*Projekat iz veštačke inteligencije*

*Tema:*

Havannah

I Faza

*Članovi tima:*

Ana Stojanović 15905

Damjan Trifunović 15948

Milica Todorović 15937

***Sadržaj***

[**Predstavljanje table** 3](#_Toc530321231)

[**Globalne promenljive I konstante** 3](#_Toc530321232)

[**Generisanje matrične reprezentacije table** 4](#_Toc530321233)

[**Učitavanje dimenzija table I postavljanje table** 5](#_Toc530321234)

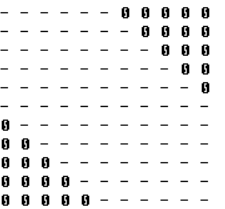
[**Štampanje table** 5](#_Toc530321235)

[**Igranje poteza** 6](#_Toc530321236)

[**Odabir igrača** 7](#_Toc530321237)

[***Reference:*** 8](#_Toc530321238)

# **Predstavljanje table**

Tabla je interno predstavljena u vidu modifikovane matrice (videti sliku). Modifikacija je izvršena da bi elementima moglo da se pristupa direktno preko indeksa matrice. Polje [A,0] table se slika u polje [0,0] matrice i predstavlja validno polje. Validno polje može imati vrednost X, O ili -, gde je X prvi igrač, O je drugi igrač a – je prazno polje. Polje [G, 0] table se slika u polje [6, 0] matrice, sto je nevalidno polje. Širina matrice je [2\*n-1, 2\*n-1].

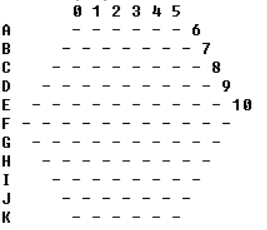


Tabla je zapamćena u matričnoj reprezentaciji kao lista redova matrice, pri čemu je svaki red predstavljen kao lista odgovarajućih elemenata datog reda.

Primer table dimenzije 4:

(( - - - - 0 0 0) ( - - - - 0 0) ( - - - - - - 0 ) ( - - - - - - -) (0 - - - - - -) ( 0 0 - - - - -) ( 0 0 0 - - - -))

# **Globalne promenljive I konstante**

**emptyField** –konstanta koja definiše prazno polje u matrici kao znak “-”

**invalidField** – konstanta koja definiše nevalidno polje u matrici kao broj “0”

**N** – dimenzija table matrice koju unosi korisnik

**matrixDim** – dimenzija matrice u matričnoj reprezantaciji table (2n - 1)

**board** – globalna promenljiva koja čuva matričnu reprezentaciju table

**firstPlayer**  - konstanta koja definiše simbol prvog igrača kao znak “X”

**secondPlayer** – konstanta koja definiše simbol drugog igrača kao znak “O”

**human** –globalna promenljiva koja definiše igrača koji je čovek

**computer** - globalna promenljiva koja definiše igrača koji je računar

# **Generisanje matrične reprezentacije table**

**(generateMembers element count)**

*Ulaz:*

Element – Element koji se ponavlja

Count – Broj koliko puta se ponavlja element

*Izlaz:*

Primer liste na izlazu ‘(0 0 0 0)

*Opis:*

Generiše listu koja sadrži uzastopno dati *element* određeni *count* broj puta.

**(generateRow firstMember firstCount secondMember secondCount)**

*Ulaz:*

firstMember – Prvi element

firstCount – Broj ponavljanja prvog elementa

secondMember – Drugi element

secondCount – Broj ponavljanja drugog elementa

*Izlaz:*

Primer liste na izlazu ‘(0 0 0 0 1 1 1 1 1 1 1)

*Opis:*

Generiše listu koja sadrži *firstMember* element uzastopno ponovljen *firstCount* broj puta I *secondMember* element uzastopno ponovljen *secondCount* broj puta. Ova funkcija koristi funkciju *generateMembers* I njome se generise jedan red u matrici koja predstavlja tablu.

**(generateMatrix firstCount secondCount lowerPart)**

*Ulaz:*

firstCount – Broj ponavljanja prvog elementa

secondCount – Broj ponavljanja drugog elementa

lowerPart – Fleg koji pokazuje da li se generise “gornja” ili “donja” polovina table

*Izlaz:*

Generisana matrica koja predstavlja tablu opisana u delu *Predstavljanje table*

*Opis:*

Generiše matricu opisanu u delu Predstavljanje table koja predstavlja tablu. *Generisanje svakog reda gornjeg dela table:* prvi element je \**emptyField*\* koji se ponavlja *firstCount* puta, dok je drugi element \**invalideField*\* koji se ponavlja *secondCount* puta pri čemu se *firstCount* u svakoj iteraciji inkrementira dok se *secondCount* dekrementira, sve dok se ne stigne do središnjeg reda table.

*Generisanje svakog reda donjeg dela table:* prvi element je \**invalideField*\*, dok je drugi element \**emptyField*\*; *firstCount* I *secondCount* zamene mesta dok dekrementiranje I inkrementiranje ostaje isto kao u generisanju gornjeg dela table, sve dok se ne stigne do kraja table.

# **Učitavanje dimenzija table I postavljanje table**

**(setDimension )**

*Izlaz:*

Učitanje dimenzije table I generisana matrična reprezentacija table.

*Opis:*

Sa standardnog izlaze se učitava dimenzija table, a zatim se setuje globalna promenljiva koja predstavlja dimenziju matrice (2\*dimenzija\_table - 1) I generiše se tabla korišćenjem funkcije *generateMatrix*.

# **Štampanje table**

**(printFirstRow count)**

*Ulaz:*

Count – Redni broj elementa u redu koji se štampa. Uvek se prosledjuje -1.

*Izlaz:*

Funkcija vraća null i štampa na standardni izlaz.

*Opis:*

Funkcija štampa novi red (~%) i prvi red matrice. Prvi red se sastoji od n+1 blanko znakova i brojeva od 0 do n-1. Koristi globalnu promenljivu \*n\*.

**(printRow rowList count rowIndex)**

*Ulaz:*

RowList – Red koji se štampa, u formatu (<nule> <validna\_polja>)

Count – Redni broj elementa koji se štampa. Uvek se prosledjuje -1.

RowIndex – Indeks reda koji se štampa.

*Izlaz:*

Funkcija vraća null i štampa na standardni izlaz.

*Opis:*

Funkcija štampa novi red, slovo, red matrice i broj (po potrebi). Transformiše nule iz matrice u blanko znake, i određuje slovo i broj na osnovu indeksa reda. Koristi globalne promenljive \*n\* i \*matrixDim\*.

**(printBoardRecursive board row)**

*Ulaz:*

Board – Tabla koja se štampa

Row – Indeks reda koji se štampa, uvek se prosledjuje -1

*Izlaz:*

Funkcija vraća null i štampa na standardni izlaz.

*Opis:*

Funkcija štampa tablu red po red, uključujući specijalni prvi red. Poziva funkcije printRow i printFirstRow, i koristi globalnu promenljivu \*n\*.

**(printBoard)**

*Ulaz:*

Nema

*Izlaz:*

Null

*Opis:*

Poziva funkciju printBoardRecursive i prosledjuje joj parametre. koristi globalnu promenljivu \*board\*. Proverava da li su odgovarajuće globalne promenljive setovane pre poziva.

# **Igranje poteza**

**(setElement el i j matrix)**

*Ulaz:*

el – element koji se ubacuje u matricu

i – indeks reda matrice

j – indeks kolone matrice

matrix – matrica u koju se element unosi

*Izlaz:*

Matrica sa unetim elementom na [i, j] poziciji

*Opis:*

Funkcija služi za ubacivanje prosleđenog elementa na poziciju [i, j] u prosleđenoj matrici i kao rezultat vraća izmenjenu matricu.

**(validateMove i j )**

*Ulaz:*

i – index vrste

j – index kolone

*Izlaz:*

True ili false u zavisnosti od toga da li je potez validan.

*Opis:*

Funkcija validira igranje poteza na datoj poziciji u matrici. Index vrste I index kolone su validni ukoliko su pozitivni brojevi, dok je potez validan ukoliko se na poziciji [I,j] nalazi \**emptyField*\*.

**(playMove player)**

*Ulaz:*

player – oznaka igrača koji igra potez

*Izlaz:*

Globalna promenljiva \*board\* dobija vrednost matrice sa odigranim potezom.

*Opis:*

Funkcija zahteva od korisnika da unese koordinate polja (slovo za red, broj za kolonu). Ako su uneti podaci validni, matrica table se ažurira tako da sadrži potez koji je odigran i ta vrednost se upisuje u globalnu promenljivu koja čuva matricu table.

# **Odabir igrača**

**(choosePlayer)**

*Ulaz:*

/

*Izlaz:*

Postavljaju se globalne promenljive koje određuju da li je prvi igrač računar ili čovek.

*Opis:*

Funkcija zahteva od korisnika da unese da li će prvi igrač biti čovek ili računar (korisnik unosi “h” ili “c”). Postavljaju se globalne promenljive \*human\* i \*computer\* vrednostima \*firstPlayer\* (‘X) i \*secondPlayer\* (‘O), u zavisnosti od izbora korisnika.

# ***Reference:***

<http://havannah.ewalds.ca/static/thesis.pdf>