

Raport utilizare tool-uri AI pentru testarea unui bot de Tetris

Hodoroagă Andrei - grupa 351
Gheorghe Marius-Cătălin - grupa 351

Tema abordată în proiectul nostru este crearea unui bot care să joace Tetris folosind algoritmi de reinforcement learning. Întrucât am experimentat cu mai mulți algoritmi (și în cadrul fiecărui algoritm cu diverși parametri, timp de antrenare etc.), am avut nevoie de câteva metode pentru a evalua cât mai ușor și cât mai bine performanța fiecărui algoritm.

Astfel, am implementat un set de teste unitare în care algoritmul trebuie să plaseze o piesă într-o configurație specificată de noi, comparând apoi rezultatul cu ce credeam noi că este optim. Un alt mod de testare a performanței implementat a fost prin rularea unui algoritm pentru un număr fixat de piese (în general 1000), iar apoi calcularea anumitor metrici precum înălțimea medie a board-ului.

Atât în implementarea algoritmilor, cât și în implementarea testelor, am folosit tool-ul **GitHub Copilot** pentru a genera anumite porțiuni de cod. Noi l-am folosit în editorul Visual Studio Code, dar acesta funcționează și în alte programe. Github Copilot oferă sugestii în timp ce scrii, prin completarea rândului curent sau chiar generarea unui bloc întreg de cod [1].

Pentru a obține alte sugestii de testare am folosit **ChatGPT** [2], oferindu-i o descriere a proiectului și întrebând ce alte metode de testare ale algoritmilor putem aborda. Una dintre sugestii era să generăm board-uri aleator și să testăm algoritmi pe ele, dar nu putem face acest lucru întrucât nu avem de unde să știm care e cea mai bună mișcare pe un board aleator. O altă sugestie era să testăm botul împotriva altor boti, lucru pe care îl făceam deja manual, dar nu am automatizat această variantă întrucât algoritmi noștri nu sunt încă la un nivel suficient de înalt.

Un alt mod în care am folosit ChatGPT a fost să îi cerem să mai creeze cinci configurații de board-uri cât de "grele" se poate, pentru a face o evaluare mai bună a algoritmilor. I-am oferit și două exemple scrise de noi pentru a avea un model după care să genereze. Din păcate nu a reușit să genereze nicio configurație interesantă, două din ele fiind greșite, iar două erau cu board-uri goale. Deși l-am instruit cum să își corecteze erorile, nu a reușit să facă acest lucru.

```
[[1, 1, 1, 1, 1, 1, 1, 1, 1, 1],  
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
```

Board generat

```
[[1, 1, 1, 1, 1, 1, 1, 1, 1, 1],  
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
```

Rezultatul așteptat al testului

Așa cum se poate vedea în imaginile de mai sus, unul dintre board-urile generate conținea două rânduri pline, configurație care e imposibil să apară în Tetris, deoarece orice rând plin este imediat curățat. În imaginea din dreapta este ce se aștepta să rezulte în urma plasării piesei, dar este aceeași configurație imposibilă.

Două alte metode de testare sugerate de ChatGPT pe care le implementaserăm deja erau testarea umană (verificarea performanței de către noi) și testarea adversarială, în care doi boți joacă unul contra celuilalt.

Deși au mai fost și alte propuneri care sunt utile în alte contexte (testare de integrare, testare “edge case” etc.), am considerat că nu sunt relevante pentru testarea performanței algoritmilor noștri.

Referințe

[1] - Github, *About Github Copilot*,

<https://docs.github.com/en/copilot/about-github-copilot>, Data ultimei accesări: 05.05.2024

[2] - <https://chatgpt.com/>