Colegiul Național “Roman-Vodă”

Roman

­­­

**Jocul Life**

Lucrare pentru obținerea atestatului profesional la informatică

Candidat, Profesor coordonator,

Iacob Marcel-Emanuel Florin Moldovanu

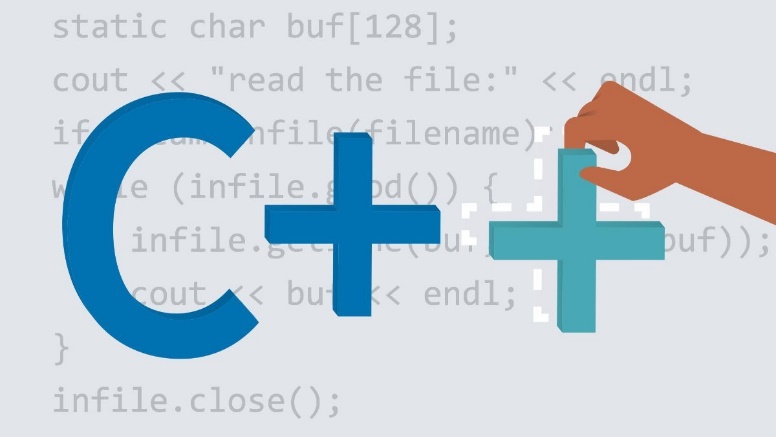
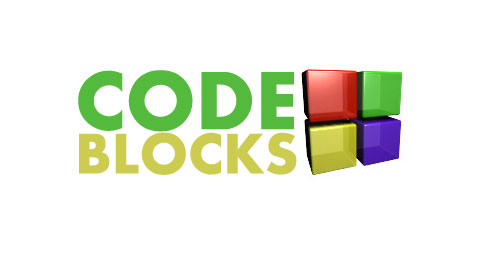
Cuprins

1. *Argument…………………………………………………………………………….pg.3*
2. *Scurtă prezentare a limbajului C++…………………………………………….pg.4*
3. *Prezentarea regulilor Jocului Life…………………………………………….....pg.5*
4. *Descrierea algoritmului utilizat pentru simularea jocului…………………...pg.6*
5. *Detalierea secvenței de cod utilizate………………………………………..pg.7*
6. *Ilustrarea interfeței grafice……………………………………………………pg.8*
7. *Bibliografie…………………………………………………………………….pg.10*
8. Argument

Am ales această temă în vederea implementării jocului într-o interfață user-friendly care folosește grafica în loc de simpla citire a matricii din fișier,urmând ulterior ca matricea modificată după un număr de zile să fie afișată într-un alt fișier.

Am considerat interesantă urmărirea evoluției în timp real a celulelor matricei,acest lucru fiind util pentru persoanele ce doresc să găsească diverse pattern-uri utilizând regulile acestui joc,vizualizarea configurației inițiale și schimbarea acesteia cu un singur click fiind mult mai intuitivă și mai simplă.

1. Scurtă prezentare a limbajului C++/a programului Code::Blocks



C-ul este un limbaj de programare de nivel mediu, dezvoltat de Denis Ritchie in 1970.

Se regareste in nucleul sistemelor UNIX si Linux, iar sintaxa sta la baza altor limbaje precum: C++, Java, JavaScript, C#, si D.

C++ -ul este o imbunatatire a vechiului C, standardizat in 1989, care acopera diferite lipsuri ca programarea orientata pe

obiecte si programarea generica, care astazi se intalneste in majoritatea soft-urilor.

Cele doua limbaje ofera o mare performanta si eficienta, din aceasta cauza sunt folosite in jocuri 3D si multe interfete grafice

Pentru crearea de programe în C++ este necesar un mediu de dezvoltare integrat (integrated development environment – IDE) care să uşureze pe deoparte procesul de programare, iar pe de altă parte, să integreze o serie de instrumente care să facă posibilă și să eficientizeze crearea de software.

Code::Blocks este un IDE pentru limbajele de programare C++, C și Fortran ce a fost lansat în versiune stabilă în 2008.

Acesta este gratuit, open-source și cross-platform (compatibil cu Windows, Mac și Linux), având proprietatea de a suporta folosirea de multiple compilatoare, inclusiv pe cel specific IDE-ului Microsoft Visual Studio C++, ceea ce face ca dezvoltatorul să aibă un control mai mare asupra software-ului creat și asupra portabilității codului sursă de pe alte IDE-uri.

Programul furnizează toate elementele necesare unui IDE performant:

Editor de cod – permite evidențierea codului în funcție de sintaxă, sugerarea de cod la tastare, posibilitatea de reformatare textuală a codului, personalizarea interfeței grafice, a fontului și a mărimi de font, organizarea facilă a fișierelor deschise – sub formă de file ș.a.

Depanator (debugger) – facilitează depistarea rapidă a erorilor de programare, prin inserția de breakpoints, vizualizarea funcțiilor locale și a argumentelor acestora, vizualizarea regiștrilor CPU, definirea de watches ș.a.

Designer GUI – permite proiectarea interfețelor grafice într-un mod vizual, de tipul WYSIWYG (What You See Is What You Get). Designerul se numește wxSmith și este derivat din librăria wxWidgets, librărie ce permite crearea de interfețe grafice cross-platform.

Posibilitatea migrării proiectelor de pe alte IDE-uri, de exemplu de pe DEV-C++ și Visual Studio.

Posibilitatea extinderii funcționalității prin intermediul plugins-urilor; în plus, plugins-urile pot fi create direct din cadrul programului.

Existența unei comunități numeroase de dezvoltatori care conlucrează la îmbunătățirea programului sau/și discută dificultățile întâlnite în cadrul acestuia. Mai mult decât atât, IDE-ul este open-source, deci oricine poate crea versiuni noi, particularizate după nevoile proprii.

Documentație completă cu privire la folosirea și configurarea IDE-ului (în limba engleză).

1. Prezentarea regulilor Jocului Life

Jocul Life, cunoscut și sub numele de Life, este un joc ce presupune simularea evoluției unor celule într-un spațiu bidimensional,conceput de matematicianul britanic John Horton Conway în anul 1970.

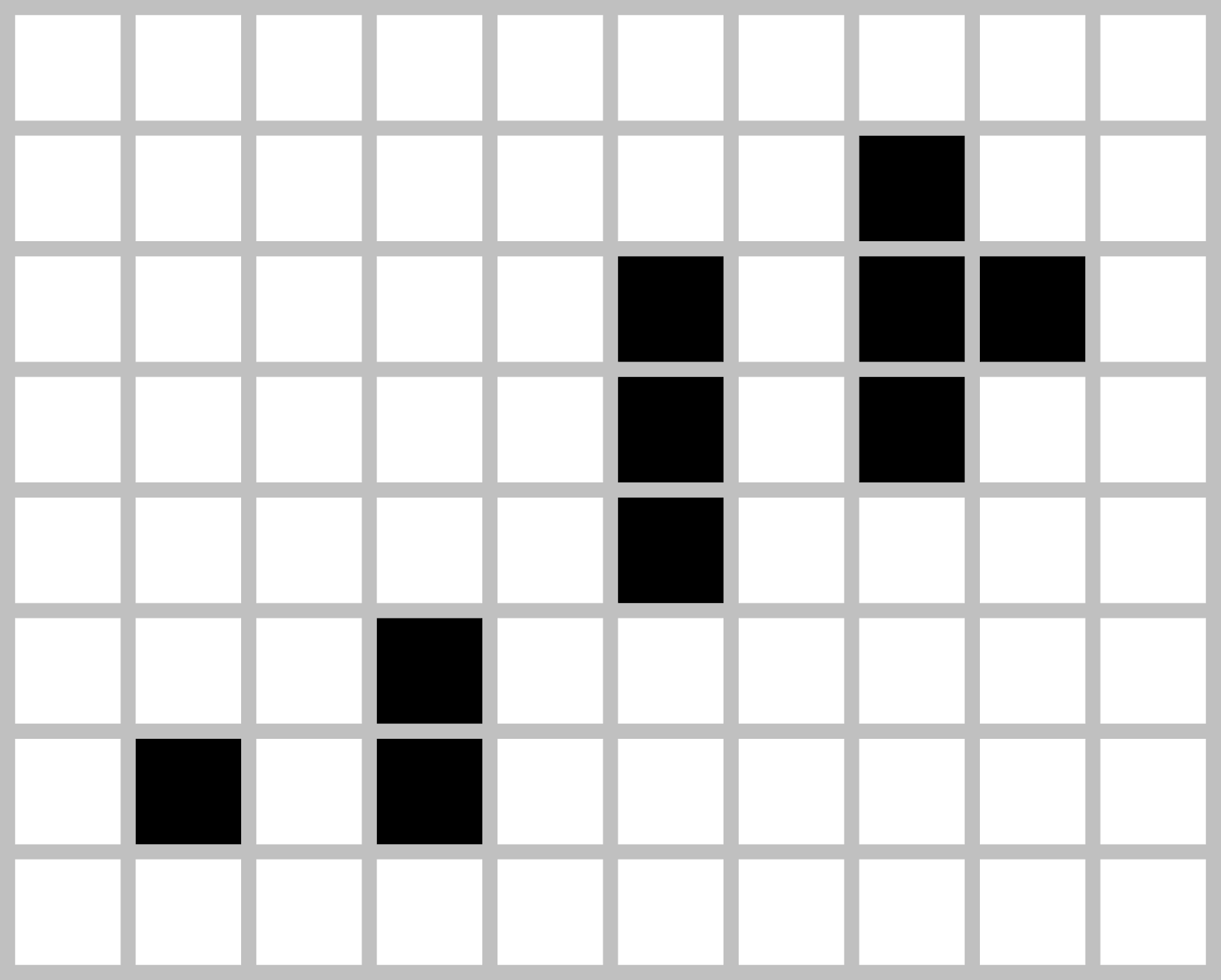
Jocul este un joc “fără jucători”,ceea ce înseamnă că evoluția acestuia este determinată de starea sa inițială, fără a fi nevoie de un input ulterior. Jucătorul interacționează cu Jocul Life, creând o configurație inițială și observând cum evoluează sau,pentru jucătorii avansați, prin crearea de modele cu proprietăți speciale.

Universul Jocului Life este o rețea ortogonală,bidimensională,de celule pătrate,fiecare dintre ele fiind în una dintre cele două stări posibile: vie sau moartă. Fiecare celulă interacționează cu cei opt vecini,care sunt celule ce sunt orizontal,vertical sau diagonal adiacente.

Multiplicarea celulelor urmează câteva reguli:

* Orice celulă cu mai puțin de două celule vecine moare de singurătate
* Orice celulă cu mai mult de trei celule vecine moare din cauza supraaglomerării
* Orice celulă cu două sau trei celule vecine supraviețuiește
* O celulă nouă este creată dacă are exact trei celule vecine

Spațiul simulării poate fi reprezentat fie ca un plan (celulele marginale au 5,respectiv 3 vecini),fie ca un toroid (toate celulele au 8 vecini), această adaptare a jocului utilizând reprezentarea plană.



1. Algoritmul Jocului Life utilizat

În primul rând,două matrici a[100][100] și b[100][100] sunt declarate global și astfel bordate cu 0. Aceste matrici vor reprezenta spațiul de joc,fiecare dintre ele fiind utilizate alternativ la fiecare etapă.

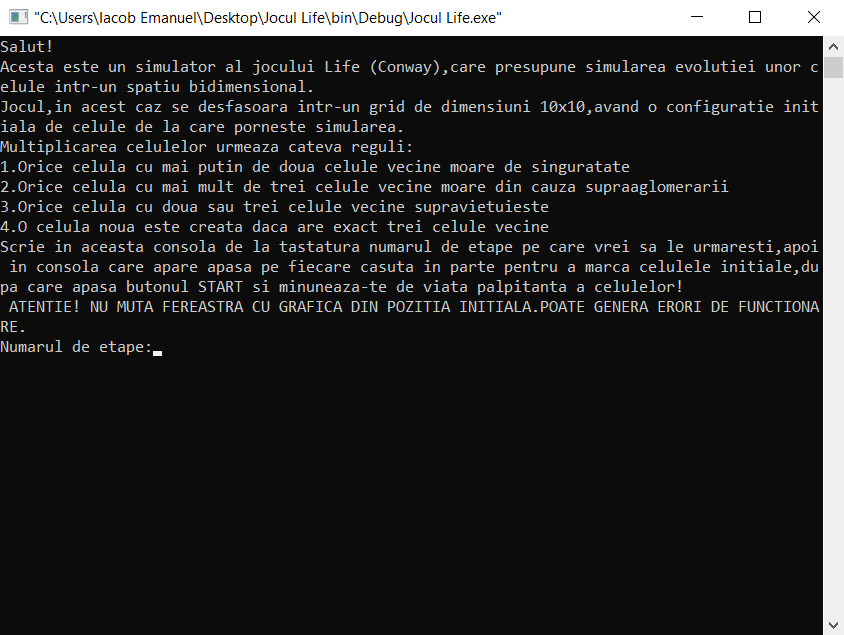
Utilizându-ne de funcția “verifvecini” ce are ca parametri matricea c și două coordonate i și j și returnează numărul de vecini adiacenți cu celula a[i][j],determinăm prin parcurgerea matricii dacă o celulă trebuie eliminată,trebuie păstrată sau alta nouă trebuie creată.

Astfel,parcurgem etapele până ajungem la etapa k (citită de la tastatură) și în funcție de paritatea etapei ne utilizăm de matricea a pentru a crea matricea b sau invers,utilizând condițiile menționate mai sus.

1. Programul și detalierea funcțiilor

* Funcția intro face o prezentare scurtă a Jocului Life și cere utilizatorului să scrie un număr ce reprezintă numărul de etape de evoluție ale planșei
* Funcția verifvecini este descrisă anterior
* Funcția deseninit desenează pe ecran o planșă de 10x10 cu căsuțe goale,precum și butonul de start care pornește evoluția celulelor,porrnind de la matricea c dată ca parametru
* În main este inițializată fereastra de grafică utilizând funcția initwindow,care primește rezoluția ferestrei,precum și coordonatele colțului din stânga sus. Utilizându-se de funcții precum settextstyle,outtextxy sau setcolor,acesta afișează text pe ecran,modifică mărimea sau stilul acestuia și modifică diverse culori pe ecran.
* While-ul ce urmează se execută până când este apăsat butonul start,afișat anterior pe ecran.Prin utilizarea funcției GetCursorPos,primim poziția cursorului,iar prin utilizarea funcției GetAsyncKeyState împreună cu click stânga sau click dreapta de la mouse verificăm dacă cursorul este în interiorul unei celule din matrice.Dacă da,celula se umple cu alb prin utilizarea funcției floodfill împreună cu setfillstyle (dacă nu e deja umplută cu alb),se calculează coordonata corespunzătoare din matricea din memorie (datorită faptului că am inițializat fereastra de 800x600,se pot calcula proporțiile) si se pune 1,si se umple cu negru daca este apasat click dreapta,iar in matrice se pune 0
* Următorul for parcurge simultan planșa de pe ecran,precum și matricea din memorie,verificând dacă celula trebuie păstrată sau ștearsă,utilizându-ne de funcția floodfill. De asemenea,după fiecare modificare a matricei se pune un delay ca să fie observată matricea curentă.
* După terminarea for-ului,se afișează un mesaj pe ecran care indică terminarea programului.



6. Interfața propriu zisă

1.Consola

Mai jos avem fereastra care apare atunci când utilizatorul execută programul

Sunt prezentate în această fereastră regulile jocului,precum și un advertisment ca utilizatorul să nu mute fereastra care apare ulterior (acest lucru generează funcționarea incorectă a programului).

De asemenea,i se cere să scrie numărul de etape a căror evoluție dorește să le urmărească. După scrierea acestui număr,se afișează fereastră de grafică

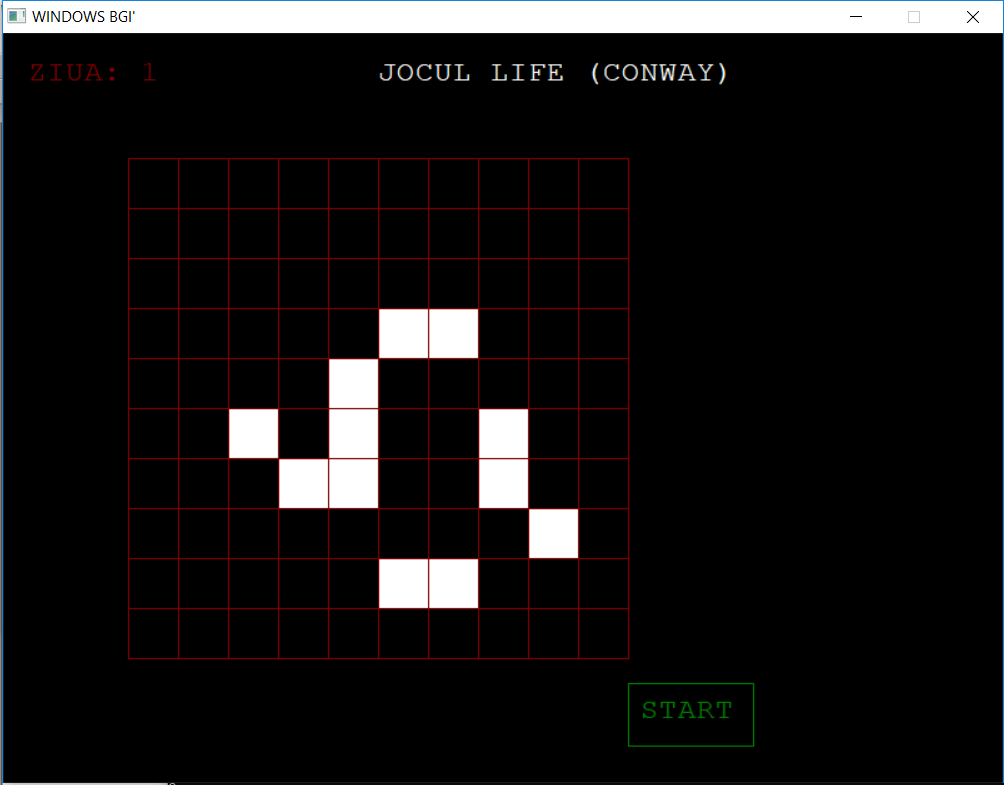
2. Fereastra de grafică

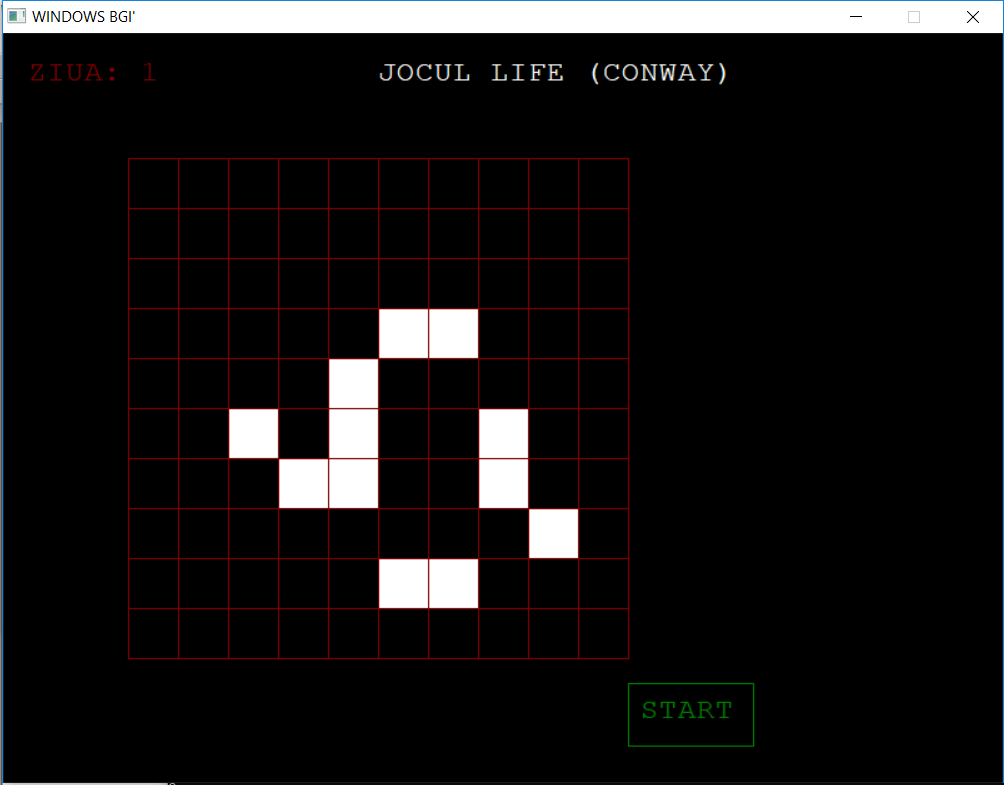
Fereastra de grafică arată inițial în felul următor:



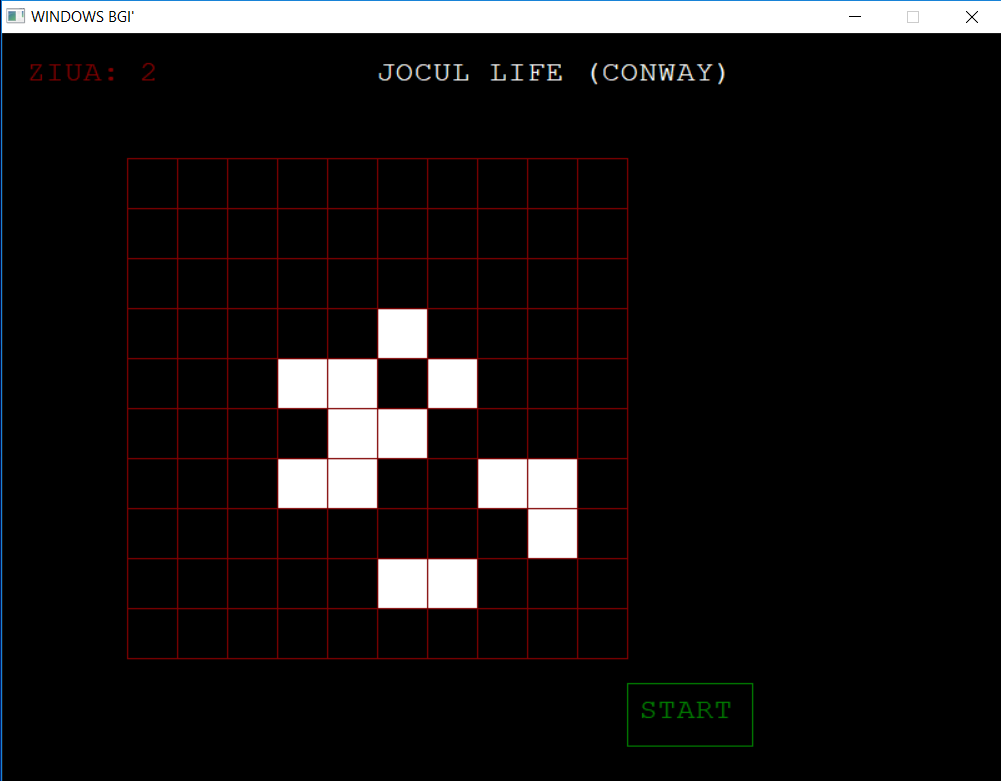
Utilizatorul poate apăsa click stânga pe orice celulă neumplută,urmând ca aceasta să se umple cu alb și să semnaleze o celulă existentă pe planșă (acest lucru actualizându-se și în matricea din memorie). Similar poate apăsa și click dreapta pe o celulă existentă pentru a o șterge.

După ce utilizatorul este mulțumit cu configurația curentă,acesta poate apăsa pe butonul de start din partea dreapta jos a ecranului,urmând ca planșa să se modifice corespunzător. În partea din stânga sus este afișată ziua curentă.

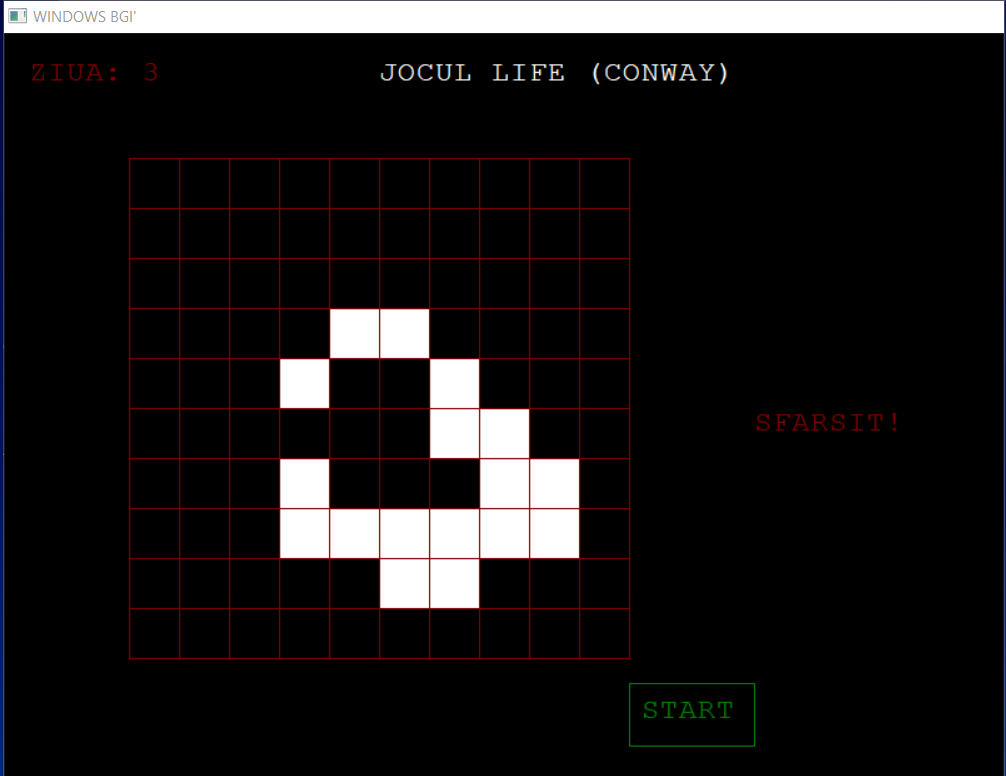
**Exemplu de evoluție a unei planșe pe parcursul a trei zile**



Planșa inițială a utilizatorului



A doua etapă a evoluției



Ultima etapă (se observă mesajul sugestiv)

Bibliografie

<https://programare.tech/blog/articole/scurta-prezentare-c-c-->

<http://www.cplusplus.com/>

<https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life>

<https://ctrl-d.ro/tips-and-tricks/codeblocks-un-mediu-de-dezvoltare-integrat-ide-pentru-c/>

<https://www.linkedin.com/learning/c-plus-plus-essential-training-2>