

AmbulanceServices - specyfikacja funkcjonalna

Aleksandra Kowalczyk, Kacper Seredyn, Kacper Achramowicz

Grudzień 2020

1 Opis tematyki projektu

1.1 Nazwa programu

Program obsługi kursów Karetok Pogotowia (KP) do pacjentów na terenie całego kraju, a następnie do Ośrodków Medycznych(OM).

1.2 Poruszany problem

Do programu dostarczana jest mapa zawierająca informacje o szpitalach OM, ilości dostępnych łóżek LŁ i drogach między szpitalami. Dystrybutor Pogotowia DP dostarcza kolejno informacje o pacjentach potrzebujących pomocy. Program sprawdza czy pacjent P znajduje się na terytorium kraju. Karetka Pogotowia KP nie przyjmuje pacjentów spoza granic. Karetka ma za zadanie zawieźć pacjenta do najbliższego szpitala. W przypadku gdy szpital nie ma już wolnych łóżek, karetka przewozi pacjenta do kolejnego szpitala. Czynność jest powtarzana dopóki pacjent nie zostanie przyjęty do szpitala lub karetka pogotowia sprawdzi wszystkie możliwe szpitale i żaden nie będzie w stanie przyjąć pacjenta.

1.3 Użytkownik docelowy

Użytkownikiem docelowym jest Dyspozytor Pogotowia (DP) który zarządza Karetkami Pogotowia KP z Służby Ochrony Zdrowia.

2 Opis funkcjonalności

2.1 Środowiska konieczne do uruchomienia programu

Niezbędne jest zainstalowanie na sprzęcie, na którym program będzie uruchamiany oprogramowania JDK (Java Development Kit).

2.2 Uruchamianie programu

Program przechowywany będzie w pliku .jar. Taki format pozwala przekazywać skompilowany już kod. Program będzie można uruchomić z terminala następującym poleceniem:

```
java -jar AmbulanceServices.jar
```

2.3 Jak korzystać z programu?

Po uruchomieniu programu wczytujemy do niego mapę kraju. Następnie kolejno wczytujemy pacjenta. Program umożliwia śledzenie drogi pacjenta do szpitala/szpitali w poszukiwaniu wolnych łóżek. Po umiejscowieniu pacjenta w szpitalu lub sprawdzeniu wszystkich możliwych opcji możliwe jest dodanie kolejnego pacjenta.

3 Format danych i struktura plików

3.1 Słownik

- plik mapy - plik z informacjami na temat szpitali, obiektów rozszerzających mapę oraz dróg
- DP - dystrybutor pogotowia
- KP - karetki pogotowia
- OM - ośrodki medyczne (szpitale)
- LL - liczba dostępnych łóżek
- obiekty- punkty na mapie, które służą jedynie wyznaczeniu granic kraju, nie pełnią żadnych funkcji użytecznych dla DP

3.2 Struktura katalogów

Nasz projekt zawiera się w czterech katalogach:

- src - katalog przechowujący kod źródłowy
- tests - katalog przechowujący testy jednostkowe
- icons - katalog przechowujący ikony wykorzystywane w GUI
- data - katalog, w którym znajdują się przykładowe pliki z danymi

Plik wykonywalny *.jar znajdować się będzie w katalogu zawierającym powyższe katalogi.

3.3 Przechowywanie danych w programie

W programie będziemy przechowywać dane szpitali, pacjentów oraz połączeń między szpitalami.

Informacje o szpitalach przechowywane są w liście obiektów szpitali, a dane pacjentów w liście obiektów pacjentów. Połączenia między szpitalami umieszczone są w grafie połączeń.

3.4 Dane wyjściowe

Program nie zwraca żadnych danych wyjściowych, wynikiem jego działania są zmiany GUI.

4 Dane wejściowe

4.1 Plik mapa

4.1.1 Szpitale

Plik zawiera informacje o szpitalach oddzielone od siebie "|" w następującym formacie:

id | nazwa | wsp. x | wsp. y | Liczba łóżek | Liczba wolnych łóżek.

Program będzie wyświetlał komunikat o błędzie, gdy dane nie będą spełniały podanych założeń:

- id szpitali muszą być unikalnymi liczbami całkowitymi dodatnimi
- nazwa nie może zawierać "|"
- Współrzędna x musi być liczbą wymierną
- Współrzędna y musi być liczbą wymierną
- nie może istnieć szpital, którego współrzędne x i y są takie same jak dla innego szpitala
- liczba łóżek musi być liczbą całkowitą dodatnią
- liczba wolnych łóżek musi być liczbą całkowitą dodatnią

4.1.2 Obiekty

Plik zawiera informacje o obiektach oddzielone od siebie "|" w następującym formacie:

id | nazwa | wsp. x | wsp. y

Program będzie wyświetlał komunikat o błędzie, gdy dane nie będą spełniały podanych założeń:

- id obiektów muszą być unikalnymi liczbami całkowitymi dodatnimi
- nazwa nie może zawierać "|"
- Współrzędna x musi być liczbą wymierną
- Współrzędna y musi być liczbą wymierną

4.1.3 Drogi

Plik zawiera informacje o drogach między szpitalami oddzielone od siebie "|" w następującym formacie:

id | id_szpitala | id_szpitala | odległość

Program będzie wyświetlał komunikat o błędzie, gdy dane nie będą spełniały podanych założeń:

- id dróg muszą być unikalnymi liczbami całkowitymi dodatnimi
- id_szpitala musi być liczbą całkowitą dodatnią, taką, którą oznaczony był jakiś szpital w sekcji szpitale
- powinno istnieć tylko jedno połączenie między danymi szpitalami
- nie może istnieć połączenie między jednym, a tym samym szpitalem
- odległość musi być liczbą wymierną dodatnią

4.2 Pacjenci

4.2.1 Plik pacjenci

Plik zawiera informacje o pacjentach oddzielone od siebie "|" w następującym formacie:

id | wsp. x | wsp.y

4.2.2 Pacjenci dodawani manualnie

Program umożliwia dodanie pacjenta, poprzez wybranie prawym przyciskiem myszki dowolnego punktu na mapie.

4.2.3 Komunikaty błędów

Program będzie wyświetlał komunikat o błędzie, gdy dane nie będą spełniały podanych założeń:

- id pacjentów muszą być unikalnymi liczbami całkowitymi dodatnimi
- Współrzędna x musi być liczbą wymierną
- Współrzędna y musi być liczbą wymierną

Przykładowy plik mapy wygląda w poniższy sposób:

```
# Szpitale (id | nazwa | wsp. x | wsp. y | Liczba łózek | Liczba wolnych  
łózek)  
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 | idSzpitala | idSzpitala | 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
```

Przykładowy plik pacjenci wygląda w poniższy sposób:

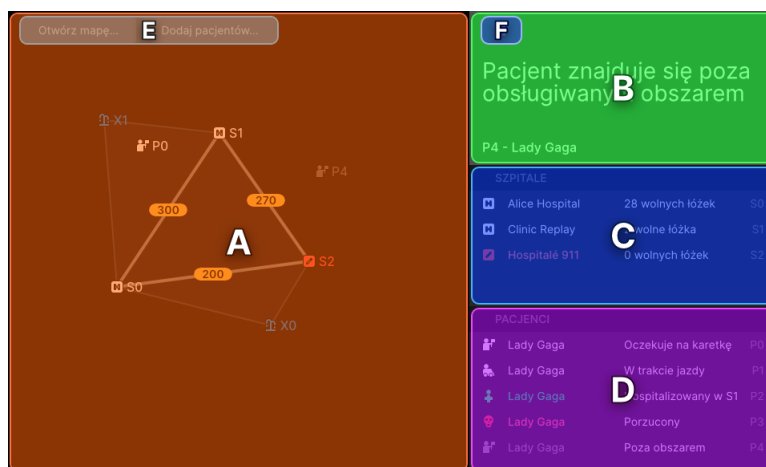
```
# Pacjenci (id | wsp. x | wsp.y)  
1 | 20 | 20  
2 | 99 | 105  
3 | 23 | 40
```

5 Działanie programu

5.1 Schemat okna programu

5.1.1 Ogólny podział elementów interfejsu

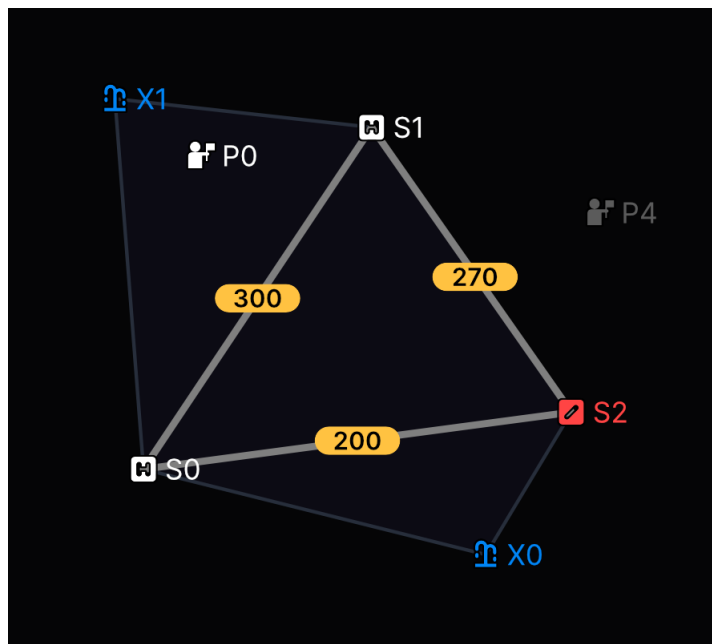
Główny interfejs programu jest podzielony na następujące sekcje:



- A - mapa szpitali, obiektów i położenia pacjentów,
- B - sekcja informacyjna,
- C - lista szpitali,
- D - lista pacjentów,
- E - przyciski ładowania plików,
- F - przycisk kontroli symulacji ("Uruchom"/"Wstrzymaj")

5.1.2 Mapa

Mapa przedstawia następujące informacje:

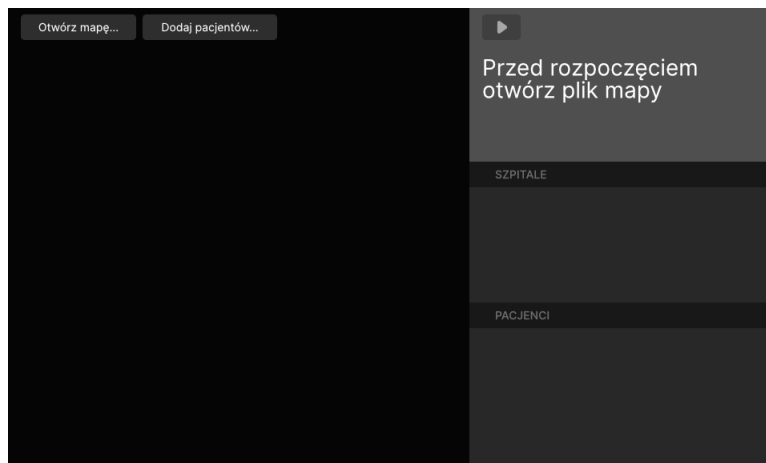


- Szpitale (oznaczone literą S),
- Pacjenci (oznaczeni literą P),
- Inne obiekty (oznaczone literą X),
- Drogi między szpitalami (gruba szara linia),
- Obsługiwany region (ciemnoszara linia oraz wypełnienie),
- Długości odcinków dróg (żółte bloki)

Kolory niektórych obiektów przekazują dodatkowe informacje: czerwone szpitale nie mogą przyjmować więcej chorych, wyszarzeni pacjenci znajdują się poza obszarem.

Użytkownik może zmieniać widoczny na mapie region przesuwając widok (ruch myszki z naciśniętym prawym przyciskiem) lub zwiększając i zmniejszając widoczny obszar (Ctrl+kółko w górę, Ctrl+kółko w dół).

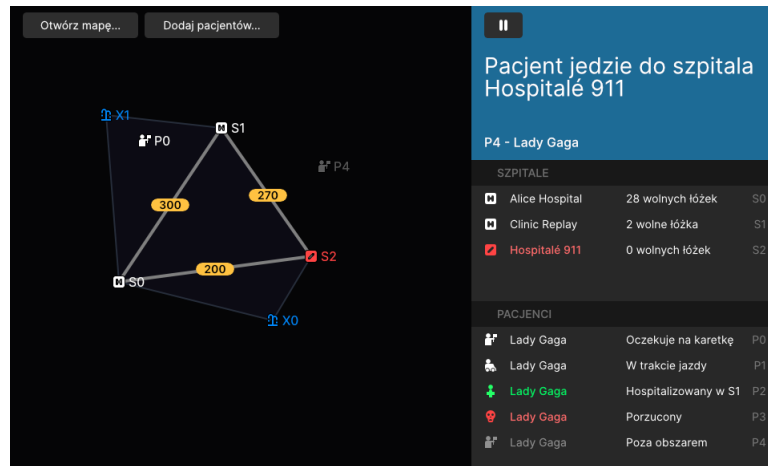
5.2 Scenariusz działania



1. Dostarczenie pliku mapy oraz pacjentów.
Program nie może rozpocząć symulacji bez obecności odpowiednich danych. Użytkownik musi wprowadzić do systemu:
 - Plik mapy przez wciśnięcie przycisku "Otwórz mapę..." i wybranie odpowiedniego pliku,
 - Dane pacjentów przez wciśnięcie przycisku "Dodaj pacjentów..." i wybranie odpowiedniego pliku, bądź kliknięcie lewym przyciskiem myszy na mapie, powodując dodanie pacjenta w punkcie pod wskaźnikiem.

W przypadku, gdy wybrany plik wejściowy (mapy bądź pacjentów) jest niepoprawnie sformatowany, program wyświetli odpowiedni komunikat w sekcji informacyjnej:

- *Niepoprawny plik mapy* - podany plik mapy jest niepoprawnie sformatowany bądź zawiera błędy.
- *Niepoprawny plik pacjentów* - podany plik danych pacjentów jest niepoprawnie sformatowany bądź zawiera błędy.



2. Rozpoczęcie symulacji.

Jeśli wprowadzone dane były poprawne oraz kompletne, użytkownik może rozpocząć symulację przy użyciu przycisku "Uruchom". W trakcie trwania, ponowne wciśnięcie przycisku powoduje wstrzymanie działania algorytmu.

Algorytm w trakcie trwania będzie przedstawiał kolejne stany działania zarówno na mapie, jak i w sekcji informacyjnej:

- *Pacjent jedzie do szpitala X*
- *W szpitalu X brakuje łózek*
- *Pacjent został przyjęty w X*
- *Pacjent znajduje się poza obsługiwanym obszarem*
- *Pacjent został porzucony*

6 Testowanie

Do testów jednostkowych wykorzystamy narzędzie JUnit 5, natomiast GUI będziemy sprawdzać w trakcie pisania programu.