



Connect 4 Engine

Igor Majić – Fakultet tehničkih nauka

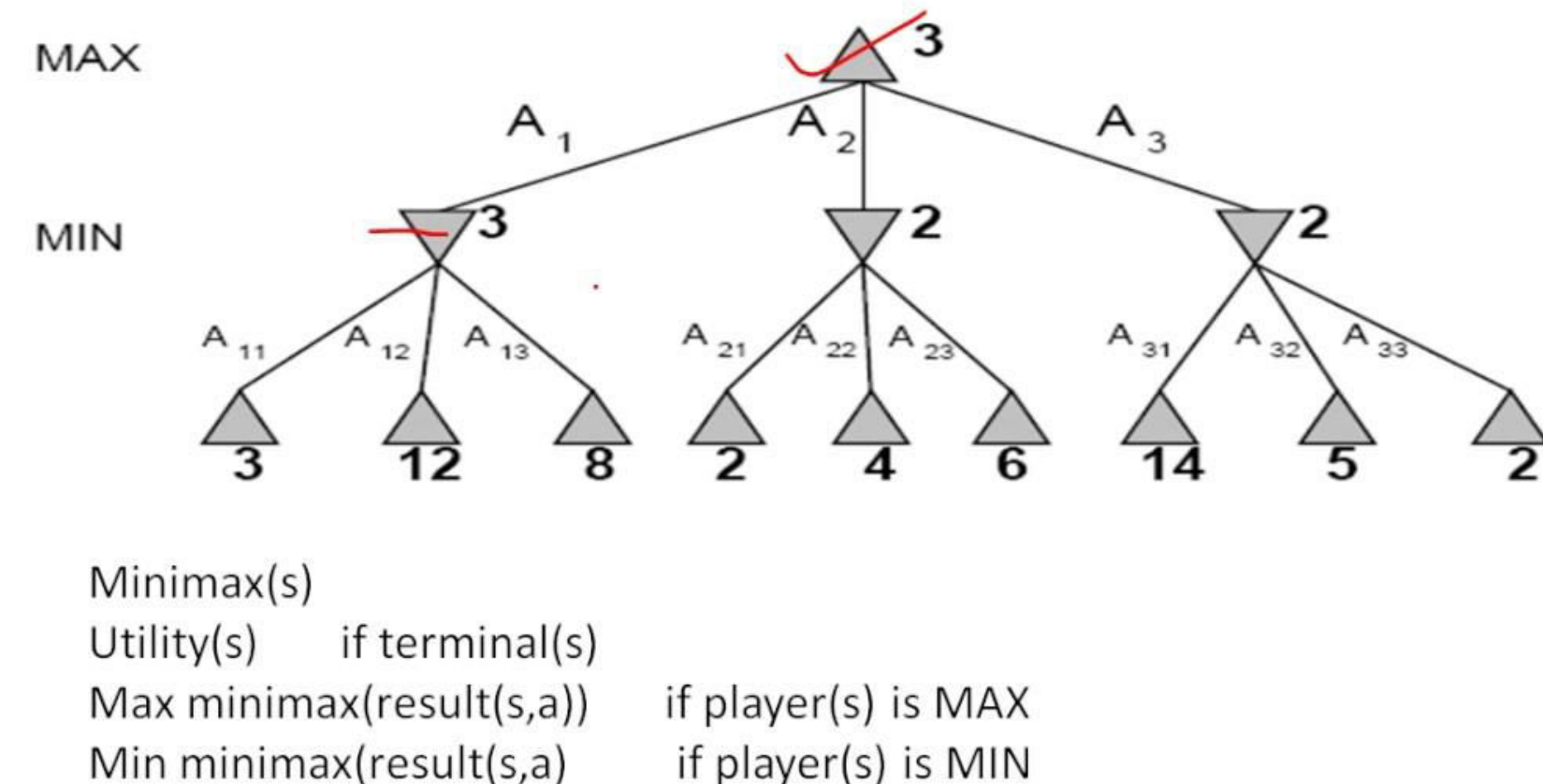
Opis problema

Moj cilj je bio da kreiram AI za igru Connect 4 koristeći Minimax algoritam sa alpha-beta pruningom i drugim optimizacijama. Connect 4 je igra u kojoj igrači po potezima bacaju obojene žetone kroz otvore na vertikalno postavljenoj tabli. Žetoni propadaju kroz redove odabrane kolone dok se ne pozicioniraju neposredno iznad poslednje ubačenog žetona u toj koloni. Ako je kolona prazna, žeton propada do poslednjeg reda. Tipična tabla je dimenzije 6x7 i sadrži otvore koji pokazuju koje je polje popunjeno. U igri pobeđuje onaj igrač koji poređa četiri uzastopna žetona na neka od tri načina: vertikalno, horizontalno ili dijagonalno (bilo koji smer).

Algoritam

Pošto je ovo igra nulte sume, bazirana na potezima, minimax algoritam je logično i prirodno rešenje. Minimax algoritam je tzv. *backtracking* algoritam koji se koristi da pronađe optimalno rešenje za igrača, pod pretpostavkom da i protivnik igra optimalno. Algoritam pravi stablo pretrage (*game tree*) u kome se smenjuju stanja u kojima su igrači odigrali poteze. Svako stanje ima svoj skor, koji utiče na to hoće li odgovarajući igrač odabrati potez koji do njega vodi. Za većinu realnih igara ovo stablo je preduboko da bi se čitavo držalo u memoriji. U minimaxu postoje dva igrača, maksimizujući i minimizujući. Maksimizujući igrač pokušava da maksimizuje svoj skor, dok minimizujući pokušava da poboljša svoje stanje tako što će ga minimizovati. Glavna optimizacija ovog algoritma predstavlja alpha-beta pruning koji orezuje (ne istražuje) one delove stabla za koje je siguran da mu neće doneti optimalan potez. Ovo orezivanje nema nikakav efekat na konačne poteze koje agent bira.

Minimax Strategy



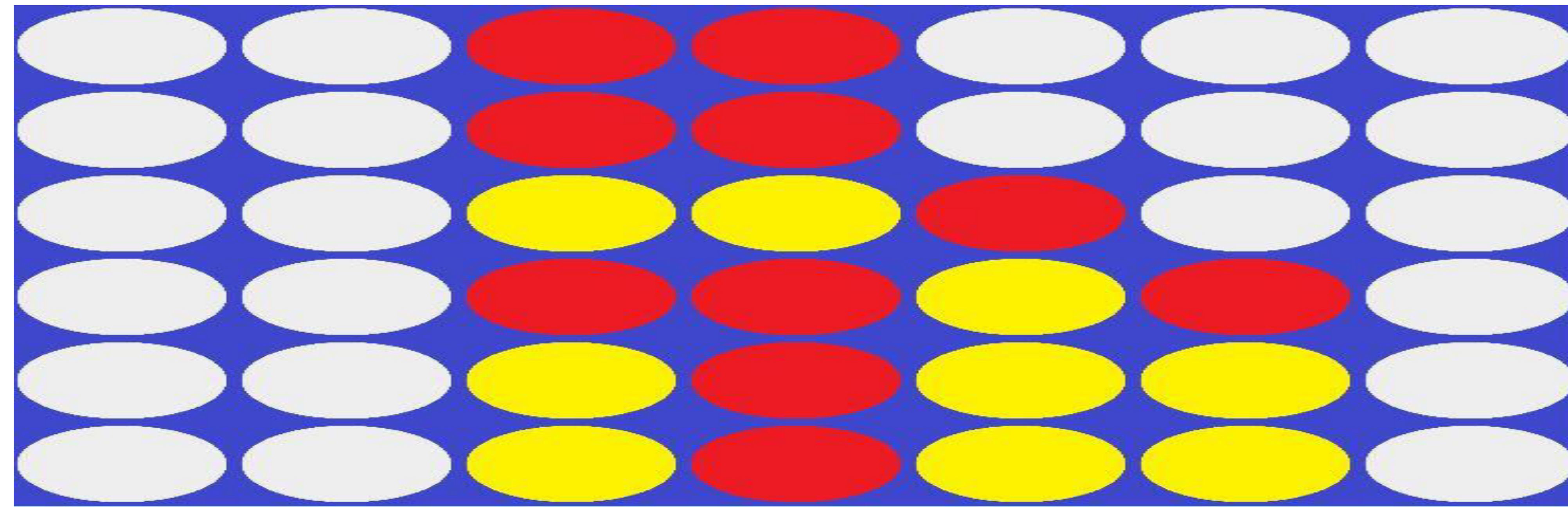
Implementacija

Implementiran je engine koji je u stanju da podrži igru između čoveka i računara, računara i računara kao i između čoveka i čoveka. Implementirani su sledeći agenti:

- Dummy agent koji nasumično ubacuje žetone.
- Minimax agent predstavlja implementaciju klasičnog minimax algoritma.
- Alpha-beta agent je po implementaciji sličan minimax agentu, ali sa alpha beta odsecanjima.
- Player agent omogućava čoveku da igra igru.

Moguće je podešavati parametre poput dubine i evalucione funkcije. Evalucione funkcije procenjuju ona stanja koja su zbog ograničene dubine postala listovi. Implementirane su dve evalucione funkcije:

- NullHeuristic, trivijalna funkcija koja vraća 0 za svako stanje.
- MyHeuristic, koja svakom stanju daje odgovarajući broj bodova, ukoliko ima žeton na nekoj poziciji.



Rezultati

Zbog velikog zauzeća memorije implementirane su bitboard reprezentacije stanja i pametniji redosled razmatranja poteza.

Nakon svih optimizacija po pitanju broja čvorova dobijeni su sledeći rezultati (otprilike broj čvorova po potezu):

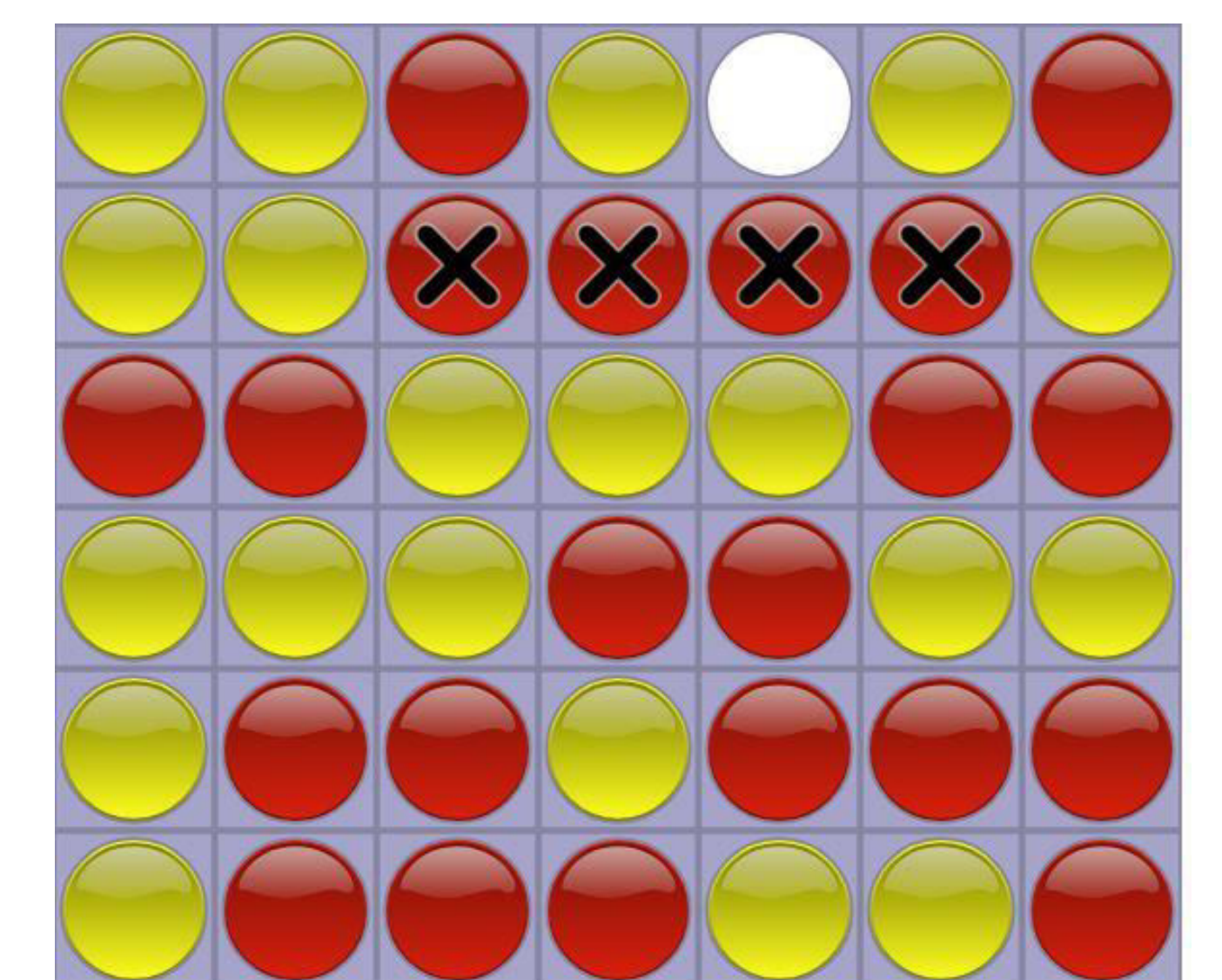
| Dubina | Minimax | Alpha-beta |
|--------|---------|------------|
| 6 | 137257 | 1249 |
| 8 | 6634027 | 8713 |
| 10 | ??? | 58550 |

Diskusija

- Koje parametre bi evalucione funkcije trebalo da uzmu u obzir?
- Da li bi iterativno produbljivanje moglo da se koristi i kako bi taj AI funkcionisao?
- Kako bi se AI skalirao na table većih dimenzija?

Zaključak

Primenom ovih optimizacija dobijen je prilično efikasan AI. U budućnosti bi mogao dobiti epitet „savršenog“ igrača ukoliko bi se upotrebile optimizacije poput transpozicionih tabli i dinamičkog razmatranja poteza koje bi moglo maksimizovati alpha-beta odsecanje.



Savršena partija