

The Game Of Life

Autor: Bâgiu Laurențiu-Ionuț 334AA

Cuprins

1.Introducere.....	3
2.Suport tehnic.....	5
3.Implementare.....	6
4.Mod utilizare	9
5.Concluzii	12
6.Bibliografie.....	13

1.Introducere

Jocul vieții, “The Game Of Life”, a fost creat de către matematicianul John Horton Conway în anul 1967. Datorită analogiilor sale despre creșterea, scăderea și alternanța unei societăți de organisme vii, acest joc aparține unei clase de jocuri numite “jocuri de simulare” (simulation-games), jocuri ce se aseamănă cu viața reală.

Pentru a juca acest joc, avem nevoie de o tablă de șah destul de mare și de o resursă suficientă de jetoane plate de două culori. Ideea jocului este să începem cu o simplă configurație de jetoane(organisme), unul la o celulă, și apoi să observăm cum se schimbă în timp ce aplicăm legiile genetice pentru naștere, moarte și supraviețuire ale lui Conway.

Conway a ales aceste reguli, după o lungă perioadă de experimente, pentru a atinge trei puncte esențiale înainte de a le stabili:

- 1. Nu trebuie să existe un model inițial pentru care există o dovadă că populația poate crește la infinit.*
- 2. Ar trebui să existe un model inițial pentru care în aparență crește fără limite.*
- 3. Ar trebui să existe modele inițiale care cresc și se schimbă pentru o perioadă considerabilă de timp înainte de a se sfârși în trei căi posibil: să dispară absolut fiecare celulă, să rămână într-o configurație stabilă care nu se mai poate schimba, sau să intre într-o fază de oscilare unde celulele vor repeta un ciclu de două sau mai multe perioade.*

Pe scurt, regulile aplicate trebuie să facă populația de celule să aibă un comportament neprevăzut.

Este important de precizat faptul că toate nașterile și toate morțile se petrec simultan. Împreună, celulele constituie o singură generație, o “mișcare” în completa “istorie a vieții” a configurației inițiale.

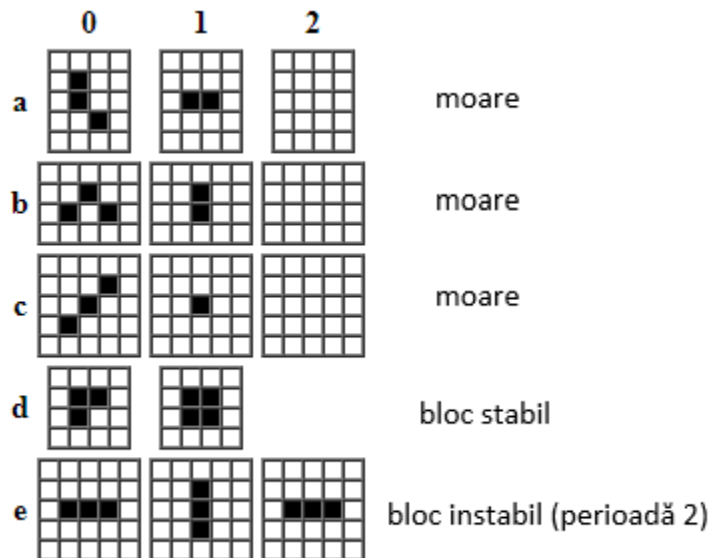
Ca și recomandări pentru a realiza mutări avem:

- 1. Incepeți cu un model format din jetoane negre.*
- 2. Identificați toate jetoanele ce vor muri. Identificați-le punând un jeton negru deasupra fiecărei celule.*
- 3. Localizați toate celulele unde se va produce o naștere. Puneți un jeton alb pe fiecare celulă.*
- 4. După ce modelul a fost verificat, pentru a ne asigura că nu au fost făcute erori, trebuie să dați la o parte toate celulele moarte și să înlocuiți toate organismele albe nou-născute cu jetoane negre.*

Regulile implementate pentru a face posibilă funcționarea jocului sunt următoarele:

1. *Toata celulele în viață care au mai mult de 3 vecini mor.*
2. *Orice celulă moartă care are 3 vecini în viață, devine vie.*
3. *Orice celulă cu mai puțini de 2 vecini în viață moare.*

Ca și exemple putem urmări mai jos ce se întâmplă cu o varietate de tipare simple:

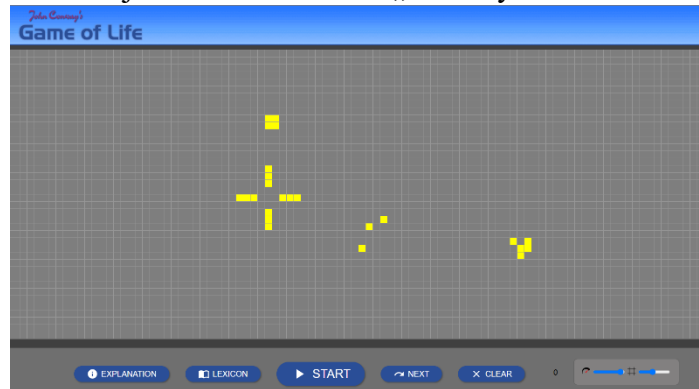


(Img 1- exepmple)

2.Suport tehnic

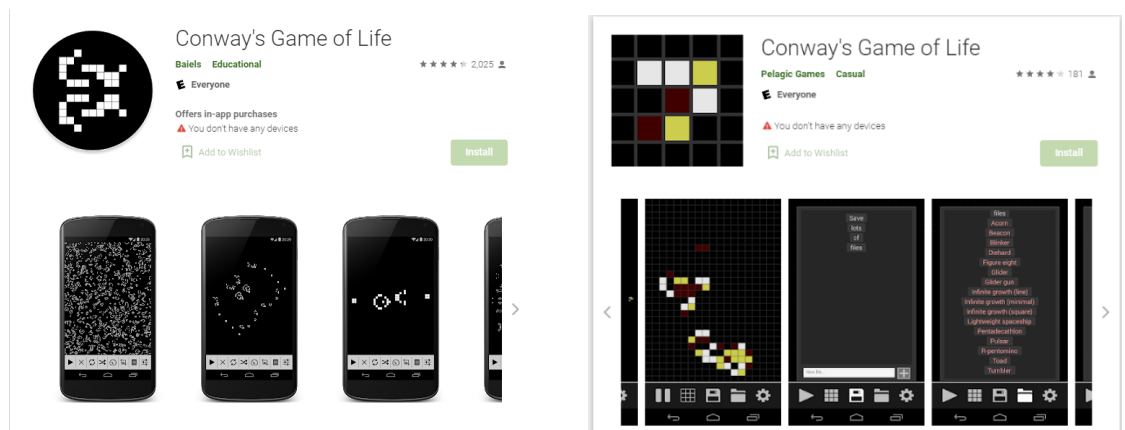
Proiectul realizat nu este singurul de acest fel. Există numeroase implementări ale acestui joc, atât ca aplicații desktop, web sau aplicații mobile. Majoritatea sunt numite „Conway’s Game Of Life”. De exemplu:

Web: <https://playgameoflife.com/>



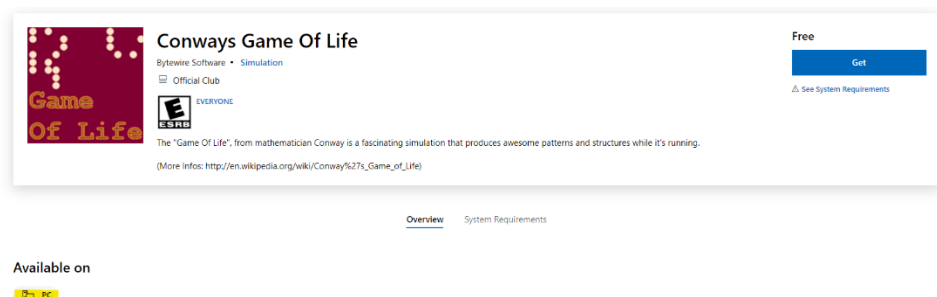
(img 2 -exp web)

Mobile:



(img 3,4 -exp mobile)

Desktop:



(img 5 -desktop)

Suportul principal in realizarea proiectului au fost aplicațiile web de tipul „Conway’s Game of Life”.

3.Implementare

Aplicația a fost realizată folosind:

- Javascript
- Node.js
- HTML

Au fost create doua pagini: o pagină principală pentru meniu-ul de start, ce conține titlul, regulile jocului și un buton de start care duce către a doua pagină.

```
function pageInit() {  
  
    textAlign(CENTER);  
  
    background(211, 254, 253);  
    textSize(width / 9);  
    fill(99, 193, 185);  
  
    text("The Game of Life", width / 2, 200);  
  
    textAlign(LEFT);  
  
    stroke(0, 0, 0, 20);  
    strokeWeight(4);  
    fill(187, 85, 107);  
    rect(width / 9, 300, 35, 35);  
  
    noStroke();  
    textSize(width / 30);  
    fill(187, 85, 107);  
    text("Celule in viata", width / 9 + 45, 330);  
  
    stroke(0, 0, 0, 20);  
    strokeWeight(4);  
    fill(255);  
    rect(width / 2 + width / 13, 300, 35, 35);  
  
    noStroke();  
    textSize(width / 30);  
    fill(99, 193, 185);  
    text("Celule moarte", width / 2 + width / 13 + 40, 330);  
  
    textAlign(CENTER);  
  
    textSize(width / 31);  
    fill(187, 85, 107);  
    text("1.Orice celula moarta care are 3 vecini in viata, devine vie", width / 2, 450);  
    fill(99, 193, 185);  
    text("2.Orice celula cu mai putini de doi vecini in viata moare", width / 2, 500);  
    fill(187, 85, 107);  
    text("3.Orice celula cu mai mult de 3 vecini in viata moare", width / 2, 550);  
  
}
```

(Img 6)

Cea de-a doua fereastră conține instrucțiunile necesare pe care utilizatorul trebuie să le urmeze pentru a începe jocul, cât și jocul în sine, adică un array bidimensional cu dimensiunea 35x35 de celule, pe care am aplicat regulile menționate mai sus.

```
var squares = 35;
var offsetY = 30;

//functia pentru a creea un array bidimensional
function make2DArray() {

    var arr = new Array(squares);
    for (var i = 0; i < arr.length; i++) {
        arr[i] = new Array(squares);
    }

    return arr;
}
```

(Img 7)

Există două moduri diferite de joc: **Primul mod** este modul random, când utilizatorul poate apăsa tasta „r” pentru a genera un număr aleatoriu de celule, așezate în poziții aleatoare.

```
function keyTyped() {

    if (key == 'r') {
        if (!cheiePress && butonPress) {
            for (let i = 0; i < squares; i++) {
                for (let j = 0; j < squares; j++) {
                    grid[i][j] = floor(random(2));
                }
            }

            cheiePress = false;
            rndm = true;
        }
    }

    if (key == 'c' && butonPress) {
        for (let i = 0; i < squares; i++) {
            for (let j = 0; j < squares; j++) {
                grid[i][j] = 0;
            }
        }

        rndm = false;
        cheiePress = false;
    }

    // keyCode = 13, reprezinta de fapt tasta ENTER, schimb valorile pentru 'cheiePress' in cazul in care pornesc sau
    //pun jocul pe pauza, cu 'false' si 'true'
    if (keyCode == 13 && butonPress) {
        if (!cheiePress) {
            cheiePress = true;
        } else if (cheiePress) {
            cheiePress = false;
        }
    }
}
```

(Img 8)

Cel **de-al doilea** mod de joc este reprezentat de modul liber, unde utilizatorul poate să selecteze folosind mouse-ul orice celulă dorește, așa oferindu-i viață. În acest fel putem să încercăm să găsim propriile modele inițiale pentru care sistemul de celule evoluează cel mai mult timp, sau care la final formează cele mai multe organisme stabile/diverse.

Am folosit funcția de mai jos pentru a realiza umplerea celulelor în acest mod.

```
function mousePressed() {  
  for (let i = 0; i < squares; i++) {  
    for (let j = 0; j < squares; j++) {  
      if (verifica && mouseX > i * matr && mouseX < i * matr + matr && mouseY > j * matr + offsetY && mouseY < j * matr + matr + offsetY) {  
        if (grid[i][j] == 0) {  
          grid[i][j] = 1;  
        } else {  
          grid[i][j] = 0;  
        }  
        break;  
      }  
    }  
  }  
}
```

(Img 9)

Codul a fost scris în Javascript și de asemenea am folosit bibliotecile externe „p5.js” și „p5.dom.js” pentru implementarea jocului. Acestea sunt incluse în fisierul HTML.

```
<!DOCTYPE html>  
<html>  
<head>  
  <meta charset="UTF-8">  
  <meta http-equiv="X-UA-Compatible" content="IE=edge">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  
  <title>Game of Life</title>  
  
  <script type="text/javascript" src="https://cdn.jsdelivr.net/npm/p5@1.3.0/lib/p5.min.js"></script>  
  
  <script type="text/javascript" src="sketch.js"></script>  
</head>  
<body>  
</body>  
</html>
```

(Img 10)

4.Mod utilizare

Utilizarea acestei aplicații este destul de intuitivă. Avem pagina principală, în care sunt menționate regulile, semnificația culorilor și un buton de start.



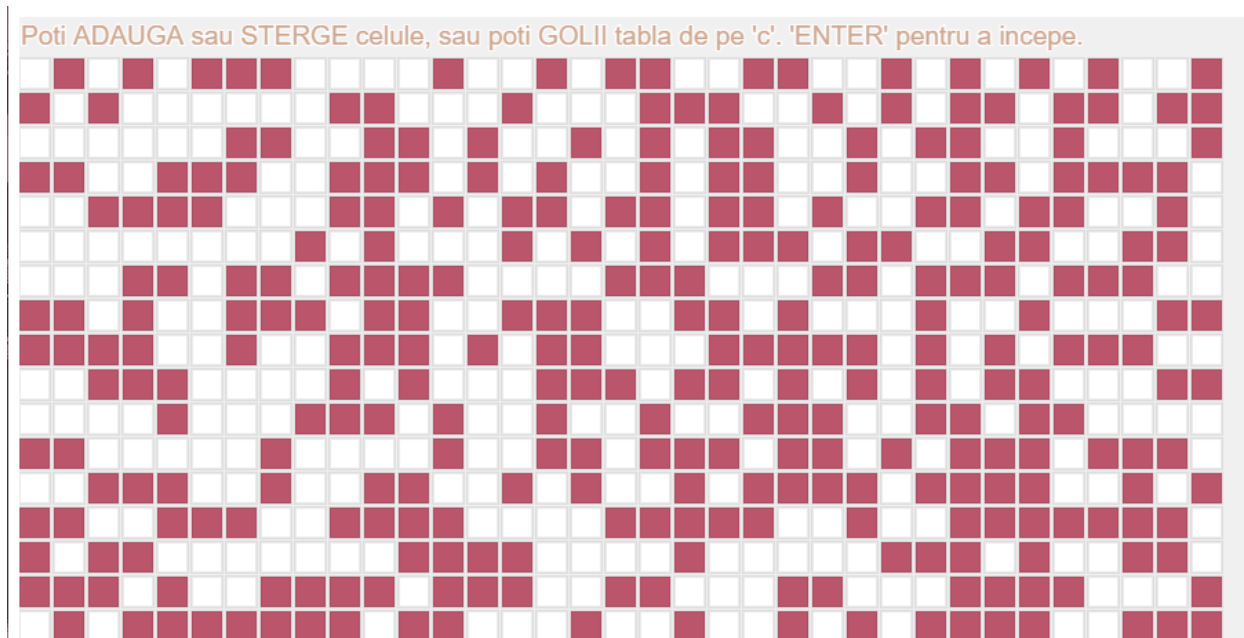
(Img 11 – pagina initiala)

După apăsarea butonului de start, fereastra se schimbă, iar utilizatorul poate începe efectiv jocul. În partea de sus a ecranului sunt menționate următoarele aspecte: Dacă apăsăm pe tasta „c” (clear), putem golii tabla, pe tasta „r” avem modul random, iar utilizatorului îi este specificat faptul că poate adăuga celule singur din mouse.



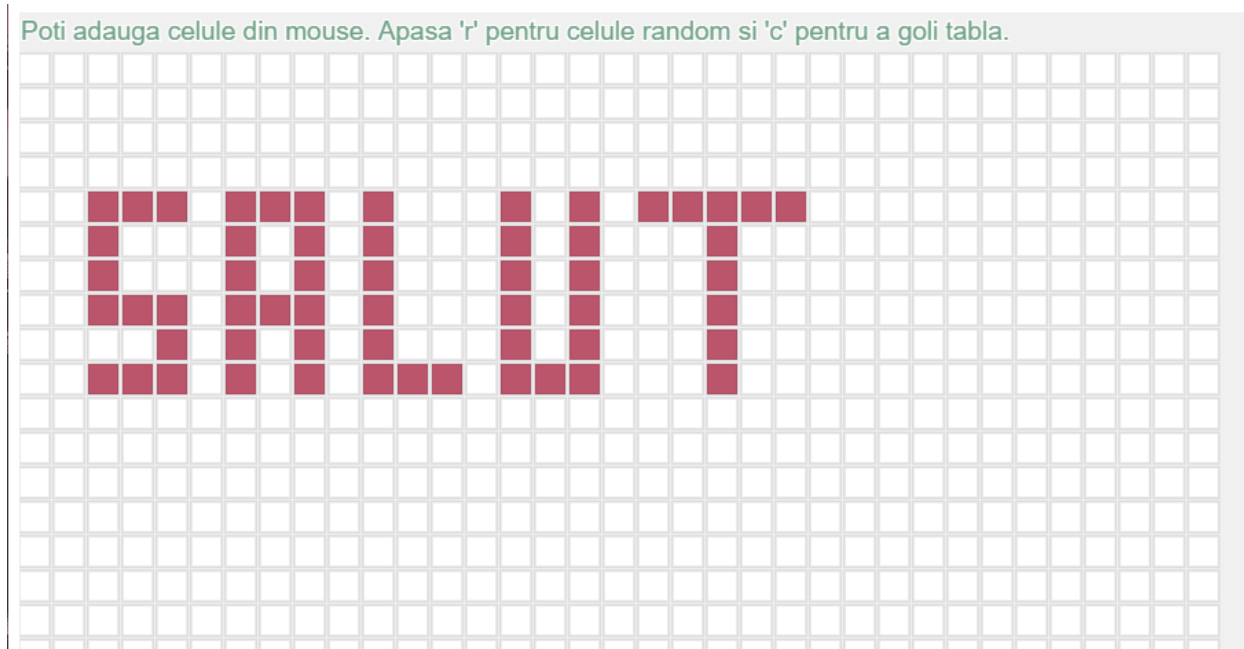
(Img 12 - instructiuni)

Daca apăsăm pe „r”, celulele se vor genera random și vor apărea noi instrucțiuni, adică faptul că putem adăuga sau șterge celule, acest lucru se face dând click pe ele, putem golii tabla apăsând pe „c”, sau puntem începe jocul apăsând pe tasta „ENTER”.



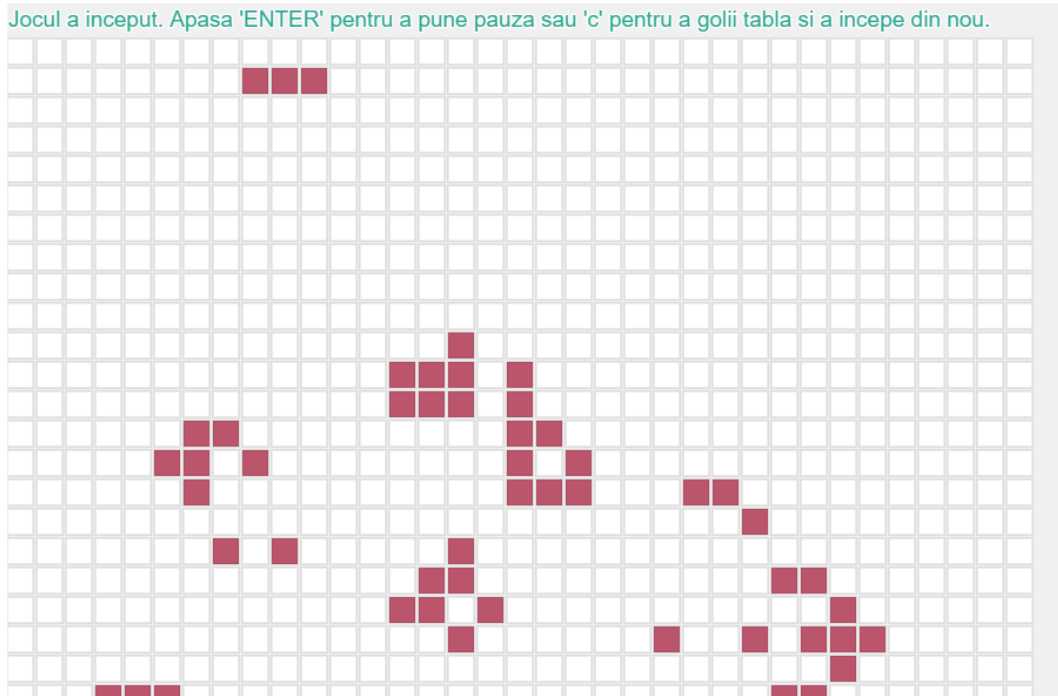
(Img 13 - instructiuni)

În cazul în care nu dormim modul random, și dorim să creem noi celulele, pur și simplu le selectăm din mouse și apoi apăsăm ENTER pentru a începe, la fel ca mai sus.



(Img 13 – instructiuni)

În momentul în care jocul a început, apar noi instrucțiuni. Dacă apăsăm din nou pe ENTER putem pune pauză și putem să mai adăugăm/ștergem celule în starea în care am rămas, sau putem să golim tabla. De asemenea, putem să facem aceste lucruri și în timp ce jocul rulează, acesta va funcționa în continuare.



(Img 15 – instrucțiuni)

Pentru a revenii la pagina inițială, putem da un simplu refresh.

Jocul poate fi deschis direct din fisierul HTML „index.html” sau poate fi rulat pe localhost:8080.

5.Concluzii

În urma realizării acestui proiect, am reușit să obțin o aplicație funcțională și interesantă din punctul meu de vedere, ce poate fi utilă unei persoane care dorește să experimenteze diverse moduri prin care poate face o civilizație de celule să evolueze sau care pur și simplu dorește să observe evoluția unui astfel de sistem, și să vada ce fel de organisme noi și stabile pot crea organismele inițiale.

Față de celelalte aplicații asemănătoare, aplicația mea are un design simplu dar plăcut, și este ușor de utilizat și experimentat.

Un alt plus, este faptul că, aceasta poate fi accesată de pe laptop/PC și este independentă de sistemul de operare.

Aplicația este la început și prezintă numeroase posibilități de dezvoltare. Pe lângă adăugarea altor funcționalități, următorul pas ar fi adăugarea unui mod în care utilizatorul poate să aleagă spawn-arearea unor celule predefinite, putând să aleagă numărul lor. De asemenea, poate fi implementat un algoritm de prezicere al celor mai frecvente organisme stabile, și colorarea acestora în culori specifice. Astfel, utilizatorul îi va fi mai ușor să observe evoluția celulelor, și va putea experimenta mai ușor.

6.Bibliografie

1. Gardner, Martin (October 1970). "[Mathematical Games - The Fantastic Combinations of John Conway's New Solitaire Game 'Life'](#)" (PDF). Scientific American (223)
2. The Coding Train. 2D Arrays in JavaScript - p5.js Tutorial -
<https://www.youtube.com/watch?v=OTNpiLUSiB4>
3. The Coding Train. The Game of Life - The Nature of Code –
<https://www.youtube.com/watch?v=tENSCEO-LEc>
4. Cornell Math Explorers' Club -
<http://pi.math.cornell.edu/~lipa/mec/lesson6.html>
5. Conway's Game of Life – Scholarpedia -
http://www.scholarpedia.org/article/Game_of_Life