

Jocului Vietii

Proiectul 6

Capisizu Cosmin Louis
IS1.1

January 9, 2022

1 Cerința

Proiectarea și implementarea unui sistem multi-agent pentru simularea jocului vietii, game of life. Se va defini și analiza cel puțin un joc de tip automat celular diferit de cel din programul demonstrativ pentru jocul vietii. Pentru jocul vietii se vor defini și experimenta și alte configurații initiale, pe lângă cele predefinite în programul demonstrativ.

2 Prezentarea jocului

Jocul Vietii, cunoscut și sub numele de Viata, este un automat celular conceput de matematicianul britanic John Horton Conway în 1970. Este un joc fără jucători, ceea ce înseamnă că evoluția sa este determinată de starea sa inițială, nefiind nevoie de alte informații.

Se interacționează cu Jocul Vietii creând o configurație inițială și observând cum evoluează. Este Turing complet și poate simula un constructor universal sau orice altă mașină Turing.

Universul Jocului Vietii este o rețea ortogonală infinită, bidimensională, de celule patrulate, fiecare dintre acestea fiind într-una dintre cele două stări posibile, vie sau moartă (sau populată și, respectiv, nepopulată). Fiecare celulă interacționează cu cei opt vecini ai săi, care sunt celulele care sunt adiacente orizontal, vertical sau diagonal.

La fiecare pas în timp, au loc următoarele tranziții

- Orice celulă vie cu mai puțin de doi vecini vii moare, ca prin subpopulare.
- Orice celulă vie cu doi sau trei vecini vii trăiește în generația următoare.
- Orice celulă vie cu mai mult de trei vecini vii moare, ca prin suprapopulare.
- Orice celulă moartă cu exact trei vecini vii devine o celulă vie, ca prin reproducere.

La fiecare pas în timp, au loc următoarele tranziții. Aceste reguli, care compară comportamentul automatului cu viața reală, pot fi condensate în următoarele

- Orice celulă vie cu doi sau trei vecini vii supraviețuiește.
- Orice celulă moartă cu trei vecini vii devine o celulă vie.

Toate celelalte celule vii mor în generația următoare. În mod similar, toate celelalte celule moarte rămân moarte.

Modelul inițial constituie sămânța sistemului. Prima generație este creată prin aplicarea regulilor de mai sus simultan la fiecare celulă din sămânța, vie sau moartă; nașterile și decesele au loc simultan, iar momentul discret în care se întâmplă acest lucru se numește uneori capusă. Fiecare generație este o funcție pură a celei precedente. Regulile continuă să fie aplicate în mod repetat pentru a crea generații viitoare.

3 Problema studiata

În cadrul acestui proiect, problema constă în implementarea și stimularea agenților din cadrul jocului vieții. Această problemă are mai multe cerințe: Simularea sistemului multi-agent, analiza comportamentului agenților, și crearea unei modalități de alterare a parametrilor simulării.

4 Metoda de rezolvare folosită

Pentru implementarea sistemului, analiza datelor dar și alterarea parametrilor simulării s-au folosit mai multe tehnologii web.

Fundația aplicației este construită folosind framework-ul "React" construit de "Facebook", dar și alte tehnologii cum ar fi "TypeScript" și "Sass".

4.1 Sistemul multi-agent

Acest sistem a fost construit folosind o clasă ce încorporează funcționalitatea dar și datele necesare simulării. Ca și structura de date ce stochează starea simulării s-a folosit matricea. Aceasta oferă un acces simplu și rapid asupra stării agenților.

4.2 Parametri simulării

Pentru a permite utilizatorului să interacționeze cu sistemul-multiagent s-au implementat următoarele funcționalități: modificarea manuală a stării unei celule folosind mouseul, modificarea dimensiunii simulării folosind interfața grafică și modificarea și resetarea regulilor folosind editorul din simulare.

4.3 Implementarea simulării

O celulă poate fi vie sau mortă, aceste stări fiind reprezentate de 1 sau 0 logic. Tabloul ce este format din mulțimea celulelor este stocat într-o matrice. Această structură este implementată la rândul ei ca fiind un vector de vectorii de stări logice.

Starea celulelor este modificată de către clasa din care fac parte, astfel celula în sine are doar funcționalitatea de a-și reține propria stare.

Regulile de evoluție sunt oferite de către un obiect ce deține o listă de reguli. Acest obiect este inițializat folosind setul de reguli de bază. Obiectul conține o listă de cerințe. Fiecare cerință are la rândul ei o listă de intervale și un set de condiții logice. Aceste condiții sunt folosite pentru a determina dacă un anumit scenariu se aplică. În cazul în care scenariul este confirmat ca fiind potrivit pentru starea actuală se folosește un alt set intern de condiții ce stabilesc starea următoare a celulei.

Evoluția celulelor este realizată de o metodă a clasei. Această funcție analizează datele ce sunt prezente în generația actuală și folosește instrucțiunile oferite de către serul de reguli pentru a genera următoare generație.

Dimensiunea simulării este predefinită dar se poate mări sau micșora folosind interfața grafică. Modificarea simulării nu duce la pierderea stării, excepție de la această regulă o face micșorarea simulării astfel încât celulele depășesc limita acesteia. Atât mărirea cât și micșorarea se face începând în colțul cel mai îndepărtat față de origine.

4.4 Implementarea interfeței grafice

Acest sistem pune accent pe implementarea modelului multi-agent, astfel interfața comanda și afișează starea simulării. Astfel arhitectura necesară implementării interfeței nu afectează sistemul simulat.

Compatibilitatea dintre sistem și interfața grafică se face folosind interfețele. Sistemul multi-agent implementează o interfață ce permite interfeței grafice să acceseze resursele necesare acesteia.

De asemenea interfața grafică are mai multe implementări ce implementează și ele la rândul lor o interfață.

Interfața grafică este implementată folosind două sisteme diferite: componente web native și implementarea grafică folosind o bibliotecă grafică specializată. Acestea două implementări sunt abstractizate și pot fi interschimbate în timp real de către utilizator.

5 Funcționarea programului pentru experimentare

6 Cazuri experimentale

7 Probleme întâlnite

Simularile multi-agent prezintă multe obstacole și probleme. Principalele dificultăți în cadrul acestui proiect sunt:

- Implementarea sistemului ce reține și modifică starea agenților. Acest sistem poate fi implementat folosind diferite structuri de date, cum ar fi: matricile sau listele. Principalul avantaj al matricii este accesul direct al oricărui agent dar și al vecinilor acestuia, un mare dezavantaj este faptul că odată cu creșterea dimensiunii simulării crește și mărimea matricii. Simularea folosind liste permite o scalare mai ușoară și o separare dintre numărul total de agenți posibil și mărimea mediului de simulare. Cel mai mare dezavantaj constă în faptul că anumite operații sunt mult mai greu de implementat și necesită mult mai mulți pași de execuție. În cazul acestui proiect matricea a fost aleasă ca și mijloc de stocare al datelor.
- Implementarea și unei interfețe grafice ce permite interacțiunea directă cu simularea. Interfața grafică și simularea necesită arhitecturi total diferite. Deci simularea sistemului este punctul de referință, interfața grafică afișează și modifică datele la cererea utilizatorului.

8 Modificările și îmbunătățirile necesare

9 Referințe

- [wiki — Conway's Game of Life](#)