

Specyfikacja funkcjonalna projektu pt.
„Patients Transport Center”

Wykonali: Antoni Malinowski, Franciszek Wysocki, Bartosz Zakrzewski
Sprawdzający: mgr inż. Paweł Zawadzki
Data: 17-12-2020

Spis treści

1	Cel dokumentu	3
2	Cel projektu	3
3	Scenariusz uruchomienia	4
4	Opis danych wejściowych	6
4.1	Plik pierwszy	6
4.2	Plik drugi	7
4.3	Struktura wierszy w sekcji	8
5	Komunikaty o błędach	9
6	Źródła	9

1 Cel dokumentu

Celem dokumentu jest przedstawienie funkcjonalności programu, który zarządza transportem pacjentów do szpitali.

Zostanie w nim zaprezentowane w jaki sposób aplikacja powinna być uruchamiana, jak mają wyglądać pliki wejściowe i jak będzie wyglądać graficzny interfejs użytkownika.

Zostanie opisane jak zachowa się program w sytuacjach niepożądanych i co możemy do takich sytuacji zaliczyć.

2 Cel projektu

Celem projektu jest symulacja w jaki sposób pacjenci mają być przewożeni do najbliższych szpitali. Program określi:

- czy pacjent znajduje się na terenie obsługiwanym przez karetki pogotowia;
- który szpital jest najbliższy;
- czy są wolne łóżka w szpitalu - jeżeli nie, to przetransportuje pacjenta do kolejnego najbliższego szpitala.

3 Scenariusz uruchomienia

1. Użytkownik uruchamia aplikację.

Pierwszy sposób uruchomienia:

- (a) Użytkownik uruchamia aplikację „patients-transport-center.jar” poprzez dwukrotne kliknięcie na jej ikonę lewym przyciskiem myszy;
- (b) Użytkownik ustawia wstępną konfigurację (załącza „plik z mapą” i „plik z pacjentami”).

Drugi sposób uruchomienia:

- (a) Użytkownik uruchamia program z poziomu terminala za pomocą:

```
java -jar patients-transport-center.jar
```

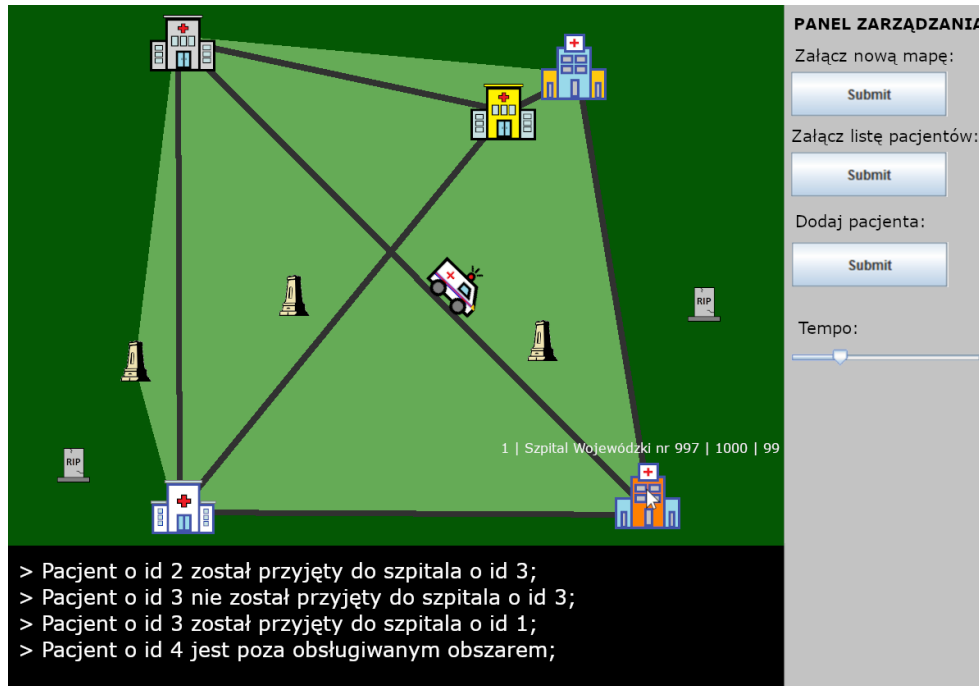
- (b) Użytkownik może podać nazwę pliku z obiektami, szpitalami i drogami oraz nazwę pliku z pacjentami, poprzez flagi „-h” i „-p”:

```
java -jar patients-transport-center.jar  
-h nazwa-pliku-ze-szpitalami-obiektami-i-drogami  
-p nazwa-pliku-z-pacjentami
```

Jeżeli jakaś flaga nie zostanie podana, użytkownik będzie musiał wczytać pliki z poziomu interfejsu użytkownika.

- 2. Użytkownik uruchamia symulację, w której na bieżąco obliczane są optymalne rozwiązania dla transportu pacjentów z pliku wejściowego;
- 3. Użytkownik może wstawić kolejnego pacjenta na mapie za pomocą prawego przycisku myszki lub w panelu zarządzania;
- 4. Użytkownik może przyspieszyć/spowolnić symulację w panelu zarządzania;
- 5. Użytkownik może zmienić plik z mapą/listą nowych pacjentów w panelu zarządzania;
- 6. Użytkownik może odczytać informacje o danym obiekcie naciśkając na niego lewym przyciskiem myszki.

Wizualizacja aplikacji stworzona za pomocą programu GIMP 2:



Rysunek 1: Wygląd programu

Mapa będzie reprezentowała obiekty zgodnie z ich położeniem w układzie współrzędnych. Pod nią znajdować się będzie panel z wyświetlanymi komunikatami, a obok niej panel zarządzania.

Transport pacjentów będzie odbywać się „pacjent po pacjencie”, aby czytelniej przedstawić symulację.

4 Opis danych wejściowych

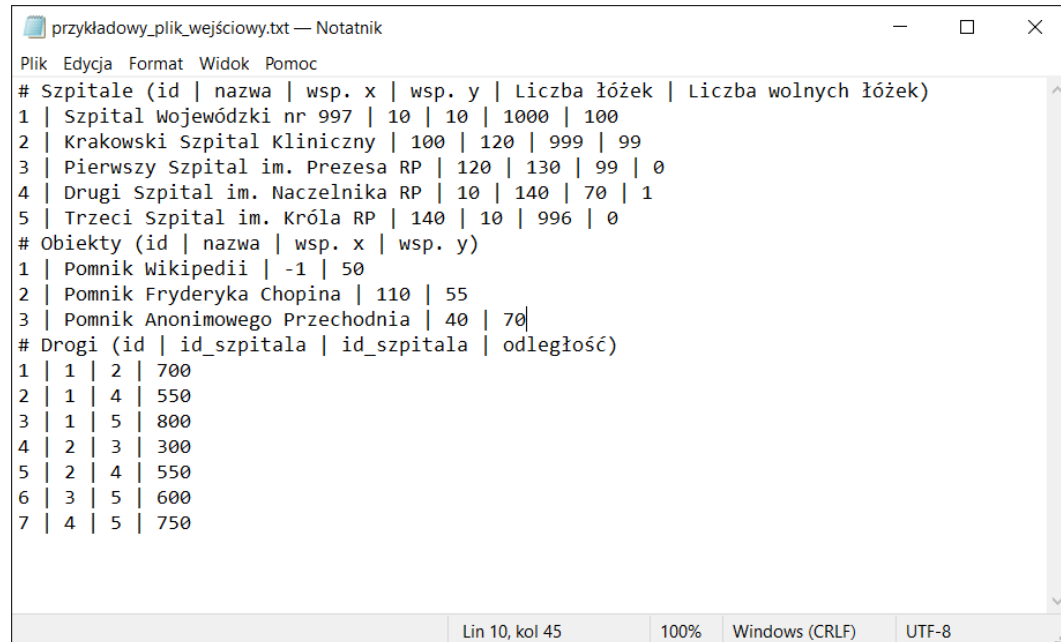
Program, jak już to wcześniej opisano w sekcji **Scenariusz uruchomienia**, wymagać będzie dostarczenia **2 plików wejściowych** (muszą być to plik tekstowe).

4.1 Plik pierwszy

Pierwszy plik wejściowy będzie się składać z następujących **trzech sekcji**:

- Szpitale (id | nazwa | wsp. x | wsp. y | liczba łóżek | liczba wolnych łóżek);
- Obiekty (id | nazwa | wsp. x | wsp. y);
- Drogi (id | id_szpitala | id_szpitala | odległość).

Nazwę każdej sekcji należy poprzedzić znakiem „#”. Każda sekcja musi zawierać wiersze, w których zapisane będą surowe dane wykorzystywane przez program. Wiersze w poszczególnych sekcjach muszą posiadać odpowiednią strukturę, zdefiniowaną w nawiasie po nazwie sekcji.



```
przykładowy_plik_wejściowy.txt — Notatnik
Plik  Edycja  Format  Widok  Pomoc
# 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
Lin 10, kol 45    100%    Windows (CRLF)    UTF-8
```

Rysunek 2: Przykładowy plik wejściowy

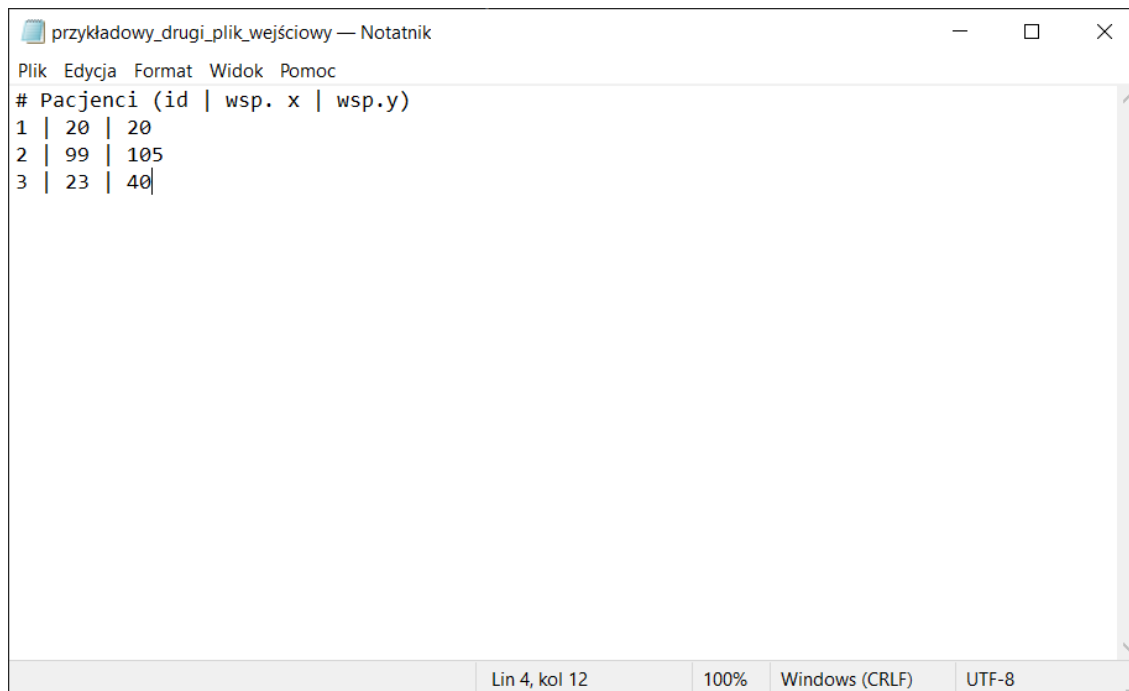
4.2 Plik drugi

Drugi plik wejściowy to lista zawierająca **pacjentów i ich współrzędne**. Pacjenci będą obsługiwani w kolejności w jakiej są w pliku. Po przetransportowaniu wszystkich pacjentów z pliku, zostaną po kolei transportowani pacjenci dodani z poziomu aplikacji.

W pliku ma znajdować się jedna sekcja:

- Pacjenci (id | wsp. x | wsp.y).

Tak jak w przypadku pierwszego pliku, nazwa sekcji musi być poprzedzona znakiem „#”, a wiersze muszą posiadać strukturę widoczną w nawiasach.



The screenshot shows a Notepad window titled "przykładowy_drugi_plik_wejściowy — Notatnik". The menu bar includes "Plik", "Edycja", "Format", "Widok", and "Pomoc". The text content is as follows:

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

The status bar at the bottom indicates "Lin 4, kol 12", "100%", "Windows (CRLF)", and "UTF-8".

Rysunek 3: Przykładowy drugi plik wejściowy

4.3 Struktura wierszy w sekcji

Każdy wiersz niebędący nazwą sekcji musi posiadać strukturę określoną po nazwie sekcji, w której się znajduje. Opis struktury zawiera etykiety, które informują o tym, co reprezentować będą dane w poszczególnych polach wiersza.

W sekcji „**Szpital**” i „**Obiekty**” w kolumnie „**nazwa**” (w kolejnych wierszach) musimy wpisać słowo (nazwa może być kilku członowa, ale nie może zawierać znaku „|”).

Dla reszty pól, w wierszu należy wpisać **liczbę całkowitą** większą bądź równą 0 (wyjątkiem są tu pola w kolumnach **wsp.x** i **wsp.y**, gdzie wartościami mogą być dowolne liczby całkowite).

Poszczególne wartości w wierszu muszą być oddzielone symbolem „|” (spacja + znak strumienia + spacja).

Sytuacje niedopuszczalne: Sytuacje, dla których program nie będzie mógł poprawnie odczytać zawartości podanych plików wejściowych to:

- niepoprawne odseparowanie od siebie wartości w poszczególnych wierszach;
- niepoprawna kolejność sekcji (np. zamiana miejscami sekcji szpitali z sekcją obiektów);
- brak znaku „#” przed nazwą sekcji;
- niepoprawnie podane wartości (liczby z wartością po przecinku, liczby całkowite mniejsze od zera dla pól innych niż wartości współrzędnych).

5 Komunikaty o błędach

W razie wystąpienia błędu, wyświetli się komunikat w wyskakującym oknie (popup). Niektóre błędy będą zmuszały użytkownika do zakończenia działania aplikacji (np. brak obrazu karetki), inne będą wskazywały błąd i oczekiwały naprawy (np. błąd w pliku wejściowym).

Komunikaty będą pojawiały się między innymi podczas:

- błędnej walidacji pliku;
- nieodnalezienia pliku;
- nieodnalezienia obrazów/sprite'ów obiektów.

6 Źródła

- Przykładowe dane wejściowe przygotował mgr inż. Paweł Zawadzki;
- Rysunek poglądowy został stworzony za pomocą programów: GIMP 2, Paint, Paint 3D;
- Ten dokument został stworzony za pomocą strony overleaf.com.