Katedra za računarstvo

Elektronski fakultet, Univerzitet u Nišu

Veštačka inteligencija

Projekat – Trouglići (*Triggle*)

Osnovne informacije

- Cilj projekta:
  - Formulacija problema
  - Implementacija algoritma za traženje (algoritma za igru)
  - Implementacija procene stanja korišćenjem pravila i zaključivanja
- Jezik: Python
- Broj ljudi po projektu: 3
- Datum objavljivanja projekta: 12.11.2024. godine
- Rok za predaju: 29.12.2024. godine

Ocenjivanje

- Broj poena:
  - Projekat nosi maksimalno 20% od konačne ocene
  - Poeni se odnose na kvalitet urađenog rešenja, kao i na aktivnost i zalaganje studenta
- Status:
  - **Projekat je obavezan!**
  - **Minimalni broj poena koji se mora osvojiti je 5!**
  - Očekuje se od studenata da ozbiljno shvate zaduženja!
  - Ukoliko ne uradite projekat u predviđenom roku, naredna prilika je tek sa sledećom generacijom, po pravilima i na temu koja će biti definisana za novi projekat!

Takmičenje/turnir

- Posle predaje projekta biće organizovano takmičenje.
- Planirani termin takmičenja je sredina januara.
- Prva tri mesta na turniru donose dodatne bodove:
  - 5 bodova za prvo mesto,
  - 3 boda za drugo i
  - 2 boda za treće mesto
- Računaju se kao dodatni bodovi se za angažovanje u toku semestra.

Pravila ponašanja

- Probajte da uradite projekat samostalno, bez pomoći kolega iz drugih timova i 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 preuzeli navedete referencu!
- Ne dozvolite da drugi prepisuje od vas, tj. da drugi koristi vaš rad i vaše rezultate!
- Ne dozvolite da član tima ne radi ništa! Dogovorite se i pronađite zaduženja koja on može da uradi. Ako mu ne ide, pronađite druga zaduženja.

Faze izrade projekta

- Formulacija problema i implementacija interfejsa
  - Rok: 1.12.2024. godine
- Implementacija operatora promene stanja
  - Rok: 15.12.2024. godine
- Implementacija Min-Max algoritma za traženje sa alfa-beta odsecanjem i algoritama za procenu stanja (heuristike)
  - Rok: 29.12.2024. godine

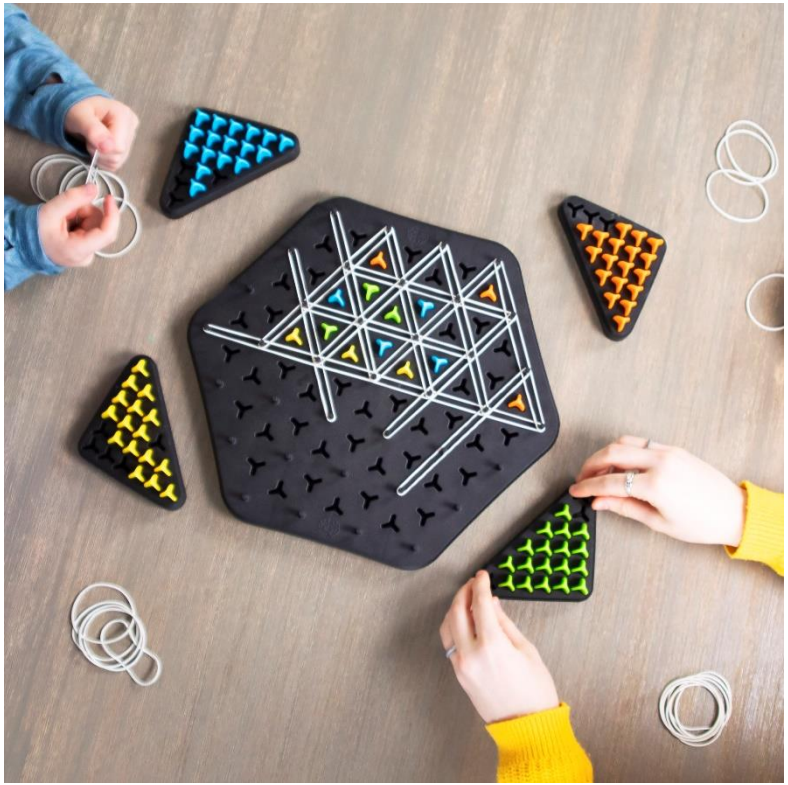
- Rezultat svake faze je izveštaj koji sadrži dokument sa obrazloženjem rešenja i datoteku (datoteke) sa kodom.

Igra Trouglići (*Triggle*)

Opis problema Trouglići (*Triggle*)

- Problem je igra Trouglići (*Triggle*).
- Strateška igra zauzimanja trouglova ograničavanjem gumicama na tabli.
- Tabla je oblika pravilnog šestougla stranice n.
  - Preporučena dužina stranice je 4
  - Maksimalna veličina je 8
- Dva igrača crni i beli (X i O) naizmenično odigravaju po jedan potez.
- Na svaku jedinicu dužine nalaze se stubići.
- Igrači razvlače gumice dužine 3 jedinice između stubića i zauzimaju trouglиće.
- Na početku je tabla prazna bez i jedne razvučene gumice.
- Pobednik je igrač koji zauzme veći broj trouglića na tabli.
- Igra čovek protiv računara i moguće izabrati da prvi igra čovek ili računar

Trouglići (*Triggle*) – Stanje u toku igre



Elementi igre Trouglići (*Triggle*)

- Gumice:
  - Sve gumice su identične i nema posebnih gumica za igrača X niti O.
  - Gumice je moguće razvlačiti između stubića na tabli.
  - Gumica se razvlači tako da je njena dužina iznosi tačno 3 (povezuje 4 stubića).
  - Ako se razvlačenjem formiraju trouglići oni se zauzimaju i označavaju od strane igrača koji je razvukao gumicu.
  - Zauzimaju se samo jednakostranični trouglići čija je dužina stranice 1.
- Pobednik je igrač koji je zauzeo više trouglića.
  - Ako je igrač zauzeo više od polovine trouglića (npr. 28 za tablu stranice 4) igra može da se prekine, jer je pobednik poznat.

Potezi u igri Trouglići (*Triggle*)

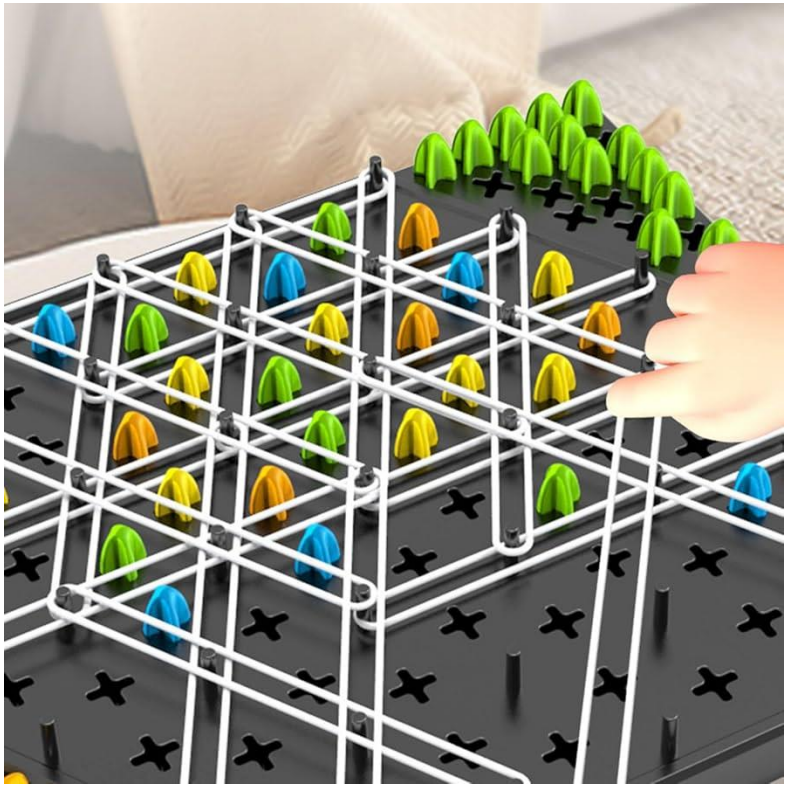
- Gumica se razvlači tako da je njena dužina iznosi tačno 3 (povezuje 4 stubića).
- Pravac rastezanja gumice može biti bilo koji od 3 moguća na tabli.
- Igrač u jednom potezu može razvući samo jednu gumicu.
- Ako se razvlačenjem formiraju trouglići, svi oni se zauzimaju od strane igrača koji je razvukao gumicu i označavaju odgovarajućom oznakom (X ili O).
- Potezi se odigravaju sve dok je moguće razvući gumicu.

Potezi u igri Trouglići (*Triggle*)

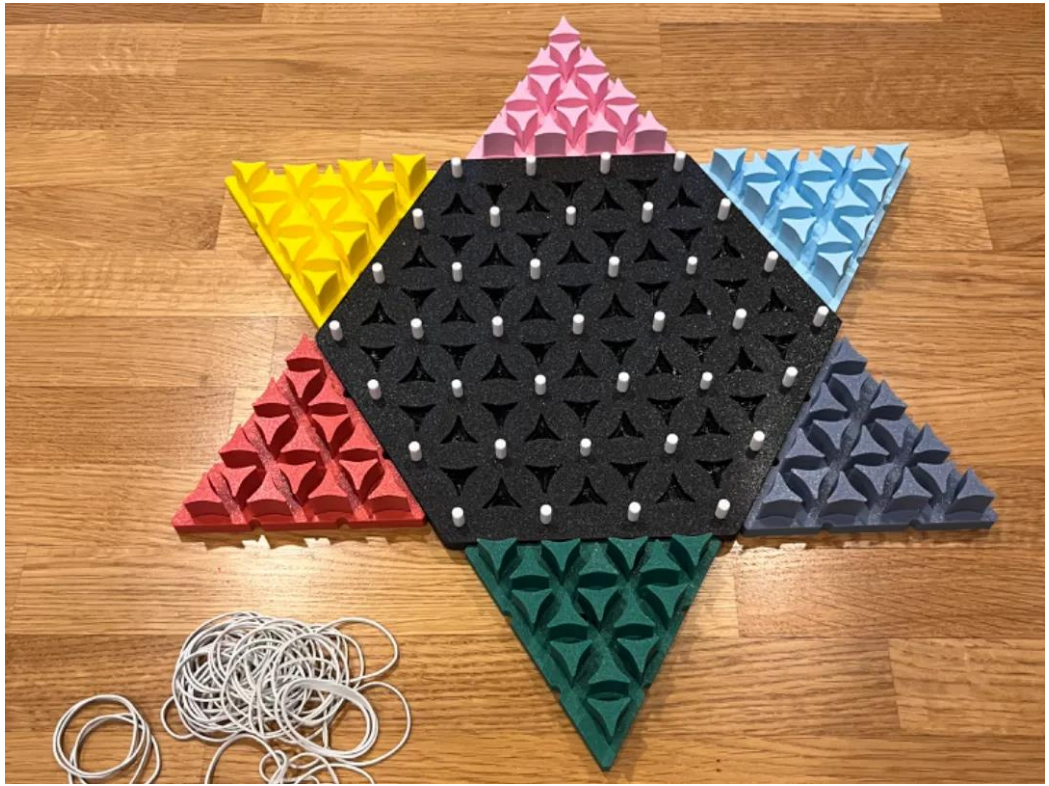
Dozvoljeni potezi u igri Trouglići (*Triggle*)

- Gumica se može razvući tako da se delom poklapa sa već razvučenim gumicama.
- Gumica se ne sme razvući između stubića tako da se celom svojom dužinom poklapa sa već razvučenim gumicama.
- Gumica se razvlači samo duž jednog pravca i nije dozvoljeno menjati pravac tokom razvlačenja (npr. razvući 2 dužine horizontalno i 1 koso).
- Igrač ne sme preskočiti potez.

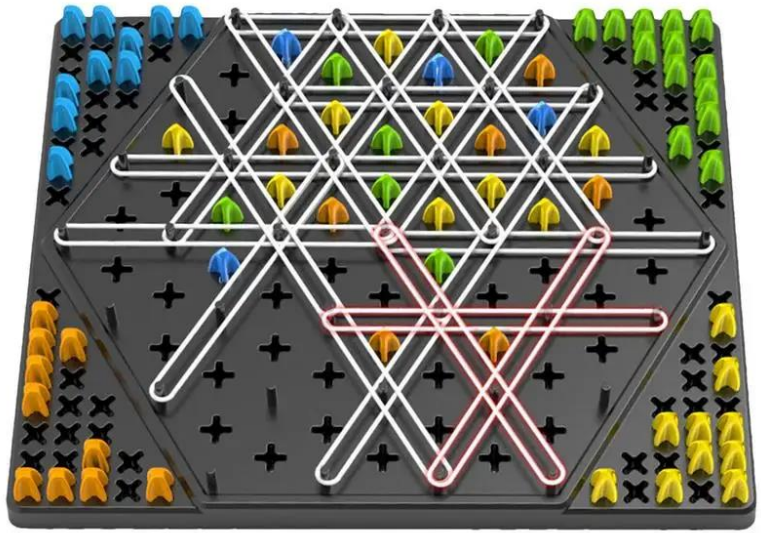
Dozvoljeni potezi u igri Trouglići (*Triggle*)



Trouglići (*Triggle*) – Početak igre



Trouglići (*Triggle*) – Primer stanja igre



Trouglići (*Triggle*) – Korisni linkovi

- Opis igre sa pravilima:
  - https://boardgamegeek.com/boardgame/405711/triggle

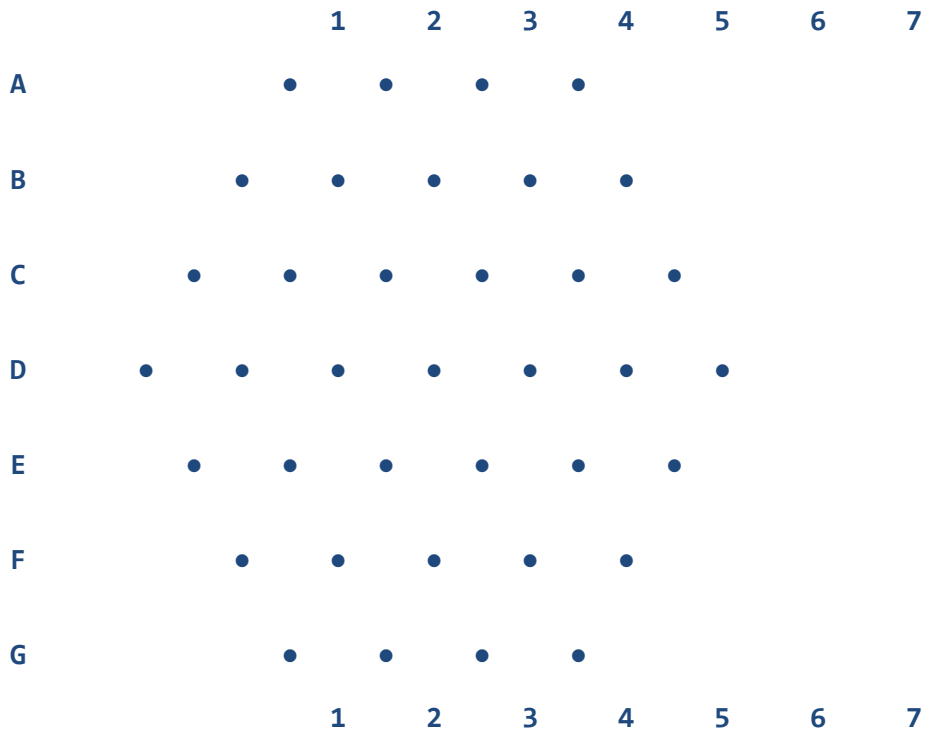
Zadatak I – Formulacija problema i interfejs

- Definisati način za predstavljanje stanja problema (igre)
  - Predstavljanje pozicija razvučenih gumica i zauzetih trouglića
- Napisati funkciju za postavljanje početnog stanja
  - Definiše se na osnovu zadate veličine table
- Napisati funkcije za proveru kraja igre
  - Tabla je popunjena razvučenim gumicama ili je igrač zauzeo više od polovine trouglića
- Napisati funkcije koje proveravaju ispravnost unosa poteza
- **NIJE POTREBNO realizovati funkcije koje proveravaju valjanost poteza i odigravaju potez (faza II)**
- **NIJE POTREBNO realizovati funkcije koje obezbeđuju odigravanje partije (faza II)**

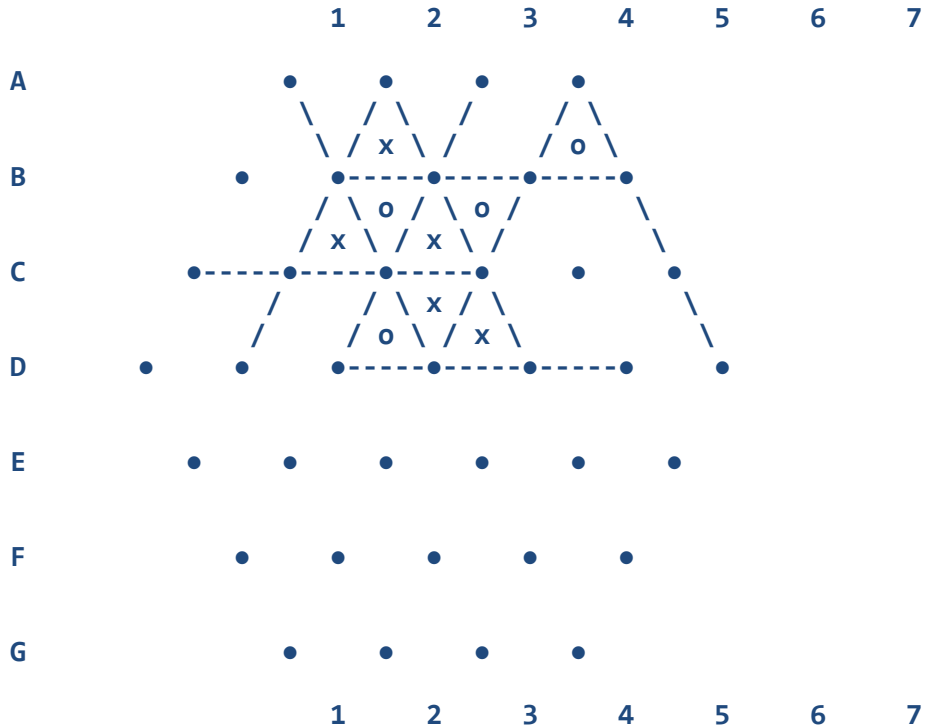
Zadatak I – Formulacija problema i interfejs

- Omogućiti izbor ko će igrati prvi (čovek ili računar)
- Omogućiti izbor koji igrač igra prvi (X ili O)
- Implementirati funkcije koje obezbeđuju unos početnih parametara igre
  - Unos dužine stranice table (n) i provera ispravnosti unosa
- Implementirati funkcije koje obezbeđuju pravljenje inicijalnog stanja problema (igre)
  - Pravljenje stanja igre na osnovu zadatih dužine stranice table (n)
- Implementirati funkcije koje obezbeđuju prikaz proizvoljnog stanja problema (igre)
  - Prikaz trenutne (proizvoljne) situacije na tabli sa pozicijama razvučenih gumica i zauzetih trouglića
- Realizovati funkcije za unos poteza
  - Potez se sastoji od pozicije polja i smera razvlačenja (D, DL, DD)
- Realizovati funkcije koje proveravaju da li je unos poteza tačan
  - Proveriti da li je zadata oznaka pozicije stubića jedna od mogućih
  - Proveriti da li je smer jedan od tri moguća

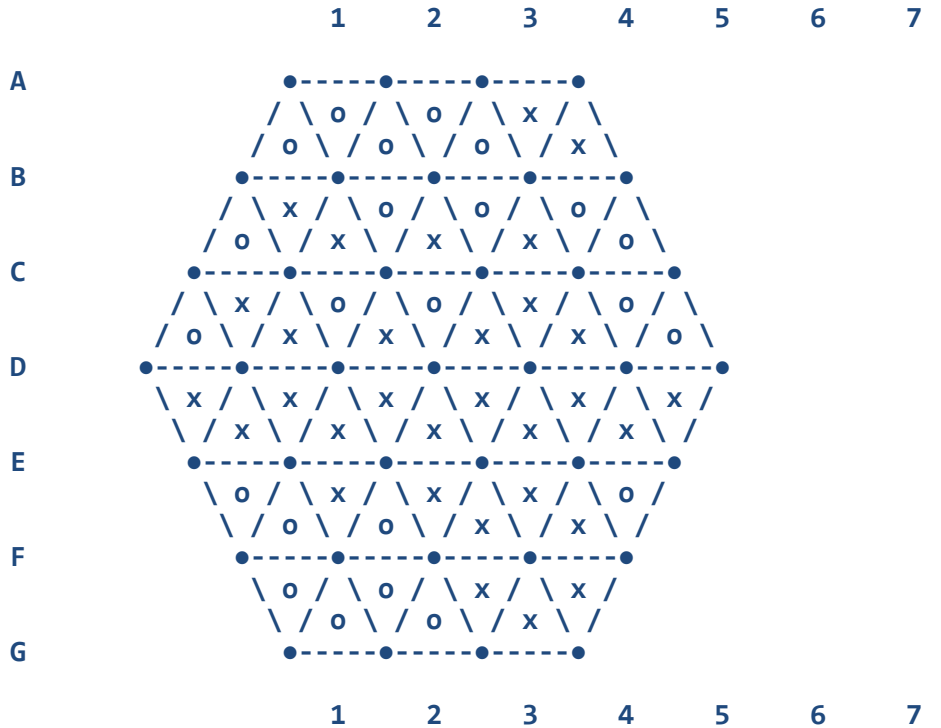
Zadatak I – Interfejs (početno stanje)



Zadatak I – Interfejs (trenutno stanje)



Zadatak I – Interfejs (krajnje stanje)

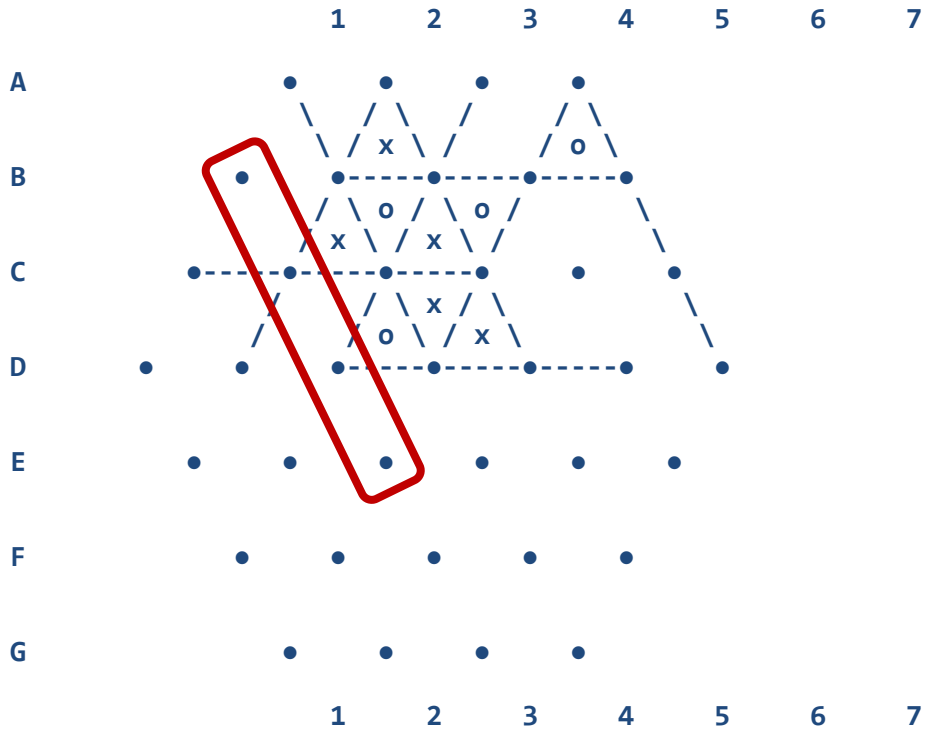


Pobeda X igrača

X: 31

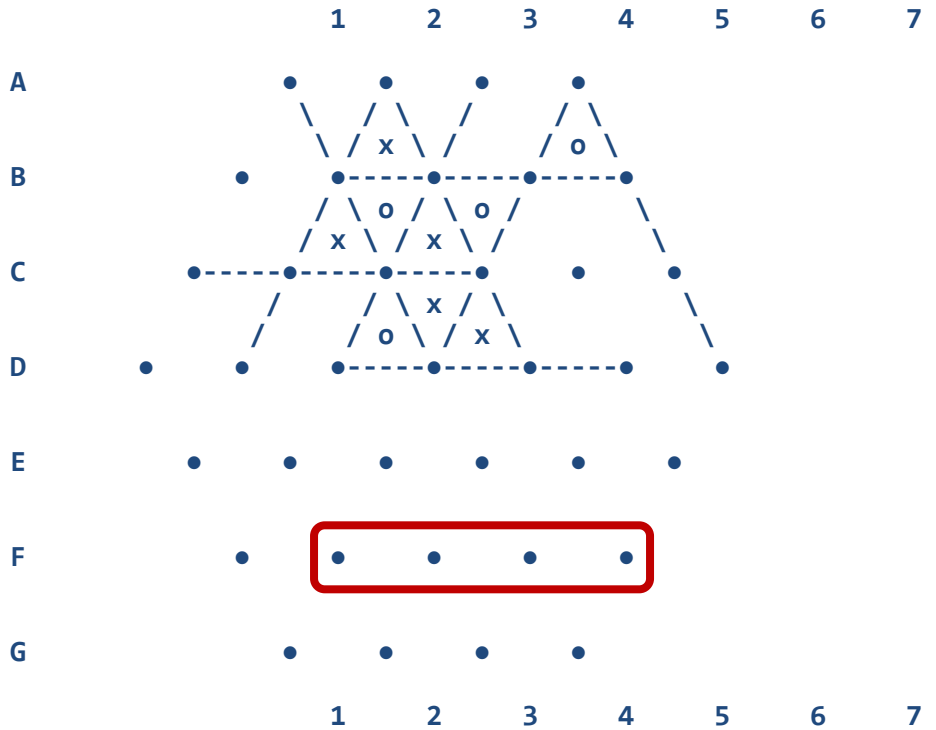
O: 23

Zadatak I – Interfejs (unos poteza)



- Potez O igrača:
- Gornja polovina:
  - ('B', 1, 'DD')

Zadatak I – Interfejs (unos poteza)



- Potez O igrača:
- Donja polovina:
  - ('F', 2, 'D')

Zadatak II – Operator promene stanja

- Napisati funkcije za proveru valjanosti poteza na osnovu konkretnog poteza i trenutnog stanja problema (igre)
- Napisati funkcije koje na osnovu konkretnog poteza menjaju stanje problema (igre)
- Napisati funkcije koje obezbeđuju odigravanje partije između dva igrača (**dva čoveka, ne računara i čoveka**)
  - Unos početnih parametara i naizmenični unos poteza uz prikaz izgleda stanja igre nakon svakog poteza
- Napisati funkcije za operator promene stanja problema (igre) u opštem slučaju (proizvoljno stanje na tabli)
  - Određivanje svih mogućih poteza igrača na osnovu stanja problema (igre)

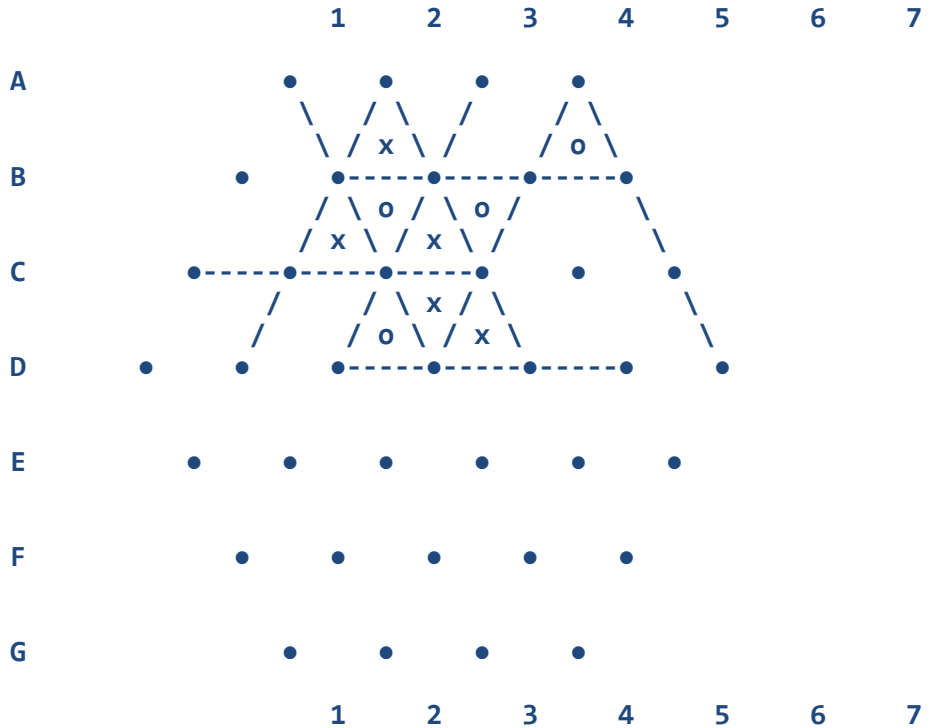
Zadatak II – Operator promene stanja

- Realizovati funkcije koje na osnovu konkretnog poteza i stanja problema (igre) proveravaju njegovu valjanost
  - Proveriti da li zadato polje postoji na tabli
  - Proveriti da li je razvlačenje gumica valjano, tj. da se ne dešava da se gumica celom dužinom poklapa sa prethodno razvučenim.
- Realizovati funkcije koje na osnovu konkretnog poteza menjaju stanje problema (igre)
  - Realizovati funkcije koje na osnovu konkretnog poteza menjaju trenutno stanje igre

Zadatak II – Operator promene stanja

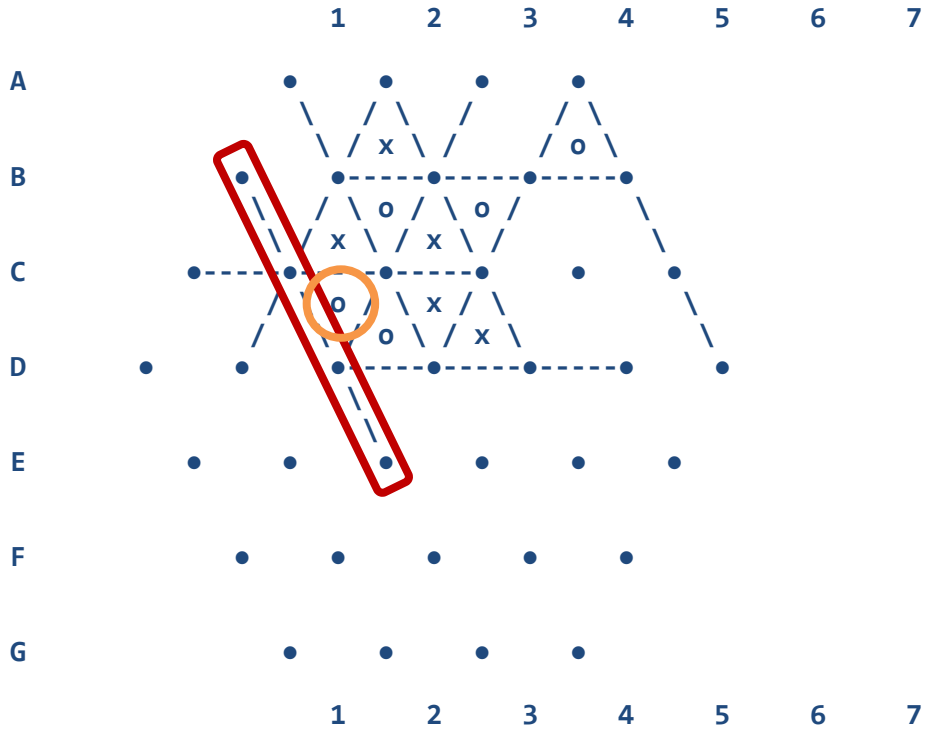
- Realizovati funkcije koje obezbeđuju odigravanje partije između dva igrača (**dva čoveka, ne računara i čoveka**)
  - Unos početnih parametara igre
  - Ponavljanje unosa novog poteza sve dok se ne unese ispravan potez
  - Odigravanje novog ispravnog poteza sa promenom trenutnog stanja igre
  - Prikaz novonastalog stanja igre nakon odigravanja poteza
  - Proveru kraja i određivanje pobednika u igri nakon odigravanja svakog poteza, odnosno promene stanja igre
- Realizovati funkcije koje implementiraju operator promene stanja problema (igre)
  - Realizovati funkcije koje na osnovu zadatog poteza i zadatog stanja igre formiraju novo stanje igre
  - Realizovati funkcije koje na osnovu zadatog igrača na potezu i zadatog stanje igre (table) formiraju sve moguće poteze
  - Realizovati funkcije koje na osnovu svih mogućih poteza formiranju sva moguća stanja igre, korišćenjem funkcija iz prethodne dve stavke

Zadatak I – Interfejs (promena stanja)



Stanje pre poteza O igrača

Zadatak I – Interfejs (promena stanja)



Novo stanje nakon poteza O igrača ('B', 1, 'DD')

Formiran 1 trouglić čije su stranice:

- C2 i D3
- C2 i C3
- C3 i D3

Zadatak III – Min-max algoritam i heuristika

- Implementirati Min-Max algoritam sa alfa-beta odsecanjem za zadati problem (igru):
  - Vraća potez koji treba odigrati ili stanje u koje treba preći
  - Na osnovu zadatog stanja problema
  - Na osnovu dubine pretraživanja
  - Na osnovu procene stanja (heuristike) koja se određuje kada se dostigne zadata dubina traženja
- Realizovati funkcije koje obezbeđuju odigravanje partije između čoveka i računara

Zadatak III – Min-max algoritam i heuristika

- Implementirati funkciju koja vrši procenu stanja na osnovu pravila zaključivanja
- Funkcija za procenu stanja kao parametre treba da ima igrača za kojeg računa valjanost stanja, kao i samo stanje za koje se računa procena.
- Procena stanja se mora vršiti isključivo korišćenjem mehanizma zaključivanja nad prethodno definisanim skupom pravila. Zadatak je formulisati skup pravila i iskoristiti ih na adekvatan način za izračunavanje heuristike.
- Za izvođenje potrebnih zaključaka (izvršavanje upita nad skupom činjenica kojima se opisuje stanje) koristiti mašinu za zaključivanje.
- Implementirati funkciju koja prevodi stanje u listu činjenica …