

AGH, Wydział IEiT, Informatyka

# Skrzyżowania ++

Piotr Skibiak, Tomasz Kwiecień, Marcin Nowak, Ksawery Głaz, Paweł Łabno

3 stycznia 2014

## **Streszczenie**

Projekt ma na celu skrócenie oczekiwania samochodów na skrzyżowaniach, poprzez zastosowanie optymalnego ustawienia czasu świateł. Optymalne (albo prawie optymalne) ustawienie świateł wyznaczone jest poprzez użycie jednego z algorytmów stadnych - algorytmu kukułki.

## Spis Treści

<b>1</b>	<b>Przedstawienie problemu</b>	<b>4</b>
<b>2</b>	<b>Cele projektu</b>	<b>4</b>
<b>3</b>	<b>Wstępne założenia</b>	<b>4</b>
3.1	Środowisko Implementacji . . . . .	4
3.2	Praca w zespole . . . . .	4
<b>4</b>	<b>Opis logiki</b>	<b>4</b>
<b>5</b>	<b>Algorytm kukulki</b>	<b>4</b>
<b>6</b>	<b>Obsługa aplikacji</b>	<b>4</b>
<b>7</b>	<b>Rezultaty</b>	<b>4</b>

## **1 Przedstawienie problemu**

## **2 Cele projektu**

Projekt ma na celu umożliwienie użytkownikowi znalezienie optymalnego ustawienia czasów świecenia świateł (światła zielonego i czerwonego) w sieci skrzyżowań ustalonych przez użytkownika. Aplikacja powinna dać użytkownikowi możliwość ręcznego wprowadzenia sieci skrzyżowań, zapisu konfiguracji do pliku, lub wczytania wcześniej zapisanej.

## **3 Wstępne założenia**

### **3.1 Środowisko Implementacji**

Aby pogodzić potrzebę szybkiego wykonywania symulacji, oraz zdążyć wykonać projekt w przeznaczonym do tego czasie, językiem wybranym do implementacji jest Java. Ustaliliśmy, że aby zmniejszyć prawdopodobieństwo wystąpienia niespójności związanej z wykorzystywaniem różnych narzędzi, aplikacja będzie pisana przy pomocy Eclipse IDE.

### **3.2 Praca w zespole**

Praca grupowa była wspomagana poprzez wykorzystanie systemu kontroli wersji GIT, oraz poprzez utworzenie Google Doc'a projektu. W trakcie realizacji przeprowadziliśmy również kilka spotkań mających na celu, kontrolę przebiegu prac, wyjaśnienie wątpliwości / niejasności wykrytych w trakcie iteracji, a także przydział kolejnych zadań.

## **4 Opis logiki**

## **5 Algorytm kukułki**

## **6 Obsługa aplikacji**

Uruchamiając nasz program wyświetla się okno, które jest podzielone na 2 sekcje obszar, w którym możemy rysować skrzyżowania i drogi oraz przedstawiana tam jest wizualnie symulacja. W drugim obszarze po prawej stronie znajdują się wszystkie przyciski do obsługi programu.

### **6.1 Dodaj Generator**

Gdy przycisk ten zostanie naciśnięty możemy za pomocą kliknięcia lewym przyciskiem myszy na obszarze do rysowania stworzyć generator. Zostanie on narysowany w klikniętym miejscu jako czerwone koło poprzez pobranie współrzędnych

kursora, a także zostanie stworzony obiekt Generator, który zostanie dodany do listy generatorów.

## 6.2 Dodaj Skrzyżowanie

Analogicznie jak w przypadku Generatora zostanie stworzony obiekt Skrzyżowanie oraz w klikniętym miejscu pojawi się czarne kółko.

## 6.3 Dodaj Drogę

Po naciśnięciu przycisku "Dodaj Drogę" użytkownik może poprzez kliknięcie na dwóch skrzyżowaniach lub generatorze i skrzyżowaniu (najpierw jednym, a następnie na drugim), które zostało wcześniej narysowane, stworzyć nowy obiekt Droga. Gdy zostaną kliknięte dwa (wcześniej narysowane) obiekty typu Generator lub skrzyżowanie, pojawi się okno z zapytaniem o to ile samochodów może pomieścić droga, a następnie ile (w ciągu godziny) samochodów może przejechać przez tą drogę czyli intensywność zatłoczenia. Jeżeli jeden z zapytanych przez program parametrów nie zostanie podany lub będzie podany, ale w złym formacie np. użytkownik wpisze jakąś literę zamiast cyfry to obiekt nie zostanie stworzony. Po poprawnym podaniu wszystkich parametrów zostanie stworzony i narysowany obiekt typu droga. Narysowany jest podwójnie by było lepiej widać poruszające się samochody w obie strony. Jedna linia oznacza drogę w jedną stronę, a druga w drugą.

## 6.4 Zaczynij od nowa

Czyści całą narysowaną bądź wczytaną mapę.

## 6.5 Zapisz

Zapisuje w wybranym przez użytkownika miejscu na dysku narysowaną wcześniej mapę w postaci rysunku i list stworzonych obiektów.

## 6.6 Wczytaj

Wczytuje z dysku wcześniej zapisaną mapę.

## 6.7 Wyślij do serwera

Gdy narysujemy już bądź wczytamy mapę, musimy ją wysłać do włączonego wcześniej serwera żądanie, podając IP hosta, na którego komputerze jest włączony serwer, lub wpisując localhost jeśli mamy postawiony serwer lokalnie u siebie. Serwer zwraca nam ID jakie nadał temu żądaniu.

## **6.8 Sprawdź rozwiązanie**

Klikając ten przycisk, a następnie podając ID żądania, pytamy serwer czy dla danego żądania zostało wyliczone optymalne rozwiązanie. Serwer zwraca nam odpowiednią informację, która zostaje wyświetlona w okienku.

## **6.9 Rozpocznij Symulację**

Rozpoczęta zostanie symulacja poruszających się samochodów przedstawionych w postaci zielonych kółek. Symulacja jest przedstawiona wizualnie na narysowanej mapie. Współrzędne samochodów są pobierane co jakiś (bardzo krótki) czas przez co mamy wrażenie, że samochody poruszają się płynnie.

## **6.10 Zatrzymaj Symulację**

Wstrzymuje wcześniej uruchomioną symulację.

## **6.11 Wyjdź**

Przyciśnięcie tego przycisku powoduje całkowite wyjście z programu.

# **7 Rezultaty**