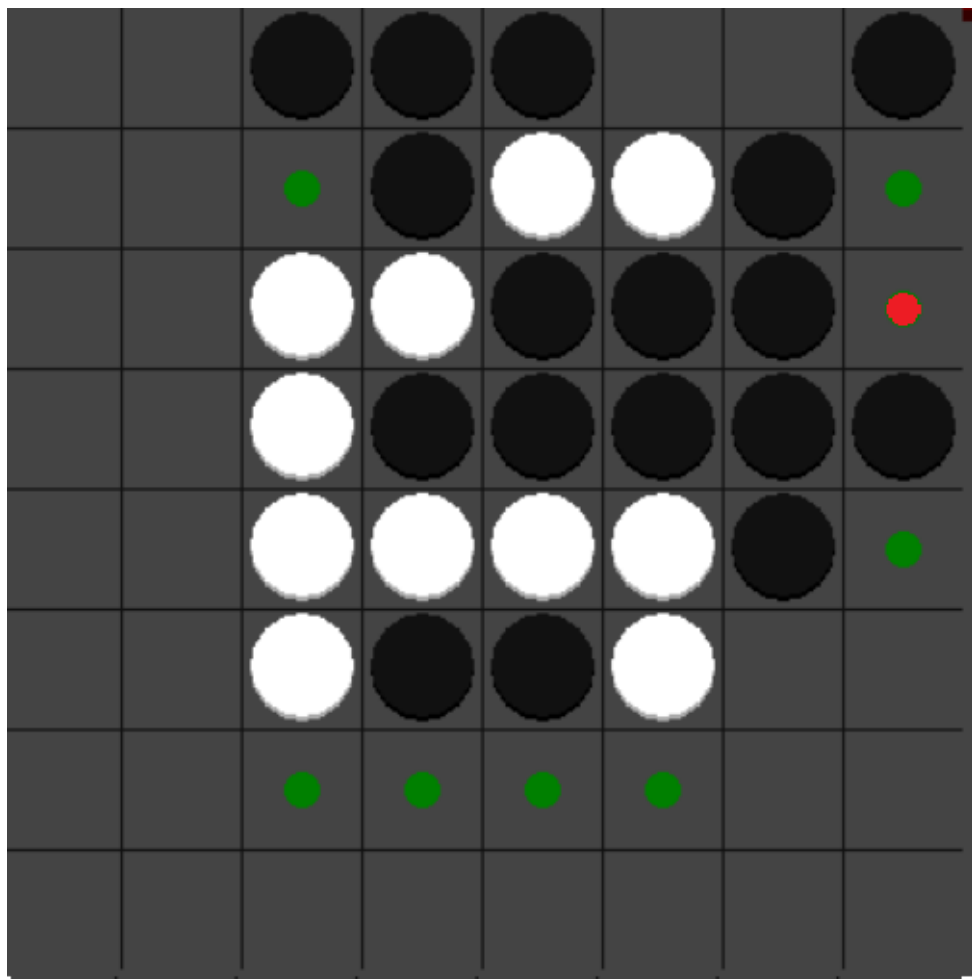


Slika 4. Primjer svih mogućih poteza 1

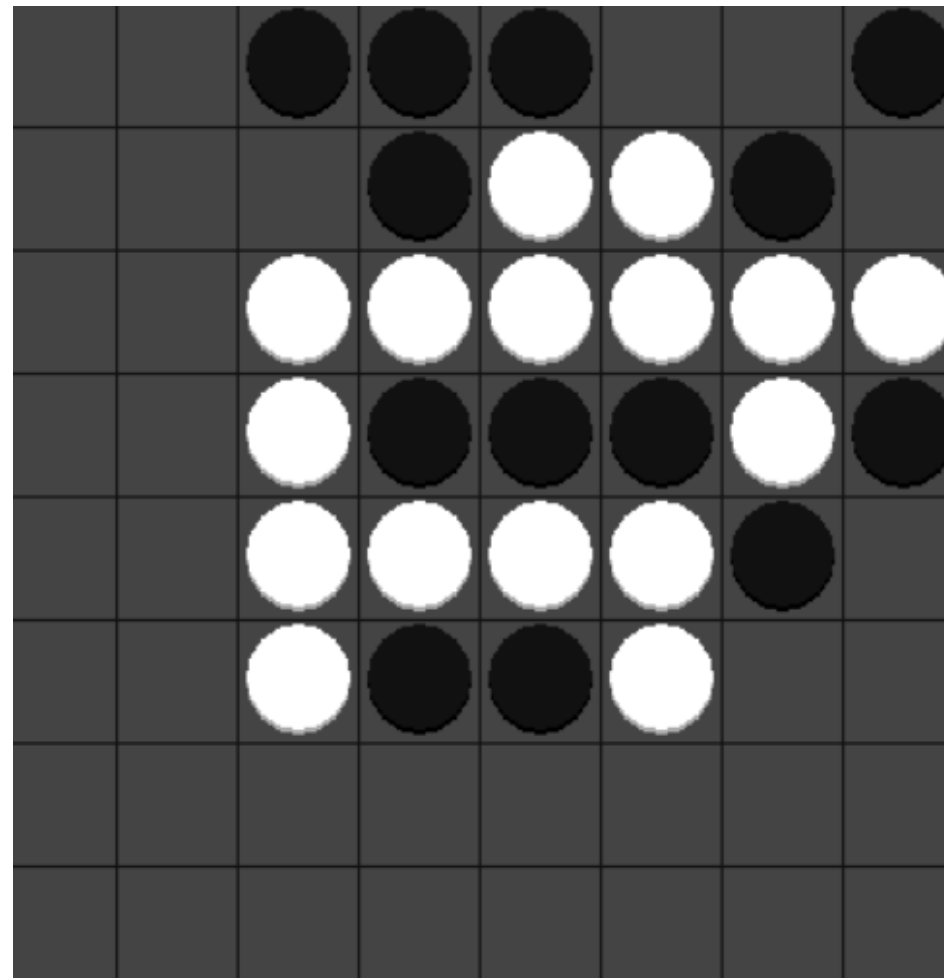


Slika 5. Primjer svih mogućih poteza 2

# Primjer mogućih poteza



Slika 6. Odigravanje poteza 1



Slika 7. Odigravanje poteza 2

# Ideja

- Alfa-beta podrezivanje koje kao heurističku funkciju koristi neuronsku mrežu
- Neuronsku mrežu trenirati genetskim algoritmom
- Agente trenirati igranjem međusobno
- Razvijena dva modela:
  - Linearni model koji koristi poznato dobra svojstva
  - Kompleksniji model koji ne koristi nikakva znanja o igri

# Linearni model

- 12x1 neuronska mreža (nema skrivenog sloja)
- Statička evaluacija ploče (10), ocjena slobode igrača (1) i bonus za zauzete kutove (1).
- Ploča je simetrična s obzirom na horizontalnu, vertikalnu i dijagonalne osi
- Ako pretpostavimo da su svi „trokuti” jednako važni možemo važnost svakog polja sažeti u 10 parametara
- Crni žetoni u implementaciji imaju vrijednost -1, bijeli 1, a prazno polje 0

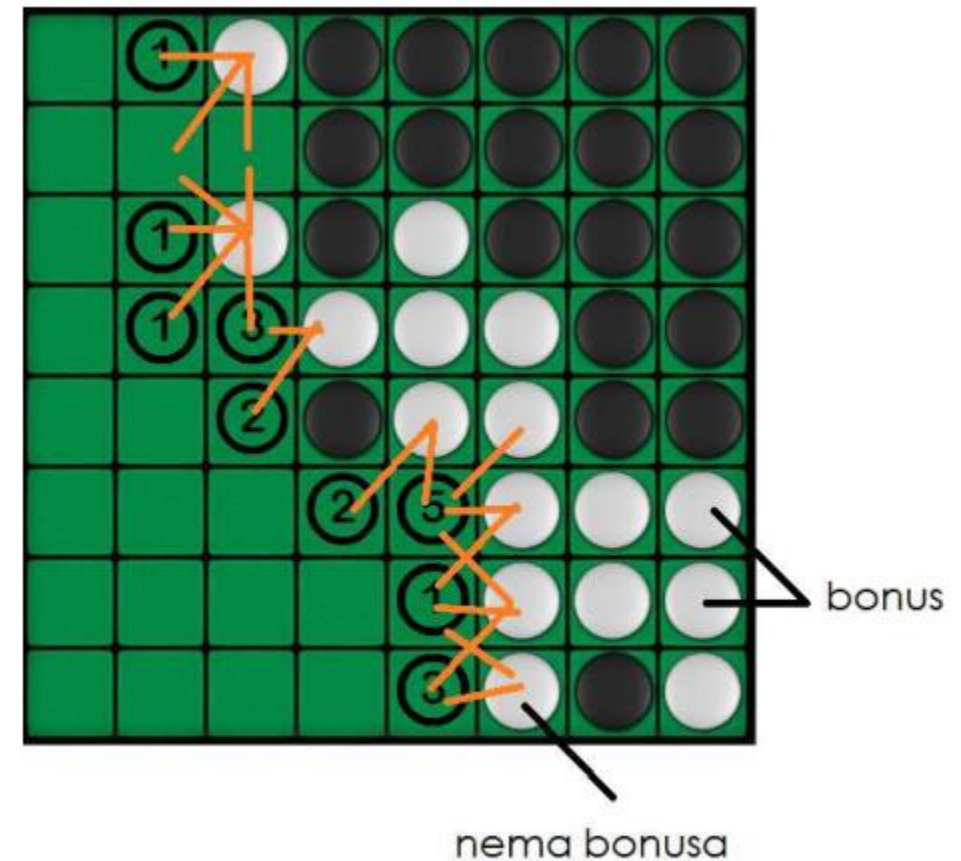
1	5	6	7						
5	2	8	9						
6	8	3	10						
7	9	10	4						

Slika 8. Statička evaluacije ploče



# Linearni model

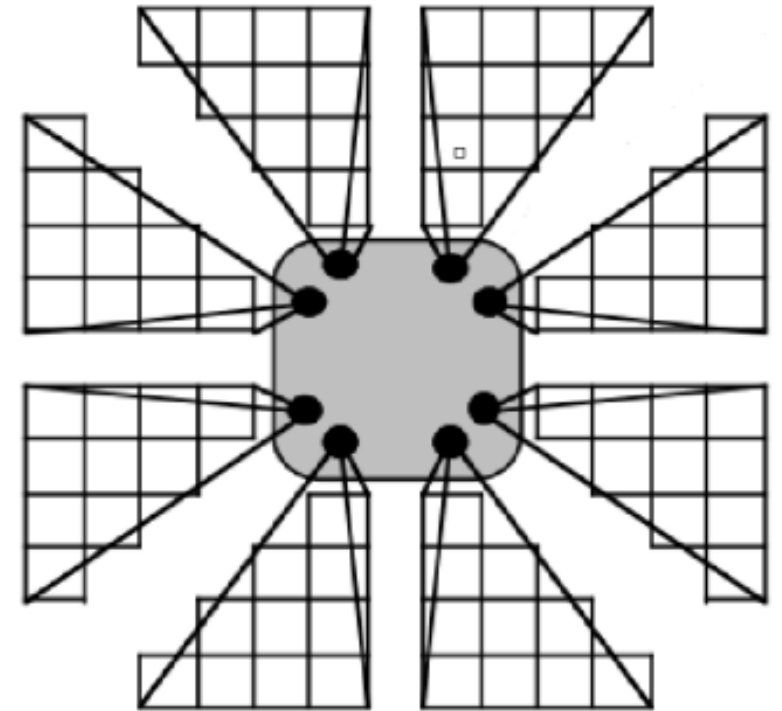
- Sloboda žetona – broj praznih polja neposredno pored njega (u svih 8 smjerova)
- Sloboda igrača – suma slobode svih njegovih žetona
- Ocjena slobode igrača – razlika slobode igrača i slobode njegovog protivnika
- Kutni žeton – najvrjedniji žeton jer se ne može okrenuti
- Ako je kut zauzet vrijednost susjeda u statičkoj evaluaciji je 0 jer ne postoji opasnost od njegovog zauzimanja
- Bonus za sve žetone izravno povezane s kutnim jer isto ne mogu biti okrenuti



Slika 9. Sloboda bijelog igrača i bonus za žetone povezane s kutnim

# Model GA\_NN\_bot2

- 80x8x8x1 unaprijedna neuronska mreža sa sigmoidalnom prijenosnom funkcijom
- Koristi se samo statička evaluacija ploče
- Kao i u prethodnom modelu pretpostavljamo da su svi trokuti jednako važni pa se koriste iste težine za svaki trokut
- 88 težina -> 11 težina iz ulaznog u prvi skriveni sloj
- $(10+1)+(8*(8+1))+(8+1) = 92$  težine



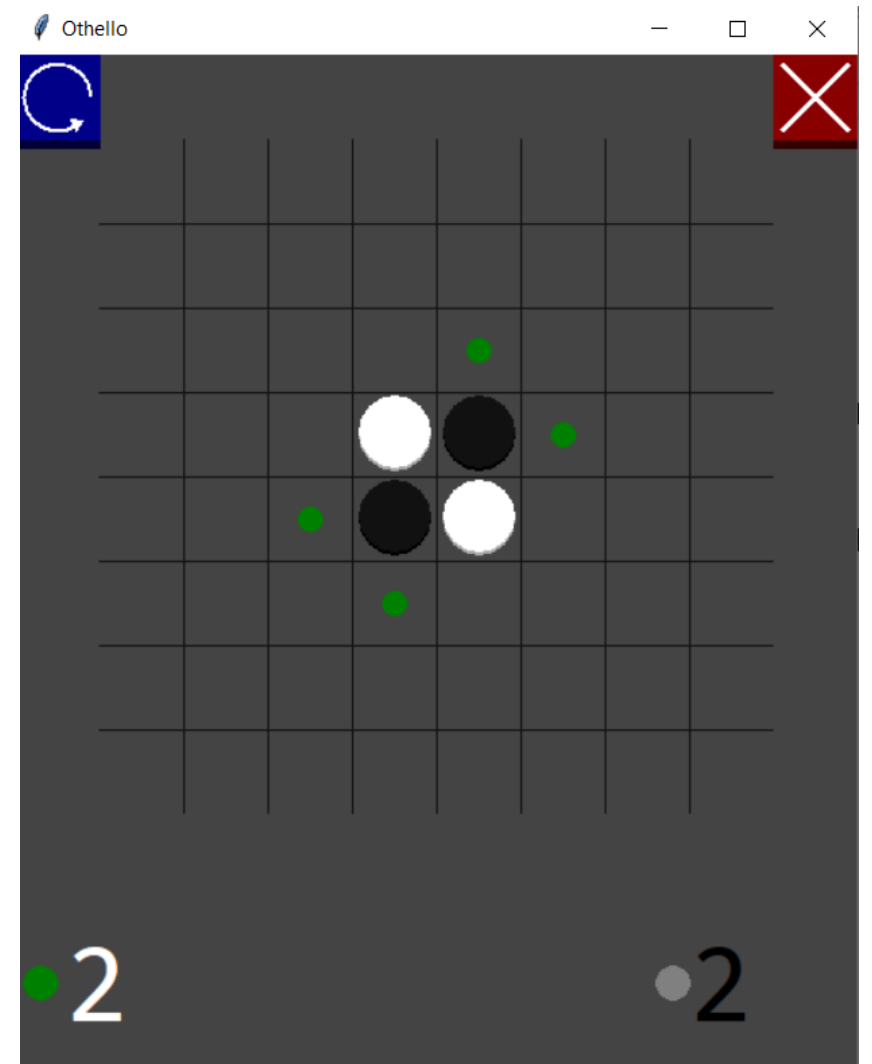
Slika 10. Ulazni i prvi skriveni sloj

# Treniranje modela

- Operator križanja – simuliranje binarno križanje (SBX)
- Operator selekcije – turnirska selekcije u kojoj se biraju po 2 jedinke
- Operator mutacije – dodavanje Gaussovog šuma
- Veličina generacije – 100 ili 300
- Vjerojatnost mutacije i križanja jedinki linearno se mijenja kroz generacije
- Funkcija dobrote – svaka jedinka igra protiv 50 drugih jedinki iz generacije
- Za neki objektivni način prikaza napretka određeni broj najboljih jedinki bi odigralo 1000 partija protiv igrača koji igra nasumične poteze

# Implementacija

- Čitava implementacija programa za treniranje agenata napravljena u C++-u
- Korištena biblioteka NumCpp za matrične operacije
- Napravljeno grafičko sučelje u Pythonu u koje se mogu učitati modeli



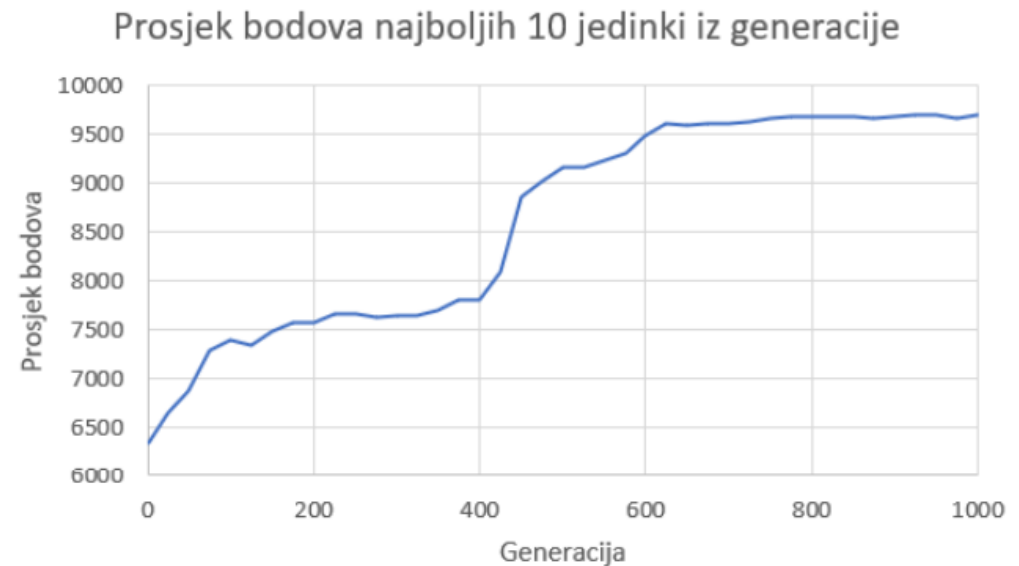
Slika 11. Grafičko sučelje

# Rezultati

- Prosječni broj bodova najboljih 10 jedinki iz generacije u 1000 odigranih partija protiv igrača koji igra nasumične poteze pri dubini 1
- Maksimalno 10 000 bodova
- Veličina generacije 300



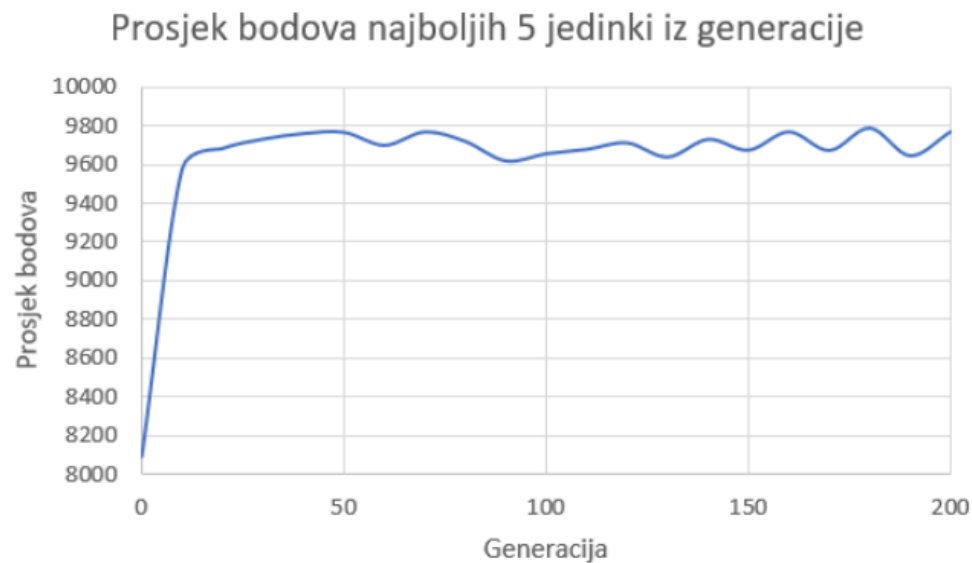
Slika 12. Rezultati linearnog modela pri treniranju na dubini 1



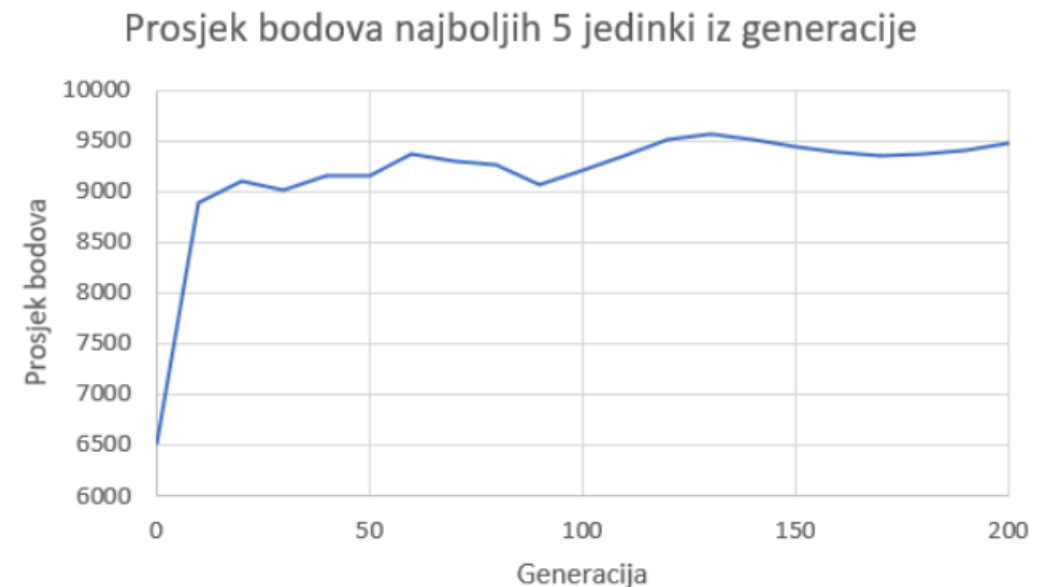
Slika 13. Rezultati modela GA\_NN\_bot2 pri treniranju na dubini 1

# Rezultati

- Prosječni broj bodova najboljih 5 jedinki iz generacije u 1000 odigranih partija protiv igrača koji igra nasumične poteze pri dubini 2
- Maksimalno 10 000 bodova
- Veličina generacije 100



Slika 14. Rezultati linearnog modela pri treniranju na dubini 2



Slika 15. Rezultati modela GA\_NN\_bot2 pri treniranju na dubini 2

# Rezultati

- Linearni model ima malo bolje performanse
- Model GA\_NN\_bot2 pri dubini 2 dostiže manji krajnji rezultat od dubine 1
- U 1000 partija protiv igrača koji igra nasumične poteze s bodovanjem:
  - Pobjeda - 10 bodova
  - Neriješeno - 1 bod
  - Poraz - 0 bodova
- Rezultat:
  - Linearni model – 9800/10 000
  - GA\_NN\_bot2 – 9700/10 000
- Prosječni faktor grananja 10

# Daljnji razvoj

- Bolji način za traženje valjanih poteza – trenutno se pokušava postaviti žeton na svako prazno polje
- Bolji način praćenja napretka agenta – igrač koji igra nasumične poteze nije dobar te već sad protiv njega agenti pobjeđuje preko 90% partija
- Pokušati trenirati agente protiv nekog poznato dobrog agenta umjesto da igraju međusobno
- Implementirati korištenje više dretvi



Hvala na pažnji