

# Tetris AI

Nenad Mišić

Fakultet Tehničkih Nauka, Novi Sad

## Uvod

Implementacija agenta koji igra igru tetris i pokušava da maksimizuje svoj skor. Ideja je da se naprave dva agenta čije se ponašanje ocenjuje i upoređuje. Jedan igra prateći Greedy search algoritam, dok drugi igra po algoritmu aproksimiranog Q-učenja.

## Cilj

Skor koji agent dobija biće računat po standardnim pravilima tetrisa, o čemu se detaljnije može videti u tabeli ispod ili putem [linka](#).

Ideja je da što se više redova spoji u jednom potezu, više se poena dobija.

Cilj projekta je agent koji igra tako da maksimizuje svoj skor.

Broj spojenih redova	Dobijeni skor
1 (single)	40
2 (double)	100
3 (triple)	300
4 (tetris)	1200

## Metode

Kako je već naglašeno u uvodu, koriste se dve metode pri implementaciji agenta, a to su Greedy search i algoritam aproksimiranog Q-učenja.

Kako i Greedy search algoritam i aproksimirano Q-učenje zahtevaju opis stanja putem feature-a, pri realizaciji projekta izdvojeni su sledeći:

1. Akumulirana visina po kolonama
2. Broj redova koji će u tom potezu biti popunjeni
3. Broj rupa na tabli
4. Razuđenost
5. Visina najviše kolone

Rupa predstavlja prazno polje iznad koga se nalazi bar jedno popunjeno polje.

Razuđenost predstavlja akumulirane razlike u visinama svake dve susedne kolone.

Ideja za ove feature-e je pronadjena u sledećem radu: [link](#)

Tehnologija koja je korišćena: Python 3 uz pygame biblioteku za iscrtavanje interfejsa Pomoć pri dizajniranju grafičkog interfejsa iskorišćena je sa sledećeg projekta: [link](#)

## Rezultati

Rezultati su dobijeni poređenjem ova dva agenta. Poređenje se izvršava sa jednakim rasporedom padanja figurica, tako da je proces merenja determinističan. Ocenjivana metrika je skor koji agenti postižu nakon 20, 50, 100 i 500 figurica. Agent koji koristi algoritam Q-učenja je pre poređenja bio treniran u 10 iteracija.

Broj figurica	Greedy search skor	Q-učenje skor
20	200	380
50	800	2020
100	1560	3760
500	7600 (457/500 figurica iskorišćeno)	16660

## Zaključak

Iz dobijenih rezultata možemo zaključiti da se agent treniran algoritmom Q-učenja, uprkos korišćenju istih feature-a, ponaša daleko optimalnije po pitanju ostvarenog skora, kako na kraćim tako i na dužim stazama. Razlog za to je to što ostavlja prostor sa strane kako bi uspeo odjednom da popuni što više redova i na taj način dobije veći skor. Međutim, ovakvim načinom igre postoji veća mogućnost upadanja u rizične situacije nego kod agenta koji radi po greedy search algoritmu i generalno igra nešto "sigurnije".

Na osnovu posmatranja težina, uočavamo da je agent treniran algoritmom Q-učenja naučen da igra tako da najviše vrednuje manjak rupa na svojoj tabli i svaki potez orijentiše prema tome. Kada je zadovoljeno to da je napravljen minimalan broj rupa, trudi se da maksimizuje ili minimizuje ostale osobine, zavisno od toga da li ih je procenio kao pozitivne ili negativne i na taj način se trudi da ostavi rupu u ćoškovima kako bi dobio što veći skor.

## Kontakt

Email: [nenad.misic997@gmail.com](mailto:nenad.misic997@gmail.com)