

Projekat

Domineering

Tema projekta: Rešavanje problema kao traženje.

Ciljevi projekta:

Student treba da demonstrira znanje iz sledećih oblasti:

1. Formulacija problema
2. Implementacija algoritama za traženje u izabranom programskom jeziku / okruženju
3. Implementacija procene stanja korišćenjem pravila i zaključivanja.

Osnovne informacije

Broj poena: Projekat nosi maksimalno **35%** od konačne ocene. Poeni se odnose na aktivnost i zalaganje studenta, kao i na kvalitet urađenog rešenja.

Status: Projekat je **obavezan**! Očekuje od studenata da ozbiljno shvate zaduženja. Ukoliko projekat ne bude urađen u navedenom roku, naredna prilika je tek sa sledećom generacijom, po pravilima koja će biti tada definisana!

Početak projekta: **19.04.2018. godine**

Rok za predaju projekta i završnog izveštaja: **julski rok**

Realizacija projekta: Projekat se radi individualno.

Podrazumevana pravila ponašanja:

1. Probajte da uradite projekat sami, bez pomoći kolega ili prepisivanja.
2. Poštujte tuđi rad! Materijal sa Web-a i iz knjiga i radova možete da koristite, ali samo pod uslovom da za sve delove koda ili rešenja koje ste uzeli od nekog navedete referencu!
3. Ne dozvolite da od vas neko prepisuje tj. da neko od kolega koristi vaš rad i vaše rezultate!

Faze izrade projekta:

1. Formulacija problema.
2. Implementacija interfejsa
3. Implementacija Min-max algoritma za traženje sa α - β odsecanjem.
4. Definicija heuristike (procena stanja)

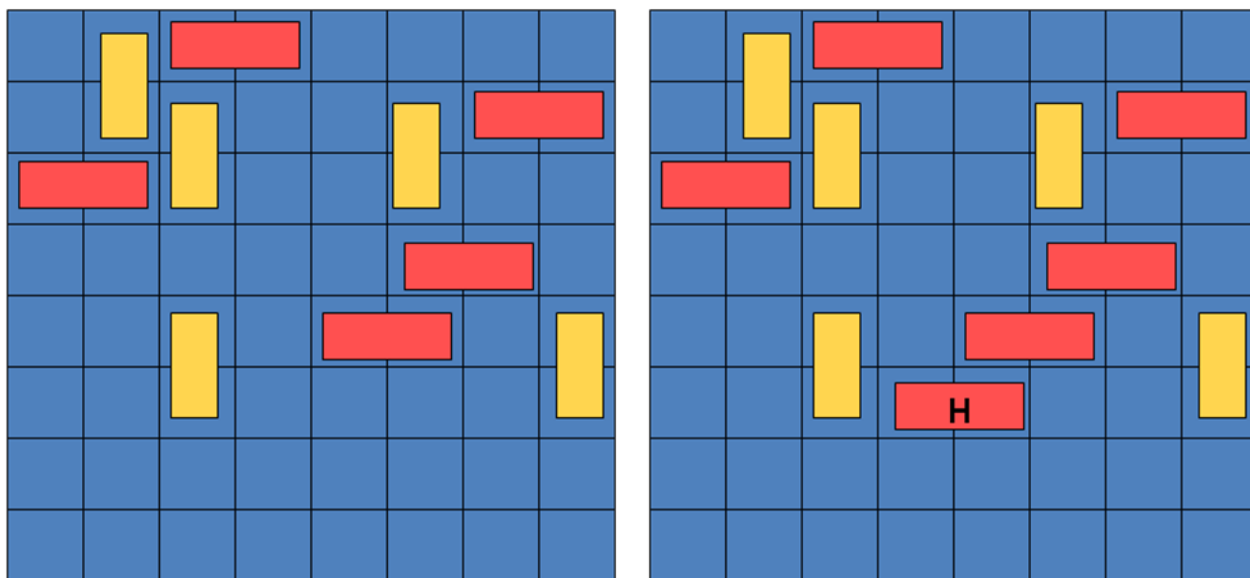
Kao rezultat svake faze projekta piše se izveštaj koji treba da sadrži opis funkcionalnosti koje su urađene. Uz izveštaj predati kod implementiranih funkcija.

Opis problema

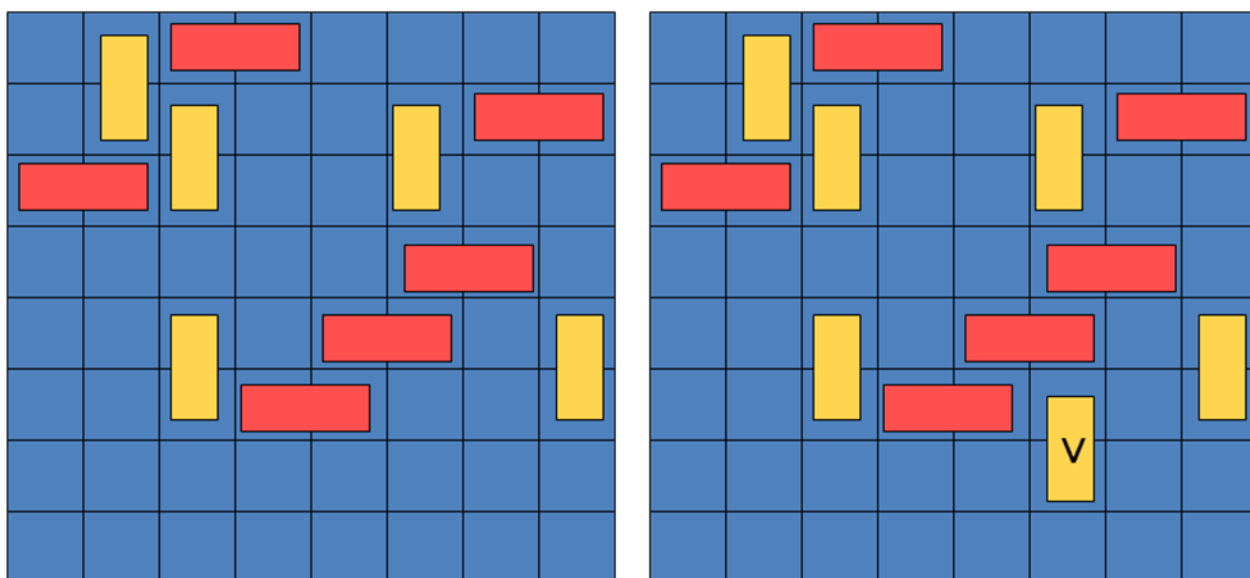
Problem je igra *Domineering*. Igra se odvija na tablu proizvoljnih dimenzija $m \times n$. Na početku igre tabla je prazna. Igraju dva učesnika tako što na tablu postavljaju pločice veličine 2×1 . Jedan igrač uvek postavlja pločice horizontalno (sl.1), dok ih drugi uvek postavlja vertikalno (sl.2). Pločice se ne smeju preklapati, niti viriti van table. Igra se završava kada jedan od igrača ne može da spusti pločicu na tablu.

NAPOMENE:

1. Obezbediti mogućnost da se izabere igrač (čovek ili računar) koji će da povlači prvi potez.
2. Prvi potez povlači igrač koji igra horizontalno.



Slika 1. Primer postavljanja horizontalne pločice



Slika 2. Primer postavljanja vertikalne pločice

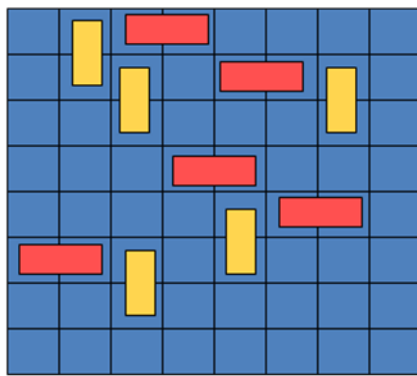
Zadatak

Potrebno je implementirati funkcije koje realizuju navedeni problem.

Student predaje izvorni kod, kao i dodatni dokument koji sadrži opis rešenja.

Konkretno, realizacija projekta podrazumeva sledeće elemente:

1. Formulacija problema.
 - a. Definirati predstavljanje stanja problema i operatore prelaza iz jednog stanja u drugo
 - b. Definirati reprezentaciju stanja u izabranom programskom jeziku
 - c. Napisati funkcije za operatore koje ste definisali
 - d. Napisati funkcije za testiranje ciljnog stanja (provera da li je moguće odigrati potez ili ne).
2. Implementirati interfejs
 - a. Interfejs treba da omogući definisanje veličine table (kvadratnog oblika $n \times n$)
 - b. Interfejs treba da omogući izbora ko će igrati prvi potez (čovek ili računar)
 - c. Interfejs treba da omogući prikaz trenutnog stanja i unos sledećeg poteza
 - d. Prikaz trenutnog stanja treba da bude implementiran na način koji je ilustrovan na slici 3.
 - e. Unos poteza realizovati jednostavnom funkcijom koja čita podatak sa standardnog ulaza.
3. Implementacija Min-Max algoritma za traženje
 - a. Implementirati Min-Max algoritam sa alfa-beta odsecanjem za navedeni problem.
 - b. Obezbediti da funkcija za Min-Max ima ulazni parametar koji definiše dubinu pretraživanja.
 - c. Obezbediti da funkcija za Min-Max vraća potez koji treba odigrati u odgovarajućem formatu.
4. Definicija heuristike
 - a. U implementaciju Min-Max-a dodati funkciju za procenu stanja kada se dostigne zadata dubina traženja.
 - b. Implementirati funkciju koja vrši procenu stanja na osnovu pravila i zaključivanja.
 - c. Funkcija za procenu stanja kao parametre treba da ima oznaku igrača za kojeg računa valjanost stanja, kao i samu tablu za koju se računa procena.
 - d. Procena stanja se mora vršiti isključivo korišćenjem mehanizma zaključivanja nad prethodno definisanim skupom pravila. Vaš zadatak je formulacija skupa pravila i iskoristiti ih na adekvatan način za izračunavanje heuristike.
 - e. Za izvođenje potrebnih zaključaka (izvršavanje upita nad skupom činjenica kojima se opisuje stanje table) koristiće se mašina za zaključivanje koja će vam biti dostavljena.
 - f. Implementirati funkciju koja prevodi stanje table u listu činjenica koje koriste samo predikate (On <pločica> i j) i (Empty i j). Za sopstvene figure koristiti konstantu X, a za protivničke O. Indeks i odgovara vrstama (1, ..., 8), a j kolonama (A->1, ..., H->8). Primer je dat na slici 3.



8	-	V	H	H	-	-	-	-
7	-	V	V	-	H	H	V	-
6	-	-	V	-	-	-	V	-
5	-	-	-	H	H	-	-	-
4	-	-	-	-	V	H	H	-
3	H	H	V	-	V	-	-	-
2	-	-	V	-	-	-	-	-
1	-	-	-	-	-	-	-	-
	A	B	C	D	E	F	G	H

```
(
  (On H 3 1) (Empty 5 1) (Empty 7 1)
  (Empty 1 1) (On H 3 2) (Empty 5 2) (On V 7 2)
  (Empty 1 2) (On V 2 3) (Empty 5 3) (On V 7 3)
  (Empty 1 3) (Empty 3 4) (On H 5 4) (Empty 7 4)
  (Empty 1 4) (On V 3 3) (On H 5 5) (On H 7 5)
  (Empty 1 5) (Empty 3 6) (Empty 5 6) (On H 7 6)
  (Empty 1 6) (Empty 3 7) (Empty 5 7) (On V 7 7)
  (Empty 1 7) (Empty 3 8) (Empty 5 8) (Empty 7 8)
  (Empty 1 8)
  (Empty 4 1) (Empty 6 1) (Empty 8 1)
  (Empty 2 1) (Empty 4 2) (Empty 6 2) (On V 8 2)
  (Empty 2 2) (Empty 4 3) (On V 6 3) (On H 8 3)
  (On V 2 3) (Empty 4 4) (Empty 6 4) (On H 8 4)
  (Empty 2 4) (On V 4 5) (Empty 6 5) (Empty 8 5)
  (Empty 2 5) (On H 4 6) (Empty 6 6) (Empty 8 6)
  (Empty 2 6) (On H 4 7) (On V 6 7) (Empty 8 7)
  (Empty 2 7) (Empty 4 8) (Empty 6 8) (Empty 8 8)
  (Empty 2 8)
)
```

Slika 3. Primer prevođenja stanja table u listu činjenica

prof. dr Edin Dolićanin,
Aldina Avdić, dipl. inž.