

Uber Heals - specyfikacja funkcjonalna

Algorytmy i Struktury Danych

Paweł Cegielski, Piotr Szumański, Jakub Matłacz

data utworzenia: 3 grudnia 2020
data ostatniej zmiany: 14 grudnia 2020

1 Wstęp

Program udostępnia mapę obiektów, szpitali oraz dróg pomiędzy nimi. Pozwala wczytać listę pacjentów z pliku lub przy pomocy interfejsu graficznego. Wizualizuje transport tychże osób do najbliższych niepełnych szpitali.

2 Opis ogólny

2.1 Nazwa programu

Nazwa programu to Uber Heals.

2.2 Poruszany problem

Problemem jest optymalna dystrybucja pacjentów, jednocześnie przestrzegając nałożonych ograniczeń takich jak istnienie dróg między lokacjami, ilość wolnych łóżek w szpitalach oraz położenie najbliższego szpitala od lokacji pacjenta.

2.3 Użytkownik docelowy

Program stworzono dla pracowników służby ochrony zdrowia odpowiedzialnych za transport pacjentów.

3 Opis funkcjonalności

3.1 Jak korzystać z programu?

Po uruchomieniu pliku .jar wyświetla się interfejs graficzny z pustą mapą oraz możliwością podania plików wejściowych: danych do naniesienia na mapę oraz listy pacjentów. Następnie można dane te wczytać najpierw tworząc mapę, a następnie, z zachowaniem kolejności, automatycznie dodawać kolejnych pacjentów. Ponadto użytkownik może dodać pacjenta na określonych współrzędnych z poziomu interfejsu. Można także zresetować pacjentów poprzez reset mapy.

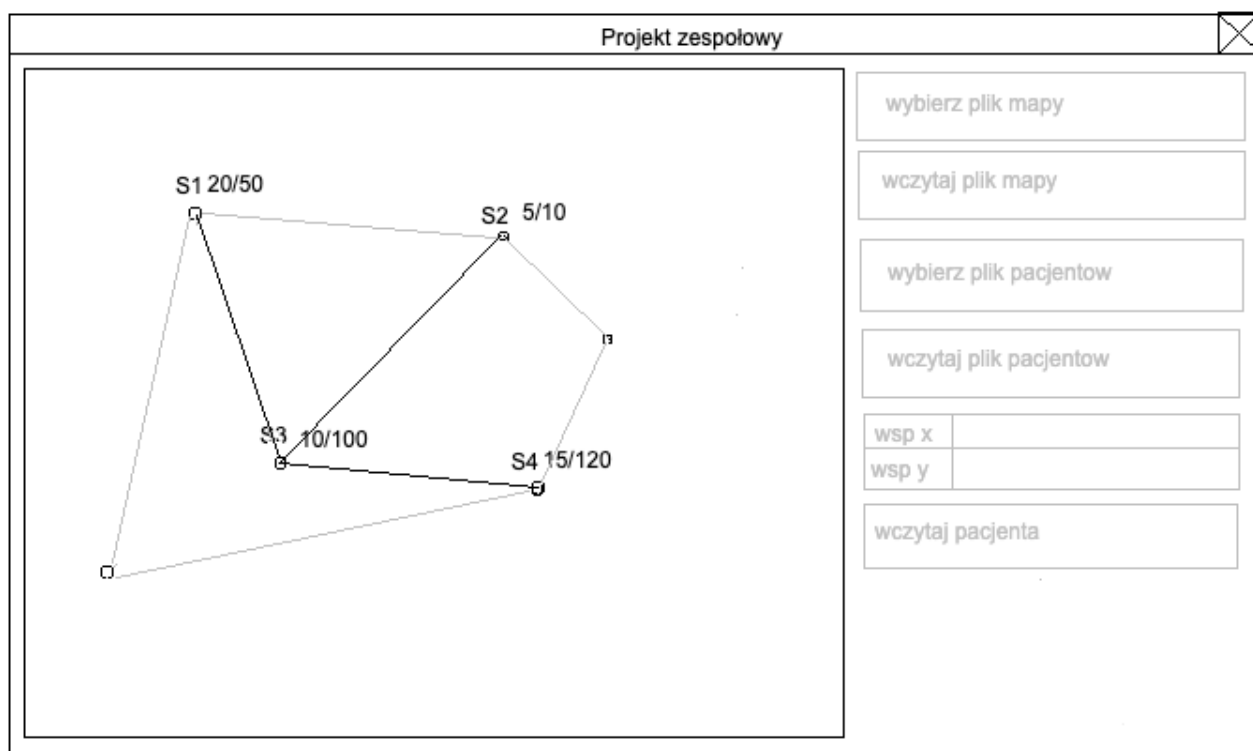
3.2 Uruchomienie programu

Należy dwukrotnie kliknąć w plik .jar i czekać na otwarcie okna z GUI.

3.3 Możliwości programu

- Wczytanie mapy z pliku.
- Wczytanie listy pacjentów z pliku.
- Stwierdzenie poprawności wczytanych danych.

- Stwierdzenie spójności i logiczności wczytanych danych.
- Wygenerowanie pliku z logami.
- Obliczenie dystrybucji pacjentów do szpitali.
- Wizualizacja dystrybucji pacjentów.
- Wyświetlanie intuicyjnych komunikatów błędów.



Rysunek 1: Schematyczny wygląd okna programu

4 Format danych i struktura plików

Pliki wejściowe powinny mieć rozszerzenie .txt. Logi programu mają to samo rozszerzenie co pliki wejściowe.

4.1 Struktura katalogów

Taka jak na Rysunku 2.

4.2 Dane wejściowe

- Dane mapy

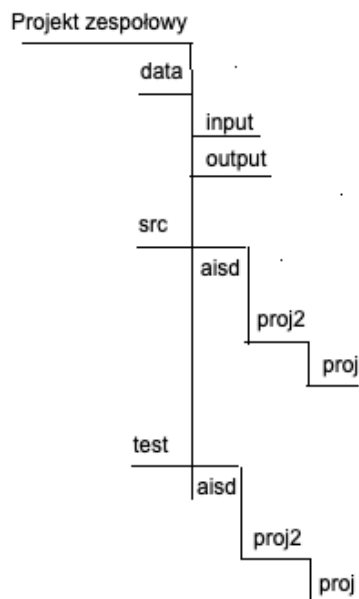
```
# Szpitale (id | nazwa | wsp. x | wsp. y | Liczba łóżek | Liczba wolnych łóżek)
1 | Szpital Wojewódzki nr 997 | 10 | 10 | 1000 | 100
2 | Krakowski Szpital Kliniczny | 100 | 120 | 999 | 99
3 | Pierwszy Szpital im. Prezesa RP | 120 | 130 | 99 | 0
4 | Drugi Szpital im. Naczelnika RP | 10 | 140 | 70 | 1
5 | Trzeci Szpital im. Króla RP | 140 | 10 | 996 | 0
```

```
# Obiekty (id | nazwa | wsp. x | wsp. y)
1 | Pomnik Wikipedii | -1 | 50
2 | Pomnik Fryderyka Chopina | 110 | 55
3 | Pomnik Anonimowego Przechodnia | 40 | 70

# Drogi (id | id_szpitala | id_szpitala | odległość)
1 | 1 | 2 | 700
2 | 1 | 4 | 550
3 | 1 | 5 | 800
4 | 2 | 3 | 300
5 | 2 | 4 | 550
6 | 3 | 5 | 600
7 | 4 | 5 | 750
```

- Dane pacjentów

```
# Pacjenci (id | wsp. x | wsp.y)
1 | 20 | 20
2 | 99 | 105
3 | 23 | 40
```



Rysunek 2: Struktura katalogów

4.3 Dane wyjściowe

Dane wyjściowe będą przedstawiać wszystkie zdarzenia, zachodzące w procesie dystrybucji pacjentów, z zachowaniem kolejności. Ich przeznaczeniem jest umożliwienie odtworzenia ciągu zdarzeń w programie.

5 Scenariusz działania programu

5.1 Scenariusz ogólny

1. Uruchomienie pliku .jar.
2. Pojawienie się okna z GUI.

3. Wczytanie mapy z pliku.
4. Wczytanie pacjentów z pliku lub ręcznie.
5. Animacja dystrybucji pacjentów.
6. Ewentualne zresetowanie mapy w trakcie.
7. Zakończenie animacji.
8. Zamknięcie okna aplikacji przez użytkownika.

6 Środowisko

Do uruchomienia potrzebny jest system operacyjny Windows w wersji 10. Ponadto należy mieć zainstalowaną najnowszą Javę (wersja 8).

7 Testowanie

Program podlegał testowaniu przy użyciu biblioteki JUnit 4.