Reinforcement Learning

Pacman

Manea Andrei

Negru Mihai-Vlad

Petreacă Vlad-Florin

Rădescu Ioana

* Tema proiectului

Acest proiect are ca temă implementarea unui joc de tip Pacman în cadrul căruia se folosește învățarea prin recompensă pentru colectarea de hrană și ocolirea inamicilor, scopul principal fiind atingerea unui scor cât mai mare.

* Principiu implementare

Am utilizat Python pentru a implementa atât interfața grafică cât și partea de antrenare.

Logica de joc este aceeași cu a jocului Pacman clasic. Există un labirint în care se găsește hrana pe care eroul trebuie să o colecteze. De asemenea, există și inamici de care acesta trebuie să se ferească. Direcția în care se deplasează inamicii este determinată în funcție de poziția curentă a pacmanului în labirint, aceștia urmărind să îl captureze.

In scopul implementării învățării prin recompensă am utilizat algoritmul de Q-learning.

Pentru început am definit stările în care se poate afla pacmanul și pentru fiecare stare am setat o recompensă.

Stări posibile:

1. **Dot** = s-a colectat un element de hrană
2. **Captured by ghost** = pacmanul a fost prins de o fantomă și a pierdut o viață
3. **None** = nu s-a efectuat nicio acțiune care să influențeze modificarea scorului
4. **Wall** = a fost lovit un perete
5. **Won** = pacmanul a colectat toată hrana disponibilă pe hartă fără a pierde toate cele trei vieți
6. **Lost** = s-au pierdut toate cele trei vieți

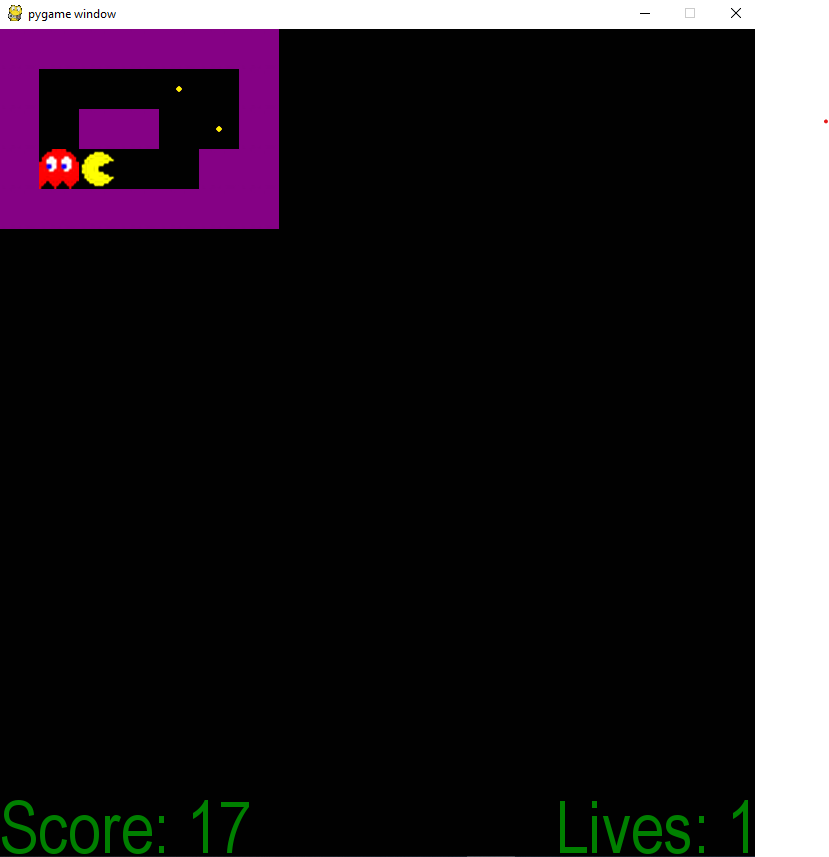
Pentru faza de antrenare a jocului am dezvoltat o metoda prin care se alege următoarea acțiune. Conform acesteia, pacmanul are o probabilitate de 20% să efectueze o acțiune aleatorie și 80% probabilitate să efectueze o acțiune optimă.

Cea mai importantă etapă a algoritmului de antrenare constă în actualizarea tabelei Q. Aceasta este realizată în cadrul fiecărui episod, în funcție de starea curentă și de acțiunea aleasă, respectând formula clasică de Q-learning.

În urma mai multor încercări am setat valoarea factorului de atenuare δ = 0.8 și valoarea ratei de învățare α = 0.2.

* Rezultate obținute

Primul set de teste le-am efectuat pe o harta destul de mica cu puține pacdot-uri si cu o singura fantoma inamic.

* In urma rulării unui singur episod se observa faptul ca pacman-ul nostru se mișca in mare partea aleatoriu sau uneori sta pe loc. Scorurile obținute in aceasta etapa sunt majoritar negative influențate de mișcările fantomei.
* In urma rulării a 10 episoade se poate observa deja o evoluție in comportamentul acestuia, acesta începând sa obțină scoruri pozitive prin adunarea de pacdot-uri dar încă nu este capabil sa se ferească de fantoma, astfel pierzând mereu.
* In urma rulării a 50 de episoade observam că pacman după aproximativ 30 de episoade a reușit sa câștige pentru prima oara, însa comportamentul sau încă este uneori imprevizibil si nu câștiga mereu, dar se obțin punctaje pozitive destul de mari. 
* In urma rulării a 100 de episoade pacman a reușit sa optimizeze destul de bine mișcările sale, câștigând in mod constant, uneori atingând scorurile maxime (30-40).
* In urma rulării a 500 de episoade pacman este optimizat la maxim, acesta găsind mereu soluția optima pentru situația in care se afla (influențate de primele mutări ale fantomei). Scorurile obținute după 500 de rulări sunt intre 37-40. 

**Harta nr -1:** Am efectuat acelasi numar de teste cu aceiasi parametrii ca pe harta trecuta cu rezultate asemanatoare, scalabilitatea fiind cam aceeasi cu rezultatele obtinute pe prima harta de test.

**Harta nr -2 :** Replica hartii originale de pacman. Rezultate asemanatoare cu celalate harti, doar ca am intampinat probleme vizand memoria, fiind o harta destul de mare q\_table-ul generat era imens, astfel load\_time-ul acesteia este unul mare, la fel si antrenarea lui.

