

Zaawansowane programowanie w C++

Prosty symulator ruchu miejskiego

Dokumentacja wstępna

*Bogdan Shkola
Andrii Khmelovskyi
Wojciech Ceret*

21.10.2014 r.

Cel:

Celem projektu jest zaprojektowanie oraz implementacja prostego symulatora ruchu drogowego w mieście. Program ma pozwalać na stworzenie prostego modelu miasta złożonego z kilku ulic, po których będą poruszać się pojazdy oraz piesi, a także na rejestrację stanu modelowanego układu. Informacje zebrane w trakcie działania będą mogły posłużyć jako dane wejściowe innych systemów.

Projekt zakłada stworzenie graficznego interfejsu użytkownika.

Funkcjonalność aplikacji:

- 1) Edytor schematu miasta:
 - a) wyznaczanie ulic,
 - b) rozmieszczanie samochodów (2 typy) oraz pieszych,
 - c) definiowanie punktu docelowego dla każdego samochodu oraz pieszego,
 - d) rozmieszczanie oraz konfiguracja kamer,
 - e) zapisywanie wyników pracy do pliku.
- 2) Symulacja ruchu miejskiego w czasie rzeczywistym.
- 3) Automatyczne znalezienie najkrótszej drogi dla każdego z samochodów i pieszych.
- 4) Wykrywanie obiektu przez kamerę wraz z zapisem informacji o zdarzeniu w bazie danych.
- 5) Odczyt informacji z bazy danych.
- 6) Prezentacja stanu aplikacji oraz wyników symulacji za pośrednictwem graficznego interfejsu użytkownika.

Opis aplikacji:

Aplikacja będzie realizować funkcję symulatora ruchu miejskiego. Założeniem jest modelowanie ruchu dwóch typów samochodów oraz pieszych. Modelowany ruch drogowy będzie obserwowany przez kamery monitoringu miejskiego. Dane pochodzące z obserwacji będą zapisywane w bazie danych, co umożliwiać będzie ich późniejsze przetwarzanie przez inne, kompatybilne systemy.

Miasto, na planie siatki kwadratowej, składać się będzie z prostych ulic, przecinających się pod kątem prostym. Projektowanie miasta zakłada umieszczanie fragmentów drogi na kolejnych kwadratach siatki. Na podstawie rodzaju sąsiednich elementów program będzie automatycznie wybierał typ drogi. Istnieć będą 4 typy dróg: prosta, skrzyżowanie typu X, skrzyżowanie typu T, ślepa uliczka. Każda z ulic składać się będzie z dwupasmowej jezdni (po jednym pasie dla każdego kierunku) oraz z chodnika po każdej stronie jezdni. Przy każdym skrzyżowaniu będą istnieć przejścia dla pieszych.

Po mieście będą poruszać się dwa typy pojazdów oraz piesi. Wszystkie obiekty będą korzystać z przeznaczonych dla nich pasów drogi (samochody z jezdni, piesi z chodnika). Każdy będzie miał ręcznie określony cel przejazdu/przejścia, natomiast trasa będzie wyznaczana automatycznie. Typy pojazdów będą charakteryzować się różnymi średnimi prędkościami. Zakłada się dynamiczne dostosowanie prędkości ruchu do położenia na drodze (np. zwalnianie przed skrzyżowaniem). Projekt nie zakłada implementacji wykrywania kolizji - pojazdy i piesi będą się przenikać.

W mieście mogą zostać zainstalowane kamery monitoringu miejskiego. Każda z kamer obserwuje ruch drogowy i co określony czas generuje informację o zarejestrowanych obiektach (współrzędne, typ, odległość). Kamera posiada unikatowe właściwości takie, jak położenie, kierunek obserwacji, kąt widzenia i zasięg. Zakłada się, że im dalej od kamery będzie znajdował się zarejestrowany obiekt, tym mniej dokładne będą zebrane informacje. Jeżeli obiekt pozostanie poza zasięgiem kamery, nie zostanie zarejestrowany.

Wszystkie dane zbierane przez system monitoringu miejskiego będą składowane w bazie danych.

Interfejs aplikacji będzie składał się z trzech elementów: edytora miasta, symulatora oraz prostej przeglądarki rekordów w bazie danych.

Edytor miasta będzie udostępniał możliwość stworzenia planu ulic, ustawienia pojazdów i pieszych wraz z określeniem celów przejazdów/przejęć oraz konfigurację i rozmieszczenie kamer. Wyniki pracy będzie można zapisać do pliku oraz wczytać uprzednio utworzony projekt.

Symulator będzie w czasie rzeczywistym modelował stan ruchu w mieście. Bieżący stan będzie wizualizowany. Poruszające się obiekty będą animowane.

Przeglądarka rekordów w bazie danych będzie udostępniać podstawową funkcjonalność, pozwalającą na zapoznanie się z zebranymi danymi.

Lista zadań z szacowanym czasem realizacji:

	Zadania	Osoba odpowiedzialna	Szacowany czas realizacji
1.	Utworzenie struktury programu	Bogdan Shkola	3h
2.	GUI programu (panele, przyciski, etc.)	Bogdan Shkola	4h
3.	Zaprojektowanie struktur danych mapy miasta	Wojciech Ceret	3h
4.	GUI schematu miasta	Bogdan Shkola	3h
5.	Zaprojektowanie struktury grafu połączeń drogowych	Andrii Khmelovskyi	2h
6.	Opracowanie konwertera struktur danych (mapa \leftrightarrow graf)	Andrii Khmelovskyi	4h
7.	Opracowanie klas dla obiektów ruchomych	Wojciech Ceret	3h
8.	Opracowanie klasy kamery	Wojciech Ceret	1h
9.	Zapis/odczyt struktur danych do/z plików	Wojciech Ceret	3h
10.	Algorytm wyszukiwania drogi	Andrii Khmelovskyi	3h
11.	Opracowanie i implementacja algorytmu wykrywania obiektów przez kamery	Wojciech Ceret	4h
12.	Opracowanie modułu wprowadzającego niedokładność pomiaru kamery	Wojciech Ceret	2h
13.	Zapis/odczyt do/z bazy danych	Andrii Khmelovskyi	3h
14.	Prezentacja danych z bazy	Andrii Khmelovskyi	5h
15.	Opracowanie i implementacja silnika symulacji	cały zespół	8h
16.	Implementacja animacji symulacji	Bogdan Shkola	5h
	Razem		56h