

Specyfikacja implementacyjna

Projekt zespołowy

Optymalizacja przewozu pacjentów

Damian Wróblewski
Bartłomiej Łukasik
Karol Kociołek

14 grudnia 2020

1 Informacje ogólne

Program napisany będzie w języku Java
Po uruchomieniu wyświetla GUI do obsługi programu

2 Struktura projektu

Klasy:

- Main - klasa sterująca programem
- GUI - klasa odpowiadająca za graficzny interfejs użytkownika
- Utils - klasa obsługująca wejścia i wyjścia i dodająca potrzebne metody
- AnimationPanel - klasa przedstawiająca animacje
- SettingsPanel - klasa odpowiadająca za interfejs ustawień
- Algorithms - klasa zawierająca algorytmy
- Node - węzeł grafu
- Szpital, Obiekt, Pacjent, Droga - klasy służące do przechwywania informacji o obiektach

3 Struktury danych

- lista - listy zawierające Szpitale, Obiekty, Pacjentów, Drogi
- tablice
- kolejka priorytetowa w postaci kopca - struktura do obsługi algorytmu Dijkstry

4 Algorytmy

- Algorytm tworzący mapę: Algorytm Chana
Wybraliśmy algorytm Chana, ponieważ ze wszystkich znanych algorytmów ma najniższą złożoność wynoszącą $O(n \log h)$ - gdzie h to liczba wierzchołków zewnętrznych
Jest on połączeniem dwóch prostych do implementacji algorytmów o większej złożoności przez co skutkuje mniejszą złożonością (algorytmy Gift wrapping oraz Grahams scan)
- Algorytm szukania skrzyżowań
Dla każdej drogi wyznaczamy funkcje $ax + b$
Dla każdej pary dróg sprawdzamy czy parametr a różni się, jeśli tak wyznaczamy punkt przecięcia i sprawdzamy czy znajduje się w dziedzinie dróg
- Algorytm znajdujący najkrótszą drogę: Zmodyfikowany Algorytm Dijkstry
Wybraliśmy algorytm Dijkstry, ponieważ ze wszystkich znanych algorytmów ma najniższą złożoność wynoszącą $O(n \log n)$
Użyjemy kolejki priorytetowej ponieważ w stosunku do tablicy zmniejsza złożoność z $O(n^2)$ do $O(n \log n)$
W programie algorytm będzie wywoływany dla każdego wierzchołka w którym zatrzyma się pacjent
Algorytm będzie zwracał tablice najkrótszych dróg do szpitali które nie zostały odwiedzone.
Z racji tego, że algorytm oparty jest o graf skrzyżowania traktujemy jak odwiedzone szpitale
W celu optymalizacji czasowej będziemy zapamiętywać dla każdego punktu tablice najkrótszych dróg

5 Testowanie

Testy zostaną wykonane z pomocą narzędzia JUnit.

GUI testowane ręcznie