

Opis Klas

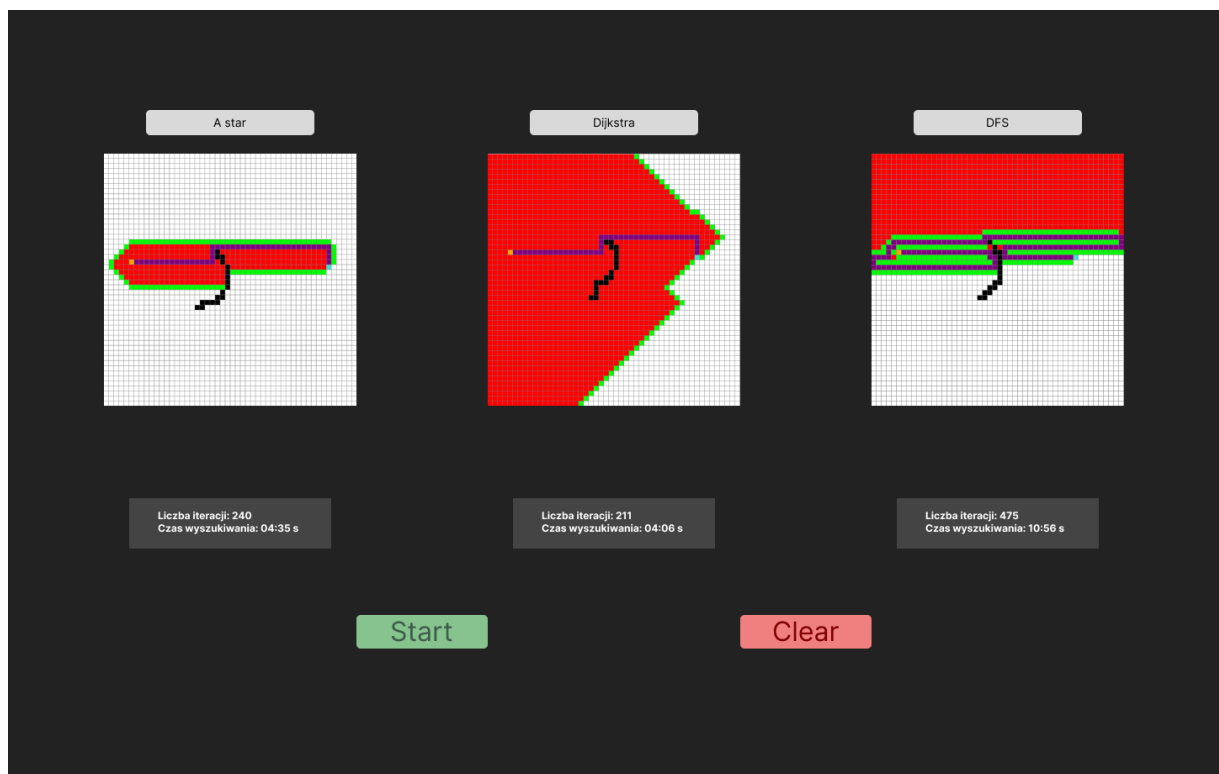
Opis Tematu

Tematem projektu jest wizualizacja algorytmów szukania drogi. Program będzie umożliwiać użytkownikowi rysowanie przeszkód oraz porównywanie czasów wyszukiwania drogi. Backend napisany zostanie w języku c++, który skompilowany zostanie to WASM, natomiast część graficzna napisana będzie przy użyciu HTML, CSS oraz JS.

Interfejs programu oraz funkcje

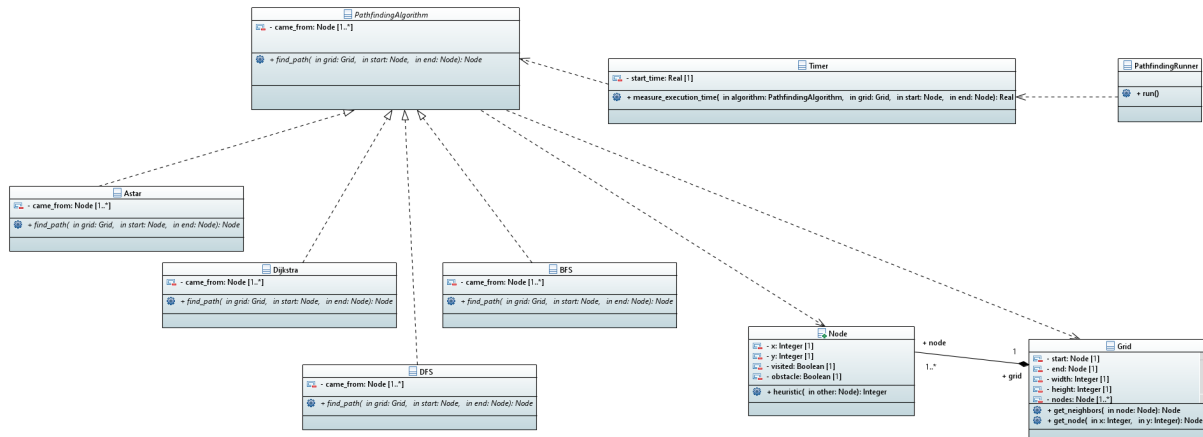
Interfejs będzie składał się z 4 plansz po których użytkownik przy pomocy lewego przycisku myszki będzie mógł rysować przeszkody oraz zaznaczać punkty: początkowy oraz końcowy. Natomiast przy pomocy prawego przycisku myszy możliwe będzie usuwanie przeszkód. Edytowanie którejkolwiek z plansz powodować będzie zmianę na pozostałych. Po naciśnięciu przycisku *start*, na każdej z plansz zostanie wyszukana droga przy użyciu jednego z algorytmów: A*, Dijkstra, BFS, DFS, natomiast po naciśnięciu przycisku *clear* wszystkie plansze zostaną wyczyszczone.

Interfejs użytkownika wykonany zostanie jako strona internetowa z wykorzystaniem HTML, CSS, JS, React oraz Bootstrap. Poniżej zamieszczony został wstępny oraz poglądowy wygląd strony. Zostanie ona w miarę możliwości rozwinięta.



Najważniejsze klasy

Projekt napisany zostanie w paradygmacie obiektowym. Główną częścią projektu są cztery klasy algorytmów wyszukiwania drogi. Dziedziczą one po klasie wirtualnej *PathfindingAlgorithm*. Umożliwiają one znalezienie drogi. Wymagają one dwóch klas: *Node* oraz *Grid*. Ta pierwsza reprezentuje pojedynczą komórkę grafu, natomiast druga cały graf. Mniej znaczące klasy to: *Timer*, *PathfindingRunner*. Pierwsza służy do pomiaru czasu wyszukiwania przez algorytm, natomiast druga służy do wywoływania wszystkich algorytmów wielowątkowo. Diagram w większym formacie zamieszczony jest na ostatniej stronie.



Wykorzystywane tematy laboratoryjne

W projekcie wykorzystane zostaną *modules*. Każda klasa będzie osobnym modułem. Z biblioteki *ranges* wykorzystaną algorytmy operujące na kontenerach STL. Wielowątkowość zostanie zaimplementowana przy użyciu biblioteki *thread* oraz *barrier*.

