Analiza si grafice Tema 1

Tufa Adriana 343 C4

Scenariul folosit pentru grafice este urmatorul:

Camera 0:

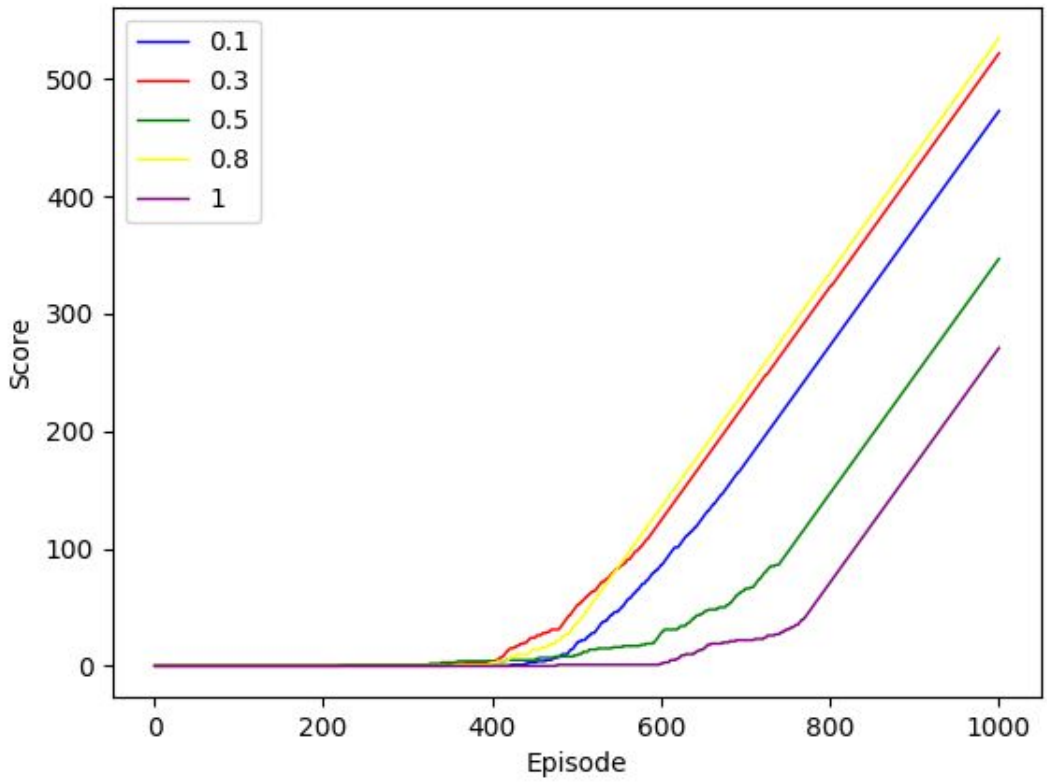
Camera1:

Raza de perceptie a gardianului (B) este 3 iar a lui Gigel este 8.
$ sunt comorile, O iesirea iar P portalurile.

Influenta learning rate:

Cu cat learning rate e mai mare, cu atat algoritmul tine cont mai putin de informatia anterioara despre perechile (stare, actiune). Daca learning rate e mai mic, atunci considera intr-o masura mai mica informatiile noi si deci va invata mai greu.

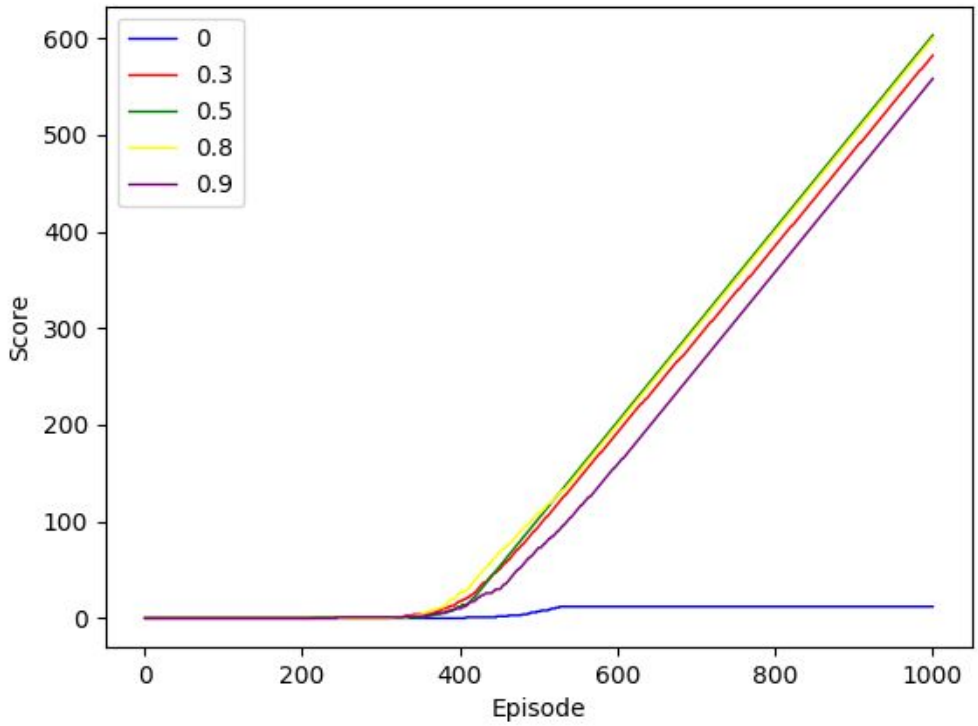
Pentru scenariul dat am contorizat numarul de jocuri castigate in cele 1000 de episoade de antrenare pentru mai multe valori ale ratei de invatare si am obtinut urmatorul rezultat:



Se observa ca in acest caz o rata de 0.8 va aduce in final cele mai multe castiguri dar la inceput varianta cu 0.5 a invatat putin mai repede. Cele mai slabe rezultate s-au obtinut cu learning rate = 1, care invata cel mai greu. Dupa un prag de aproximativ 800 de epsioade, toate variantele se imbunatatesc in aceeasi proportie.

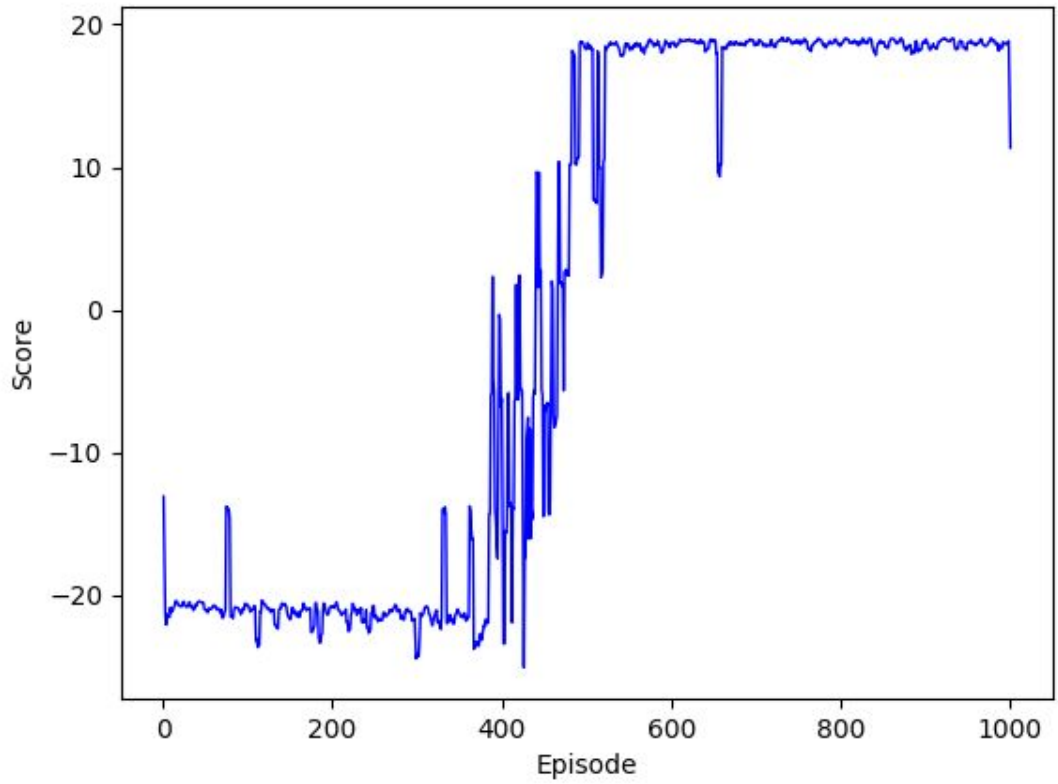
Influenta discount-ului:

Considerand aceeasi parametri ai graficului ca anterior, diferitele valori pentru numarul de jocuri castigate in functie de discount arata astfel:



Algoritmul invata cel mai repede o solutie folosind un discount de 0.5. Pentru discount 0 (deci ia in calcul doar recompensa imediata), nu invata aproape nimic.

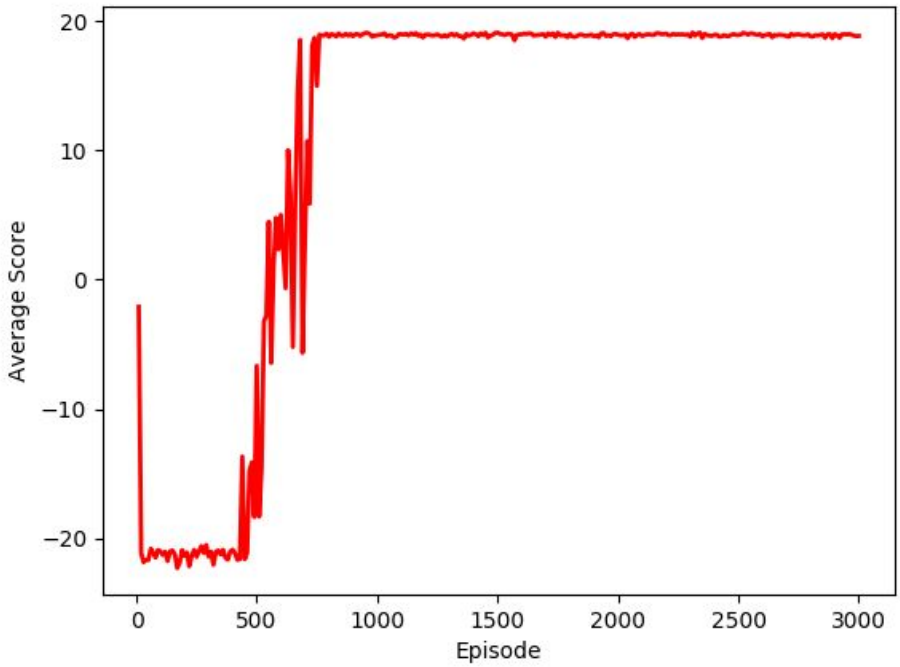
Scorul in functie de numarul episodului:



Aceste valori s-au obtinut pentru learning_rate = 0.1, discount = 0.9 si strategia MaxFirst. Dupa aprox 400 de episoade, scorul incepe sa fie pozitiv si sa sa se obtina si mai multe castiguri. Aprope de episodul 600, player-ul urmeaza aproximativ aceeasi cale pe care a gasit-o afi optima, scorul fluctuand foarte putin.

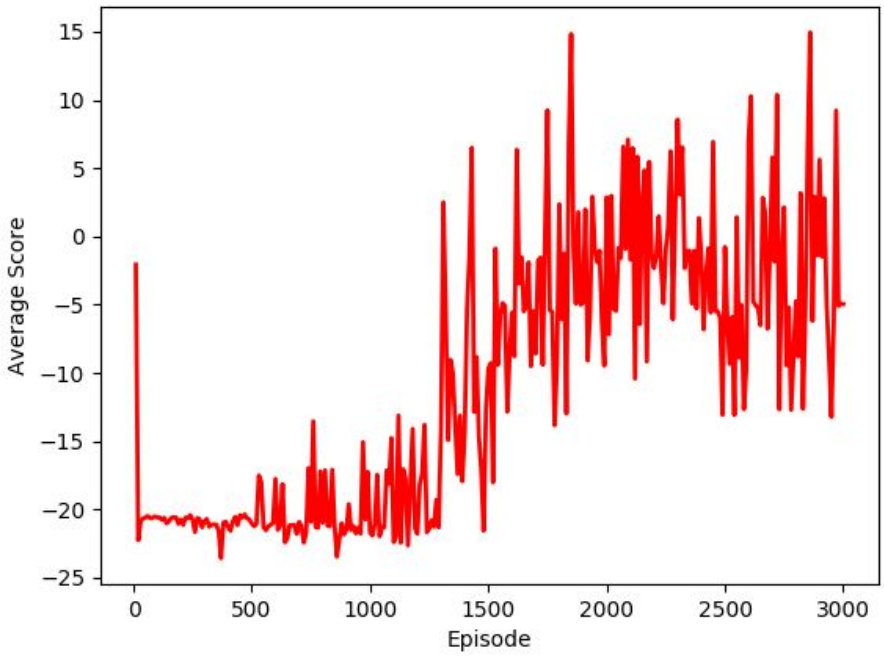
Influenta strategiei:

MaxFirst:



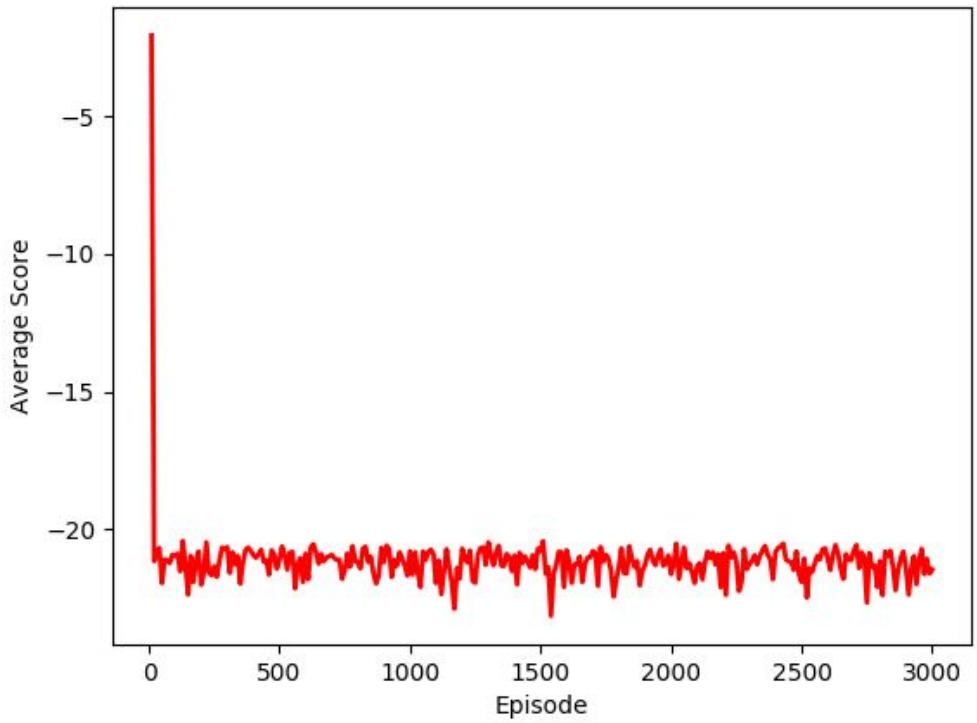
Scorul creste destul de repede si ajunge la un maxim de aproape 20 de puncte dupa 1000 de episoade.

Epsilon-greedy: (epsilon = 0.15)



Scorul este oscilant datorita factorului de explorare, dar scorurile se imbunatatesc dupa 1500 de episoade.

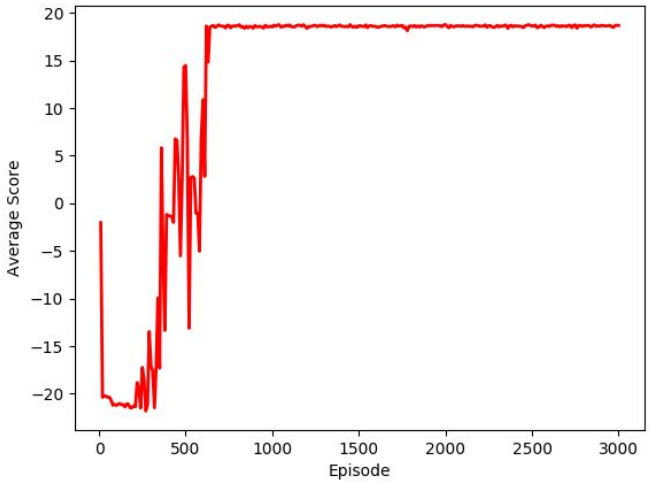
Random:



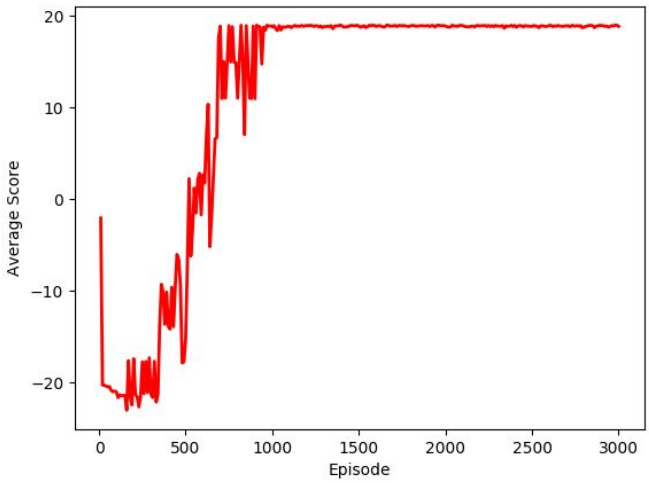
Pentru acest scenariu, strategia random nu se dovedeste a fi prea folositoare, scorul fiind constant negativ si nereusind in final sa invete o cale buna.

Discutie algoritm

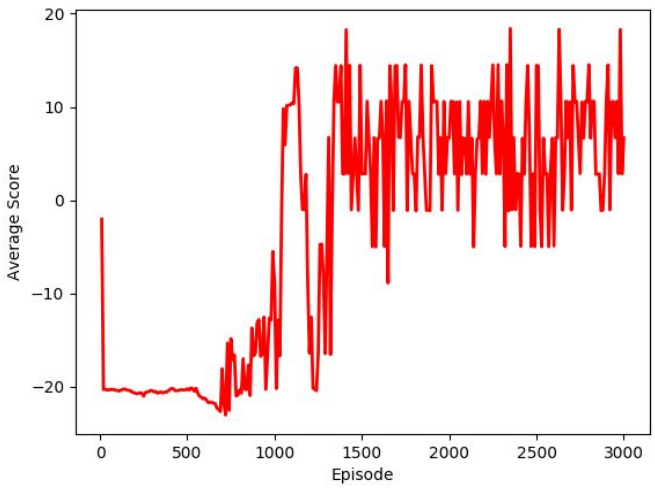
Raza de perceptie a lui Gigel nu influenteaza foarte mult numarul de stari deoarece intr-o stare este salvata si pozitia curenta (camera + coordonate), astfel incat doua camere separate similare vor produce stari diferite oricat de mica ar fi raza. Ea este folositoare atunci cand gardianul apare in cadru si Gigel invata sa se fereasca de el: cu cat il vede mai din timp, cu atat invata mai repede actiunea cea mai buna. Pentru a evidentia cat mai bine acest aspect am testat pe scenariul de mai sus dar in care raza de vizibilitate a gardianului este 8, indeajuns incat sa cuprinda toata camera.



Cand Gigel are raza 8, algoritmul invata destul de repede cu MaxFirst.

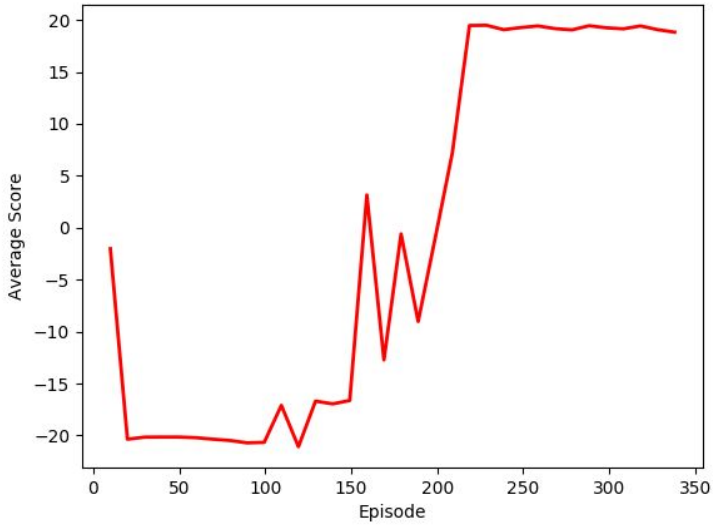


Pentru raza = 4, se observa ca se stabilizeaza la scorul maxim dupa mai multe episoade.

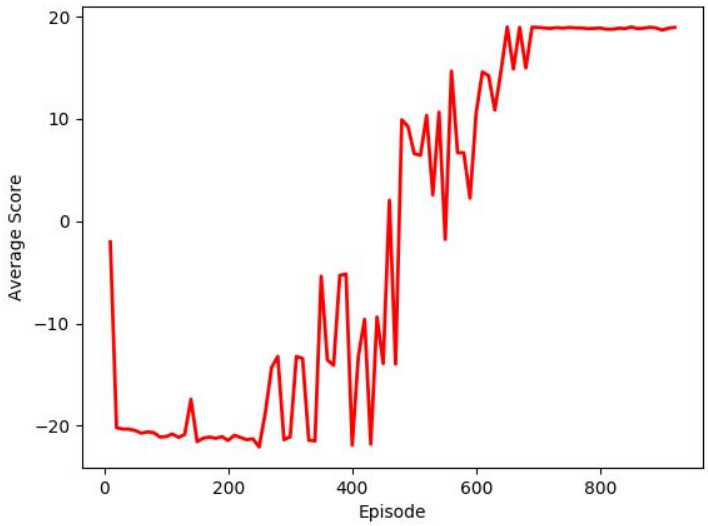


In schimb, cu o raza de 1 Gigel nu invata la fel de bine nici dupa 3000 de episoade, scorul osciland destul de mult.

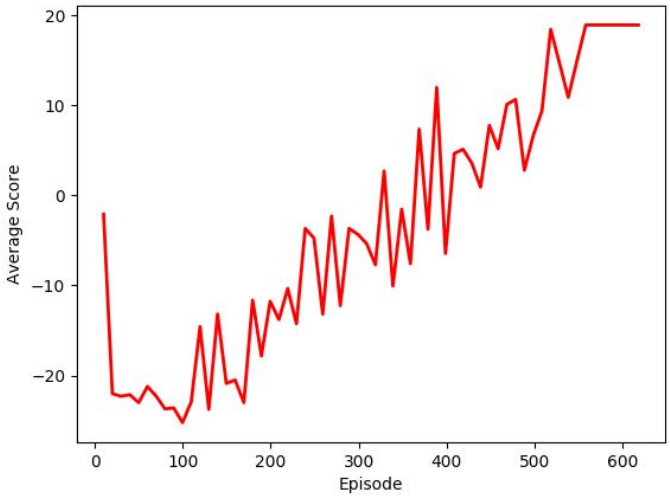
Numarul de camere influenteaza, bineinteles, complexitatea jocului: cate perechi (stari, actiune) sunt invatate si cat timp ii ia algoritmului se invete valori relevante. Pentru a arata diferenta, am rulat algoritmul pe mai multe scenarii si l-am lasat sa ruleze pana cand numarul de castiguri este 50% din numarul total de jocuri. Strategia este max first iar razele de actiune a gardienilor si a lui Gigel sunt de 8.



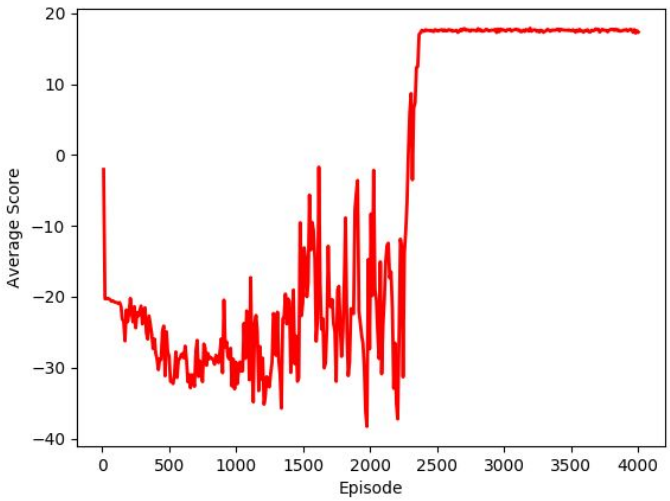
Aici scenariul a avut o camera de dimensiune 10, cu un gardian. In mai putin de 250 de episoade, Gigel invata drumul bun.



La 2 camere de dimensiune 8, numarul de episoade de antrenare creste mai mult de 2 ori.

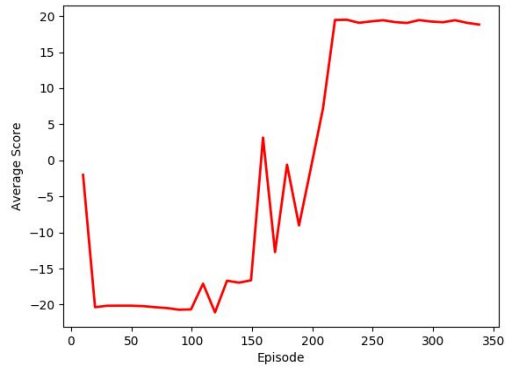


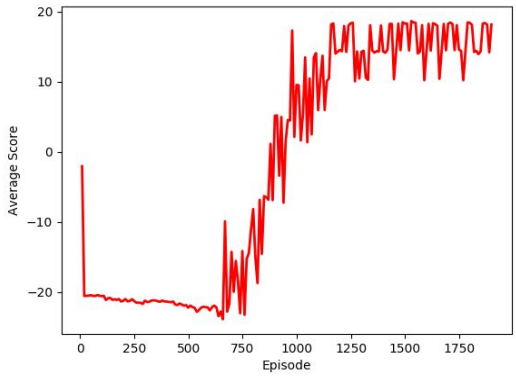
La 3 camere si o configuratie mai favorabila probabil, invata mai repede decat in cazul cu 2 camere.



Pentru un scenariu cu 6 camere si 3 gardieni este nevoie de un timp de antrenare considerabil mai mare: 4000 de episoade.

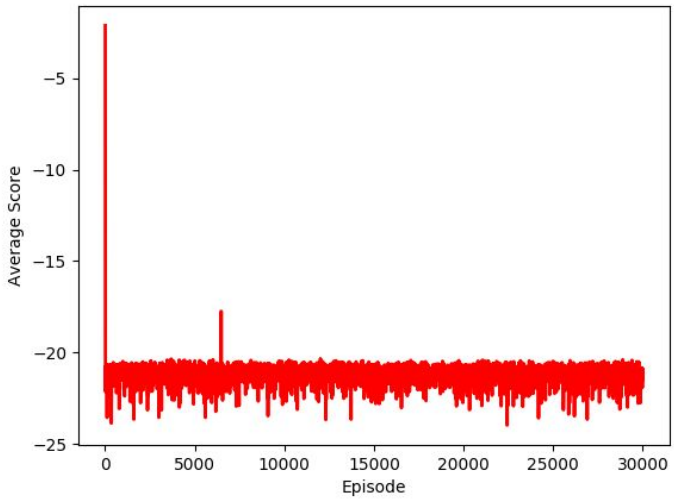
Distributia elementelor pe harta influenteaza foarte mult eficienta algoritmului chiar si cand e vorba de o singura camera. In primul grafic sunt reprezentate scorurile atunci cand gardianul nu se afla direct intre pozitia initiala si iesire iar in al doilea atunci cand se afla. Se observa si de cate episoade de antrenare este nevoie pentru fiecare caz.





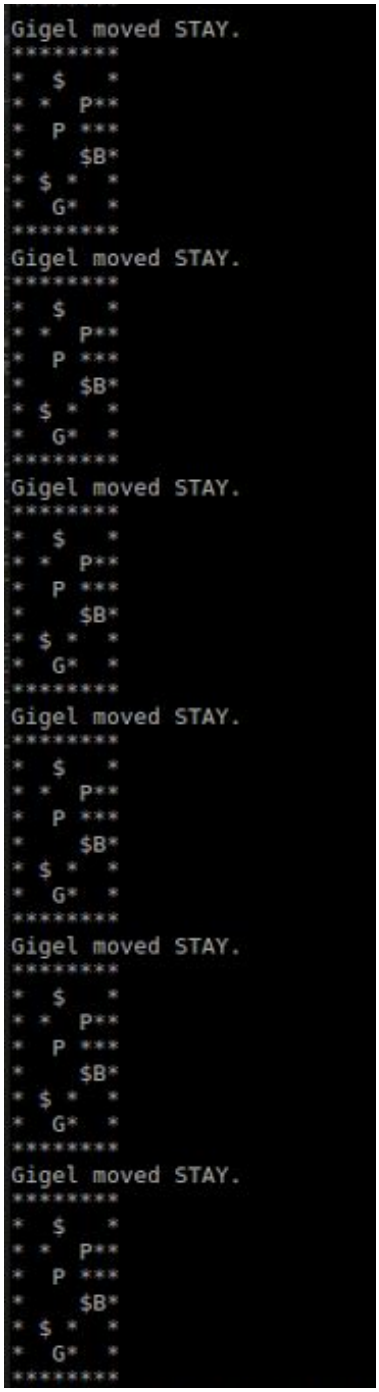
Dimensiunea camerei modifica proportional numarul de stari ce pot fi “invatate” si afecteaza mai vizibil algoritmul atunci cand se foloseste strategia random sau epsilon greedy, strategii ce se bazeaza pe explorare.

Daca este lasat destul timp sa se antreneze, algoritmul ar trebui sa gaseasca mereu o cale folosind max first si stiind ca exista o modalitate de a trece de gardieni. Random si epsilon greedy ar putea antrena si ele bine algoritmul dar intr-un numar mult mai mare de episoade, in mare parte pentru ca sunt foarte multe stari si nu sunt grupate dupa similaritate. In plus, random ar putea ajunge mult prea greu la iesire selectand mereu actiuni la intamplare si ar putea chiar sa nu invete nimic. De exemplu, pentru primul scenariu, strategia random nu da roade nici dupa 30k episoade:



O alta situatie speciala ar fi cand castelul are un numar foarte mare de camere si comori iar recompensa pentru castig nu este destul de mare astfel incat sa il determine pe Gigel sa se indrepte spre iesire. El ar putea sa stranga comorile din jur si apoi sa stea pe loc sau sa se fereasca de gardian la infinit intrucat acea stare are o utilitate mai mare decat daca s-ar

deplasa. Pentru a exemplifica am rulat algoritmul pe scenariul cu 6 camere, pentru 3000 de episoade si o recompensa de castig mai mica iar in jocul final a ramas blocat pe o pozitie:



Precizari:

- Am implementat max first astfel incat intai sa utilizeze toate actiunile posibile dintr-o stare si apoi sa aleaga mereu actiunea cea mai buna.
- Nu am lasat algoritmul sa ruleze pana la cand se atinge un anumit procent de jocuri castigate deoarece pentru random ar putea lua foarte mult.
- O stare e data de campul vizual a lui Gigel plus pozitia lui.
- Daca Gigel vede iesirea dar vrea sa o ia printr-un portal in loc sa se indrepte spre ea, este penalizat