Politechnika Warszawska

Ambulance Path Advisor - Specyfikacja implementacyjna

Wydział: Elektryczny
Kierunek: Informatyka stosowana
Przedmiot: Algorytmy i struktury danych
Edvin Suchodolskij
Konrad Žilinski
Mateusz Pietrzak

Warszawa 2020 r.

Spis treści:

Opis ogólny 3
Struktura folderów 3
Opis klas 3
Opis algorytmu 3
GUI 4
Scenariusz działania programu 4
Testowanie 4

Opis ogólny

Program służy dla Służby Ochrony Zdrowia, której celem jest pomoc pacjentom dotkniętych nową pandemią, które znajdują się na terenie kraju. Program wyznacza drogę dla karetki pogotowia, czyli kieruje karetką razem z pacjentem do najbliższego ośrodka medycznego, w celu znalezienia najkrótszej drogi do szpitala mającego swobodne łóżko dla pacjenta. W wypadku, gdy w danym szpitalu nie zostało się więcej swobodnych łóżek, program naprawia pacjenta do najbliższego szpitala, w którym jeszcze dany pacjent nie był.

Struktura folderów

W projekcie rozróżniamy 3 główne foldery:

1) src – folder, w którym jest umieszczony cały kod liczący.
2) test – folder, w którym jest umieszczony kod testujący.
3) doc – folder, w którym są umieszczone specyfikacja implementacyjna i funkcjonalna.

Opis klas

W programie jest 7 klas:

- Klasa Hospital – przechowuje informacje o id, nazwie, położeniu, łóżkachszpitala i wszystkich możliwych bezpośrednich drogachwychodzących od tego szpitala do innychobiektów(szpitali lub skrzyżowań).
- Klasa Object – przechowuje informacje o id, nazwie i powożeniuobiektów.
- Klasa Cross– przechowuje informacje o id, położeniu skrzyżowań i wszystkich możliwych bezpośrednich polaczeniach do innych obiektów (szpitali i innych skrzyżowań).
- Klasa Patient – przechowuje aktualne położeniepacjenta.
- Klasa Map – szczytuje dane szpitali, dróg oraz objektów, a także tworzy mapę kraju.
- Klasa FindHospital – znajduje szpital dla pacjenta.
- Klasa Main – zarządza programem, szczytuje oraz wypisuje dane o pacjentach.

Opis algorytmu

W danym projekcie użyjemy algorytmu ***Dijkstry***. Wybraliśmy go, ponieważ pomaga w znalezieniu najkrótszej drogi do każdego objektu, nie sprawdzając wszystkich możliwych dróg. Algorytm wygląda następująco:

1. Program otrzymuje punkt z krótego rozpoczyna obliczenia.
2. Znajduje najkrótszą drogę do sąsiedniego oczka.
3. Przemieszcza się do danego oczka, zapisując jaką pokonał drogę, oraz najkrótszą drogę do konkretnego oczka.
4. Kontynuje kroki 2-3 do momentu, aż nie znajdziemy najkrótszą drogę do każdego oczka.
5. W wypadku, gdyodległość do zapisanych oczek jest równa, program bierze dowolną z wcześniej zapisanychdróg i wraca, aż okarze się w oczku, do którego drogę wybrał.
6. Jeżeli nowa droga do już przeanalizowanego oczka jest dłuższa od dotychczas zapisanej, to zostawiamy krótszą drogę.

GUI

GUI służy do wybrania ścieżki obu plików wejściowych oraz pliku wyjściowego. Pliku z danymi mapy szpitali oraz pliku z pacjentami. Dodatkowo, użytkownik będzie mógł zaobserwować przebieg działania programu w postaci animacji wyświetlanej w oknie.Cykle animacji uruchamiane będą po sobie bądź manualnie. wybraniu wszystkich tych elementów i wciśnięcie przycisku SUBMIT, program stworzy nowe okno, w którym zostanie wyświetlona animacja.

Przebieg animacji - Karetka będzie się przemieszczała do miejsc docelowych w trakcie jednego ustalonego cyklu.

Scenariusz działania programu

Po uruchomieniu GUI i kliknięciu przycisku SUBMIT program poczyna następujące kroki:

1. Sprawdza poprawność pliku wejściowego.
2. Czyta dane.
3. Tworzy mapę składającą się z szpitali i objektów.
4. Szuka potencjalnych skrzyżowań dróg.
5. Sprawdza poprawność pliku wejściowego 2.
6. Pobiera dane o pacjencie.
7. Znajduje nową lub wykorzystuje już znaną drogę do najbliższego szpitala, aż znajdzie wolne miejsce.
8. Zapisuje pokonaną drogę pacjenta do pliku wyjściowego.
9. Powtarza kroki 7-8 dla wszystkich pacjentów.

Po wyczerpaniu pacjentów zostaje wyświetlone repezentacja graficzna pokonanych dróg wszystkich pacjentów.

Testowanie

Do przetestowania będziemy używali narzędzia JUnit 4.0 w celu znalezienia błędów w programie lub udowodnienia, że program działa we właściwy sposób. GUI zostanie przetestowane ręcznie podczas tworzenia aplikacji. Program wykrywa następujące błędy:

- błędną ścieżkę lub brak pliku.
- niepoprawną ilość nagłówków w pliku wejściowym.
- niepoprawne indeksowanie.
- nadmiar elementów w poszczególnych klasach.
- niepoprawny format pojedynczej linii w pliku wejściowym.
- niepoprawny typ elementu w pliku wejściowym.