

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Filip Drvoderić

IMPLEMENTACIJA IGRE S PREDIKTIVNIM
AI PROTIVNIKOM: KORITENJE
ALGORITMA MINIMAX ZA STVARANJE AI
PROTIVNIKA U IGRI ČOVJEČE NE LJUTI SE

PROJEKT

VIŠEAGENTNI SUSTAVI

Varaždin, 2024.

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

Filip Drvoderić

Matični broj: 0016150980

Studij: Informacijski i poslovni sustavi

**IMPLEMENTACIJA IGRE S PREDIKTIVNIM AI PROTIVNIKOM:
KORISTENJE ALGORITMA MINIMAX ZA STVARANJE AI PROTIVNIKA
U IGRI ČOVJEČE NE LJUTI SE**

PROJEKT

Mentor:

dr. sc. Bogdan Okreša Đurić

Varaždin, siječanj 2024.

Izjava o izvornosti

Izjavljujem da je ovaj projekt 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

Fokus projekta je na razvoju i implementaciji umjetne inteligencije za popularnu društvenu igru "Čovječe ne ljuti se". Cilj je stvoriti AI protivnika donosi odluke temeljem minimax algoritama. AI donosi odluke, uzimajući u obzir sigurnost svojih figura, mogućnost uzimanja protivničkih figura i najbolje poteze za pobjedu. Projekt uključuje programiranje u Pythonu. Korištene su funkcije za bacanje kocke, pomicanje figura, donošenje AI odluka, inicijalizaciju figura, uzimanje protivničkih figura, provjeru poteza, ispisivanja ploče i provjera pobjedničkih uvjeta. Testiranjem AI protivnika u različitim scenarijima igre osigurao sam njegovu funkcionalnost. Ovaj pristup ne samo da prikazuje primjenu AI u društvenim igrama, već i pokazuje kako se Minimax algoritam može koristiti za stvaranje inteligentnih i izazovnih AI protivnika, otvarajući nove mogućnosti u svijetu igara i umjetne inteligencije.

Ključne riječi: Umjetna inteligencija, Minimax algoritam, Čovječe ne ljuti se, Python programiranje, Strategijske igre, AI ponašanje u igrama, Razvoj igara,

Sadržaj

1. Uvod	1
2. Metodike i tehnike rada	2
2.1. Programski Jezik i Razvojno Okruženje	2
2.2. Implementacija Igre	2
2.3. Minimax Algoritam	2
2.4. Heuristička Evaluacija	2
2.5. Testiranje i Validacija	2
3. Kritički osvrt	3
3.1. Praktična Izvedivost	3
3.2. Primjena u Stvarnom Svijetu	3
4. Opis Implementacije	4
4.1. Generiranje Slučajnih Brojeva	4
4.2. Pomicanje Figura	4
4.3. Provjera Stanja Igre	4
4.4. Inicializacija Figura	4
4.5. Interakcija s Igračima	5
4.6. Prikazivanje Stanja Igre	5
4.7. Glavna Petlja Igre	5
5. Prikaz rada aplikacije	6
5.1. Upute za Igranje	7
6. Zaključak	8
Popis literature	9
Popis slika	10

1. Uvod

U središtu projekta je implementacija AI protivnika korištenjem algoritma Minimax. Igra koju sam odabrao je igra "Čovječe ne ljuti se". Odabir ove teme ima nekoliko ključnih razloga. Prvo, "Čovječe ne ljuti se" je jedna od najpopularnijih društvenih igara s dobro postavljenim pravilima i strategijama koje je moguće implementirati u minimax algoritmu. Drugo, Minimax algoritam je već dobro poznat u svijetu teorije igara, ali primjena u ovoj igri otvara nove mogućnosti za inovativna rješenja. Treće, osobna motivacija za ovaj projekt proizlazi iz ljubavi prema igranju "Čovječe ne ljuti se" s obitelji i prijateljima.

Cilj projekta je razviti AI koji može ne samo sudjelovati u igri, već odgovoriti na poteze ljudskog igrača, čime se stvara realistično i izazovno iskustvo. Kroz ovaj projekt, istražujemo kako različite strategije i taktike utječu na odluke AI-a, te kako se AI može prilagoditi različitim stilovima igre. Osim toga, razvoj ovakve vrste AI-a pruža uvide u potencijale i ograničenja Minimax algoritma u realnom vremenu i dinamičnim igrama. Konačno, ovaj projekt služi kao platforma za daljnje istraživanje u području umjetne inteligencije, otvarajući put za naprednije aplikacije u igrama i izvan njih.

2. Metodike i tehnike rada

2.1. Programski Jezik i Razvojno Okruženje

Programski jezik koji sam koristio kako bih ostvario ovo rješenje bio je Python. Razlog tome je što taj jezik učimo na nastavi. Razvojno okruženje koje sam koristio bilo je Visual Studio Code i Google Colab.

2.2. Implementacija Igre

Igra "Čovječe ne ljuti se" implementirana je kroz konzolu, simulira se modificirana igra "Čovječe ne ljuti se". U igri je moguće pomicanje figura, bacanje kocke, odabir figura i izbacivanje protivničkih figura.

2.3. Minimax Algoritam

Minimax algoritam je ključni dio ove igre. Ovaj algoritam omogućava AI-u da donesi odluke temeljene na analizi mogućih ishoda igre. AI gleda na to ako može figuru pospraviti na sigurno mjesto, pojesti protivničkoga igrača, staviti još jednu figuru na ploču ili pak odlučiti kojom će figurom ići dalje.

2.4. Heuristička Evaluacija

Heuristička evaluacija igre igra ključnu ulogu u odlučivanju AI protivnika. Heuristike su dizajnirane da ocijene trenutačno stanje igre i pomognu AI-u da izabere najbolji potez. Ove heuristike uključuju razmatranje sigurnosti figura, mogućnost pojeđanja protivničkih figura i brzinu kojom igrači mogu doći do cilja. Pravilna kalibracija heuristika ključna je za učinkovitost AI-a u stvarnom vremenu.

2.5. Testiranje i Validacija

Testirao sam AI protivnika u mnogobrojnim simulacijama igre. Testiranje se sastojalo od testova oduzimanja protivničkih figura, analize koju će figuru AI uzeti ako može uzeti više, hoće li AI ići dalje s trenutnom figurom ili će uzeti novu figuru van, hoće li izaći van na sigurno ili će uzeti protivničku figuru i slično.

3. Kritički osvrt

3.1. Praktična Izvedivost

Ideja implementacije AI protivnika u igri "Čovječe ne ljuti se" je u teoriji zanimljiva, ali nosi svoje izazove:

- **Heuristička Ocjenjivanja:** Glavni element implementacije je korištenje efikasne heurističke funkcije. Funkcija je ključna za procjenu vrijednosti određenih poteza i odabira najboljeg poteza. Potrebno je uzeti u obzir položaj igrača na ploči, broj sigurnih i ugroženih figura, mogućnost napadanja protivničke figure, mogućnost pobjede, mogućnost izlaska s još jednom figurom te situaciju kada nije moguće odigrati onaj broj poteza koliko je bačeno na kocki.

3.2. Primjena u Stvarnom Svijetu

Primjena AI protivnika u igrama poput "Čovječe ne ljuti se" ima šire implikacije:

- **Edukativna Upotreba:** Takav AI može poslužiti kao alat za učenje i razumijevanje strategija i odlučivanja u igrama, što može biti korisno u edukativne svrhe.
- **Zabava i Igre:** AI protivnik pruža bogatije i izazovnije iskustvo u igrama, što može poboljšati zabavnu vrijednost i privući veći broj igrača.

4. Opis Implementacije

Projekt implementira igru "Čovječe ne ljuti se" s AI protivnikom koristeći Python. Implementacija se sastoji od nekoliko ključnih komponenti:

4.1. Generiranje Slučajnih Brojeva

Funkcija `roll_dice` koristi se za generiranje slučajnih brojeva koji predstavljaju bacanje kocke. Funkcija omogućuje element slučajnosti u igri.

```
def roll_dice():  
    return random.randint(1, 6)
```

4.2. Pomicanje Figura

Funkcija `move_piece` odgovorna je za logiku pomicanja figura na ploči. Uzima trenutnu poziciju figure i rezultat bacanja kocke te vraća novu poziciju figure.

```
def move_piece(position, roll):  
    if position + roll > 30:  
        return -1  
    elif position + roll == 30:  
        return 100  
    return position + roll
```

4.3. Provjera Stanja Igre

Funkcija `game_over` provjerava jesu li sve figure jednog igrača stigle do cilja, što označava kraj igre.

```
def game_over(positions):  
    return all(pos == 100 for pos in positions)
```

4.4. Inicializacija Figura

Funkcija `initialize_pieces` postavlja početne pozicije figura za svakog igrača.

```
def initialize_pieces():  
    return [-1, -1, -1, -1]
```

4.5. Interakcija s Igračima

Funkcije `choose_piece_human` i `choose_piece_ai_minimax` omogućavaju igraču, odnosno AI protivniku, da odaberu koju figuru žele pomaknuti na temelju trenutnog bacanja kocke i stanja na ploči.

4.6. Prikazivanje Stanja Igre

Funkcije `display_positions` i `print_board` koriste se za prikazivanje trenutnih pozicija figura na ploči i stanja igre.

Figure od ljudskog igrača su prikazane oznakama: P1, P2, P3, P4 , a AI-jeve figure su prikazane oznakama: A1, A2, A3, A4

4.7. Glavna Petlja Igre

Glavna petlja igre upravlja tijekom igre, uključujući bacanje kocke, odabir figura, pomicanje figura i provjeru uvjeta za kraj igre.

Implementacija ovih komponenti stvara funkcionalnu simulaciju igre "Čovječe ne ljuti se", u kojoj se AI protivnik natječe protiv ljudskog igrača koristeći strateške odluke temeljene na Minimax algoritmu.

5. Prikaz rada aplikacije

```
Igrač je bacio 1.
AI protivnik je bacio 3.
Igrač je bacio 2.
AI protivnik je bacio 4.
Igrač je bacio 4.
AI protivnik je bacio 1.
Igrač je bacio 2.
AI protivnik je bacio 1.
Igrač je bacio 5.
AI protivnik je bacio 2.
Igrač je bacio 3.
AI protivnik je bacio 1.
Igrač je bacio 5.
AI protivnik je bacio 3.
Igrač je bacio 5.
AI protivnik je bacio 2.
Igrač je bacio 4.
AI protivnik je bacio 4.
Igrač je bacio 5.
AI protivnik je bacio 4.
Igrač je bacio 6.
Pozicije Igrača: [0, -1, -1, -1]
Pozicije AI protivnika: [-1, -1, -1, -1]
|P1| | | | | | | | | | | | | | | | | | | | | |
Igrač je na potezu i bacio je 3.
Igračeve figure su na pozicijama: [0, -1, -1, -1]
Bacili ste 3.
Odaberite figuru koju želite pomaknuti (0-3) ili unesite -1 za preskakanje poteza:
0
```

Slika 1: Početak igre

```
Igrač pomaknuo figuru 0 na poziciju 13.
Pozicije Igrača: [13, -1, -1, -1]
Pozicije AI protivnika: [1, -1, -1, -1]
| |A1| | | | | | | | |P1| | | | | | | | | | | |
AI protivnik je na potezu i bacio je 6.
AI protivnik pomaknuo figuru 1 na poziciju 0.
Pozicije Igrača: [13, -1, -1, -1]
Pozicije AI protivnika: [1, 0, -1, -1]
|A2|A1| | | | | | | | |P1| | | | | | | | | | |
AI protivnik je na potezu i bacio je 3.
AI protivnik pomaknuo figuru 0 na poziciju 4.
Pozicije Igrača: [13, -1, -1, -1]
Pozicije AI protivnika: [4, 0, -1, -1]
|A2| | |A1| | | | | | | |P1| | | | | | | | | |
Igrač je na potezu i bacio je 6.
Igračeve figure su na pozicijama: [13, -1, -1, -1]
Bacili ste 6.
Odaberite figuru koju želite pomaknuti (0-3) ili unesite -1 za preskakanje poteza:
█
```

Slika 2: Sredina igre

5.1. Upute za Igranje

Igra je dizajnirana da bude interaktivna i jednostavna za korisnike. Evo koraka kako korisnik igra:

1. **Početak Igre:** Kada se program pokrene, igra automatski započinje. Na početku igre, program se vrti samostalno sve dok igrač ne baci broj 6. Ovo je preduvjet za početak, jer igrač mora baciti 6 kako bi mogao postaviti figuru na početno polje.
2. **Odabir Figure:** Nakon što igrač baci 6, igra se prebacuje u interaktivni mod. Igrač može birati koju od svojih figura (označenih brojevima 0-3) želi pomaknuti, ovisno o broju koji je bačen kockom. Ova odluka se unosi putem tipkovnice.

```
print("Odaberite figuru koju želite pomaknuti (0-3)  
ili unesite -1 za preskakanje poteza:")
```

3. **Preskakanje Poteza:** U slučaju da igrač nema valjanih poteza (na primjer, ako nijedna figura ne može biti pomaknuta s obzirom na broj koji je bačen), igrač mora preskočiti potez upisivanjem broja -1.
4. **Nastavak Igre:** Igra se nastavlja naizmjeničnim bacanjem kocke između ljudskog igrača i AI protivnika. Svaki igrač na svom potezu bira figuru za pomicanje, vodeći računa o trenutnom stanju na ploči.
5. **Kraj Igre:** Igra završava kada jedan od igrača uspije dovesti sve svoje figure do cilja. Program tada ispisuje pobjednika igre.

```
winner = "Igrač" if game_over(players[0]) else "AI protivnik"  
print(f"Igra je završila. Pobjednik je {winner}!")
```

6. Zaključak

Implementacijom igre s prediktivnim AI protivnikom, spojili smo staru društvenu igru koja se igra na ploči sa umjetnom inteligencijom. Projektom je dokazana efikasnost Minimaxa u analizi poteza i odlučivanju najboljeg mogućeg poteza u danoj situaciji. Pokazali smo da je heurističko ocjenjivanje bilo ključno za uspješno implementaciju AI protivnika. Implementacijom Minimax algoritma dobivamo igru koja je laka za razumijeti, ali je istovremeno izazovna. Rezultati projekta ne samo da doprinose polju umjetne inteligencije u igrama, već i otvaraju mogućnosti za daljnja istraživanja u razvoju AI sustava.

Bibliografija

- [1] R. Gibson, *Introduction to Minimax*, Google Books. https://books.google.hr/books?hl=hr&lr=&id=xRT-AgAAQBAJ&oi=fnd&pg=PP1&dq=minimax&ots=BAZwpiw__5&sig=JPF5F6W3Eqa0R6loedCXQqogf94&redir_esc=y. (Pristupljeno 21.01.2024.).
- [2] N. M. Gotts and B. Polhill, *Simulating Evolutionary Games: A Python-Based Introduction*, *Journal of Artificial Societies and Social Simulation*, vol. 11, br. 3, str. 8, 2008. <https://www.jasss.org/11/3/8.html>. (Pristupljeno 21.01.2024.).
- [3] *Minimax Algorithm in Game Theory - Introduction*, GeeksforGeeks. <https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/>. (Pristupljeno 21.01.2024.).

Popis slika

1.	Početak igre	6
2.	Sredina igre	6