

# Symulacja ruchu na IV obwodnicy Krakowa

Jan Godlewski, Joanna Partyka

Elementem symulacji jest załączony na Githubie plik **map.gif**, który należy dodać do IDE lub folderu z kodem, aby wizualizacja mogła działać poprawnie.

Celem naszego projektu było utworzenie symulacji ruchu samochodowego na IV obwodnicy Krakowa. Skorzystaliśmy z modelu **Nagela-Schreckenberga**. Symulacja została napisana w języku **Python**, przy użyciu modułów: **time**, **turtle** i **math**.

Z obliczeń wynika, że długość całej IV obwodnicy to ok. 57 kilometrów. Długość istniejącego odcinka to ok. 37 kilometrów. Przy używanej w modelu Na-Sch długości komórki 7,5 metra, istniejący fragment IV obwodnicy składa się z 4880 komórek, a całość z **7587** komórek.

Komórka to obiekt klasy **Cell**, zawierający atrybuty: **cond** (0 jeśli komórka jest pusta, 1 jeśli jest zajęta), **exists** (0 jeśli istnieje, 1 jeśli nie istnieje) oraz **id**. Komórki są przechowywane w listach: **firstlane**, **secondlane**, **thirdlane**. Kolejną listą jest lista **cars** zawierająca obiekty klasy **Car** (atrybuty to: **v** - prędkość, **category** - truck/car, **id** oraz **exists**). Najważniejsze zmienne obiektów to **id** - unikalny numer każdego pojazdu i **cond** - wartość 0 oznacza, że obiekt klasy **Cell** jest pusty, wartość 1 oznacza, że obiekt klasy **Cell** jest zajęty.

Kategoria pojazdu wpływa na jego prędkość maksymalną. Prędkość maksymalna ciężarówki jest o 1 niższa niż samochodu. Prędkość jest liczona w komórkach/sekundę. 1 komórka/sekundę to ok. 25 km/h.

Do utworzenia list **firstlane**, **secondlane** i **thirdlane** wykorzystujemy funkcję **creation**, która wypełnia listy komórkami. Do komórki **thirdlane[3266].exist = 0**, zaznaczamy w ten sposób, że do tego odcinka istnieją tylko dwa pasy ruchu.

Zapełnianie autostrady samochodami trwałoby bardzo powoli i funkcja **random\_auto** rozmieszcza pojazdy na obwodnicy, żeby w momencie uruchomienia symulacji była ona zapełniona. Nowe pojazdy są przypisywane do komórek na obwodnicy przez zajęcie ich oraz dodanie id tworzonych pojazdów. Oprócz tego nowe pojazdy trafiają na listę **cars**. Można tą funkcjonalność wyłączyć zmieniając wartość zmiennej **randomauto** z 1 na 0.

Liczba pojazdów znajdujących się obecnie na wybranym odcinku obwodnicy jest obliczana za pomocą funkcji **car\_counter**, która przeiterowuje listy **firstlane**, **secondlane** i **thirdlane**, sprawdzając ile komórek jest zajętych. Oprócz tego używane są mniej istotne liczniki liczby kroków i id nowo tworzonych samochodów. Lista **thirdlane** dotyczy tylko fragmentu obwodnicy, który ma trzy pasy na jezdni Łagiewniki-Nowa Huta i budowanego Nowa Huta-Modlnica.

Właściwa część programu działa w oparciu o nieskończoną pętlę.

## Sprawdzanie możliwości wyprzedzania.

1. Sprawdzanie zajęcia czterech komórek przed każdym pojazdem. Jeśli przestrzeń przed pojazdem jest zajęta, rozpoczyna się sprawdzanie możliwości zmiany pasa.

2. Sprawdzanie zajęcia komórek na sąsiednim pasie przed samochodem. Jeśli jezdnia ma trzy pasy, w pierwszej kolejności sprawdzany jest przeskok ze środkowego na prawy pas.
3. Jeśli jest miejsce przed pojazdem, następuje "spojrzenie w lusterko" i sprawdzenia miejsc na sąsiednim pasie z tyłu samochodu.
4. Ewentualna zmiana pasa wiąże się z obniżeniem prędkości pojazdu o 1.

#### **Dostosowanie prędkości pojazdu do zajętości drogi.**

1. Sprawdzenie, czy 4 komórki przed każdym pojazdem są zajęte.
2. Jeśli są zajęte i prędkość pojazdu jest większa od 0, jest ona obniżana o 1 w celu uniknięcia kolizji. Prędkość nie może być obniżona poniżej 0.
3. Jeśli droga jest wolna i pojazd nie porusza się z maksymalną prędkością, jest ona podnoszona o 1.

#### **Usuwanie samochodów wyjeżdżających poza granicę listy.**

1. Jeśli jedna z kilku ostatnich komórek listy jest zajęta, jest ona zwalniana, zmienna `exists` pojazdu jest zmieniana na 0 i przestaje on być częścią symulacji. Przez to pojazd nie może wyjechać poza listę i nie dochodzi do przerwania programu przez błąd "list index out of range".

#### **Przesuwanie pojazdu.**

1. Samochód jest przenoszony o liczbę komórek równą wcześniej obliczonej prędkości. Nowa komórka jest zajmowana i jest do niej przypisywane id pojazdu. Komórka dotychczas zajmowana przez pojazd jest zwalniana, jej id jest zerowane.

#### **Wjazd pojazdów na węzłach**

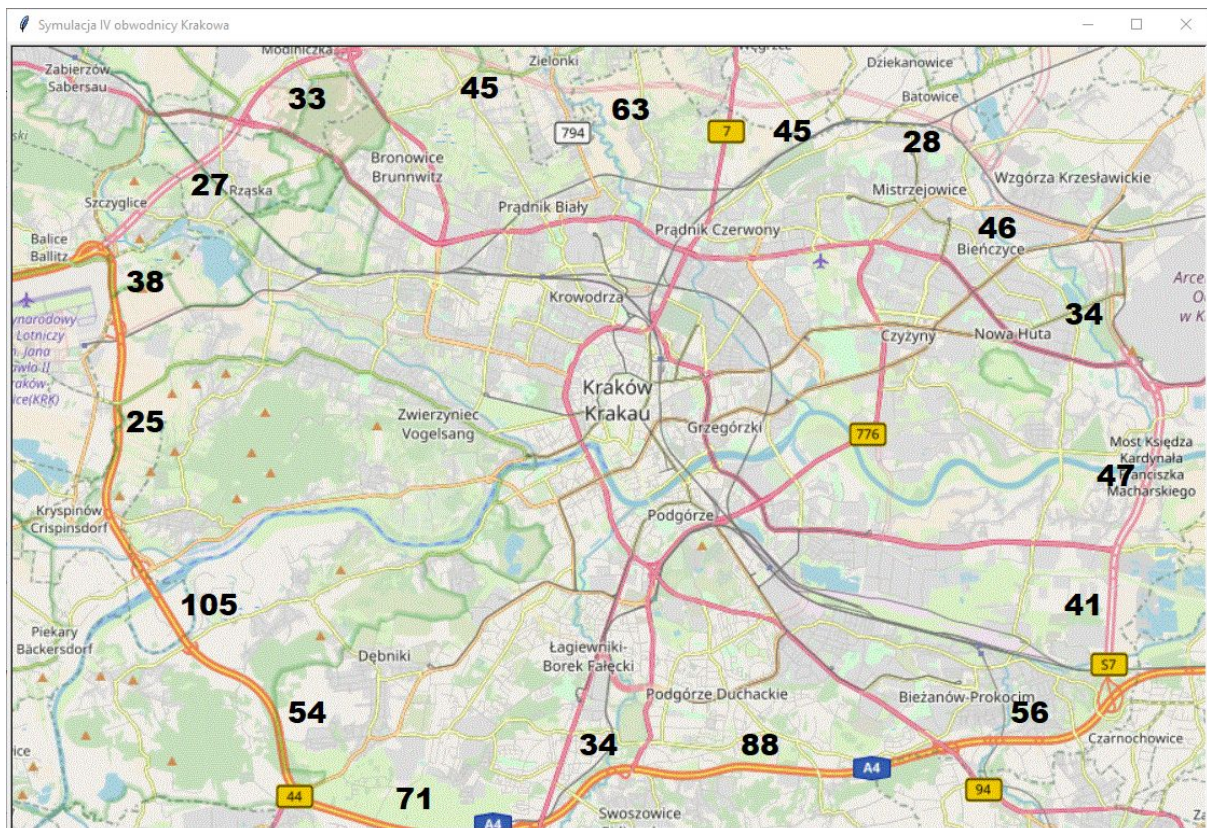
1. Na niektórych węzłach co jakiś czas próbują wjeżdżać pojazdy. Sprawdzają przy tym, czy prawy pas jest wolny. Funkcjonalność nie dotyczy budowanego odcinka, dla którego nie są dostępne dane dotyczące natężenia ruchu.
2. W takiej sytuacji tworzony jest kolejny obiekt klasy `Car`, który jest przypisywany do listy samochodów i prawej jezdni na węźle.

#### **Wyjazd pojazdów na węzłach**

1. Na niektórych węzłach co jakiś czas pojazdy chcą zjechać z obwodnicy. Funkcjonalność nie dotyczy budowanego odcinka, dla którego nie są dostępne dane dotyczące natężenia ruchu.
2. W takiej sytuacji komórka jest zwalniana, zmienna `exists` pojazdu jest zmieniana na 0 i przestaje on być częścią symulacji.

#### **Wizualizacja**

Przy użyciu modułu **turtle** wykonana została wizualizacja, pokazująca ile jest pojazdów na konkretnych odcinkach obwodnicy w danym momencie. Co sekundę wypisywana jest w konsoli liczba pojazdów na całej obwodnicy jak i na poszczególnych jej odcinkach. **turtle** korzysta ze swojej wbudowanej pętli `turtle.mainloop()`. Pojawiające się na wizualizacji liczby są obiektami klasy `turtle`.



Run: symobw x

Suma: 887

Ilość pojazdów na odcinkach obwodnicy:

Modlnica - Modlniczka:	32
Modlniczka - Balice I:	25
Balice I - Balice II :	38
Balice II - Bielany:	27
Bielany - Tynec:	108
Tynec - Skawina:	55
Skawina - Południe:	73
Południe-Łagiewniki:	34
Łagiewniki-Wieliczka:	88
Wieliczka - Bieżanów:	57
Bieżanów - Przewóz:	39
Przewóz - Nowa Huta:	47
Nowa Huta - Grębałów:	35
Grębałów - Mistrzejowice:	46
Mistrzejowice - Batowice:	28
Batowice - Węgrzce:	46
Węgrzce - Zielonki:	62
Zielonki - Modlnica:	46
Suma:	891