Projekat – *Dots*

Aleksandar Milosavljević
Vladan Mihajlović

Osnovne informacije

- Cilj projekta:
  - Formulacija problema
  - Implementacija algoritama za traženje (algoritama za igre)
  - Implementacija procene stanja korišćenjem pravila i zaključivanja
- Jezik: Lisp
- Maksimalan broj ljudi po projektu: 3
- Datum objavljivanja projekta: 16.03.2011.
- Rok za predaju: 16.05.2011.

Ocenjivanje

- Broj poena:
  - Projekat, zajedno sa laboratorijskim vežbama nosi maksimalno 35% od konačne ocene.
  - Poeni se odnose na aktivnost i zalaganje studenta, kao i na kvalitet urađenog rešenja.
- Status:
  - Projekat je obavezan! Minimalni broj poena koji se mora osvojiti je 5!
  - Očekuje od vas da ozbiljno shvatite zaduženja.
  - Ukoliko ne uradite projekat u navedenom roku, naredna prilika je tek sa sledećom generacijom, po pravilima koja će biti tada definisana!

Takmičenje/turnir

- Posle predaje projekta biće organizovano takmičenje.
- Planirani termin takmičenja je kraj maja.
- Prva tri mesta na turniru donose dodatne poene: 5 za prvo mesto, 3 za drugo i 2 za treće mesto (računaju se kao poeni za angažovanje u toku semestra).

Pravila ponašanja

- Probajte da uradite projekat sami, bez pomoći kolega ili prepisivanja.
- 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!
- Ne dozvolite da od vas neko prepisuje, tj. da neko od kolega koristi vaš rad i vaše rezultate!
- Ako radite u timu, ne dozvolite da vaš kolega iz tima ne radi ništa! Nađite mu zaduženja koja može da uradi – ako mu nešto ne ide, nađite mu druga zaduženja.

Faze izrade projekta

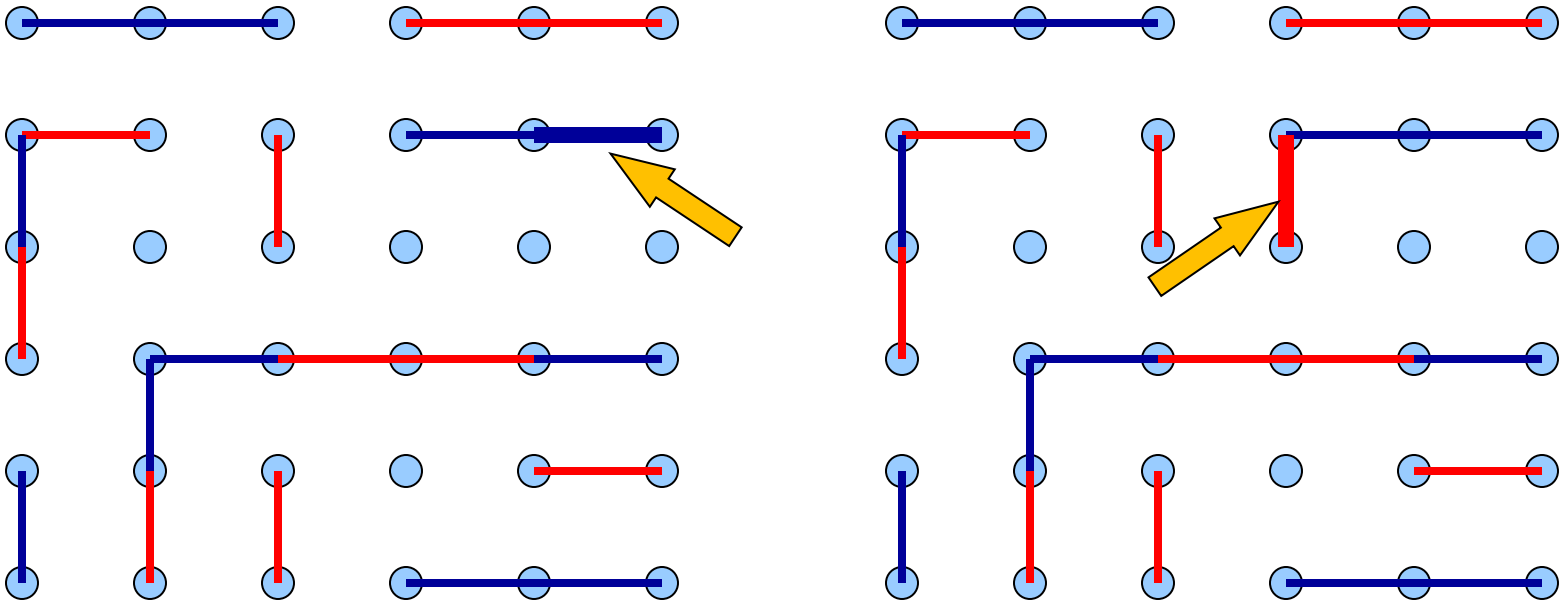
- Implementacija interfejsa ka korisniku
  - Rok: 28.03.2011. god
- Formulacija problema i promene stanja
  - Rok: 11.04.2011. god
- Implementacija Min-Max algoritma za traženje sa alfa-beta odsecanjem
  - Rok: 02.05.2011. god
- Definicija heuristike (procena stanja)
  - Rok: 16.05.2011. god

- Rezultat svake faze je izveštaj

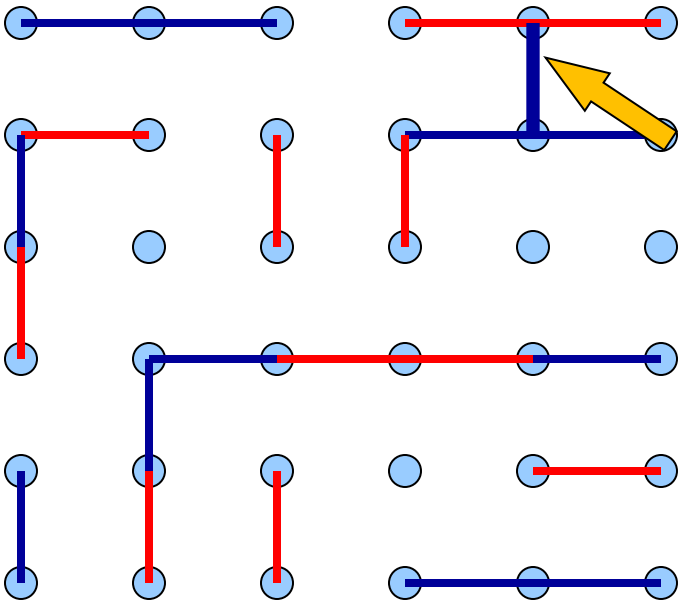
Opis problema

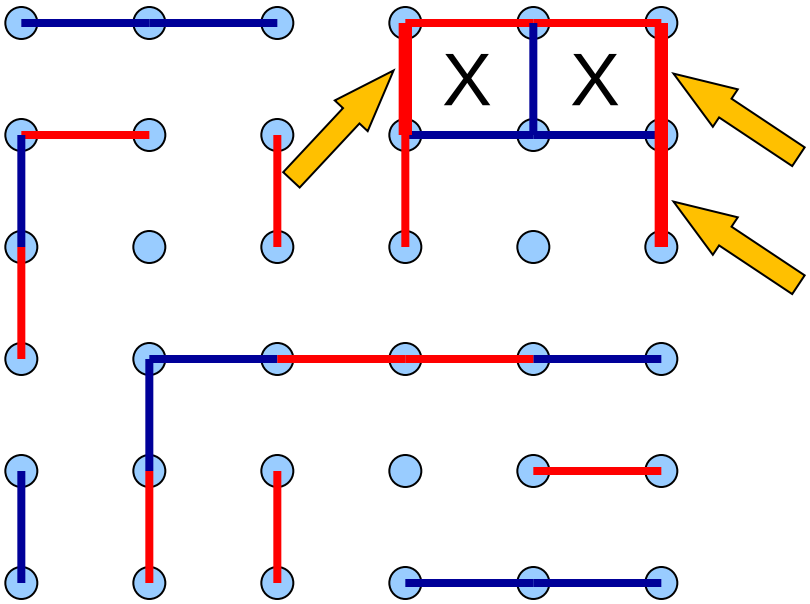
- Problem je igra *Dots*.
- Broj tačaka je n *x* n.
- Na početku igre ne postoji nijedna veza između tačaka.
- Postoje dva igrača koji naizmenično postavljaju linije između bilo koje dve susedne tačke
- Ukoliko se nakon postavljanja linije zatvara kvadrat u polje se upisuje vrednost svoju oznaku (x ili o). Nakon toga isti igrač ima pravo na novi potez.
- Dozvoliti izbor da li prvi igra čovek ili računar.
- Pobednik je onaj igrač koji ima više označenih polja.

Dots – primer igre



Dots – primer igre



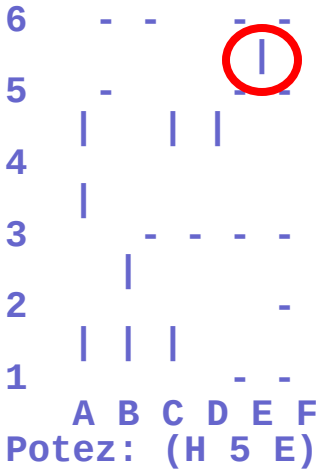


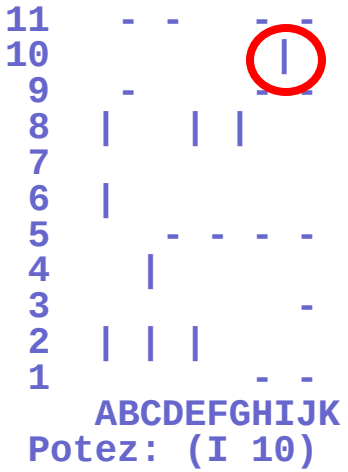
Zadatak

- Potrebno je implementirati funkcije u Lisp-u koje realizuju navedeni problem.
- Student predaje izvorni kod u Lisp-u, kao i dodatni dokument koji sadrži opis rešenja.
- Konkretna realizacija projekta podrazumeva sledeće elemente =>

Zadatak I
Implementacija funkcija za interfejs

1. Obezbediti definisanje veličine polja na kome se igra (broj tačaka u vrsti).
2. Interfejs treba da omogući prikaz trenutnog stanja i unos poteza.
3. Obezbediti izbor ko će igrati prvi (čovek ili računar).
4. Prikaz trenutnog stanja treba da bude implementiran na jedan od načina koji je ilustrovan sa sledeća dva primera:





5. Unos poteza realizovati jednostavnom funkcijom koja čita podatak (listu) sa standardnog ulaza.
6. Potrebno je izvršiti proveru validnosti poteza, tj. da li je polje u predviđenom opsegu i da li postoji potez na zadatom mestu.

Zadatak II
Formulacija problema

1. Definisati operatore prelaza iz jednog stanja u drugo
2. Napisati Lisp funkcije za operatore promene stanja koje ste definisali
3. Napisati funkcije za testiranje ciljnog stanja (provera da li je peostalo neko prazno polje)
4. Napisati funkcije za odigravanje niza poteza kada unos jednog poteza izazove popunjavanje više polja

Zadatak III
Implementacija Min-Max algoritma za traženje

1. Implementirati Min-Max algoritam sa alfa-beta odsecanjem za navedeni problem
2. Obezbediti da funkcija Min-Max sa alfa-beta odsecanjem ima ulazni parametar kojim se definiše dubina pretraživanja
3. Funkcija Min-Max sa alfa-beta odsecanjem treba da vrati stanje u koje treba preći

Zadatak IV
Definicija heuristike (procena stanja)

1. U implementaciju Min-Max-a sa alfa-beta odsecanjem dodati funkciju za procenu stanja kada se dostigne zadata dubina traženja.
2. Implementirati funkciju koja vrši procenu stanja na osnovu pravila i zaključivanja.
3. 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.
4. 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.
5. Za izvođenje potrebnih zaključaka (izvršavanje upita nad skupom činjenica kojima se opisuje stanje table) koristiće se mašina za zaključivanje koja će vam biti dostavljena.
6. Implementirati funkciju koja prevodi stanje table u listu činjenica …