GIS PLAN REALIZACJI

Kamil Dębski Michał Przyłuski

13 stycznia 2009

Projekt zostanie zrealizowany w oparciu o mapę Warszawy z zaczerpniętą z opensourceowego projektu[1]. Mapa ta udostępnia wszystkie ulice Warszawy oraz najbliższych okolic, w postaci plików tekstowych — umożlwiając zatem ich wykorzystanie w naszym projekcie.

Zastosowany zostanie algorytm Dijkstry do wyszukania najlepszej ścieżki w grafie, pomiędzy zadanymi przez użytkownika wierzchołkami.

1 Baza danych

Planujemy korzystać z mapy Warszawy. Struktura danych wejsciowych jest następująca:

- nazwa ulicy
- współrzędne geograficzne łamanej opisującej jej położenie — najkrótszy opis składa się z dwóch punktów — początku oraz końca ulicy.
- pewne informacje pomocnicze jak kategoria, które umożliwią oszacowanie *czasu* przejazdu, oraz posłuża do wizualizacji
- Na podstawie współrzędnych geograficznych zostaną wyznaczone odległości (w metrach) pomiędzy kolejnymi węzłami danej ulicy, które posłużą jako wagi dla jedengo z dwóch systemów wag do zaimplementowania w projekcie.

Mapa zostanie reprezentowana w postaci grafu, wierzchołkami będą punkty występujące na mapie Warszawy. Odpowiada to faktycznym skrzyżowaniom, bądź zmianom kierunku drogi (z założenia droga (fizyczna) między dwoma punktami w świecie rzeczywistym będzie zakrzywiona; a wszelkie zakrzywienia są na mapie symulowane przez łamaną składającą sie z wielu odcinków).

Każda krawędź odpowiada zatem fragmentowi (lub całej) pewnej ulicy.

Warto dodać, że takie przedstawienie ulic, jako niezależnych od siebie odcinków niesie ze sobą liczne korzyści. Podstawowym ułatwieniem jest należyte oddanie ulic, których kategoria (jakość, ranga) zmiania się wraz z jej biegiem, a to bezpośrednio wpływa na szacowany czas przejazdu. Dzięki takiemu podziałowi możemy lepiej oddać rzeczywistość.

2 Struktura danych

Graf zostanie reprezentowany w postaci listy następstwa. Jednakże, standardowa implementacja tej struktury nie jest wystarczająca do naszych celów, gdyż musimy przechowywać pewne informacje na krawędziach (nazwę ulicy, długość/czas przejazdu przez tę krawędź). Z tego względu zostanie zastosowana uogólniona lista incydencji — dla każdego wierzchołka przechowywana będzie lista objektów reprezentujących krawędzie do niego incydentne. Obiekty te będą zawierać odpowiednie informacje, niezbędne do poprawnego działania algorytmu.

Konieczne jest także przechowywanie pewnych zmiennych pomocniczych w wierzchołkach — na potrzeby algorytmu przeszukującego.

Pomocnicze struktury: Vertex i Edge. Vertex będzie przechowywać dwie współrzędne geograficzne wierzchołka, oraz listę obiektów Edge z nim związanych. Obiekt Edge będzie zawierać wymienione wyżej pola (nazwę ulicy, długość/czas przejazdu, rangę ulicy — pomocne do wizualizacji grafu, i inne jeśli zajdzie taka potrzeba).

Główna struktura, Graf, powinna udostępniać następujące metody:

- Dodaj_Ulicę(nazwa, ranga, wskaźnik na listę wierzchołków) — metoda ta, doda odpowiednią ulicę do grafu, poprzez dodanie wierzchołków do grafu (jeśli jakieś wierzchołki już występują w grafie — nie zostaną zduplikowane) oraz utworzenie nowych krawędzi je łączących o odpowiednich wartościach pól informacyjnych.
- Iterator — iterator umożliwiający iterowanie się po wierzchołkach, każdy wierzchołek zostanie zwrócony dokładnie raz.
- Szukaj_Ścieżki(Vertex początkowy, Vertex końcowy, bool wagi) — metoda, która wyszuka ścieżkę (według jednego z dwóch systemów wag), oraz zwróci listę wierzchołków, przez które ona przebiega.
- Pobierz_Listę_Ulic() — zwraca listę, w której każda nazwa ulicy występuje dokładnie jeden raz.

3 Algorytm

Zastosowany zostanie algorytm Dijkstry[3]. Ze wstępnej analizy problemu wynika, że takie rozwiązanie będzie wystarczająco szybkie. Dla grafu o ok. 30000 wierzchołków, w przypadku poszukiwania ścieżki pomiędzy wierzchołkami umiarkowanie oddalonymi, algorytm potrzebuje poniżej 7 sekund, aby zwrócić odpowiedź (na procesorze ponad rocznym 2.4GHz).

4 Przebieg wykonania

Projektowany przebieg wykonania programu jest następujący:

1. Program ładuje plik z bazą danych o mapie. Dla każdego wpisu wywołując metodę dodającą ulicę do grafu.
2. Procedura wizualizująca graf iteruje się przez wszystkie wierzchołki, rysując mapę.

3. Użytkownik wybiera z listy ulic jakiej trasy szuka, oraz jakie są jego kryteria. Możliwe będzie zarówno wybranie ulicy na mapie (udostępniającej operacje przesuwania oraz przybliżania widoku), a także korzystając z listy podpowiedzi z nazwami ulic, która przeniesie widok mapy do interesującej użytkownika ulicy.

4. Metoda SZUKAJ_ŚCIEŻKI zwraca listę wierzchołków, które zostają zwizualizowane na mapie. Jednocześnie, dodatkowa informacja o szacowanym czasie przejazdu oraz długości (w kilometrach) trasy, jest wyświetlana użytkownikowi. Użytkownik zostanie poinformowany o przebiegu trasy — lista ulic, oraz czas i dystans, który musi każdą z nich przebyć. Ponadto, obliczona trasa zostanie zaznaczona odmiennym kolorem na mapie.

Literatura

[1] Projekt *mapa prawie całej Polski*: WWW.UMP.WAW.PL

[2] Stephen C. Perry: *Core C# i .NET*, Helion 2006

[3] Thoman H. Corman, et al: *Introduction to Algorithms*, The MIT Press, 2001