

# Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

---

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki



## Symulacja ruchu samochodów na I Obwodnicy Krakowa

Oskar Kocjan  
Maciej Koch

# Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
1.1	Opis problemu . . . . .	2
1.2	Cel naszej symulacji . . . . .	2
1.3	Podstawowy model Na-Sch . . . . .	2
1.4	I Obwodnica Krakowa . . . . .	3
<b>2</b>	<b>Analiza modelu oraz modyfikacje podstawowego modelu Na-Sch</b>	<b>4</b>
2.1	Automaty komórkowe . . . . .	4
2.2	Reguły poruszania pojazdów . . . . .	4
<b>3</b>	<b>Wybór technologii</b>	<b>6</b>
<b>4</b>	<b>Symulacja zjawiska - implementacja</b>	<b>6</b>
4.1	Przygotowanie siatki obwodnicy . . . . .	6
4.2	Stworzenie logiki modelu . . . . .	7
4.3	Tworzenie wizualizacji symulacji . . . . .	8
4.4	Analiza wyników końcowych . . . . .	8

# 1 Wprowadzenie

## 1.1 Opis problemu

W wielu miastach codziennie możemy spotkać się z dużym ruchem na drogach publicznych co niestety w przypadku większych miejscowości skutkuje korkami, częstymi wypadkami, wstrzymaniami komunikacji miejskiej. Innymi powodami wyżej wymienionych sytuacji mogą być również planowane przez urzędy miast plany rozbudowy infrastruktury czy też samych dróg. Przeprowadzanie skutecznych i realistycznych symulacji pozwala na lepsze rozwiązanie tych problemów.

## 1.2 Cel naszej symulacji

W naszym projekcie zajmiemy się I obwodnicą Krakowa. Głównym celem będzie przeprowadzenie takiej symulacji, aby w jak najbardziej realistyczny sposób zasymulować dynamikę ruchu miejskiego. Przedstawić ruch samochodów jednoosobowych oraz ciężarowych. Odzwierciedlimy faktyczny układ dróg z uwzględnieniem najważniejszych skrzyżowań, dróg jednokierunkowych, wielopasmowych oraz węzłów z sygnalizacją świetlną. W tym celu posłużymy się modelem Nagela-Schreckenberga - klasyczny sposób oparty na automatach komórkowych, który służy do opisu ruchu samochodów. Uwzględnimy specyfikę pojazdów oraz ich ruch w oparciu o automaty komórkowe. Określimy sąsiedztwo (w ujęciu Moore'a), przyspieszenie, hamowanie, losową zmianę prędkości, regułę prawej ręki.

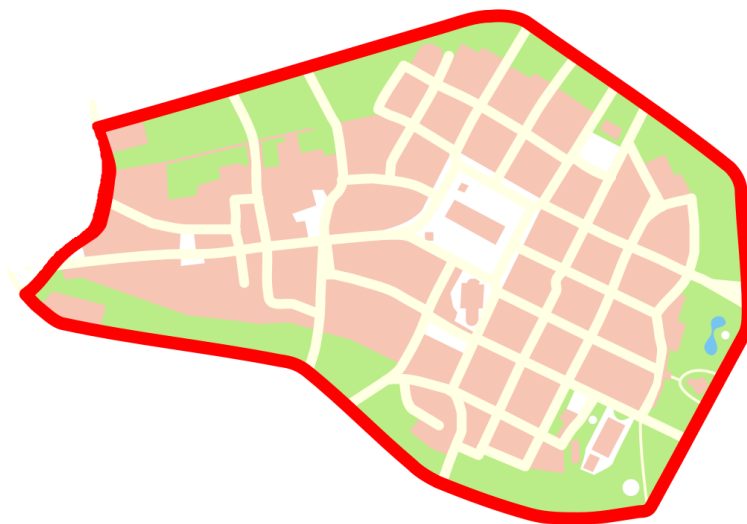
## 1.3 Podstawowy model Na-Sch

W 1992 r. Nagel i Schreckenberg przedstawili model - automat komórkowy opisujący ruch pojazdów. Założenia modelu to:

- podział ulicy na komórki długości 7.5m
- droga jest jednokierunkowa jedno pasowa
- komórka może być zajęta przez jeden pojazd lub wolna
- każdej komórce przypisana jest wartość prędkości
- ruch pojazdów w tym modelu opisują 4 pojęcia:
  1. przyspieszanie - pojazd zwiększa swoją prędkość
  2. hamowanie - zmniejszenie szybkości w komórce poprzedzającej znajduje się pojazd
  3. hamowanie - na drodze bywają zdarzenia losowe np. wtargnięcie pieszego na jezdnię
  4. przesunięcie - o wykonaniu powyższych czynności następuje przesunięcie wszystkich pojazdów do następnej komórki i zwiększenie zmiennej czasu

## 1.4 I Obwodnica Krakowa

Mianem pierwszej obwodnicy określa się w Krakowie ciąg ulic otaczających Stare Miasto wzdłuż plant. Ruch na obszarze wewnątrz pierwszej obwodnicy jest wysoce ograniczony



W skład obwodnicy wchodzi niżej przedstawiony wykaz ulic (zgodnie z ruchem wskazówek zegara) :

- ul. św. Idziego (ograniczenie ruchu w jednym kierunku)
- ul. Podzamcze (ograniczenie ruchu w jednym kierunku)
- ul. F. Straszewskiego (ruch jednokierunkowy na odcinku ulicy)
- ul. Podwale (ograniczenie ruchu na odcinku ulicy i ruch jednokierunkowy na całości)
- ul. J. Dunajewskiego (ograniczenie ruchu i ruch jednokierunkowy)
- ul. Basztowa (ograniczenie ruchu na odcinku i ruch jednokierunkowy na całej ulicy)
- ul. Westerplatte
- ul. św. Gertrudy

## 2 Analiza modelu oraz modyfikacje podstawowego modelu Na-Sch

### 2.1 Automaty komórkowe

Automaty komórkowe (ang. cellular automata) mogą być prezentowane w postaci czwórki  $A = (\alpha, S, N, f)$  :

$\alpha$  - definiuje regularną, uporządkowaną siatkę złożoną z jednakowych komórek

$S$  - skończony zbiór stanów, jaki może przyjąć komórka

$N$  - zbiór sąsiadów

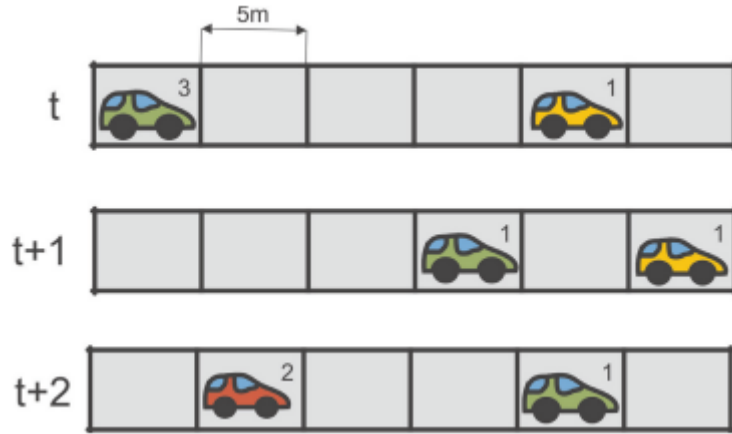
$f$  - funkcja przejścia która definiuje ewolucje automatu w kolejnych jednostkach czasu

W modelu Nagel–Schreckenberga droga jest podzielona na komórki. Każda z nich może być w dwóch stanach tj. albo pełna (czyli zawiera jeden samochód) lub pusta (nie zawiera wcale). Każdy pojazd ma przypisaną prędkość od 0 do  $V_{max}$  ustalonego gdzie ta wartość jest liczbą całkowitą. Informuje ona o liczbie komórek, o które przesunie się pojazd w kolejnym kroku. Jest to model symulacyjny autostrady opracowany do badania warunków spontanicznego powstawania korków, jednak idealnie nada się on również do opisu ruchu miejskiego. W naszym projekcie ustaliliśmy trzy rodzaje prędkości, gdyż zakładamy, iż samochody nie przekraczają pewnej granicy tym bardziej na I obwodnicy, gdzie ruch jest i tak z powodów oczywistych utrudniony. Dodatkowo w naszym przypadku, komórki mają długość 5m.

### 2.2 Reguły poruszania pojazdów

W każdym dyskretnym kroku dla wszystkich pojazdów są wykonywane równolegle kolejno następujące reguły poruszania się:

- przyspieszenie  $v(t+1) \rightarrow \min(v(t) + 1, v_{max})$ , gdzie  $v(t)$  to prędkość aktualna pojazdu
- hamowanie  $v(t+1) \rightarrow \min(v(t), g(t) - 1)$ , gdzie  $g(t)$  jest liczbą pustych komórek między pojazdami
- element losowy, prawdopodobieństwo  $p$ , że zajdzie  $v(t+1) \rightarrow \max(v(t) - 1)$ , jeżeli  $v(t) \geq 1$
- ruch (zmiana położenia w czasie)  $x(t+1) = x(t) + v(t)$



Rysunek 1: Ruch modelu Nagela-Schreckenberga na pasie ruchu kolejnych chwilach czasowych. W lewym górnym rogu każdej komórki można zauważyć aktualną prędkość pojazdu podaną w komórkach na chwilę czasową.

## Sygnalizacja świetlna

W naszym projekcie chcemy zmodyfikować model poprzez dodanie dodatkowych parametrów w postaci sygnalizacji świetlnej co wpłynie na element ruchu:

jeżeli:  $s(t) = 1$  i  $x_i(t) < x_s$ , to  $v_i(t) \leftarrow \min(v_i(t), x_s - x_i(t) - 1)$   
gdzie:

$s(t)$  - oznacza sygnał wyświetlany dla pojazdów w chwili  $t$   
( $s(t) = 1$ , gdy wyświetlany jest sygnał czerwony)

$x_s$  - jest numerem komórki, która znajduje się na modelowanym wlocie skrzyżowania, bezpośrednio za linią warunkowego zatrzymania

## Reguła prawej ręki

Kolejną modyfikacją podstawowego modelu będzie dodanie reguły prawej ręki, która będzie działać w następujący sposób dla danego skrzyżowania:

jeżeli  $(x(t) = p \in A) \wedge (\forall z \in B)$ , jeżeli  $\exists z_n = 1 \Rightarrow v(t) = 0$   
gdzie:

$A$  - oznacza zbiór punktów (komórek) linii warunkowego zatrzymania na skrzyżowaniu w przypadku świateł, bądź ustąpienia pierwszeństwa.

$B$  - oznacza zbiór punktów (komórka) *otoczenia* skrzyżowania, co w praktyce oznacza punkty skrzyżowania oraz punkty odcinka pasa ruchu na którym poruszają się samochody, którym być może będzie należało ustąpić pierwszeństwa. Każdy element tego zbioru może przyjmować wartość 0 lub 1 co oznacza kolejno iż punkt jest wolny lub zajęty.

### 3 Wybór technologii

Do stworzenia naszego projektu, wykorzystaliśmy następujące technologie:

- Środowisko PyCharm
- OpenStreetMap
- Oprogramowanie JOSM
- Github

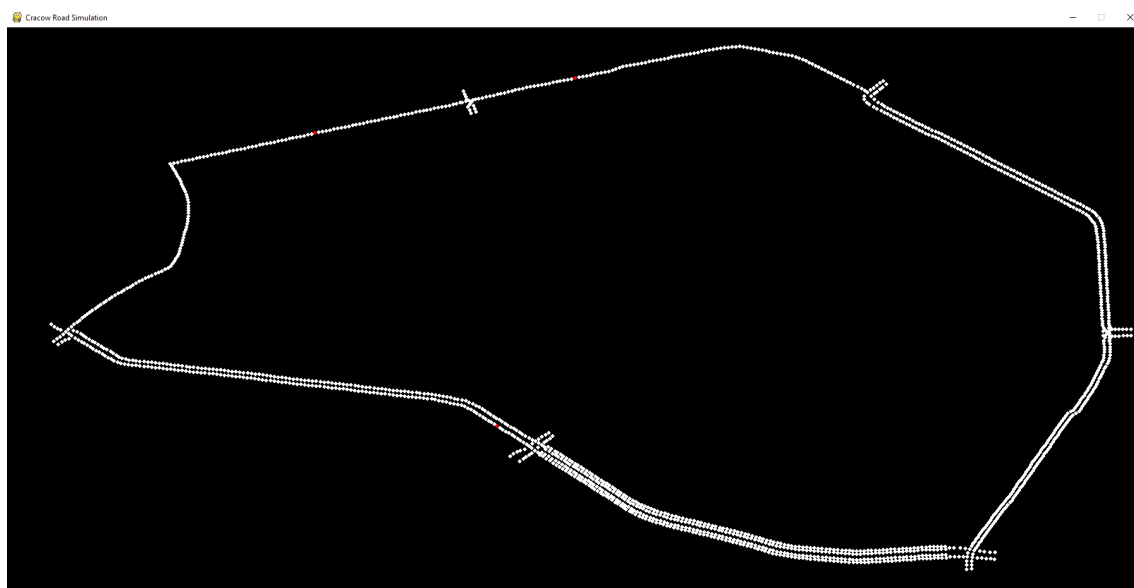
Zdecydowaliśmy się na wykorzystanie Pythona, gdyż jest prostym językiem, idealnie nadającym się do wykonywania różnych nieskomplikowanych matematycznie operacji. Istnieje wiele przydatnych bibliotek, które ułatwiają pracę. W naszym projekcie do samej wizualizacji użyliśmy PyGame, a do implementacji wszelkich zachowań oraz warunków ruchu wykorzystaliśmy m. in. `os`, `json`, `threading`, `math`, czy też `time`.

Dzięki stronie OpenStreetMap mogliśmy uzyskać rzeczywiste współrzędne interesujących nas punktów, a przy użyciu JOSM mogliśmy je odpowiednio edytować i eksportować do przyjaznego formatu `.json`. Całość była regularnie aktualizowana na zdalnym repozytorium na Github'ie.

## 4 Symulacja zjawiska - implementacja

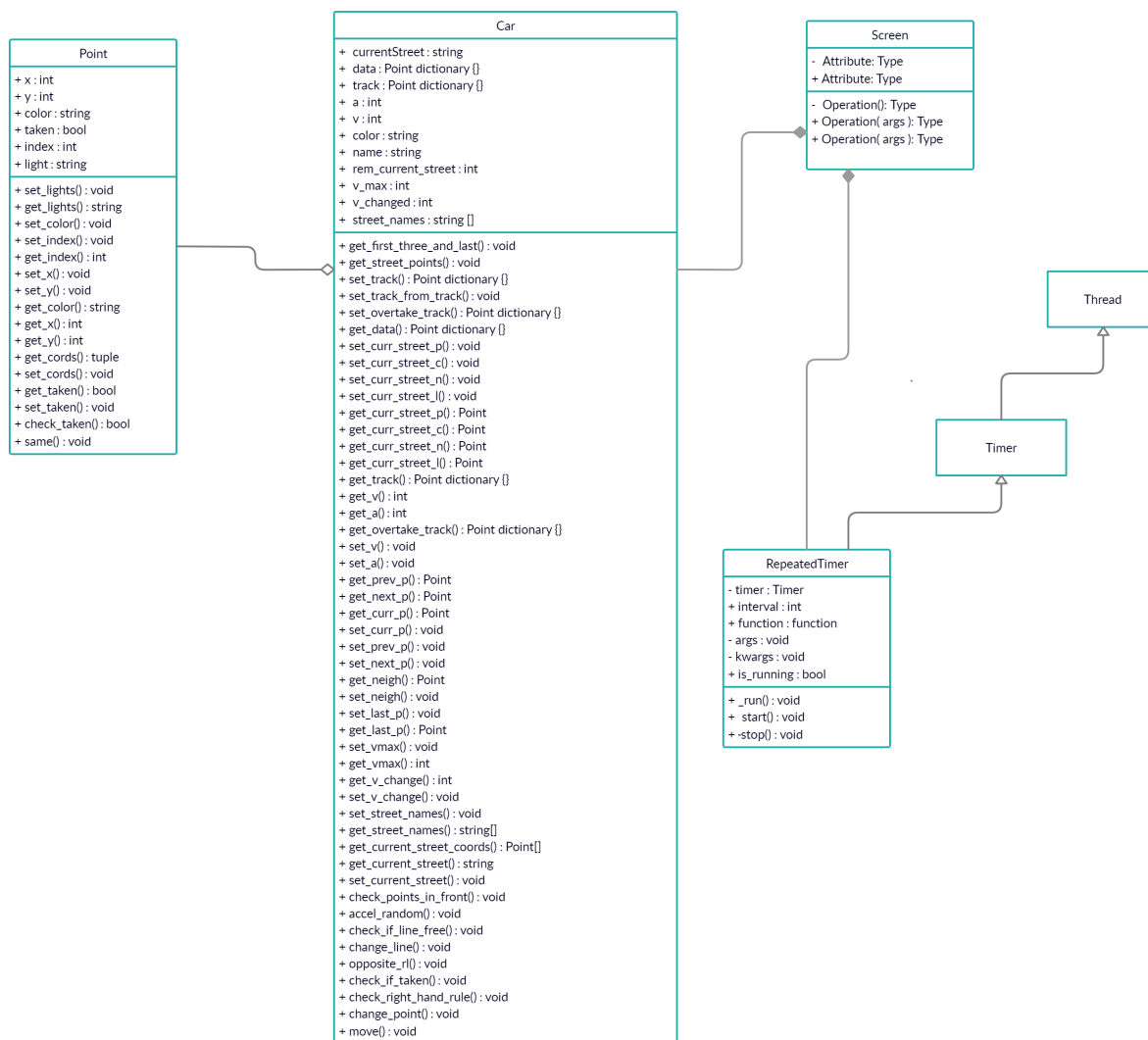
### 4.1 Przygotowanie siatki obwodnicy

Początek pracy rozpoczęliśmy od przygotowania siatki naszej obwodnicy. W tym celu wykorzystaliśmy oprogramowanie JOSM, do którego zaimportowaliśmy mapę w formacie `mapa.osm`, a następnie nanieśliśmy punkty otrzymując w formacie `mapa.json` zbiór koordynatów. Współrzędne dzięki oprogramowaniu zostały ustawione w taki sposób, że każda komórka jest oddalona od siebie o 5m, gdyż tak założyliśmy w naszym modelu. Kolejnym krokiem była odpowiednia ich konwersja w Pythonie oraz wstępna wizualizacja poprzez wykorzystanie biblioteki Pygame.



## 4.2 Stworzenie logiki modelu

Poniżej został przedstawiony diagram UML klas naszego projektu, zasadniczo jak wi-  
dać najwięcej się dzieje w obiektach klasy Car, gdzie zostały zaimplementowane wszystkie  
warunki dotyczące ruchu, klasa Point posłużyła do pracy z punktami i stworzeniem siatki  
z komórek, klasa Screen odpowiada jedynie za wizualizację, a jako ostatnia użyta kla-  
sa RepeatedTimer odpowiada za liczenie czasu i m. in. obsługę działania świateł. Czas  
niejako jest liczony w osobnym wątku, tak aby nie ingerować w sam ruch pojazdów na  
obwodnicy.

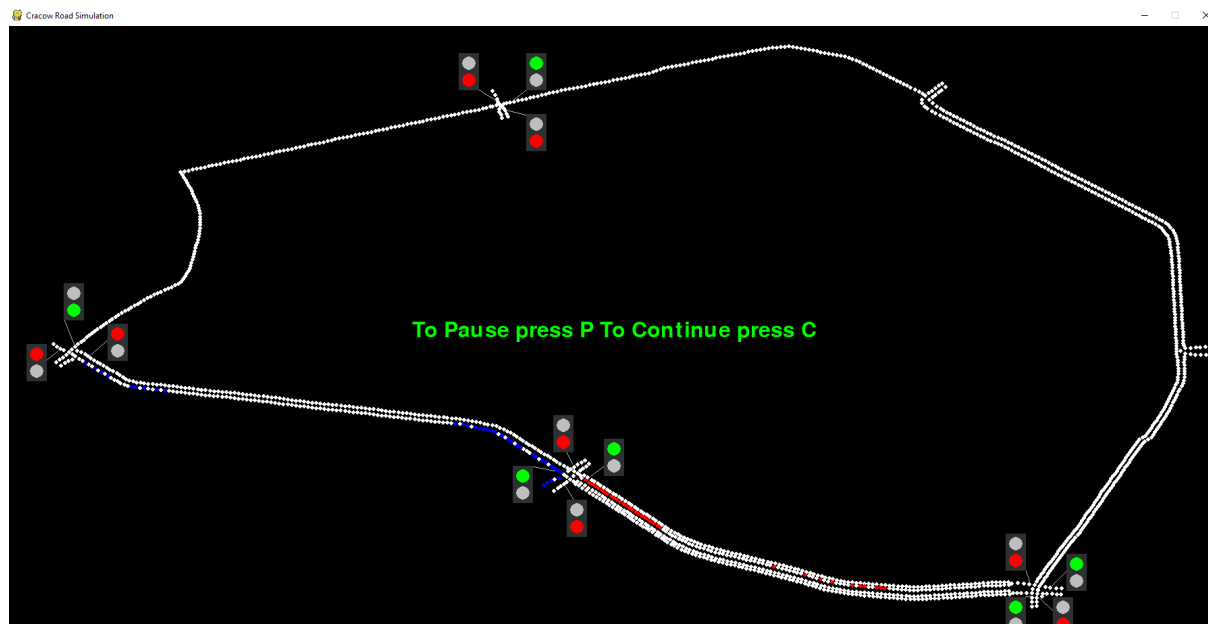


Rysunek 2: Diagram UML klas



### 4.3 Tworzenie wizualizacji symulacji

W ostatniej wersji programu umieściliśmy przyciski do obsługi symulacji oraz zostały dodane światła, które się odpowiednio zmieniają. Całość ostatecznie prezentuje się w następujący sposób:



Rysunek 3: Ostateczny wygląd

### 4.4 Analiza wyników końcowych