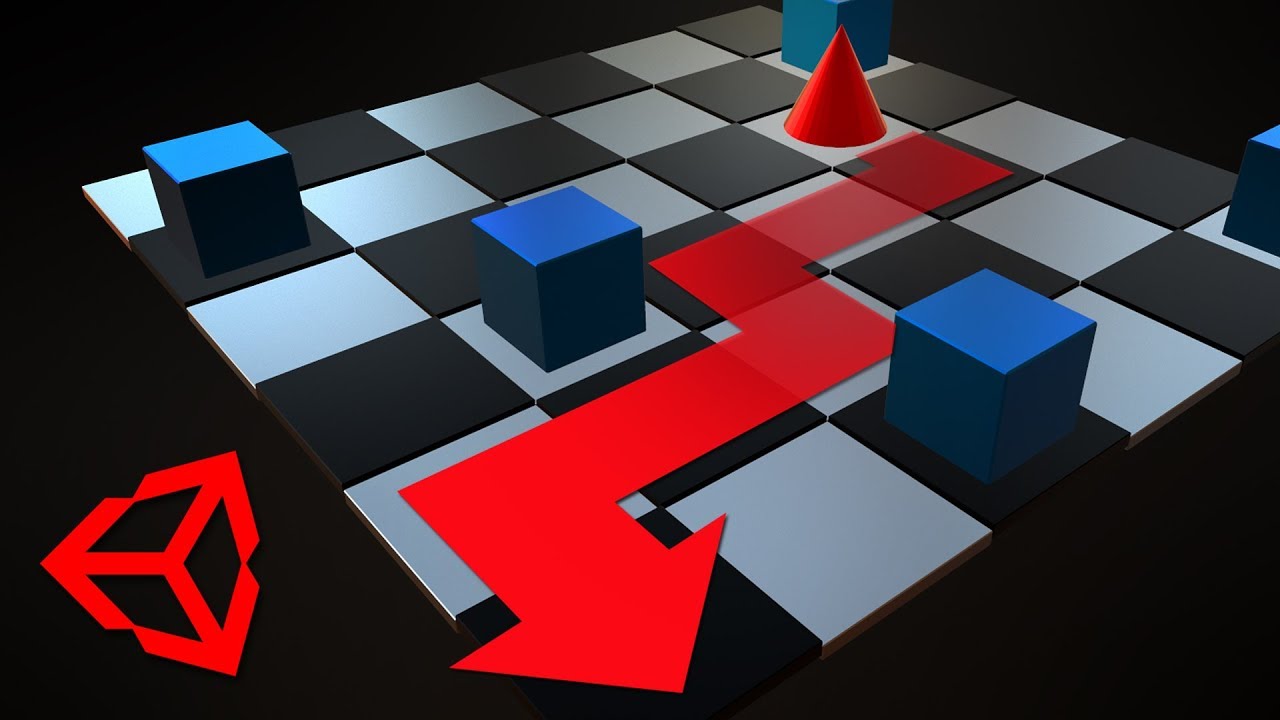
Colegiul Național “Eudoxiu Hurmuzachi ”, Rădăuți

**Pathfinding Visualizer**



**Elev: Gherasim Teodor-Samuel**

**Prof. îndrumător: Lăzărescu Antoneta**

Mai 2020

**Introducere**

Pentru un elev poate fi foarte dificil să învețe concepte noi de programare dacă acestea nu îi sunt explicate în detaliu și exemplificate. Tocmai de aceea, elevii ce invață informatică au nevoie de site-uri web precum **”Pathfinding Visualizer”.**

Am creat acest website pentru a putea înțelege și vizualiza algoritmii de căutare a drumului de distanță minimă între două puncte, în mod deosebit al algoritmului lui Dijkstra.

Algoritmii prelucrare și interpretare a grafurilor sunt foarte complecși, dar totuși sunt necesari a fi cunoscuți de orice programator în devenire. Prin metode interactive, învățarea devine mai ușoară și mai distractivă.

**Scopul lucrării**

Scopul site-ului este acela de a-l învăța pe utilizator modul de funcționare al algoritmilor de căutare cu ajutorul unei interfețe prietenoase și intuitive.

El va putea învăța într-un mod foarte intuitiv concepte legate de teoria grafului și va putea experimenta testându-le în contexte diferite.

**Cui i se adresează?**

Site-ul se adresează tuturor categoriilor de vârstă, în special elevilor și studenților care studiază informatica, sau celor care doresc să vizualizeze cum anume calculează aplicațiile de hărți și navigare distanța cea mai mică până la destinație.

**Modul de realizare și funționare**

Site-ul a fost realizat utilizând limbajul **HTML**, limbajul de stilizare **CSS** şi limbajul de programare **JavaScript** (în mod particular, librăria **Node.js** a acestuia).

Codul a fost scris în mediul de programare **Microsoft Visual Studio**, având 6 fișiere JavaScript (.js/.jsx) și 5 fișiere CSS interconectate (.css) pentru a face mai ușoară repartizarea rolurilor fiecărui bloc de cod.

Am utilizat programarea orientată pe obiecte și am realizat clase de obiecte pentru a putea face interfața cât mai eficientă și mai clar definită.

Întreaga funcționalitate a site-ului este accesibilă din pagina principală. Prin intermediul claselor, am definit nodurile unei matrici și am atribuit acestora 6 stări posibile, identificate în funcție de culoarea explicată în **Legendă**. Aceste stări sunt modificate de către utilizator prin acționarea butoanelor programate cu ajutorul limbajului HTML și acționate prin funcția onClick a librăriei Node.js.

Utilizatorului îi este explicat modul de utilizare a site-ului prin accesarea butonului de Help **( ? )** din partea dreaptă sus a ecranului. Folosind o altă librărie JavaScript (react-module.js), meniul de ajutor se deschide într-un modul stilizat prin limbajul CSS.

Utilizatorul poate interacționa cu matricea, desenând zidurile unui labirint sau mutând nodul de început și cel de final. De asemenea, prin apăsarea butonului **Create Maze,** va fi animat un labirint predefinit pe care utilizatorul îl poate folosi și customiza.

Dacă este satisfăcut cu labirintul creat, acesta poate vizualiza modul de funcționare al algoritmului de căutare prin apăsarea butonului **Visualize Dijkstra’s Algorithm**.

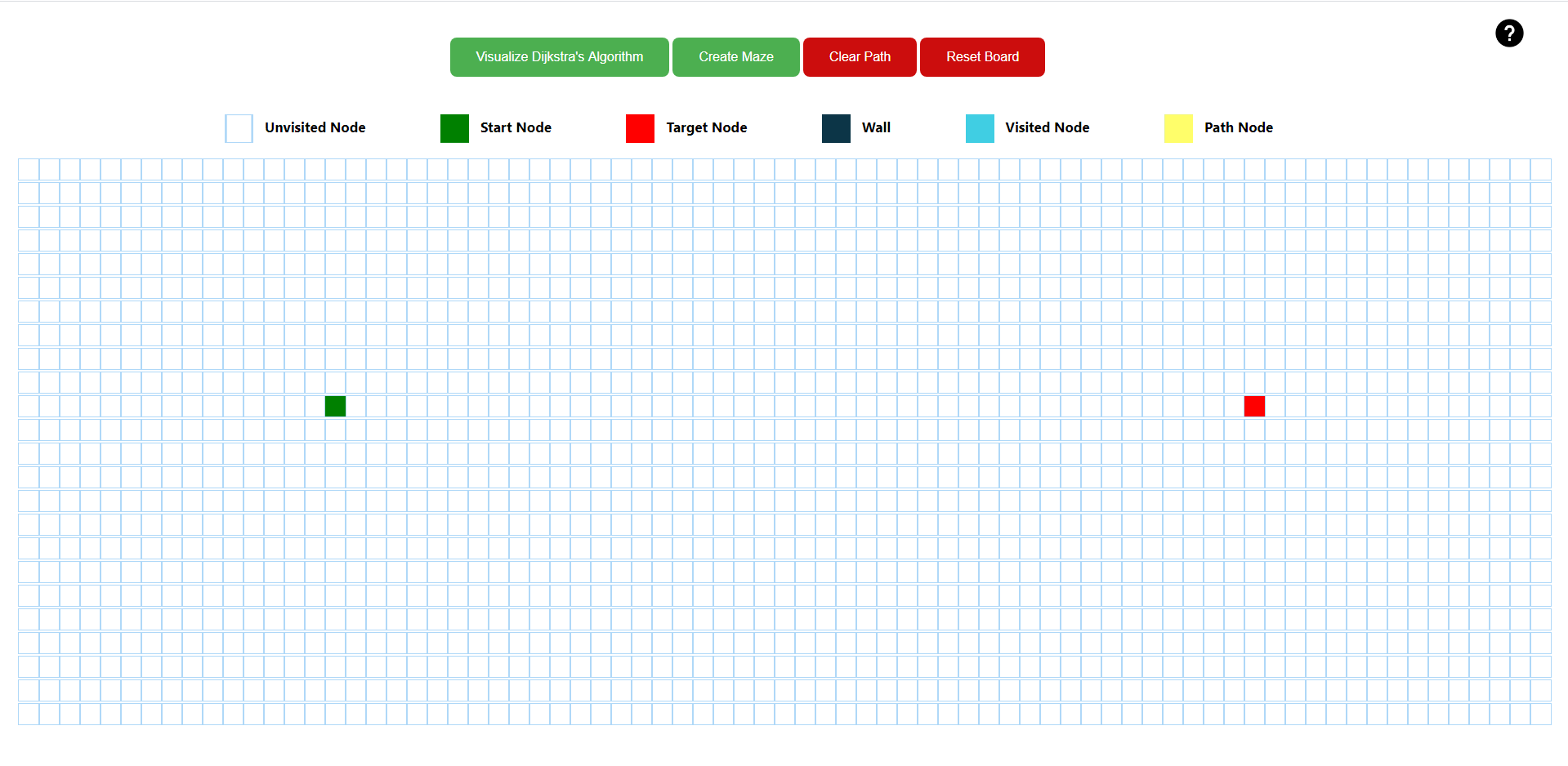
Utilizatorul este împiedicat din a face orice acțiune până la sfârșirea animației, după care poate să continue fie prin a șterge drumul parcurs, fie prin a reseta labirintul (butoanele **Clear Path** și **Reset Maze**).

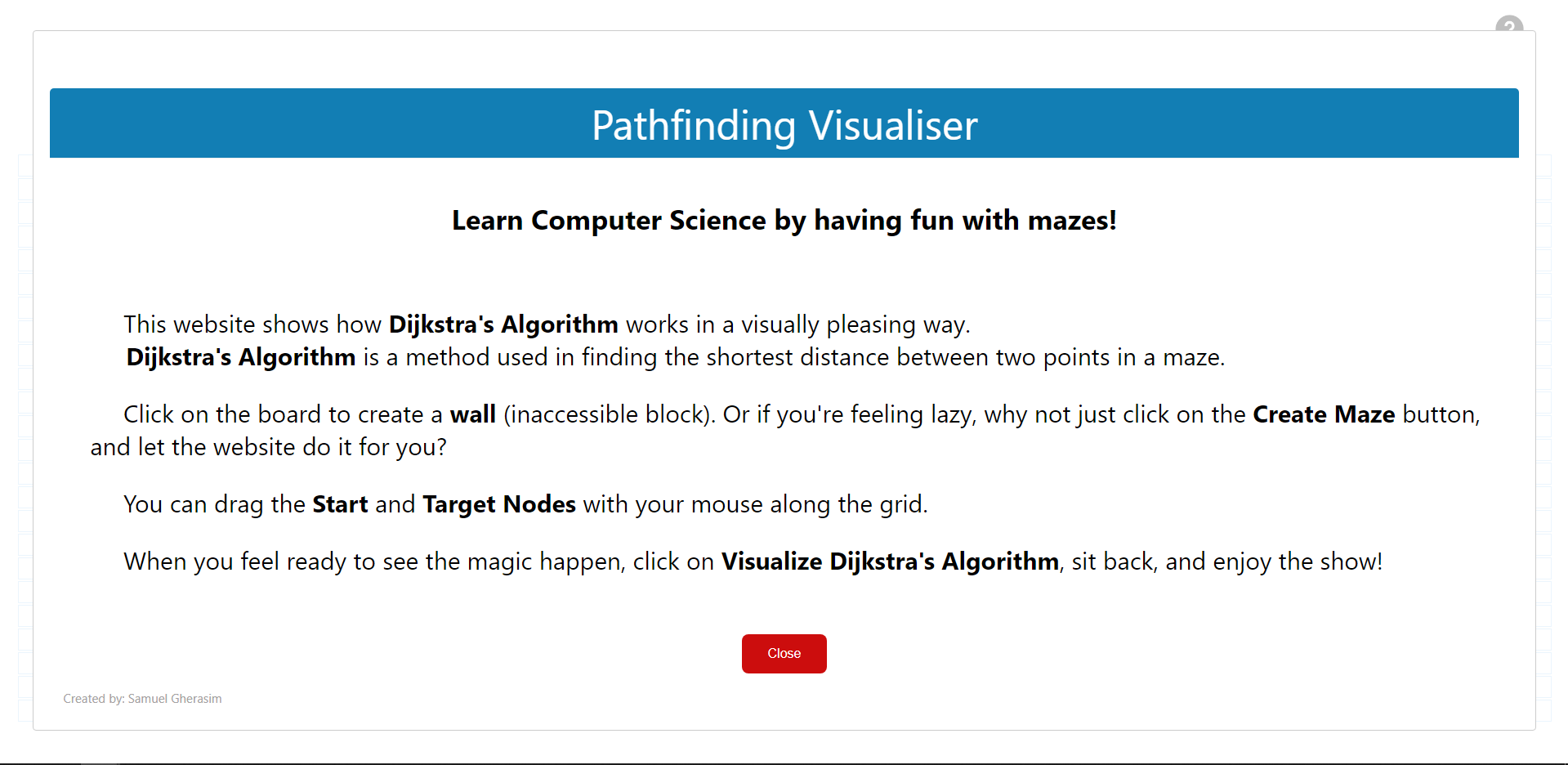
În aproximativ **900 de linii de cod**, obiectele interactive au fost stilizate și animate utilizând limbajul CSS, iar algoritmii utilizați pentru calculare și procesare au fost scriși în limbajul JavaScript.

**Versionarea și subversionarea** acestui proiect a fost realizată prin intermediul platformei **Github** (<https://github.com/EleyondFarli/PathfindingVisualiser>), iar proiectul este încărcat și pe internet prin platforma **Netify**, având adresa url : <https://eloquent-mccarthy-8b19cb.netlify.app>

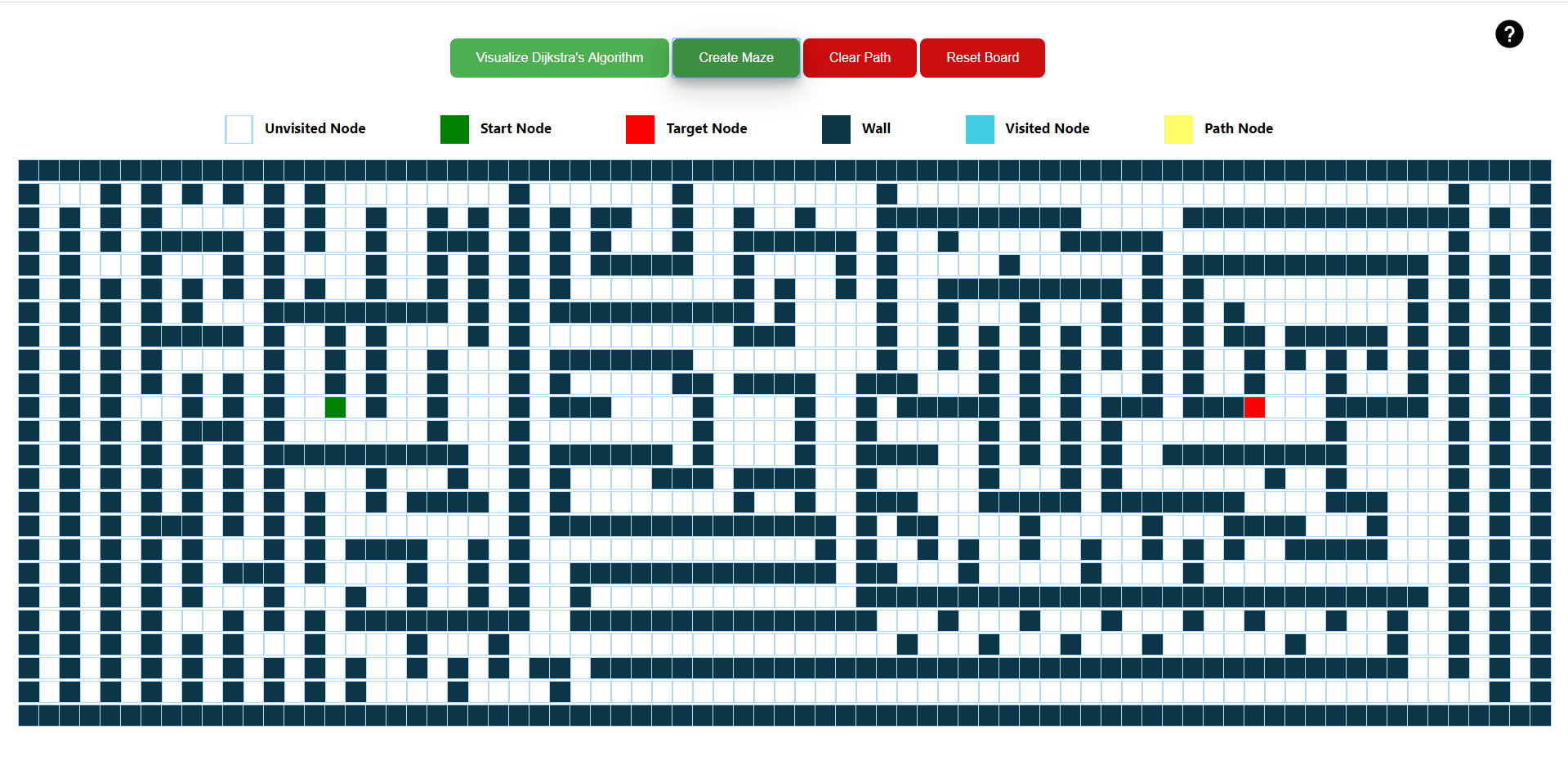
.

**Prezentarea proiectului**

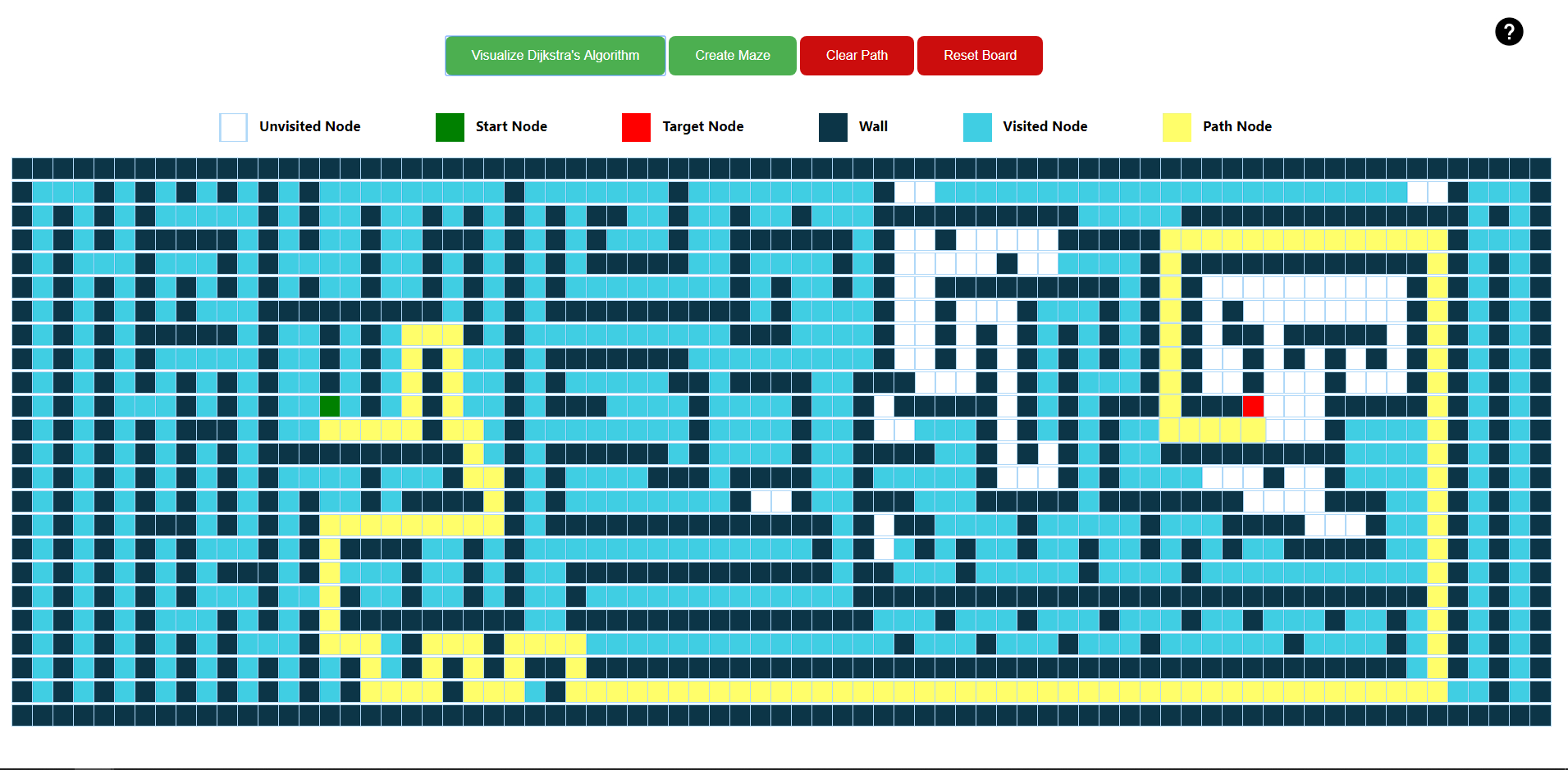
1.Pagina principală

2. Meniul de ajutor

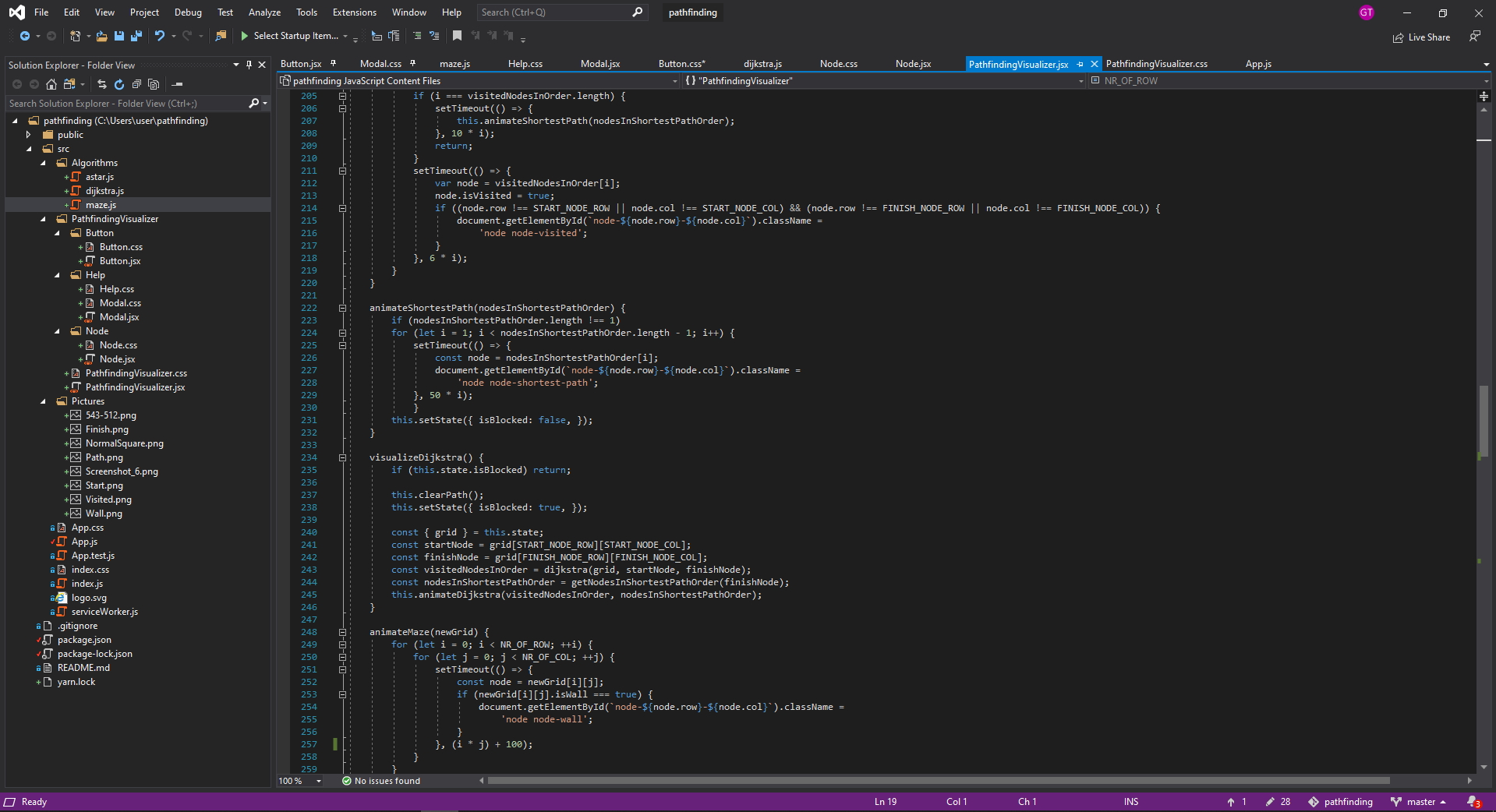
3. Labirintul creat prin butonul **Create Maze**



**4. Găsirea drumului cel mai scurt de către algoritmul de căutare**



**5. Mediul de lucru**



**Bibliografie**

1. <https://en.wikipedia.org/wiki/Dijkstra's_algorithm>