

UNIVERSITATEA "ȘTEFAN CEL MARE" SUCEAVA
FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR
SPECIALIZAREA CALCULATOARE

PROIECT DISCIPLINĂ POO

"Identificare traseu între 2 puncte"

Canevschii Daniel

Tema și motivația alegerii

Tema dată presupune găsirea celui mai scurt traseu între două puncte. Ca scop final al proiectului se cere ca programul dat să poată găsi cel mai scurt traseu posibil între două puncte de pe ecran, poziția cărora este dată de către utilizator.

Deși problema dată reprezintă în mare parte o problemă de algoritmică, implementarea acesteia în limbajul de programare C++ presupune o realizare a acestuia în stil OOP.

Motivația de a alege acest proiect se află în oportunitatea studierii mai extensiv atât al **Programării Orientate pe Obiecte**, cât și a algoritmilor de căutare bazați pe grafuri.

Inspirația alegerii:

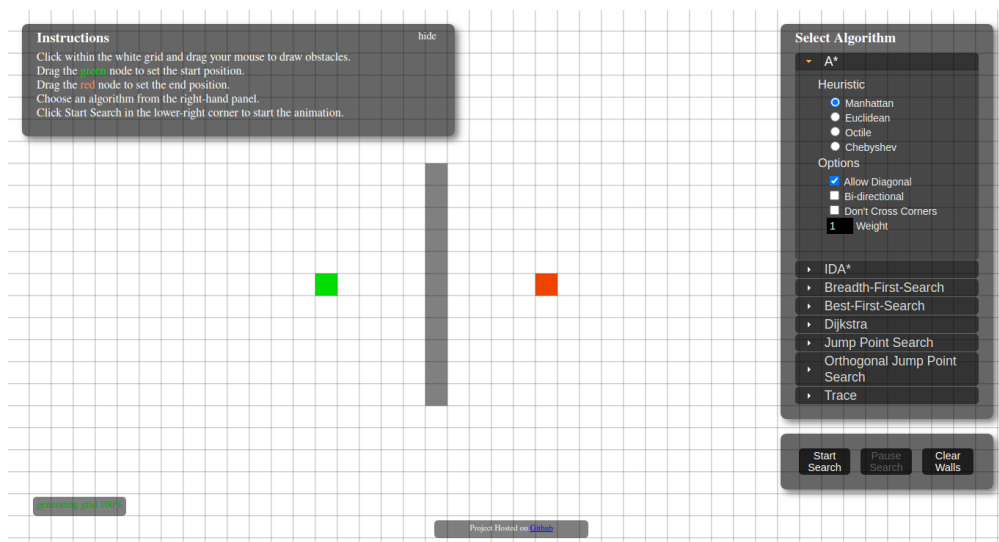


Fig.1 Proiectul de inspirație ¹ (punct de start și final cu barieră la mijloc)

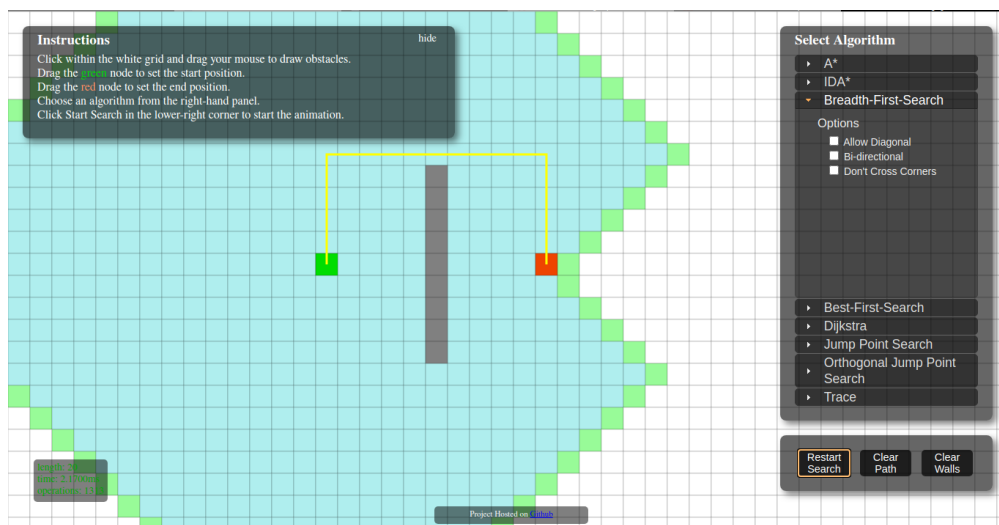


Fig.2 Proiectul de inspirație(faza finală)

¹<https://qiao.github.io/PathFinding.js/visual/>

Cuprins

1	Elemente teoretice	3
1.1	Descrierea Proiectului	3
1.2	Elemente specifice POO	4
1.3	Elemente specifice Interfață	5
2	Implementare	6
2.1	Tehnologii folosite	6
2.2	Diagrama de clase	6
3	Analiza soluției implementate	7
4	Manual de utilizare	8
5	Concluzii	9
6	Bibliografie	10
6.1	Articole	10
6.2	Surse Diverse	10

Capitolul 1

Elemente teoretice

1.1 Descrierea Proiectului

Proiectul dat prezintă cautarea și afișarea în timp real a găsirii celui mai scurt drum între două puncte de pe grid.

Interfața proiectului presupune două puncte, poziția cărora poate fi modificată de către utilizator, adițional acesta permite utilizatorului să definească obstacole ce pot modifica drumul rezultat.

Interfața, la baza căreia stă librăria open-source FTXUI ¹ pentru dezvoltarea interfețelor de tip terminal utilizând limbajul C++, are o latență de reîmprospătare de 100ms, care trece prin întreaga matrice, în care este definită starea rețelei afișată pe ecran.

În matrice fiecare celulă reprezintă o celulă afișată pe ecran, valoarea fiecărei celule din matrice e reprezentată prin numerele {0, 1, 2, 3, 4, 5} ce reprezintă respectiv {empty, start, end, wall, visited, path} care au la rândul lor asiguate culori specifice.

Partea dreaptă a ecranului presupune un mesaj de introducere și o listă de tip radiobutton care permite alegerea algoritmului de *pathfinding* dorit.

Se presupun deasemenea butoane care efectueaza actiuni de tipul:

- *START* - butonul de start a algoritmului de cautare;
- *CLEAR* - curățare screen de tot inafara de punctul de start și end;
- *RESET* - butonul de reset a gridului (poziția inițială a celulelor *start* și *end*;

¹<https://github.com/ArthurSonzogni/FTXUI>

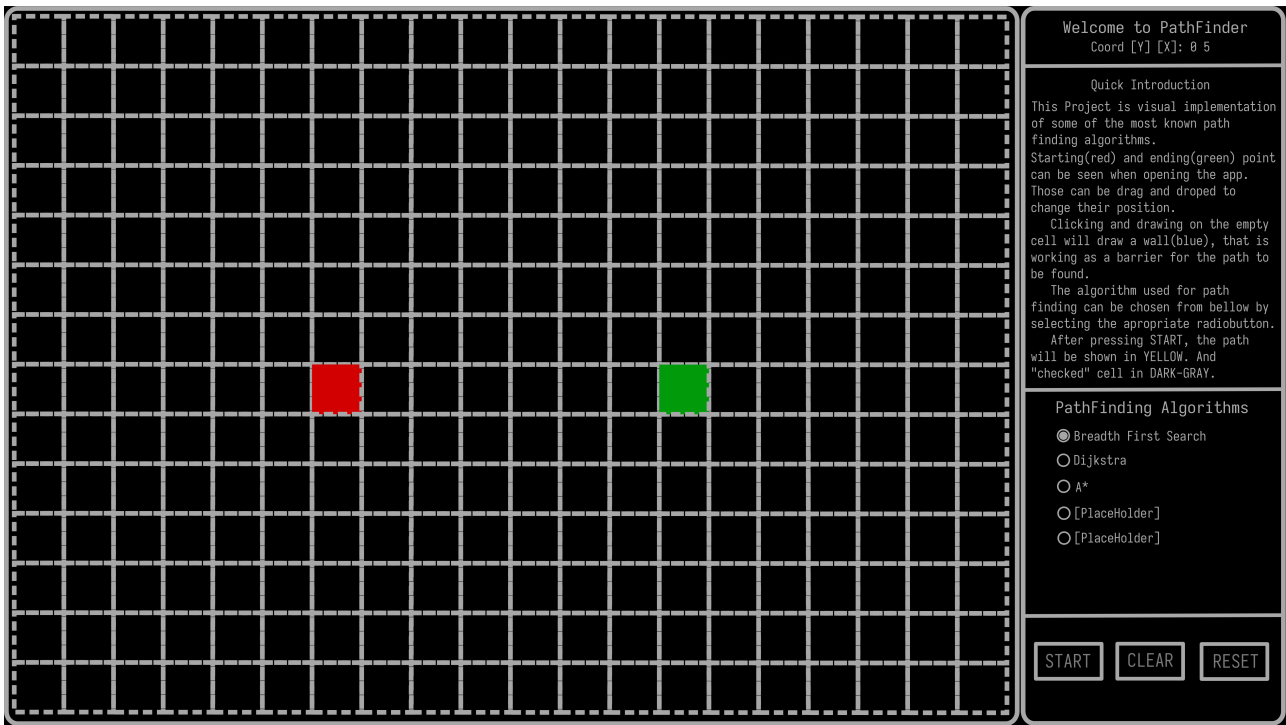


Fig. 3 Interfața propusă spre realizare

1.2 Elemente specifice POO

Fișierul *main* are la bază doar partea de interfață, care crează clasa de tip *Grid* și apelează metodele specifice a acesteia în dependență de *Event-ul* curent.

Clasa *Grid* are ca variabilă privată de tip clasă *Matrix*, și metodele acesteia efectuează modificări asupra variabilei *matrix*.

Un fișier cu funcții de calculare a traseului cel mai eficient în dependență de algoritmul de cautare ales.

Deasemenea se va crea o clasă de pastrare(matrice) a stării gridului care e modificat de către clasa de calculare a traseului și a interfeței. Datele căreia sunt preluate pentru afișare de către *thread-ul* de *refresh* a ecranului.

1.3 Elemente specifice Interfață

Canvas

Pentru partea de interfață se consideră funcție de convertire a unei matrici intrun grid de tip *ftxui::canvas*, verificând valoare fiecărei celule a matricii și atribuindu-i culoarea corespunzătoare valorii.

Aceasta face o reîmprospătare la fiecare 100ms, și la fiecare *Event* primit de la utilizator(în mare parte *Mouse Event*).

La un *Event* de tip mouse, în caz ca pointerul mouse-ului se află în zona de grid, se calculeaza poziția acestuia în referință cu celula pe care se află pe ambele axe *x* și *y*.

În cazul în care celula pe care s-a aflat pointerul mouse-ului e o celula de tip *empty*, aceasta se va schimba pe o celulă de tip *wall* și viceversa. De asemenea se poate apăsa pe o celula goală și face hover pe celule, dacă sunt de tip *empty* se convertesc în celule de tip *wall*.

Modificarea poziției celulelor de tip *start* și *end*, se efectuează utilizând drag-and-drop. Dacă drop-ul se face pe celulă de tip opus acesteia, atunci aceasta revine la poziția de unde a fost luată.

La apăsarea butonul START se începe cautarea drumului într-un nou *thread* și modifică valoarea celulelor vizitate cu o latență de 25 ms.

Iar în momentul când găsește celula de tip *end*, cătarea se oprește și se începe recreierea drumului de la final la început.

Capitolul 2

Implementare

2.1 Tehnologii folosite

Limbajul de bază a proiectului e **C++**, compilat folosit compilatorul **g++**, utilizând utilitarul de compilare și administrare a build-ului automat **CMake** și **Make**.

Drept mediu de dezvoltare se va utiliza editorul de fișiere **VIM**, iar pentru managementul modificărilor făcute în proiect se utilizează, sistemul de control a versiunii **GIT**.

Proiectul are o copie up-to-date încărcată pe **GITHUB**.

Interfața grafică a acestuia se va crea utilizând o librerie open source **FX-TUI**.

Pentru o vizualizare real-time a căutării traseului s-a utilizat tehnologia de *threading*, și încetinire a căutării acestuia.

2.2 Diagrama de clase

Capitolul 3

Analiza soluției implementate

Capitolul 4

Manual de utilizare

Capitolul 5

Concluzii

Capitolul 6

Bibliografie

6.1 Articole

<https://en.cppreference.com/w/cpp/thread/thread/detach>

https://en.cppreference.com/w/cpp/thread/sleep_for

<https://www.codespeedy.com/dictionary-in-cpp/>

<https://en.cppreference.com/w/cpp/container/map>

<https://www.redblobgames.com/pathfinding/a-star/introduction.html>

<https://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html>

6.2 Surse Diverse

<https://arthursonzogni.github.io/FTXUI>

https://github.com/ebarmas/minesweeper_ftxui/blob/main/src/minesweeper.cpp

chat.openai.com

<https://stackoverflow.com/questions/618511/a-proper-way-to-create-a-matrix-in-c>

<https://stackoverflow.com/questions/2076624/c-matrix-class>

<https://stackoverflow.com/questions/62735210/adding-a-struct-into-a-map>

<https://stackoverflow.com/questions/8963208/gdb-display-output-of-target-application-in-a-s>