



Katedra za računarstvo
Elektronski fakultet, Univerzitet u Nišu

Veštačka inteligencija

Projekat – Dominacija (*Domineering*)

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: 14.11.2022.
- ▶ Rok za predaju: 10.1.2023.



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 od studenata da ozbiljno shvatite 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
- ▶ Bodovi se računaju se kao dodatni 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.

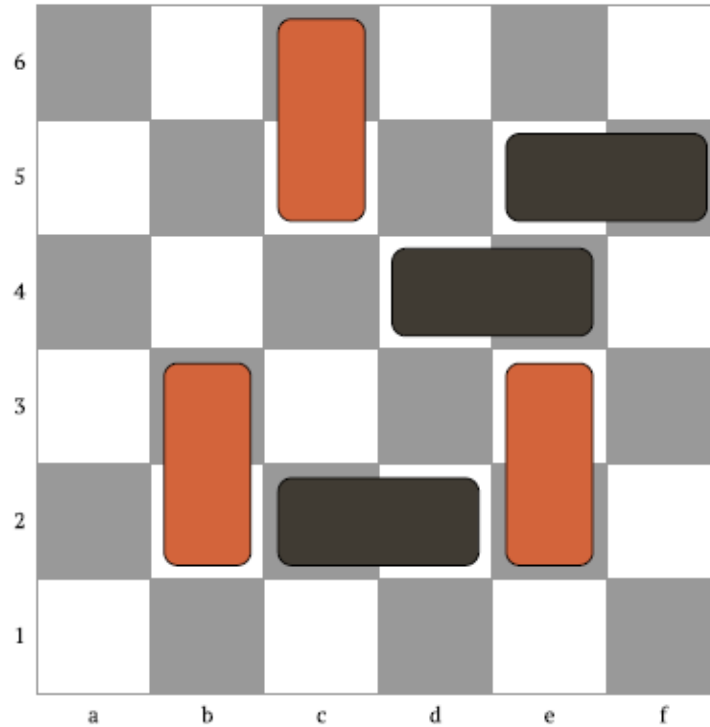


Faze izrade projekta

- ▶ Formulacija problema i implementacija interfejsa
 - ▶ Rok: 3.12.2022. godine
- ▶ Implementacija operatora promene stanja
 - ▶ Rok: 17.12.2022. godine
- ▶ Implementacija Min-Max algoritma za traženje sa alfa-beta odsecanjem i algoritama za procenu stanja (heuristike)
 - ▶ Rok: 10.1.2023. godine
- ▶ Rezultat svake faze je izveštaj koji sadrži dokument sa obrazloženjem rešenja i datoteku (datoteke) sa kodom.



Igra Dominacija (*Domineering*)



Opis problema Dominacija (*Domineering*)

- ▶ Problem je igra Dominacija (*Domineering*).
- ▶ Strateška igra postavljanja pločica na tablu.
- ▶ Tabla je proizvoljnih dimenzija $m \times n$, m vrsta i n kolona (preporučeno je 8×8).
- ▶ Pločice su veličine 2×1 .
- ▶ Dva igrača crni i beli (X i O) naizmenično odigravaju po jedan potez.
- ▶ Ne postoji ograničenje u broju pločica koje igrači poseduju.
- ▶ Jedan igrač postavlja pločice horizontalno, dok ih drugi postavlja vertikalno.
- ▶ Poražen je onaj igrač koji nema mogućnost da postavi pločicu na tablu.
- ▶ Igra čovek protiv računara i moguće izabrati da prvi igra čovek ili računar

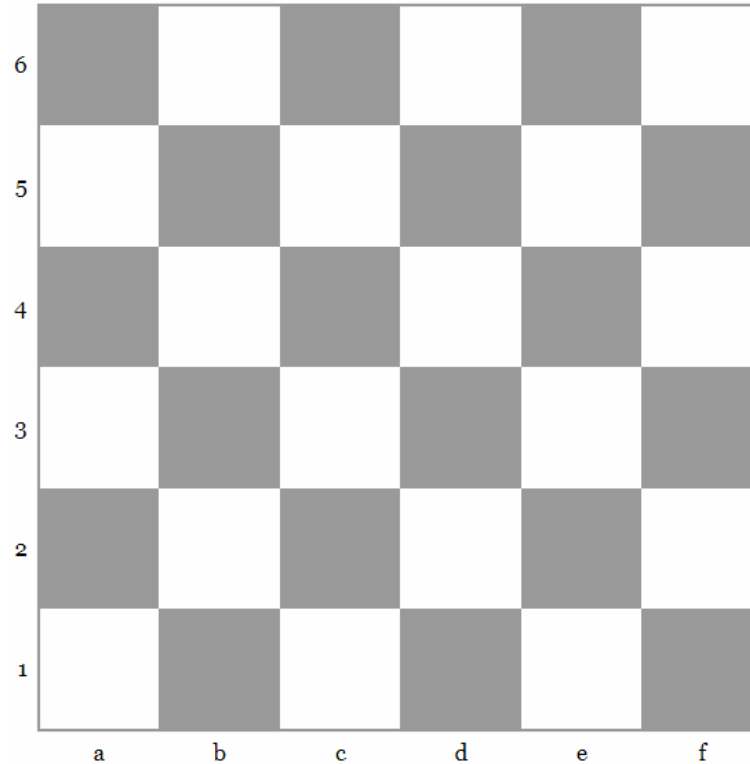


Pravila igre Dominacija (*Domineering*)

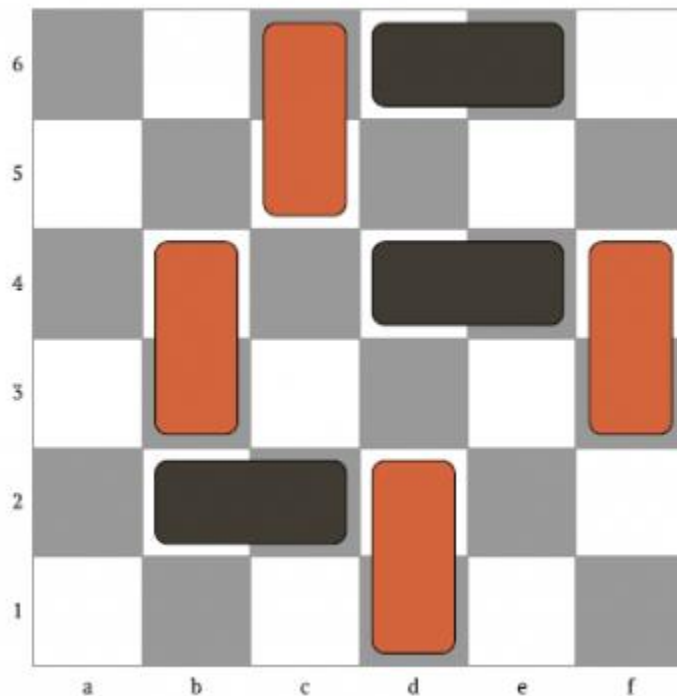
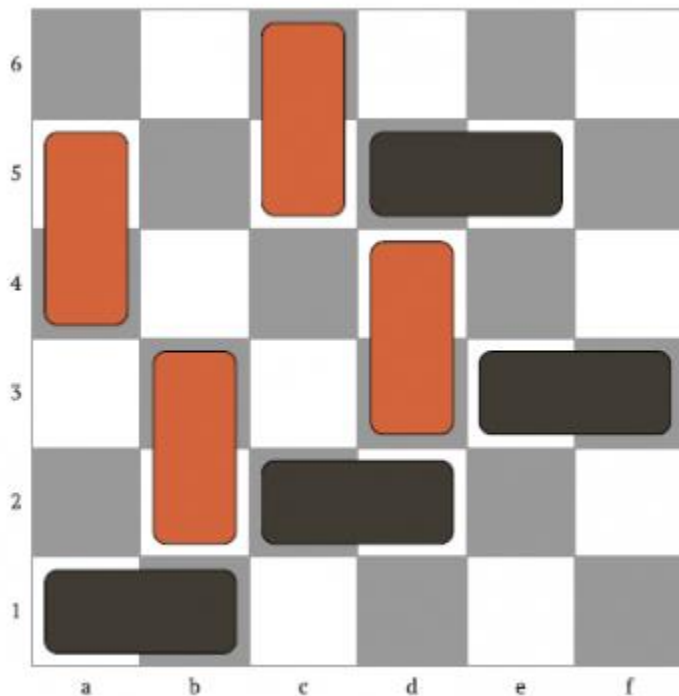
- ▶ Na početku igre tabla je prazna.
- ▶ Igrači povlače poteze naizmenično.
- ▶ Jedan igrač postavlja pločice samo horizontalno, dok ih drugi postavlja samo vertikalno.
- ▶ Prvi igrač postavlja pločice vertikalno.
- ▶ Igrač, u jednom potezu, postavlja samo jednu pločicu na tablu u odgovarajućem smeru (horizontalno i vertikalno)
- ▶ Pločice se ne smeju preklapati, niti viriti van table.
- ▶ Pločice se ne mogu postaviti tako da počinju na pola polja i završavati se na pola polja.
- ▶ Poražen je onaj igrač koji nema mogućnost da postavi pločicu na tablu.



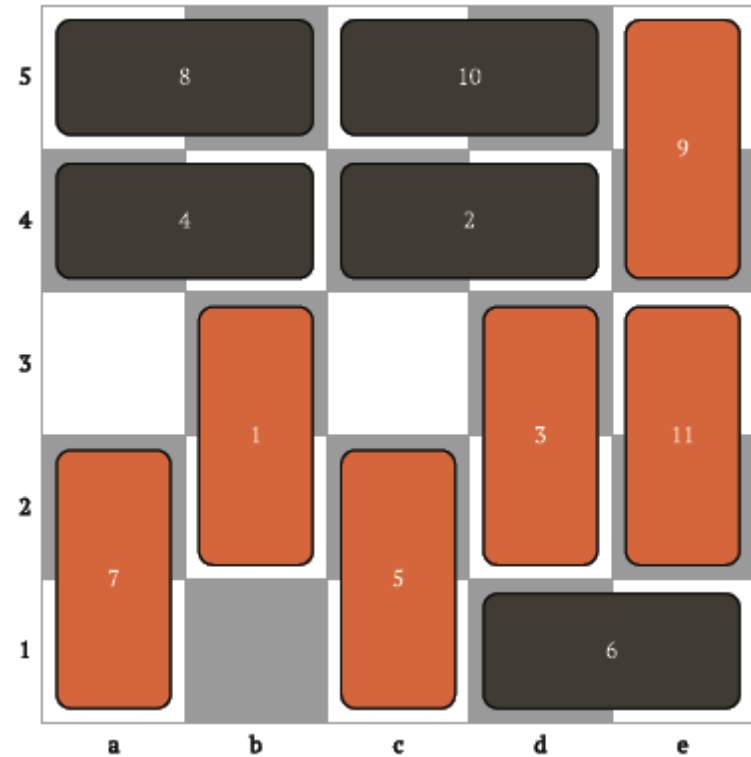
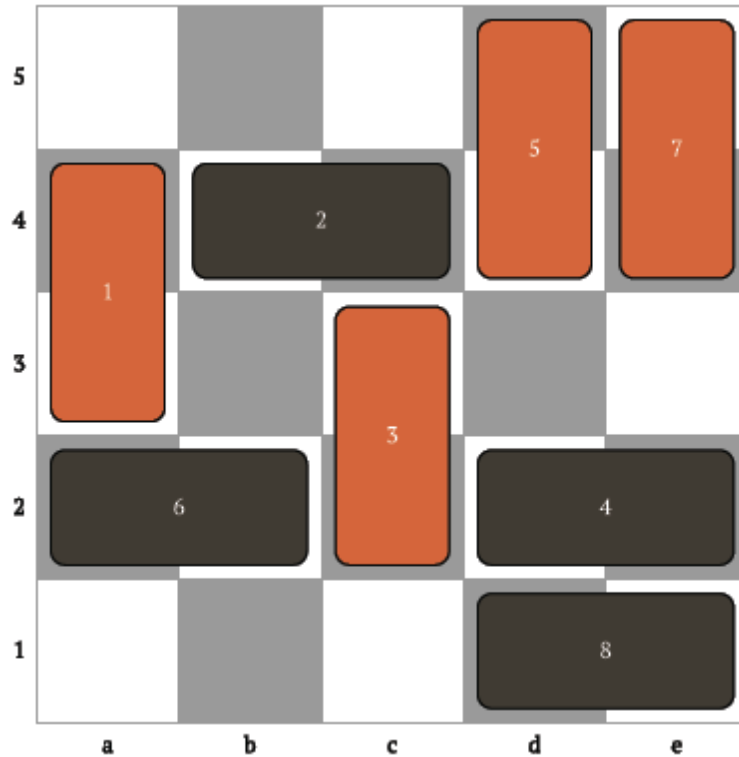
Dominacija (*Domineering*) – Početak igre



Dominacija (*Domineering*) – Primer stanja igre



Dominacija (*Domineering*) – Primer kraja igre



Dominacija (*Domineering*) – Korisni linkovi

- ▶ Osnovna verzija igre:
 - ▶ [Stop-Gate | Board Game | BoardGameGeek](#)
- ▶ Pravila igre i preporuke za pobedu:
 - ▶ [Domineering – Wikipedia](#)
- ▶ Web verzija igre:
 - ▶ [Domineering Game \(jasondavies.com\)](#)
 - ▶ [Domineering \(uni-passau.de\)](#)



Zadatak I – Formulacija problema i interfejs

- ▶ Definirati način za predstavljanje stanja problema (igre)
 - ▶ Tabla, pozicije pločica na tabli
- ▶ Napisati funkciju za postavljanje početnog stanja
 - ▶ Definiše se na osnovu zadate veličine table
- ▶ Napisati funkcije za proveru kraja igre
 - ▶ Igrač ne može postaviti ni jednu pločicu na tablu
- ▶ Napisati funkcije koje proveravaju ispravnost unetog poteza
- ▶ **NIJE POTREBNO realizovati funkcije koje odigravaju potez (faza II)**
- ▶ **NIJE POTREBNO realizovati funkcije koje obezbeđuju odigravanje partije (faza II)**

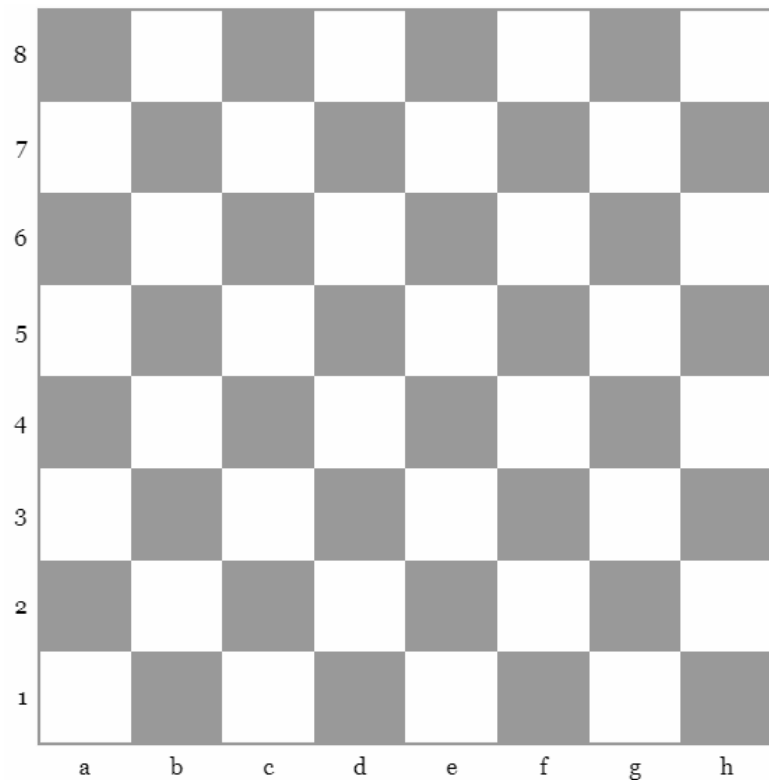


Zadatak I – Formulacija problema i interfejs

- ▶ Omogućiti izbor ko će igrati prvi (čovjek ili računar)
- ▶ Prvi igra uvek igrač X, a drugi igrač O (ili omogućiti izbor koji igrač igra prvi)
- ▶ Implementirati funkcije koje obezbeđuju unos početnih parametara igre
 - ▶ Unos dimenzija table (m i n)
- ▶ Implementirati funkcije koje obezbeđuju pravljenje inicijalnog stanja problema (igre)
 - ▶ Pravljenje prazne table na osnovu zadatih dimenzija (m i n)
- ▶ Implementirati funkcije koje obezbeđuju prikaz proizvoljnog stanja problema (igre)
 - ▶ Prikaz zadate table
- ▶ Realizovati funkcije za unos poteza
 - ▶ Potez je oblika igrač (X ili O) i pozicija donjeg levog polja gde će se naći pločica [4,C]
 - ▶ Ako je igrač X (vertikalno), pločica bi trebalo da se postavi na polja [4,C] i [5,C]
 - ▶ Ako je igrač O (horizontalno), pločica bi trebalo da se postavi na polja [4,C] i [4,D]
- ▶ Realizovati funkcije koje proveravaju da li je potez valjan
 - ▶ Proveriti da pločica nije van table
 - ▶ Proveriti da pločica ne viri van table
 - ▶ Proveriti da se pločica ne poklapa sa nekom već postavljenom pločicom

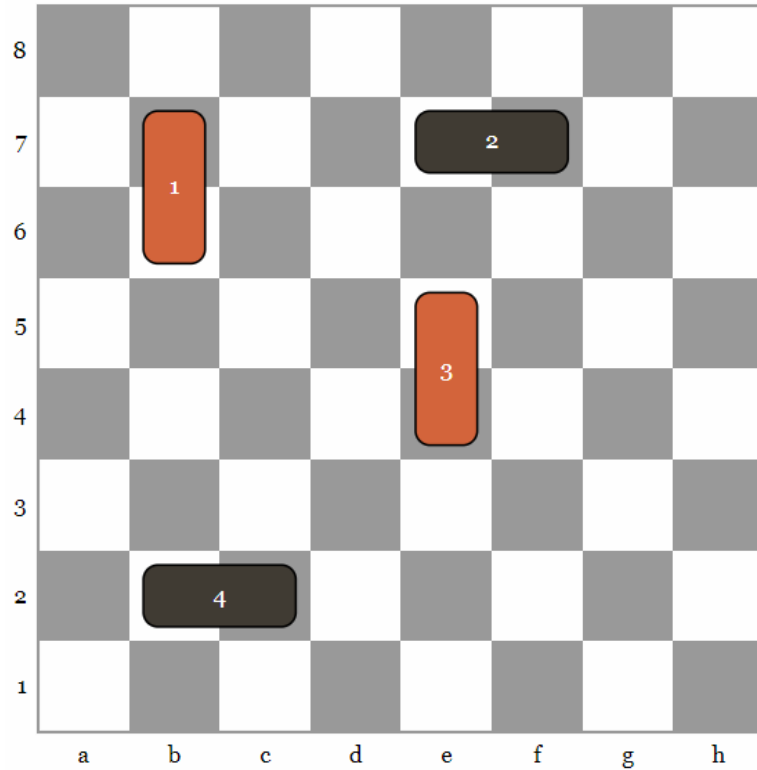


Zadatak I – Interfejs (početno stanje)



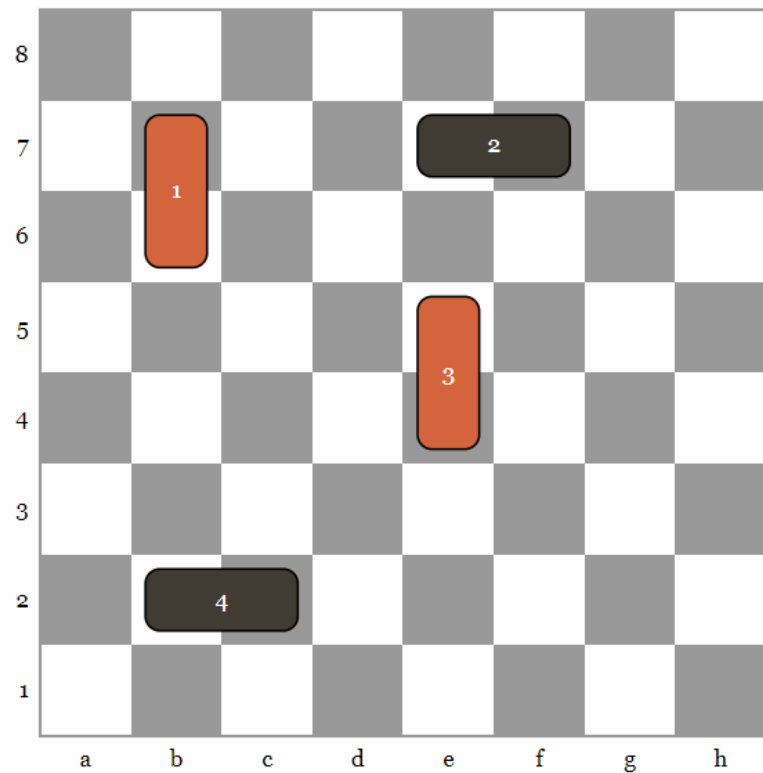
	A	B	C	D	E	F	G	H
	=	=	=	=	=	=	=	=
8								8
	-	-	-	-	-	-	-	-
7								7
	-	-	-	-	-	-	-	-
6								6
	-	-	-	-	-	-	-	-
5								5
	-	-	-	-	-	-	-	-
4								4
	-	-	-	-	-	-	-	-
3								3
	-	-	-	-	-	-	-	-
2								2
	-	-	-	-	-	-	-	-
1								1
	=	=	=	=	=	=	=	=
	A	B	C	D	E	F	G	H

Zadatak I – Interfejs (prikaz stanja)



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

Zadatak I – Interfejs (unos i provera poteza)



	A	B	C	D	E	F	G	H	
8									8
7		X		O O					7
6		X							6
5					X				5
4					X				4
3							X		3
2		O O				X			2
1									1
	=	=	=	=	=	=	=	=	
	A	B	C	D	E	F	G	H	

Loši potezi X

1. [9 F]
2. [3 K]
3. [8 B]
4. [8 G]
3. [5 E]
4. [6 F]

Dobri potezi X

1. [1 A]
2. [2 E]
3. [5 H]
4. [7 D]

Zadatak II – Operator promene stanja

- ▶ 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)

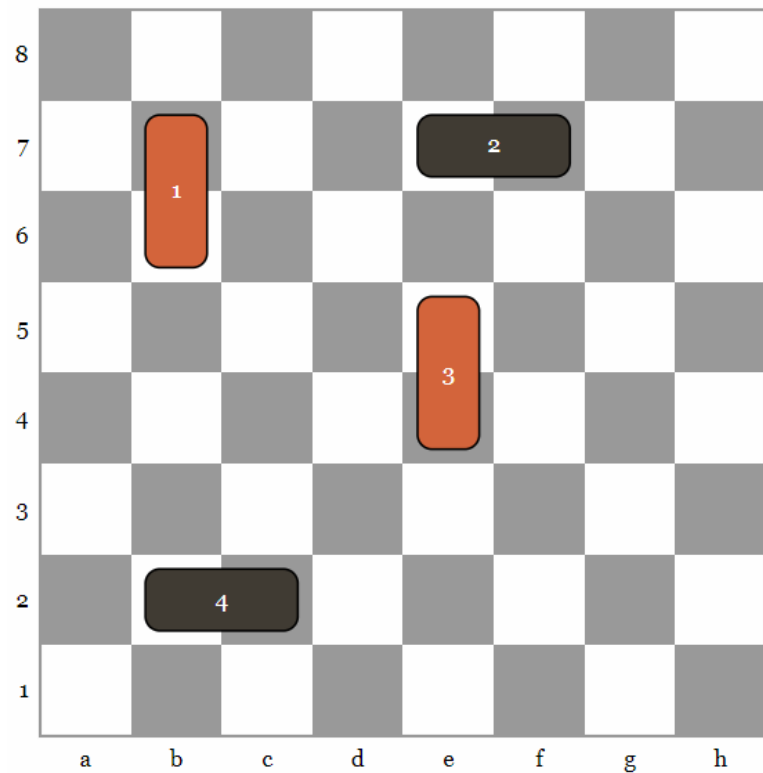


Zadatak II – Operator promene stanja

- ▶ Realizovati funkcije koje na osnovu konkretnog poteza menjaju stanje problema (igre)
 - ▶ Realizovati funkcije koje na osnovu konkretnog poteza menjaju trenutno stanje igre (table)
- ▶ Realizovati funkcije koje obezbeđuju odigravanje partije između dva igrača (**dva čoveka, ne računara i čoveka**)
 - ▶ Unos početnih parametara igre
 - ▶ Unos poteza i proveru da li je potez ispravan
 - ▶ Unos novog poteza sve dok on nije ispravan
 - ▶ Odigravanje novog poteza ako je ispravan i promenu trenutnog stanja igre (table)
 - ▶ Prikaz novonastalog stanja igre (table) nakon odigravanja poteza
 - ▶ Proveru kraja i određivanje pobednika u igri nakon odigravanja svakog poteza, odnosno promene stanja igre (table)
- ▶ Napisati funkcije za operator promene stanja problema (igre)
 - ▶ Realizovati funkcije koje na osnovu zadatog poteza i zadatog stanje igre (table) formiraju novo stanje igre (table)
 - ▶ Realizovati funkcije koje na osnovu zadatog igrača na potezu i zadatog stanje igre (table) formiraju sva moguća stanje igre (sve moguće table), korišćenjem funkcija iz prethodne stavke



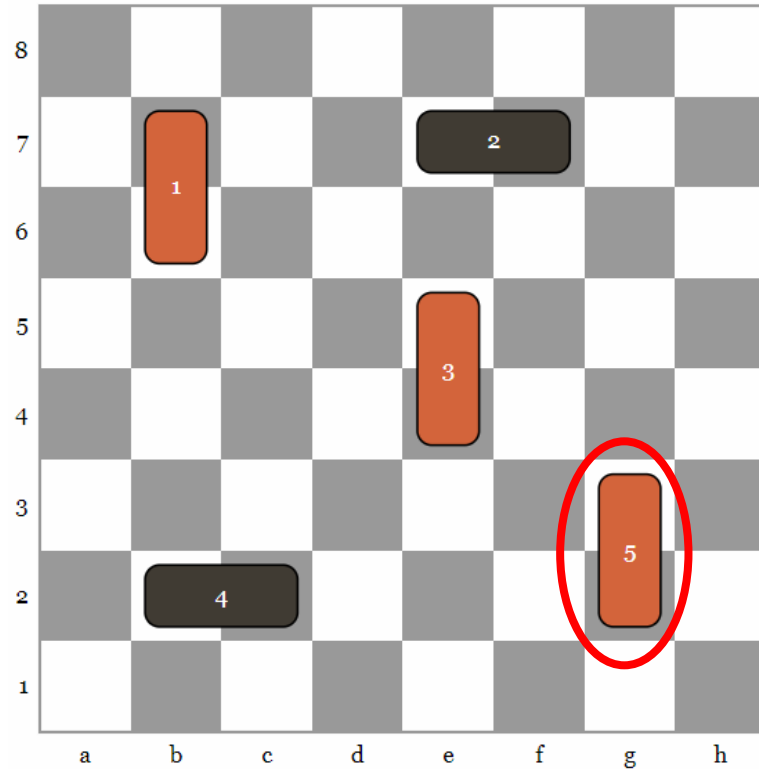
Zadatak II – Promena stanja



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

Potez X: [2 G]

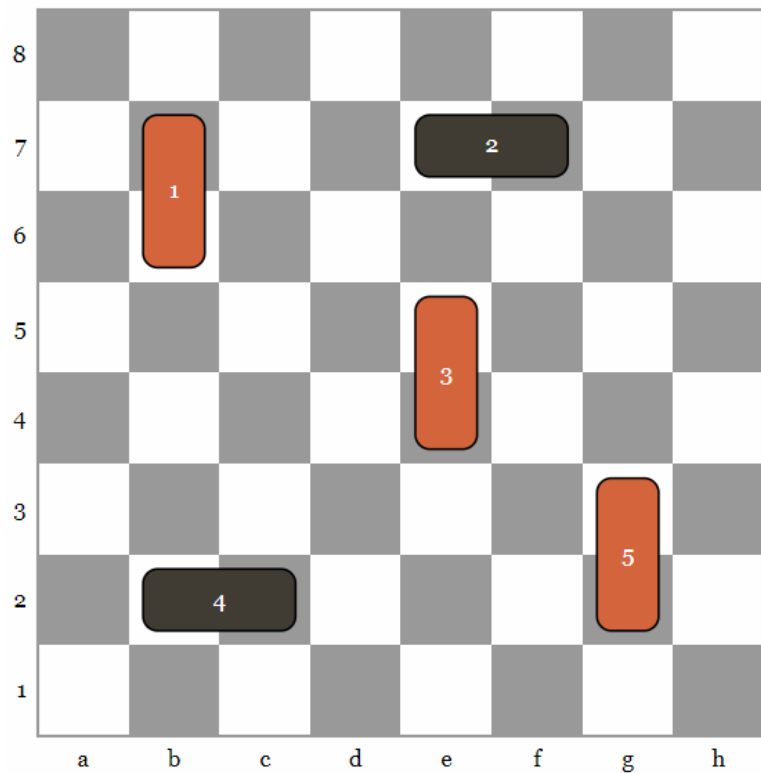
Zadatak II – Promena stanja



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

Potez X: [2 G]

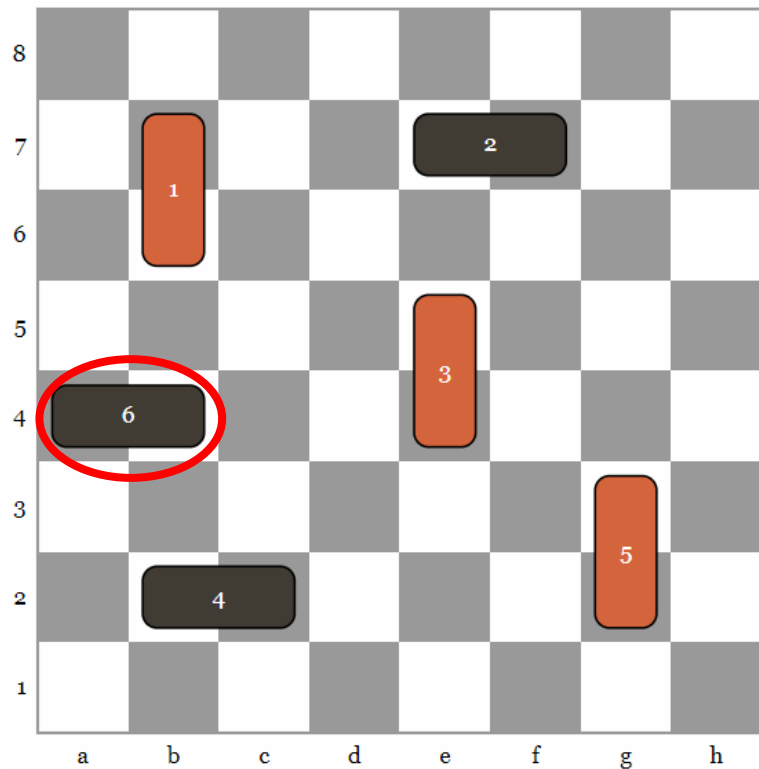
Zadatak II – Promena stanja



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

Potez O: [4 A]

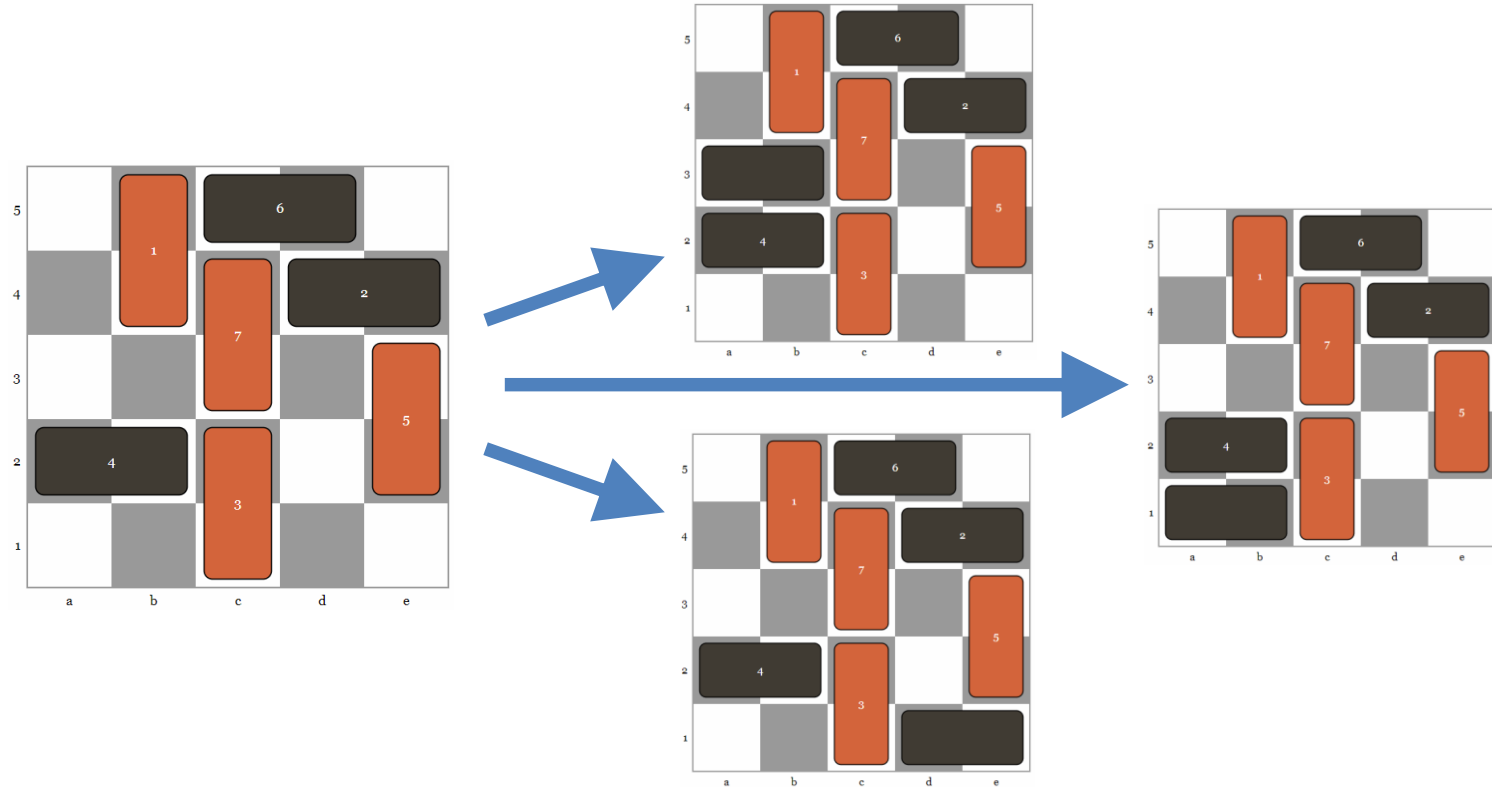
Zadatak II – Promena stanja



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

Potez O: [4 A]

Zadatak II – Operator promene stanja



Zadatak III – Min-max algoritam i heuristika

- ▶ Implementirati Min-Max algoritam sa alfa-beta odsecanjem za zadati problem (igru):
 - ▶ Na osnovu zadatog stanja problema
 - ▶ Na osnovu dubine pretraživanja
 - ▶ Na osnovu procene stanja (heuristike) koja se određuje kada se dostigne zadata dubina traženja
 - ▶ Vraća potez koji treba odigrati ili stanje u koje treba preći
- ▶ 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 ...

