

Predmet: Veštačka inteligencija

Projekat: Havana

Tim: 7A3

Faza: III

Uvodna napomena - modifikacije u fazi II:

- Traženje prstena je izmenjano. Kada je implementiran *minimax* algoritam primećeno je da detekcija postojanja prstena ipak ne radi dovoljno dobro. Zato je rešenje kompletno izmenjano. Novo rešenje funkcioniše na sledeći način: Kreće se provera prstena od poslednje odigranog poteza, u trenutku kada neki od čvorova ispunjava barem 1 od uslova navedenih u prethodnoj fazi (*Ukoliko se u toku pretrage naiđe na čvor koji ima manje od dva suseda, ili ima tačno dva suseda, ali su oni međusobno takođe susedi ovaj čvor se briše iz konekcije*) briše se iz konekcije i pretrega počinje ispočetka. Naposljetku je bitno da je u konekciji ostalo barem 6 čvorova kako bi prsten postojao.

Opis funkcija:

- ***minimax (state move alpha beta currentDepth isMyMove)***  
*Opis:* Implementacija *minimax* algoritma sa alfa-beta odsecanjem.

*Parametri:*

- *state* – stanje na tabli
- *move* – poslednje odigrani potez
- *alpha, beta* – parametri neophodni za alfa-beta odsecanje
- *currentDepth* – dubina do koje se radi ispitivanje. Kreće od zadate dubine i u trenutku kada dostigne 0 vrši se evaluacija
- *isMyMove* – logička vrednost koja opisuje ko je na potezu

*Implementacija:*

Naizmenično se igraju potezi dva igrača pozivajući funkcije *maxPlay* i *minPlay*. Za poziv ove dve funkcije se uvek prosleđuje lista mogućih poteza koje oni mogu da odigraju. U trenutku kada je dubina ispitivanja dostigla 0, ili više nema poteza u listi mogućih poteza vrši se evaluacija.

Minimax vraća parove (potez, evaluacija), sem na kranjem izlasku iz rekurzije kada vraća samo potez koji treba odigrati.

- ***maxPlay (movesList bestMove depth alpha beta isMyMove previousState)***  
*Opis:* Igra poteze maksimizirajućeg igrača.

*Parametri:*

Značenje parametara je jasno iz samih naziva i uz objašnjenje parametara iz prethodne funkcije.

*Implementacija:*

Poziva se *minimax* algoritam sa *alfa-beta* odsecanjem uz dekrementiranu dubinu za elemente iz prosleđene liste poteza kako bi se odredio naredni potez, tj. potez min-igrača. Vrš se poređenje alfa vrednosti sa dobijenom evaluacijom da se odredi da li ima mogućnosti za odsecanjem.

Ukoliko nema odsecanja vrši se provera poteza za naredni u listi poteza koja je prosleđena. Kraj rekurzije je isti kao i za *minimax* algoritam.

- ***minPlay (movesList bestMove depth alpha beta isMyMove previousState)***

Opis, parametri i implementacija su analogni funkciji *maxPlay* sem što ova funkcija igra poteze minimizirajućeg igrača.

- ***evaluate (move isMyMove state)***

Funkcija koja *imitira* evaluaciju. Biće, po planu, implementirana u narednoj fazi.

*Napomena:*

Kod nije moguće razdvojiti po fajlovima jer se za izradu projekta koristi Visual Studio Code, a ne Allegro iz tehničkih razloga.