

# Symulacja ruchu samochodów na II Obwodnicy Krakowa.

Dawid Godek, Stanisław Światłoch

## 1 Cel

Celem naszej pracy jest stworzenie wiernego modelu II Obwodnicy Krakowa. Tworzenie takich modeli ma szerokie zastosowanie w analizie istniejących oraz planowanych rozwiązań drogowych. Dostarczają one cenne statystyki oraz pozwalają na wizualizację problemów.

II Obwodnica Krakowa jest ważną częścią infrastruktury. Znajduje się na niej wiele skrzyżowań ze światłami, które znacząco wpływają na ruch drogowy. Zadaniem naszej symulacji jest zobrazować jak ten złożony system odpowiada na różne natężenia ruchu.

## 2 Analiza literatury

### 2.1 Automaty komórkowe

Automaty komórkowe można przedstawić w postaci czwórki  $(L, S, N, F)$ ;

- $L$  - przestrzeń podzielona na siatkę komórek
- $S$  - skończony zbiór stanów
- $N$  - zbiór sąsiadów danej komórki
- $f$  - funkcja zmiany konfiguracji w poszczególnych komórkach

Konfiguracja  $C_t$  to funkcja przypisująca każdej komórce stan ze zbioru  $S$ . Funkcja  $f$  zmienia konfigurację dla czasu  $t$   $C_t$  w konfigurację dla czasu  $C_{t+1}$ . Jeżeli stan komórki w czasie  $t$  oznaczmy jako  $x_t$ , a stan jej sąsiedztwa jako  $u(x_t)$ , to stan komórki w czasie  $t + 1$  można opisać wzorem:

$$x_{t+1} = f(u(x_t), x_t) \quad (1)$$

### 2.2 Model Nagela-Schreckenberga

W naszej symulacji wykorzystamy oparty na automatach komórkowych model Nagela-Schreckenberga [1]. W nim, droga jest reprezentowana przez jednowymiarową tablicę komórek. Każdy pojazd zajmuje dokładnie jedną komórkę. Prędkość jest przedstawiona jako ilość komórek, które przebywa pojazd w chwili czasu. Do każdego pojazdu stosowane są cztery kroki:

1. *przyśpieszanie* - jeżeli prędkość pojazdu nie jest maksymalna i dystans do poprzedzającego pojazdu jest większy o jeden niż aktualna prędkość, wtedy zwiększamy prędkość o jeden;
2. *hamowanie* - jeżeli dystans do poprzedzającego pojazdu jest mniejszy niż aktualna prędkość, wtedy ustawiamy ten dystans jako nową prędkość;

3. *element losowy* - z pewnym prawdopodobieństwem prędkość pojazdu zostanie zmniejszona o 1;
4. *ruch* - zmiana położenia.

Został on stworzony do symulacji ruchu na autostradzie, który znacząco się różni od ruchu miejskiego. Dlatego zastosujemy zmodyfikowany model Nagela-Schreckenberga dla ruchu w mieście [2]. W stosunku do oryginału, lepiej odwzorowuje on dynamikę samochodu, dzięki wprowadzeniu zmiennego przyspieszenia. Samochód jest reprezentowany na siatce automatu przez kilka komórek, dając lepsze odwzorowanie prędkości. Wprowadza również symulację zmiany pasa ruchu oraz skrzyżowań ze światłami. Sieć drogowa jest reprezentowana za pomocą grafu skierowanego, gdzie krawędzie reprezentują jednokierunkowe odcinki drogi, a węzły skrzyżowania. Istnieją również specjalne węzły wprowadzające i usuwające pojazdy.

## 3 Realizacja

### 3.1 Przygotowanie mapy II obwodnicy Krakowa

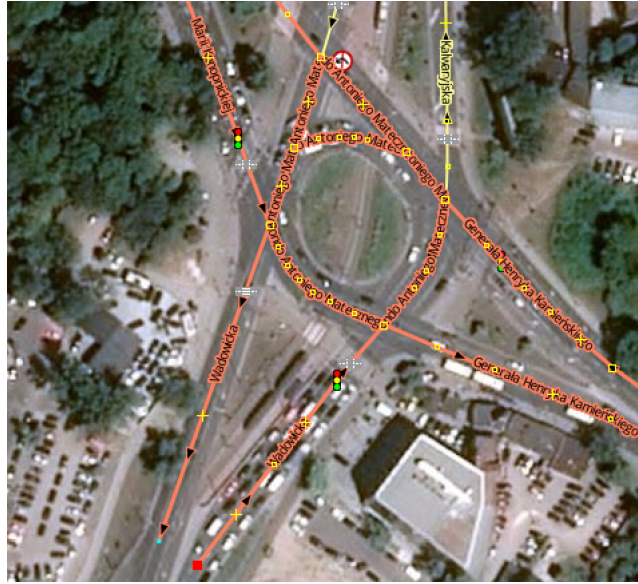
Dbając o realizm naszej symulacji, wykorzystujemy mapę OpenStreetMap [3], która poza podstawowymi danymi o lokalizacji dróg, zawiera również dużo innych informacji, jak lokalizację świateł, ilość pasów ruchu, czy dozwolone prędkości. Ponieważ potrzebujemy tylko część sieci drogowej, wykorzystaliśmy program JOSM [4] do edycji. W efekcie uzyskaliśmy kompletną mapę II Obwodnicy Krakowa w formacie .osm.

### 3.2 Procesowanie mapy i tworzenie modelu dróg na jej podstawie

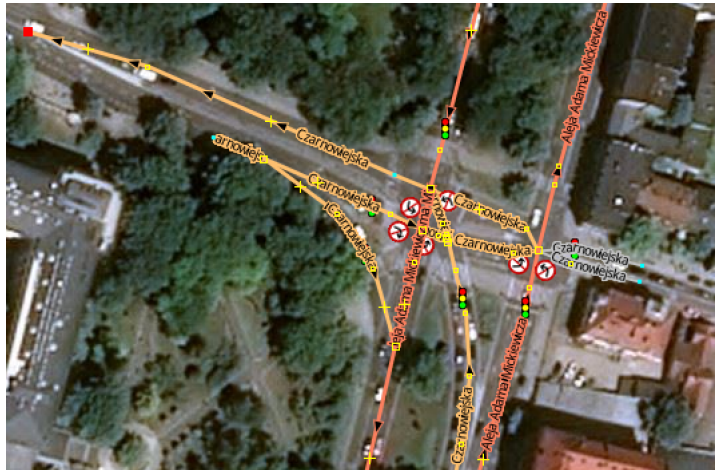
Efektem końcowym przetworzenia mapy przez nasz program jest de facto cykliczny graf skierowany, w którym:

- *węzłami* - są pojedyncze, istotne punkty na mapie (zmniejszenie ilości pasów, skrzyżowanie)
- *krawędziami* - odcinki drogi łączące powyższe punkty na mapie (posiadające informacje o ilości pasów, zajętości pasów)

Dodatkowo istnieją węzły **origin**, które odpowiadają za wprowadzanie samochodów na obwodnicę, oraz **term**, przez które samochody mogą zniknąć z obwodnicy. Dla zachowania kształtu jezdni istnieją też węzły pośrednie, które nie mają istotnego znaczenia dla stworzonego grafu. Dla końcowego stanu mapy stworzony graf posiada 485 krawędzi oraz 1674 węzły.



Rysunek 1: Przykład węzła origin na ul. Wadowickiej (lewy dolny róg); podgląd w JOSM



Rysunek 2: Przykład węzła term na ul. Czarnowiejskiej (lewy górny róg); podgląd w JOSM

### 3.3 Wprowadzanie samochodów na obwodnicę

Za wprowadzanie samochodów na obwodnicę w odpowiednich miejscach odpowiadają węzły **origin**, które z pewnym prawdopodobieństwem (modyfikowalnym na żywo w GUI) za każdym krokiem symulacji wprowadza bądź nie samochód do ruchu. Samochodowi jest nadawana prędkość, przyspieszenie oraz cel podróży (któryś z węzłów **term**). Aby samochód mógł prawidłowo odbyć swoją podróż, musi posiadać mapę nawigacji. Taka mapa to prosta funkcja:

$$f : E_c \rightarrow E \quad (2)$$

gdzie  $E_c$  jest zbiorem wszystkich krawędzi odwiedzonych podczas drogi samochodu z punktu startowego do punktu końcowego, a  $E$  zbiorem wszystkich krawędzi w utworzonym modelu obwodnicy.

Aby stworzyć taką funkcję, należy przeszukać graf w poszukiwaniu ścieżki prowadzącej z węzła **origin** do węzła **term**. Nasz program wykorzystuje do tego lekko zmodyfikowany algorytm Breadth First Search, który zaczynając od pierwszej krawędzi wychodzącej z węzła **origin** znajduje odpowiednią drogę. Trafność wyboru algorytmu i jego poprawność potwierdza fakt, że dla każdego węzła origin w krótkim czasie (niecała sekunda) jesteśmy w stanie odnaleźć drogi do wszystkich węzłów terminalnych. Użycie algorytmu DFS byłoby o wiele mniej trafne, bądź

wcale, bo nasz graf jest grafem cyklicznym i algorytm nie zakończyłby swojego działania lub znajdował bardzo długie drogi.

### 3.4 Poruszanie się samochodu

Samochody poruszają się zgodnie z modelem Nagela-Schreckengberga z lekkimi modyfikacjami. Samochód ma zadaną prędkość maksymalną, przyspieszenie. Jeżeli samochód napotka przed sobą przeszkodę (inny samochód), to:

- a) jeżeli istnieje lewy pas i nie ma żadnej przeszkody w odległości  $pozycja\_samochodu : pozycja\_samochodu + aktualna\_prędkość$  to samochód zmienia pas i będzie kontynuował nim podróż (wyprzedzanie)
- b) zwalnia do prędkości równej odległości do najbliższej przeszkody

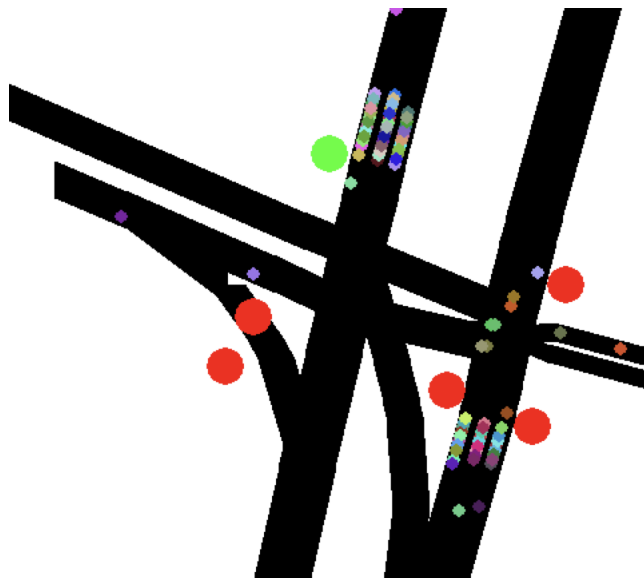
Dodatkowo, samochód jadący po pasach lewych będzie próbował zmienić pas na bardziej prawy wtedy, kiedy to będzie możliwe (na takich samych zasadach jak podczas wyprzedzania). Za przeszkodę uznaje się także sygnalizację świetlną pokazującą kolor czerwony.

### 3.5 Działanie skrzyżowań ze światłami

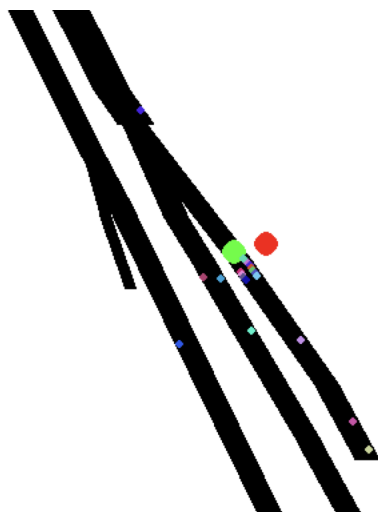
Zaimplementowano dwa rodzaje skrzyżowań:

- skrzyżowania dwugrupowe - kolejne tury zielonych świateł rozdzielone są między dwie grupy świateł
- skrzyżowania trójgrupowego - kolejne tury zielonych świateł rozdzielone są między trzy grupy świateł

Zaimplementowane działanie nie jest zbyt skomplikowane. Podczas jednej tury zielonego światła puszczana jest jedna grupa (nie kolidujących ze sobą) świateł. Następnie kolejna i tak w kółko. Dodatkowo między grupami robiona jest przerwa kilku kroków symulacji po to, aby wszystkie samochody z poprzedniej grupy mogły zjechać ze skrzyżowania. Większość skrzyżowań jest trójgrupowa.



Rysunek 3: Przykład działania skrzyżowania trójgrupowego ze światłami na ulicy Czarnowiejskiej

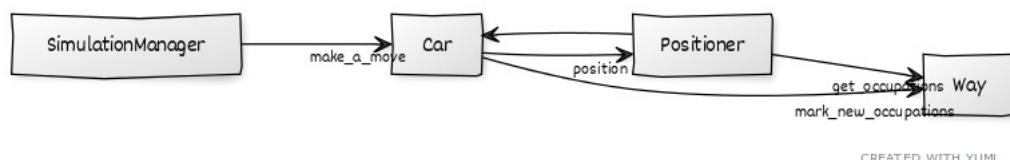


Rysunek 4: Przykład działania skrzyżowania dwugrupowego ze światłami na ulicy Konopnickiej

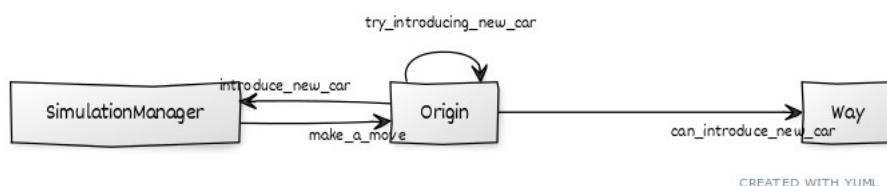
## 4 Implementacja

Do implementacji symulacji użyliśmy języka Python, ze względu na wygodę obliczeń i znajomość bibliotek przez autorów. Pisząc kod staraliśmy się trzymać podstawowych zasad programowania obiektowego, co znacznie pomogło w pracy nad projektem (szczególnie zasada Single Responsibility Principle). Przegląd najbardziej istotnych klas w projekcie:

- **Car** - reprezentacja samochodu na mapie, poruszanie się samochodu
- **DirectionsFinder** - znajdowanie drogi między origin a term z użyciem algorytmu BFS
- **Map** - parsowanie mapy z pliku *.osm* i tworzenie modelu obwodnicy na jej podstawie
- **Node** - reprezentacja istotnego punktu na mapie - zarówno węzły będące węzłami w grafie, jak i te, które służą tylko poprawnemu rysunkowi drogi
- **Origin** - wprowadzenie nowych samochodów na obwodnicę, nadawanie im początkowej wartości
- **Positioner** - wyznaczenie kolejnej pozycji samochodu, detekcja przeszkód i kolizji, odpowiada za zmianę pasa ruchu samochodu
- **Way** - reprezentacja pojedynczego odcinka drogi, krawędzi w grafie obwodnicy, przechowywanie struktur danych odpowiedzialnych za wyznaczanie zajętości pól na jezdniach
- **TrafficLightsManager** - kierowanie pojedynczym skrzyżowaniem ze światłami
- **SimulationManager** - wołanie metod kolejnych kroków na wszystkich elementach symulacji, wyznaczanie kolejnych kroków, wizualizacja symulacji



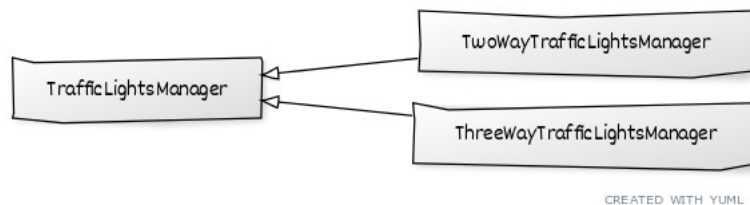
Rysunek 5: Diagram UML operacji makeamove dla klasy Car



Rysunek 6: Diagram UML operacji makeamove dla klasy Origin



Rysunek 7: Diagram UML operacji makeamove dla klasy TrafficLightsManager



Rysunek 8: Diagram klas TrafficLightsManager

Do wizualizacji symulacji użyliśmy biblioteki PyGame.

## 5 Obsługa

Symulacja zaczyna się zaraz po uruchomieniu. Na początku cała sieć drogowa jest pusta, a prawdopodobieństwo wprowadzenia nowego pojazdu na węzle **origin** wynosi 20%. Wartości prawdopodobieństwa można zwiększać lub zmniejszać o 5% za pomocą przycisków po prawej stronie. W menu po prawej znajdują się również ogólne statystyki dla całej obwodnicy, czyli liczba pojazdów oraz ich średnia prędkość.

Istnieje również możliwość wybrania konkretnego odcinka drogi, dla którego rozwinię się dodatkowe menu z nazwą ulicy i statystykami. Znajdziemy w nim ilość samochodów, średnią prędkość oraz przepływ na minutę, obliczany na podstawie samochodów zjeżdżających z drogi. Zaznaczony odcinek zmienia również kolor na szary.

Chance of car spawning:	-	25%	+
Number of cars:	4504		
Average speed:	15.7 km/h		
Statistics for:			
Kotlarska			
Number of cars:	88		
Average speed:	5.3 km/h		
Flow:	82 veh/min		

Rysunek 9: Menu programu

## 6 Wyniki

Na początkowych etapach trwania symulacji można zauważyć wzrost ogólnej średniej prędkości wraz ze wzrostem samochodów. Jednak szybko ta wartość stabilizuje się, a po chwili zaczyna stopniowo maleć. Podobny trend można zauważyć na poszczególnych odcinkach dróg, szczególnie przed dużymi skrzyżowaniami oraz drogach bardziej oddalonych od węzłów komunikacyjnych. Wraz z upływem czasu średnia prędkość oraz przepływ maleją.

Można też zobaczyć przedłużony okres narastania ruchu, nawet po zmniejszeniu prawdopodobieństwa pojawiania się nowych samochodów do zera. Jest to związane z dużą kolejką (korkami) pojazdów chcących wjechać na obwodnicę, która powstała podczas wzmożonego ruchu.

Inne właściwości można zaobserwować przy mniejszych natężeniach ruchu, kiedy pojawiają się regularne oscylacje średniej prędkości, spowodowane jednoczesnymi zmianami świateł na głównej drodze. Kolejnym interesującym zjawiskiem jest fala samochodów zatrzymujących się lub zwalniających, powstała na skutek czerwonego światła lub chwilowego zwiększenia ruchu.

## 7 Wnioski

Zaimplementowany przez nas model nie jest idealny. Nie uwzględnia on na przykład realnych tras oraz natężeń ruchu. Jest on tylko małą częścią większej sieci drogowej, która ma na niego ogromny wpływ. Jednak pomimo tego można z niego wyciągnąć ważne wnioski.

Pierwszym z nich jest efektywność użycia sygnalizacji świetlnej. W niektórych miejscach jest ona niezbędna i bilansuje ruch, zapobiegając blokadom niektórych dróg lub nawet deadlockom. Natomiast w niektórych przypadkach mogą prowadzić do efektu odwrotnego, blokowania głównej drogi na rzecz mniej ważnych. Jest to szczególnie widoczne w miejscach, gdzie jest duże zagęszczenie skrzyżowań. Innym negatywnym skutkiem jest niepotrzebne spowalnianie ruchu, kiedy jest małe natężenie. Dlatego kolejnym wnioskiem jest potrzeba wzajemnego regulowania świateł, a najlepiej ich ciągłej adaptacji do warunków na drodze.

## Literatura

- [1] Kai Nagel, Michael Schreckenberg *A cellular automaton model for freeway traffic*. 1992.
- [2] Jarosław Wąs, Rafał Bieliński, Bartłomiej Gajewski, Patryk Orzechowski *Problematyka modelowania ruchu miejskiego z wykorzystaniem automatów komórkowych*. 2009.
- [3] OpenStreetMap <https://www.openstreetmap.org>
- [4] JOSM <https://josm.openstreetmap.de/>