
DS 1989

Havannah

**Dimitrije Jović 15657
Miloš Krcobić 15683
Đorđe Nikolić 15777**

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

Faza 1

Problemi koji su rešavani u okviru faze 1:

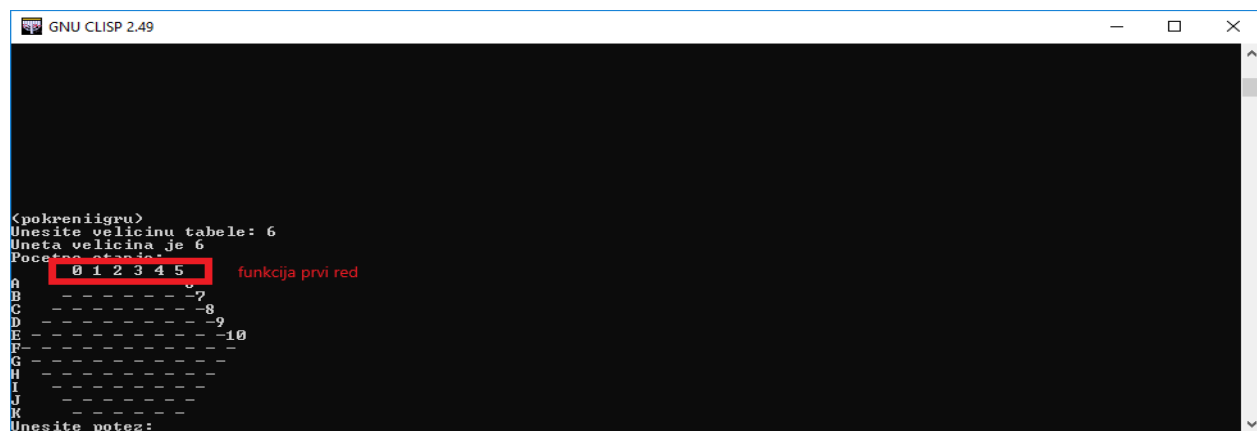
1. Kreiranje početnog stanja
2. Iscrtavanje table za igru na osnovu početnog stanja
3. Odigravanje poteza i kreiranje novog stanja
4. Provera validnosti poteza

Kreiranje početnog stanja

Početno stanje se formira pomoću pet funkcija:

1. prvired
2. gornja
3. sredina
4. donja
5. spoji

prvired – funkcija koja služi za kreiranje prvog reda u kome se nalaze brojevi, zvezdice (*) koje se dodaju koriste se kako bi se ispisao razmak, kao rezultat izvršenja dobija se lista.



```

GNU CLISP 2.49
<pokreniigru>
Unesite velicinu tabele: 6
Uneta velicina je 6
Pocetno stanje:
0 1 2 3 4 5      funkcija prvi red
A  - - - - - 7
B  - - - - - 8
C  - - - - - 9
D  - - - - - 10
E  - - - - -
F  - - - - -
G  - - - - -
H  - - - - -
I  - - - - -
J  - - - - -
K  - - - - -
Unesite potez:

```

Slika 1

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

gornja – funkcija koja služi za ispisivanje gornje polovine table, na početku se dodaje slovo a zatim naizmenično crtica (-) i zvezdica (*) a na kraju broj, kao rezultat izvršenja dobija se lista.



```

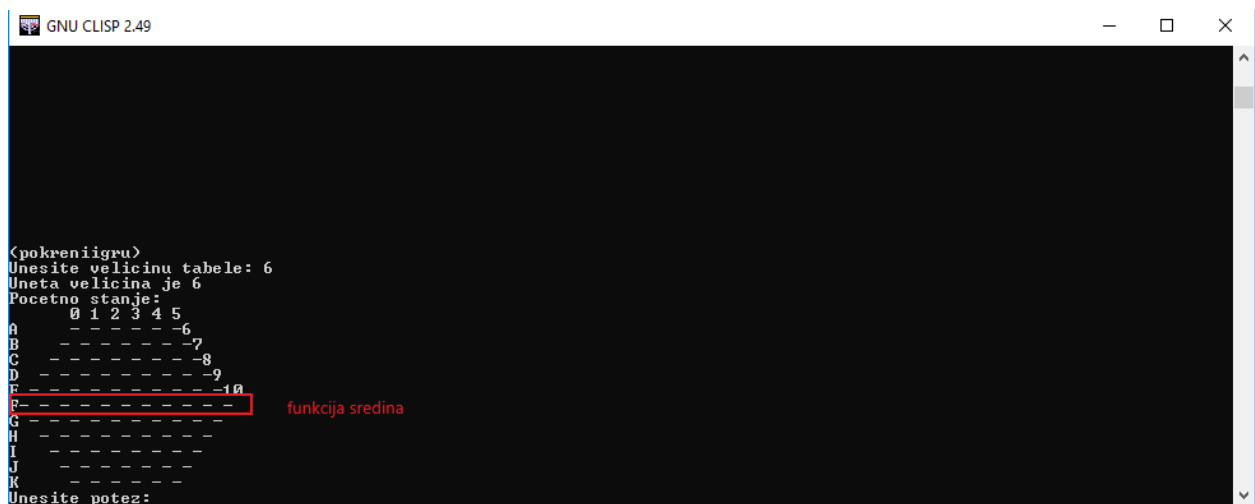
GNU CLISP 2.49
<pokreniigr>
Unesite velicinu tabele: 6
Uneta velicina je 6
Pocetno stanje:
  0 1 2 3 4 5
A - - - - -6
B - - - - -7
C - - - - -8
D - - - - -9
E - - - - -10
F - - - - -
G - - - - -
H - - - - -
I - - - - -
J - - - - -
K - - - - -
Unesite potez:

```

funkcija gornja

Slika 2

sredina - funkcija koja služi za ispisivanje srednjeg dela table, na početku se dodaje slovo a zatim naizmenično crtica (-) i zvezdica (*) a na kraju broj, kao rezultat izvršenja dobija se lista



```

GNU CLISP 2.49
<pokreniigr>
Unesite velicinu tabele: 6
Uneta velicina je 6
Pocetno stanje:
  0 1 2 3 4 5
A - - - - -6
B - - - - -7
C - - - - -8
D - - - - -9
E - - - - -10
F - - - - -
G - - - - -
H - - - - -
I - - - - -
J - - - - -
K - - - - -
Unesite potez:

```

funkcija sredina

Slika 3

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

donja - funkcija koja služi za ispisivanje donje polovine table, na početku se dodaje slovo a zatim naizmenično crtica (-) i zvezdica (*), kao rezultat izvršenja dobija se lista.

```

GNU CLISP 2.49
<pokreniigru>
Unesite velicinu tabele: 6
Uneta velicina je 6
Pocetno stanje:
  0 1 2 3 4 5
A  - - - - -
B  - - - - -
C  - - - - -
D  - - - - -
E  - - - - -
F  - - - - -
G  - - - - -
H  - - - - -
I  - - - - -
J  - - - - -
K  - - - - -
unesite potez:

```

Slika 4

spoji – funkcija koja vrši spajanje lista dobijenih prethodnim funkcijama u jednu veliku listu, samim tim se kreira početno stanje na osnovu kog će biti izvršeno crtanje, kao rezultat izvršenja dobija se lista.

Is crtavanje početnog stanja

Is crtavanje stanja na tabli se vrši funkcijom iscertaj kojoj se prosledi kao parameter lista koja sadrži trenutno stanje table i ispred svakog slova se vrši prebacivanje ostatka liste u novi red. Prilikom iscertavanja na mestu gde se nalazi zvezdica (*) vrši se ispisivanje blanko znaka ().

Odigravanje poteza i kreiranje novog stanja

Sa standardnog ulaza vrši se učitavanje pozicije na kojoj je potrebno odigrati potez. Nakon toga vrši se validacija unetog poteza i ukoliko je on validan vrši se njegovo dodavanje u listu stanja, kreiranje novog stanja a nakon toga ispisivanje tabele sa novim stanjem i promena igrača, dok ukoliko potez nije validan zahteva se unos novog poteza sve dok on nije validan.

Funkcije za odigravanje poteza su:

1. potez
2. potez1

potez – funkcija koja pronalazi zadato slovo u okviru lista i ukoliko se slovo nalazi u listi poziva funkciju potez1.

potez1 – vrši dodavanje (x ili o) na odgovarajuću poziciju. XO je promenljiva koja menja vrednost u odnosu na to ko treba da odigra potez.

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

```
GNU CLISP 2.49
(pokreniigru)
Unesite velicinu tabele: 6
Uneta velicina je 6
Pocetno stanje:
  0 1 2 3 4 5
A - - - - -6
B - - - - -7
C - - - - -8
D - - - - -9
E - - - - -10
F - - - - -
G - - - - -
H - - - - -
I - - - - -
J - - - - -
K - - - - -
Unesite potez: a 0
  0 1 2 3 4 5
A x - - - -6
B - - - - -7
C - - - - -8
D - - - - -9
E - - - - -10
F - - - - -
G - - - - -
H - - - - -
I - - - - -
J - - - - -
K - - - - -
Unesite potez:
```

Slika 5

```
GNU CLISP 2.49
I - - - - -
J - - - - -
K - - - - -
Unesite potez: a 0
  0 1 2 3 4 5
A x - - - -6
B - - - - -7
C - - - - -8
D - - - - -9
E - - - - -10
F - - - - -
G - - - - -
H - - - - -
I - - - - -
J - - - - -
K - - - - -
Unesite potez: a 3
  0 1 2 3 4 5
A x - 0 - -6
B - - - - -7
C - - - - -8
D - - - - -9
E - - - - -10
F - - - - -
G - - - - -
H - - - - -
I - - - - -
J - - - - -
K - - - - -
Unesite potez:
```

Slika 6

```
GNU CLISP 2.49
K - - - - -
Unesite potez: a 0
  0 1 2 3 4 5
A x - - - -6
B - - - - -7
C - - - - -8
D - - - - -9
E - - - - -10
F - - - - -
G - - - - -
H - - - - -
I - - - - -
J - - - - -
K - - - - -
Unesite potez: a 3
  0 1 2 3 4 5
A x - 0 - -6
B - - - - -7
C - - - - -8
D - - - - -9
E - - - - -10
F - - - - -
G - - - - -
H - - - - -
I - - - - -
J - - - - -
K - - - - -
Unesite potez: k 0
Nevalidan potez
Unesite potez ponovo:
```

Slika 7

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

Provera validnosti poteza

Validacija poteza se vrši funkcijama:

1. validan
2. validan1
3. validan2

validan – funkcija koja proverava da li uneti potez ima odgovarajuću poziciju na tabli, to jest da li je moguće odigrati dati potez na osnovu dimenzija table.

validan1 – funkcija koja pronalazi dato slovo u listi i koja poziva funkciju validan2 ukoliko je dato slovo pronađeno.

validan2 – funnkcija koja ispituje da li je na zadatom polju već odigran potez, ukoliko je odigran vraća nil, a ukoliko nije true.

```

GNU CLISP 2.49
R
Unesite potez: a 3
  0 1 2 3 4 5
A  X - - 0 - -6
B  - - - - -7
C  - - - - -8
D  - - - - -9
E  - - - - -10
F  - - - - -
G  - - - - -
H  - - - - -
I  - - - - -
J  - - - - -
K  - - - - -
Unesite potez: k 0
Nevalidan potez
Unesite potez ponovo:k 10
  0 1 2 3 4 5
A  X - - 0 - -6
B  - - - - -7
C  - - - - -8
D  - - - - -9
E  - - - - -10
F  - - - - -
G  - - - - -
H  - - - - -
I  - - - - -
J  - - - - -
K  - - - - X
Unesite potez: _

```

Slika 8

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

Pokretanje igre

Pokretanje igre se vrši pozivom funkcije pokreniigru, nakon čega se unosi veličina table.



Slika 9



Slika 10

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

Faza 2

Problemi koji su rešavani u okviru faze 2:

1. Provera završetka igre – most
2. Provera završetka igre – vila
3. Provera završetka igre – prsten
4. Generisanje svih mogućih poteza na osnovu trenutnog stanja na tabli

Generisanje grafa odigranih poteza i obilazak istog

Generisanje grafa odigranih poteza i obilazak istog vrši se sledećim funkcijama:

1. dodaj
2. dodavanje
3. dodaj_cvor
4. dodaj_pot
5. dodaj_veze
6. susedi1
7. susedi2
8. susedi
9. napraviGraf
10. ispitajCvorove
11. cvoroviGrafal
12. cvoroviGrafa
13. susediElementa
14. dodaj-potomke
15. novi-cvorovi
16. nađi-put
17. obrađen

*dodaj – funkcija koja vrši dodavanje elementa u listu koja kada se dati element ne nalazi u listi.

*dodavanje – funkcija koja vrši dodavanje novog čvora u grafu, sa njegovim potomcima i sa roditeljskim čvorovima.

*dodaj_cvor – funkcija koja vrši dodavanje čvora u grafu sa njegovim potomcima.

*dodaj_veze – funkcija koja vrši dodavanje veze između čvorova i roditeljskog čvora.

* -funkcije su preuzete sa prezentacija računskih vežbi iz predmeta Veštačka inteligencija.

susedi1 – funkcija koja određuje sve moguće susede zadatog čvora, bilo da su oni validni ili ne.

susedi2 – funkcija koja vrši validaciju pronađenih suseda za dati čvor.

susedi – funkcija koja vrši čuvanje svih validnih potomaka datog čvora u posebnu listu.

napraviGraf – funkcija koja pravi graf odigranih poteza za datog igrača (poseban graf za igrača X, poseban za igrača O).

ispitajCvorove – funkcija koja ispituje da li se zadati čvor nalazi u listi.

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

cvoroviGrafal – vraća sve čvorove koji predstavljaju odigrane poteze.

cvoroviGrafa – funkcija koja vrši čuvanje čvorova u posebnu listu (čvorovi su odigrani potezi).

susediElementa – funkcija koja pronalazi sve odigrane susede datog poteza.

*dodaj-potomke – funkcija koja izdvaja listu čvorova potomaka zadatog čvora koji do poziva nisu obrađeni algoritmom.

*novi-čvorovi – funkcija koja vraća listu koja sadrži potomke koji do poziva funkcije nisu obrađeni algoritmom.

*nađi-put – funkcija koja pronalazi put između dva zadata čvora ukoliko on postoji (korišćen je obilazak po dubini).

obrađen – funkcija koja proverava da li se dati cvor nalazi u listi obrađenih.

* -funkcije su preuzete sa prezentacija računskih vežbi iz predmeta Veštačka inteligencija.

Detektovanje ivice i ćošeta na tabli

Funkcije koje se koriste prilikom detekcije:

1. cose
2. listaIvica
3. velikaListaIvica
4. odigraneCoske

cose – funkcija koja određuje da li je zadati potez ćoš.

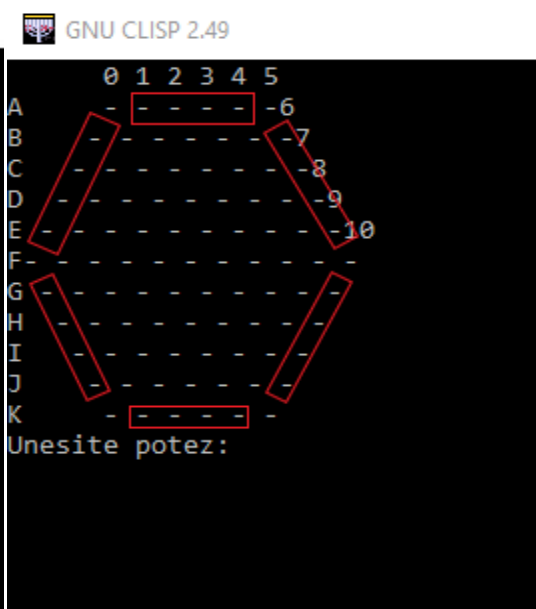
listaIvica – funkcija koja određuje da li je zadati potez ivica i čuva ga u odgovarajućoj listi (za svaku stranu table posebna lista).

velikaListaIvica – kreira listu koja sadrži podliste sa ivicama datog igrača.

odigraneCoske – funkcija koja određuje sve odigrane ćoške.



Slika 11



Slika 12

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

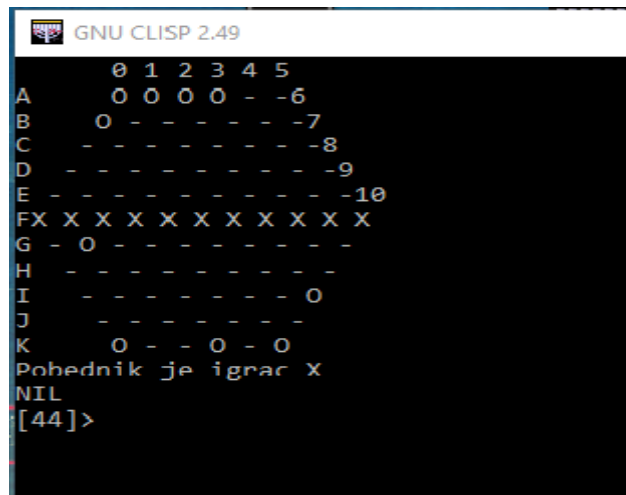
Provera završetka igre – most

Ova funkcionalnost se postiže funkcijama:

1. most1
2. most

most1 – pomoću funkcije nađi-put proverava da li postoji put između zadatog ćošeta sa bilo kojim odigranim ćošetom.

most – proverava da li postoji put između bilo koja dva ćošeta i ako postoji označava završetak igre.



Slika 13

Provera završetka igre – vila

Ova funkcionalnost se postiže funkcijama:

1. fork1
2. fork2
3. fork3
4. fork


fork1 – pomoću funkcije nađi-put proverava da li postoji put između zadate ivice sa bilo kojom ivicom na nekoj strani.

fork2 – proverava da li postoji put između bilo koje ivice na jednoj strani sa bilo kojom ivicom na drugoj strani. (dve liste koje predstavljaju odigrane ivice sa jedne i sa druge strane).

fork3 – proverava za svaku odigranu ivicu da li postoji put do neke ivice koja se nalazi na nekoj drugoj strani

fork – proverava koliko puteva postoji između ivica na suprotnim stranama, kada je broj puteva jednak 2 označava se kraj igre.

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

 GNU CLISP 2.49

```
Unesite potez: b 4
      0 1 2 3 4 5
A      X X - - 0 -6
B      0 0 0 0 0 0 7
C      - - - - - X - -8
D      - - - - - - - -9
E      - X - - - - - -10
F      - - - - - - - -
G      - - - - - X - - -
H      - - - - X - - - -
I      - - - - - - - -
J      - - - - - - - -
K      - X - X - - -
Pobednik je igrac 0
NIL
[45]>
```

Slika 14

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

Provera završetka igre – prsten

Ova funkcionalnost se postiže funkcijama:

1. disjunktniPut
2. disjunktniPut1
3. ocisti
4. ocisti1
5. testSused
6. prsten
7. test1
8. test2
9. test3
10. test4

disjunktniPut – funkcija koja proverava koliko disjunktne puteve postoji između dva čvora.

disjunktniPut1 – funkcija koja vraća sve disjunktne puteve između dva čvora.

ocisti – funkcija koja vrši brisanje čvora u grafu ukoliko se on nalazi u listi.

ocisti1 – funkcija koja ispituje da li se dati čvor grafa nalazi u listi.

testSused – funkcija koja proverava da li su dva čvora susedi.

prsten – funkcija koja proverava koliko ima disjunktne puteva koji imaju više od 3 čvora i ako je taj broj disjunktne puteva veći od 1 tada označava kraj igre pod uslovom da je ispunjen i uslov naveden u funkciji test.

test1 – funkcija koja proverava koliko dati element ima suseda u različitom disjunktne putu u odnosu na njegov.

test2 – funkcija koja proverava da li u datom disjunktne skupu svi elementi imaju manje od 2 suseda u drugom disjunktne putu u odnosu na njegov.

test3 – funkcija koja proverava da li bilo koji par disjunktne puteva ispunjava uslov naveden u test2.

test – funkcija koja proverava ispunjenost uslova u funkciju test3.

Generisanje svih mogućih poteza na osnovu trenutnog stanja na table

Ova funkcionalnost postiže se pomoću sledećih funkcija:

1. brojertica
2. bms
3. moguca

brojertica – funkcija koja određuje koliko budućih mogućih stanja ima.

bms – funkcija koja postavlja vrednost na datoj poziciji.

moguca – funkcija koja generise listu svih mogućih budućih poteza.

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

Faza 3

Problemi koji su rešavani u okviru faze 3:

1. Implementacija Min-Max algoritma sa alfa-beta odsecanjem

Min-Max algoritam

Funkcije koje su korišćene:

1. igra-min
2. igra-max
3. minimax
4. heuristika
5. vratiPotez

-minimax (stanje dubina alfa beta XO) - služi za određivanje najboljeg mogućeg poteza algoritmom Min-max sa alfa/beta odsecanjem. Funkcija se poziva za dubinu 2, alfa i beta koji su -100 i 100 respektivno kao i za igrača koji je trenutno na potezu. Zbog trenutne heuristike igra na prvo slobodno polje.

-igra-max(lp dubina alfa beta XO stanje) i igra-min(lp dubina alfa beta XO stanje)- funkcije koje paralelno pozivaju jedna drugu izvršavajući algoritam min-max za igrača koji je na potezu . Odredjuju najbolji potez sa optimizacijom u vidu alfa-beta odsecanja.

-heuristika - postavlja heuristiku za trenutno stanje. Funkcija će detaljnije biti implementirana u narednoj fazi.

-vratiPotez – služi da se odigrani potez računara smesti u graf koji je njemu namenjen.

<i>Havannah</i>	<i>Verzija:</i> 1.1
<i>Faza 1 i Faza 2</i>	<i>Datum:</i> 29.11.2018. god.

Faza 4

Problemi definisani u okviru faze 4:

1. Definicija i implementacija heuristike

Implementacija heuristike

Funkcije koje su korišćene:

1. vratiPotez1
2. heuristika

vratiPotez1- funkcija koja na osnovu trenutnog stanja na tabli vraća listu odigranih poteza u formatu (X brojSlova brojPolja)/ (O brojSlova brojPolja).

heuristika – funkcija koja na osnovu trenutnog stanja na tabli vraća heuristiku za dato stanje.

Heuristika se računa uz pomoć mašine za zaključivanje i definisanih pravila *T1-RULES*.

Pravila su definisana tako da, ukoliko je protivnik vezao dva poteza jedan pored drugog, se vrši blokiranje te sekvence poteza sa jedne od strana, zavisno od prethodno odigranih poteza.

Ukoliko pak ne postoji neka vezana sekvenca, odigraće se prvo slobodno ćoš, čine se sprečava da protivnik napravi most.