

SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN

Matija Žnidarić

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**

**Matija Žnidarić**

**Matični broj: 0016148634**

**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.**

*Matija Žnidarić*

### **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.

*Autor potvrdio prihvatanjem odredbi u sustavu FOI Radovi*

---

## Sažetak

U svom projektu "Implementacija igre s prediktivnim AI protivnikom: Korištenje algoritma minimax za stvaranje AI protivnika u igri čovječe ne ljuti se", kreirao sam zanimljivu kombinaciju igre čovječe ne ljuti se i AI tehnologije. Igra se igra s četiri figurice po igraču na ploči od 30 polja, a izazov je bio ugraditi Minimax algoritam kako bi AI igrač bio dostojan protivnik pravom igraču. Cijeli proces programiranja bio je u Pythonu, a fokus je bio na stvaranju igre koja je jednostavna za igranje, ali istovremeno pruža taktičku dubinu i izazov. Ovaj projekt nije samo zabavan, već i odličan primjer kako se AI može koristiti za poboljšanje klasičnih igara, dajući im novu dimenziju i izazov.

### Ključne riječi:

"Čovječe ne ljuti se", AI protivnik, Algoritam Minimax, Bacanje kockice, Potezi i strategija,

# Sadržaj

<b>1. Uvod</b>	<b>1</b>
<b>2. Metode i tehnike rada</b>	<b>2</b>
2.1. Izbor Programskog Jezika	2
2.2. Minimax Algoritam za AI	2
2.3. Objektno Orijentirano Programiranje (OOP)	2
2.4. Iterativno Testiranje i Poboljšanja	2
<b>3. Kritički osvrt</b>	<b>3</b>
3.1. Pozitivni aspekti	3
3.2. Područja za poboljšanje	3
3.3. Praktična primjena	3
<b>4. Opis implementacije</b>	<b>4</b>
4.1. Opis igre	4
4.2. Logika Igre	4
4.3. Implementacija AI-a	4
4.4. Korisničko Sučelje	5
4.5. Upravljanje Igrama	5
<b>5. Prikaz implementacije</b>	<b>7</b>
<b>6. Zaključak</b>	<b>8</b>
<b>7. Literatura</b>	<b>9</b>
<b>Popis literature</b>	<b>9</b>
<b>Popis slika</b>	<b>10</b>

# 1. Uvod

Odlučio sam napraviti implementaciju igre s prediktivnim AI protivnikom, koristeći Mini-max algoritam, u klasičnoj igri "Čovječe, ne ljuti se". Fasciniran sam spojem tradicionalne igre i suvremene tehnologije te sam želio istražiti kako AI može zaigrati ovu popularnu igru. Ovaj projekt predstavlja pokušaj da se klasična igra podigne na višu razinu, pružajući igračima iskustvo natjecanja s protivnikom kojeg vodi umjetna inteligencija, čime se igračima pruža novi izazov i užitak.

## **2. Metode i tehnike rada**

### **2.1. Izbor Programskog Jezika**

Za razvoj igre odabrao sam Python, popularan programski jezik poznat po svojoj čitljivosti i jednostavnosti. Python odličan izbor za prototipove i brzi razvoj zbog svojih mogućnosti i bogate biblioteke. U projektu, Python je korišten za izgradnju logike igre, upravljanje korisničkim interakcijama i implementaciju AI.

### **2.2. Minimax Algoritam za AI**

Glavna komponenta AI igrača je Minimax algoritam. Ovaj algoritam se koristi u teoriji igara i računalnim igrama za optimizaciju odlučivanja AI-a. Minimax radi tako što simulira sve moguće poteze u igri, ocjenjujući ih prema prioritetu i time znatno utječe na ishod igre. Algoritam zatim odabire potez koji maksimizira potencijalnu korist za AI.

### **2.3. Objektno Orijentirano Programiranje (OOP)**

Ovaj pristup programiranju bio je ključan u strukturiranju koda. OOP omogućava organizaciju programa kroz klase i objekte, gdje svaka klasa ima svoje atribute i metode. U kontekstu igre, svaka figurica, igrač i potezi predstavljaju objekte s definiranim svojstvima i ponašanjima, što olakšava upravljanje složenošću koda.

### **2.4. Iterativno Testiranje i Poboljšanja**

Tijekom razvoja igre, koristio sam iterativni pristup. Nakon svake promjene sam testirao igru i pokušao ju unaprijediti. Tako sam lakše i brže uočio probleme.

### 3. Kritički osvrt

Po pitanju praktične izvedivosti i primjeni mog projekta "Implementacija igre s prediktivnim AI protivnikom: Korištenje algoritma minimax za stvaranje AI protivnika u igri čovječe ne ljuti se", suočavam se s različitim aspektima koji zaslužuju pažljivu analizu. Iako je projekt u mnogim pogledima uspješan, još ima mjesta za poboljšanja u nekim djelovima.

#### 3.1. Pozitivni aspekti

1. Edukacijska vrijednost: Projekt je odličan alat za učenje, kako za nove programere, tako one koji žele istražiti primjenu AI-a u igrama.
2. Jednostavnost Pythona: Korištenje Pythona pokazalo se kao izvrstan izbor zbog njegove jednostavnosti i efikasnosti, što čini kod lako razumljivim i pristupačnim.
3. Primjena Minimax: algoritma Implementacija ovog algoritma bila je ključna za stvaranje inteligentnog i izazovnog AI protivnika, što igri daje dubinu i ponovnu igrivost.

#### 3.2. Područja za poboljšanje

1. Složenost AI-a: Iako Minimax algoritam pruža solidnu osnovu, njegova složenost može biti ograničavajuća u smislu brzine i učinkovitosti, posebno za veće dubine pretrage. Trenutno igra koristi jednostavan Minimax algoritam koji samo provjerava trenutni potez AI-a.
2. Korisničko Sučelje: Trenutno, igra koristi tekstualno sučelje, što može biti manje privlačno širem krugu korisnika. Grafičko sučelje bi značajno poboljšalo korisničko iskustvo.
3. Skalabilnost i Prilagodljivost: Igra trenutno dobro funkcionira, pitanje je kako bi se igra nosila s dodatnim mogućnostima.

#### 3.3. Praktična primjena

1. U trenutnom obliku, "Implementacija igre s prediktivnim AI protivnikom: Korištenje algoritma minimax za stvaranje AI protivnika u igri čovječe ne ljuti se" je dobra za učenje za studente ili početne programere. Igra trenutno nije za pravo tržište ili komercijalnu upotrebu, za to bi trebalo napraviti lijepše korisničko sučelje.
2. Skalabilnost i Prilagodljivost: Iako igra dobro funkcionira u trenutnom formatu, pitanje je kako bi se projekt nosio s dodatnim funkcijama ili većom složenošću igre.



## 4. Opis implementacije

### 4.1. Opis igre

Ova verzija igre čovječe ne ljuti se za jednog igrača koja koristi ploču s 30 polja, kockicu i figure koje igrači pomiču. Cilj igre je premjestiti sve figurice s početne pozicije do cilja, pri čemu se baca kocka za određivanje broja za koliko polja igrači mogu pomaknuti svoje figure. Na početku igre se naizmjenično baca kocka dok AI ili igrač ne bace 6, a nakon toga započinje igra. Ključni element igre je sposobnost igrača da bira strategiju u kombinaciji s elementom sreće koji donosi bacanje kockice.

### 4.2. Logika Igre

Osnova igre temelji se na Pythonu, gdje je svaki element igre modeliran kao objekt. Igračke figurice, ploča i pravila igre implementirani su kroz klase i funkcije. Na primjer, klasa `Figurica` sadrži attribute kao što su pozicija i status, dok klasa `Igra` upravlja tijekom igre.

- Klasa `Figurica`:

```
class Figurica:
    def init(self):
        self.pozicija = None
        self.status = "na čekanju"
```

- Klasa `Igra`:

```
class Igra:
    def init(self):
        self.figurice = [Figurica() for _ in range(4)]
        self.ploča = [None]*30
        ...
```

### 4.3. Implementacija AI-a

Za AI igrača, upotrijebljen je Minimax algoritam. Algoritam procjenjuje svaki mogući potez i odabire onaj koji maksimizira šanse za pobjedu AI-a. Složenost algoritma i moguća buduća proširenja uključuju alfa-beta rezanje za optimizaciju. Minimax algoritam daje najviši prioritet figurici da ode do cilja, nakon toga prioritet ima 'jedenje' protivničke figurice. Slijedi stavljanje nove figurice na ploču. Ako ništa od navedenog nije moguće samo pomiče figuricu.

```
def minimax(ploča, dubina, maksimizirajućiIgrač):
    if dubina == 0 or kraj_igre(ploča):
```

```

return ocjena(ploča)
if maksimizirajućiIgrač:
    maxEval = -beskonačno
    for svaki mogući potez:
        eval = minimax(ploča, dubina-1, False)
        maxEval = max(maxEval, eval)
    return maxEval
else:
    minEval = beskonačno
    for svaki mogući potez:
        eval = minimax(ploča, dubina-1, True)
        minEval = min(minEval, eval)
    return minEval

```

## 4.4. Korisničko Sučelje

Korisničko sučelje je tekstualno i omogućava igračima da unesu svoje poteze kroz konzolu. Sučelje je jednostavno i intuitivno i pruža sve potrebne informacije o trenutnom stanju igre.

```

def prikaži_ploču(ploča):
    for i in range(len(ploča)):
        if ploča[i] is None:
            print("-", end=" ")
        else:
            print(ploča[i], end=" ")
    print()

```

## 4.5. Upravljanje Igrama

Ključna funkcija `igraj` upravlja tijekom igre, uključujući bacanje kocke, odabir poteza i provjeru uvjeta za pobjedu. Ova funkcija se oslanja na prethodno definirane klase i metode za upravljanje igrom.

```

def igraj():
    while not kraj_igre():
        bacanje = baci_kocku()
        potez = odaberi_potez(bacanje)
        ažuriraj_ploču(potez)
        if provjeri_pobjedu():
            break
    prikaži_pobjednika()

```

Ovaj opis implementacije pruža uvid u tehničke aspekte projekta "Implementacija igre s prediktivnim AI protivnikom: Korištenje algoritma minimax za stvaranje AI protivnika u igri čovječe ne ljuti se", demonstrirajući kako su pojedine komponente igre dizajnirane i povezane kako bi zajedno formirale koherentnu i funkcionalnu igru.

## 5. Prikaz implementacije

```
Trenutne pozicije:  
Igrač: [None, None, None, None]  
AI: [None, None, None, None]  
AI, bacili ste: 6  
AI je pomaknuo figuricu 1  
  
Trenutne pozicije:  
Igrač: [None, None, None, None]  
AI: [0, None, None, None]  
AI, bacili ste: 1  
AI je pomaknuo figuricu 1  
  
Trenutne pozicije:  
Igrač: [None, None, None, None]  
AI: [1, None, None, None]  
Igrač, bacili ste: 4  
  
Trenutne pozicije:  
Igrač: [None, None, None, None]  
AI: [1, None, None, None]  
AI, bacili ste: 5  
AI je pomaknuo figuricu 1
```

Slika 1: Početna bacanja

```
Trenutne pozicije:  
Igrač: [16, None, None, None]  
AI: [23, 0, None, None]  
Igrač, bacili ste: 6  
Igrač, odaberite figuricu (1-4): 2  
  
Trenutne pozicije:  
Igrač: [16, 0, None, None]  
AI: [23, None, None, None]  
Igrač, bacili ste: 4  
Igrač, odaberite figuricu (1-4): 2  
  
Trenutne pozicije:  
Igrač: [16, 4, None, None]  
AI: [23, None, None, None]  
AI, bacili ste: 6  
AI je pomaknuo figuricu 2  
  
Trenutne pozicije:  
Igrač: [16, 4, None, None]  
AI: [23, 0, None, None]  
AI, bacili ste: 5  
AI je pomaknuo figuricu 1  
  
Trenutne pozicije:  
Igrač: [16, 4, None, None]  
AI: [28, 0, None, None]  
Igrač, bacili ste: 6  
Igrač, odaberite figuricu (1-4): █
```

Slika 2: Prikaz Igre

## 6. Zaključak

Projekt "Implementacija igre s prediktivnim AI protivnikom: Korištenje algoritma minimax za stvaranje AI protivnika u igri čovječe ne ljuti se" spaja klasične elemente igare na ploči "Čovječe ne ljuti se" s AI tehnologijama. Tako dobivamo igru koja je jednostavna za razumijevanje, ali izazovna za igranje. Integracija Minimax algoritma u igru donosi novu dimenziju strategije i predviđanja, čineći svaki susret s AI igračem jedinstvenim i uzbuđljivim iskustvom. Ovaj projekt daje uvid u primjenu AI u igrama, ali nudi i mogućnosti za daljnji razvoj i istraživanja u području interaktivne zabave.

## 7. Literatura

1. "Rules of Ludo", mastersofgames, adresa: <https://www.mastersofgames.com/rules/ludo-rules-instructions-guide.htm> (pogledano 20.1.2024)
2. "Laboratorijske vježbe UUI @ SUZG FOI; dio 3.", elf.foi.hr, adresa: <https://elf.foi.hr/mod/url/view.php> (pogledano 20.1.2024)
3. "Mini-Max Algorithm in Artificial Intelligence", Javapoint, adresa: <https://www.javatpoint.com/mini-max-algorithm-in-ai> (pogledano 20.1.2024)
4. "Python Code Example Handbook – Sample Script Coding Tutorial for Beginners", FreeCodeCamp, adresa: <https://www.freecodecamp.org/news/python-code-examples-sample-script-coding-tutorial-for-beginners/> (pogledano 20.1.2024)

# Popis slika

1.	Početna bacanja . . . . .	7
2.	Prikaz Igre . . . . .	7