

**Projekat “Havannah” - Prva faza  
Veštačka inteligencija**

**Izveštaj**

**Verzija 1.0**

<b>15834</b>	<b>Aleksandar Radosavljevic</b>
<b>15841</b>	<b>Anđela Randelović</b>
<b>15845</b>	<b>Luka Randelović</b>

Projekat "Havannah" - Prva faza	Verzija: 1.0
Izveštaj	Datum: 17.11.2018. god.

## I. Definisanje načina za predstavljanje stanja problema (igre)

Stanje problema (igre) će se pamtit u asocijativnoj listi gde će svaki element sadržati još jednu asocijativnu listu unutar sebe.

Primer: (A(0 X) (1 X) (2 Z)) B(2 Y) (4 Y))....

U ovoj listi prva vrednost predstavlja oznaku vrste, a svaka vrsta ima čuva trenutno zauzete pozicije tako što beleži broj kolone i oznaku igrača koji je odigrao to polje.

Problem (igra) će se predstavljati u obliku šestougla kao i što nalažu pravila igre. Da bi se to postiglo koristi se matrica kao struktura koja se kreira operacijom (**setf tabla (make-array '(60 100))**). Nakon toga se poziva metoda **spremi-tablu()** koja će na početku postaviti zvezdama sva polja matrice koja ne ulaze u šestougao a nule ukoliko ulaze. Polja sa nulama predstavljaju validne poteze koje igrač može odigrati, osim za poteze koji su zauzeti.

## II. Pisanje funkcija za postavljanje početnog stanja na osnovu zadate veličine table

Za postavljanje početnog stanja se koristi struktura matrice koja se kreira pozivanjem funkcije (**setf tabla (make-array '(60 100))**). Nakon nje će se u memoriji zauzeti prostor za matricu veličine **60x100**, što prestavlja maksimalnu veličinu table. Podrazumevano se postavlja veličina table na **6x6** tako što se postavlja promenljiva **n=6**, ali je omogućeno korisniku da sam izabere veličinu table. Nakon toga je matrica kreirana. Ona se potom koristi kako bi se kreirala tabla za igru, to se radi pozivom metode **spremi-tablu()**. Metoda radi tako što će postaviti zvezdice na sva polja koja ne ulaze u šestougao i time predstavljaju nevalidne poteze za igru, dok će nule predstavljati polja šestougla koji su validni potezi za igru.

## III. Omogućavanje izbora ko će igrati prvi (čovek ili računar)

Izbor igrača koji će igrati prvi se radi postavljanjem promenljive **prvi\_igrac**, i to za vrednost 0 prvi igra čovek a za vrednost 1 računar. Promenljiva se može postaviti pozivom funkcije **odabir\_prvog\_igraca (x)** kojoj se kao argument prosleđuje igrač koji će prvi igrati a ona će u svom telu programa da postavi vrednost za promenljivu **prvi\_igrac**. Nakon odigravanja prvog poteza poziva se funkcija **igraj\_potez()** koja će pozvati funkcije **igra\_prvi\_covek()** ili **igra\_prvi\_kompjuter()** u zavisnosti o vrednosti promenljive **prvi\_igrac**. Ove funkcije pozivaju funkciju (**mapiraj (read) (read) (nth moduo\_za\_vrednost\_polja vrednost\_polja\_igranje) )** koja predstavlja igranje poteza, tj. Igrač unosi poziciju na koju želi da odigra.

Projekat "Havannah" - Prva faza	Verzija: 1.0
Izveštaj	Datum: 17.11.2018. god.

#### IV. Implementiranje funkcija koje obezbeđuju prikaz proizvoljnog stanja problema (igre)

Za prikaz proizvoljnog stanja problema (igre) koristi se funkcija **stampaj()** koja će na osnovu trenutnog stanja igre koje je sačuvano u asocijativnoj listi štampa tablu. Prikazana tabla će predstaviti trenutno stanje igre. Ova funkcija će se ponovno pozivati nakon svakog odigranog poteza bilo kog od igrača tako da će uvek biti predstavljeno trenutno stanje igre. Na taj način će biti prikazano svako proizvoljno stanje.

#### V. Realizovanje funkcija koje na osnovu zadatog poteza igrača, u obliku (vrsta kolona), omogućavaju proveru da li je potez valjan i ako jeste, tj. promenu prosleđenog stanja problema (igre) odigravanjem poteza

Na početku će funkcija **stampaj()** da prikaže trenutno stanje igre tako da će igrač koji je na redu za igru da ima uvid u slobodna polja table. Nakon toga će se na osnovu toga koji igrač sledeći igra pozvati funkcija **igra\_prvi\_covek()** ili **igra\_prvi\_kompjuter()**. U svakoj od nje postoji naredba (**mapiraj (read) (read) (nth modulo za vrednost\_polja vrednost\_polja\_igranje) )** kojom će se zahtevati od igrača da unese poziciju na koju želi da igra (isto se radi i u metodi **igra\_prvi\_kompjuter()** jer u ovoj fazi nije implementiran na igra protiv računara pa će trenutno i drugi igrač biti čovek). Unosom polja je pozvana funkcija **mapiraj (a b igrac)** koja će prvo pozvati funkciju (**validacija a b br**) koja će proveriti da li je igrač odigrao validan potez za njega. Ukoliko je potez validan, asocijativna lista će sačuvati novi potez u novom elementu (**indeks table, X|O**), a matrica za prikaz će se ponovno ažurirati. Funkcijom **dodaj-potez ( a b igrac)** će se sačuvati odigran potez igrača. Na kraju će se pozvati funkcija **igraj\_potez()** koja će prebaciti na drugog igrača da odigra potez tako što će ažurirati promenljivu **prvi\_igrac**. Tada će drugi igrač biti u mogućnosti da odigra svoj potez na isti način koji je već opisan.