

VEŠTAČKA INTELIGENCIJA

IZVEŠTAJ ZA 3.FAZU PROJEKTA

Implementacija min-max algoritma i heuristike

18027 Bogdan Filipović

16868 Dušan Spasić

16473 Danica Antić

- Pre pokretanja programa, u terminal ukucati naredbu: *pip install colorama*

Dodate metode:

minimax(board, depth, player, alpha=(None, -10), beta=(None, 10)):

Prosleđena tabla – board, Dubina – depth, Igrač – player, alpha=(None, -10) – polazna vrednost za alfa, beta=(None, 10) – polazna vrednost za beta. Ova metoda poziva max_value ukoliko je player na potezu, odnosno min_value ukoliko je računar na potezu.

max_value(board, depth, alpha, beta, move=None):

Prosleđena tabla – board, Dubina – depth, alpha – alfa, beta – beta, Move – početni potez. Ova metoda pronalazi najbolju vrednost za igrača Max.

min_value(board, depth, alpha, beta, move=None):

Prosleđena tabla – board, Dubina – depth, alpha – alfa, beta – beta, Move – početni potez. Ova metoda pronalazi najbolju vrednost za igrača Min.

heuristic(board):

Prosleđena tabla – board. Ova metoda vraća vrednost heuristike koja za trenutno stanje table predstavlja razliku mogućih poteza X igrača I O igrača.

play_with_minimax():

Ukoliko je player na potezu, vrši se unos poteza. Ukoliko je računar na potezu, poziva se minimax metoda sa dubinom 2.

del_move(board, move, player):

Prosleđena tabla – board, Potez – move, Igrač – player. Ova metoda se koristi kod procene najboljeg poteza za računar.

best_move(board, player):

Prosleđena tabla – board, Igrač – player. Kada je na potezu računar, odigravaju se svi mogući potezi kako bi se našao potez koji protivniku ostavlja najmanje mogućnosti. Nakon isprobavanja svakog poteza, sa del_move se vraća na trenutno stanje, kako bi izvršio dalju procenu za preostalih stanja. Na kraju, odigrava se najbolji potez koji je pronađen.

Prethodno implementirane metode:

board(n, m):

Vrste – n, Kolone – m. Inicijalizuje tablu, praveći matricu, gde je svako polje blanko znak.

print_board (board):

Prosleđena tabla – board. Ova metoda štampa tablu. Reverse() metoda je iskorišćena kako bi se redosled brojeva za svaku vrstu ispisao traženim redosledom.

play_move(board, move, player):

Prosleđena tabla – board, Potez koji treba odigrati – move, Igrač (X=1, O=0) – player. Potez (move) je u format "4C". Testira se ispravnost formata unetog poteza. Nakon toga, za određeni potez se poziva metoda validate_move, i ukoliko je uspešno izvršena, potez se upisuje u listu polja tabele. Ukoliko je format poteza neispravan – traži se ponovni unos željenog poteza.

validate_move(board, row, column, player):

Prosleđena tabla – board, Vrste – row, Kolone - column, Igrač (X=1, O=0) – player. Vršiti se validacija poteza. Ispituje se da li je potez u potpunosti van table, da li je za X igrača u dozvoljenim redovima (zabranjen poslednji red), da li je za O igrača u dozvoljenim kolonama (zabranjena poslednja kolona) i da li na željenom mestu već postoji odigrani potez.

validate_end(board, player):

Prosleđena tabla – board, Igrač (X=1, O=0) – player. Vršiti se validacija kraja igre – ukoliko igrač koji je na potezu nema mesta za svoju pločicu, ispisuje se pobednik igre.

first_player():

Izbor prvog igrača – C za računar, P za čoveka.

play():

Poziva se metoda za unos dimenzija table, za izbor igrača, zatim metoda za odigravanje poteza. Nakon toga se štampa tabla.

possible_moves (player, board):

Igrač – player, Prosleđena tabla – board. Vršiti proveru svih mogućih poteza igrača koji je na redu da igra.

enter_board_dimensions ():

Ova metoda proverava ispravnost unetih podataka za veličinu table.

board_example (list_of_moves):

Dimenzije i odigrani potezi – list_of_moves. Ova metoda prikazuje tablu u nekom proizvoljnom trenutku. Prosleđeni su potezi koji su igrači upisivali redom u toku igre.