Maze2D Algo Editor

Authors: **Eran Safadel & Batya Pollack**

Project name: Algo Editor

**Project's description:**

We build a Maze2D Algo Editor that shows a three different running algorithms: BFS, DFS, A\*.

"Algo Editor"- was developed in C++ and the Graphic Library -SFML 2.5.1.

**Algo Editor Emphasis on:**

* **OOP principles**- (Abstraction, Encapsulation, Inheritance and Polymorphism).
* **Design patterns**- (Singleton and Command) to provide general solutions.
* **STL** (Containers, Iterators, Smart pointers and lambda functions)
* **Algorithms**: BFS, DFS, A\*

**Details on the main classes:**

Controller – The brain of program, manage all the main objects and run the program.

Board- Responsible for all operations on grid (Create, Refresh, Draw, Delete, Resize)

MediaSource- Responsible for media loading in the program. MediaSource implemented through "Singleton" for efficient media loading.

Button - Responsible for the buttons that perform actions: setting start point and end of the route, delete button, matrix refresh and matrix resize.

AlgoCommand – Responsible for the buttons that execute algorithms (BFS, DFS, A\*).

MenuButton – Menu that holds the different buttons.

MenuAlgo- Menu that holds the different algorithms.

Macros – file that holds the constants of the program .

**Data structures:**

* BFS

std::queue<sf::Vector2i> m\_nodeQueue; // make queue to maze

std::list <DIRECTION> m\_path; // the path

std::vector<std::vector<GraphNodeBFS>> m\_edges; // to save every node prev

* DFS

std::vector<std::vector<bool>> m\_visited;

std::vector < std::vector <std::vector < sf::Vector2i >>> m\_adjacents;

* A\*

typedef std::pair<double,std::pair<int,int> > pPair;

std::vector<std::vector<bool>> m\_closedList;

std::vector<std::vector<cell>> m\_cellDetails;

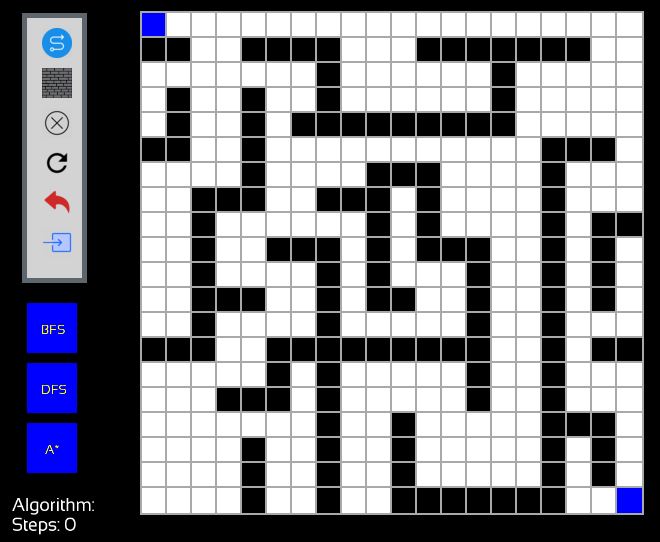
std::set<pPair> openList;

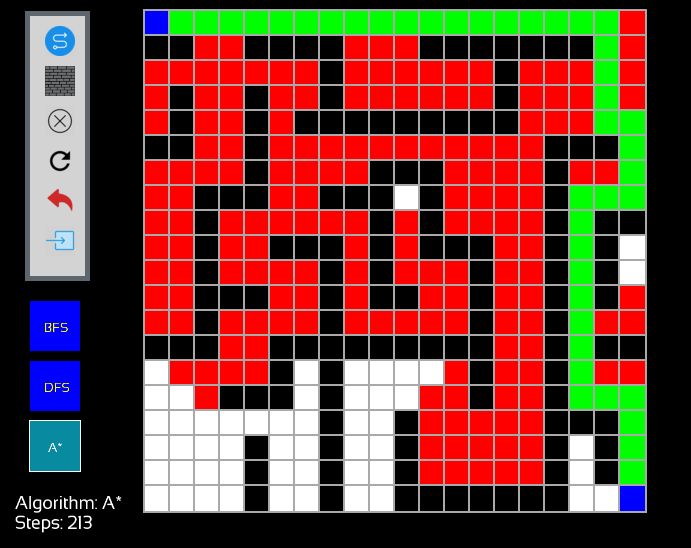
**Other notes:**

Attached a "UML" folder showing the **UML** of the program.

**Pictures from the program:**

Before:



After execute A\*****