



***Wydział Elektrotechniki Automatyki
Informatyki i Inżynierii Biomedycznej***

Informatyka
Rok studiów: II
Rok akademicki: 2019/20

***Symulacja ruchu samochodów
na II obwodnicy Krakowa***

Radosław Bielecki
Łukasz Stanik
Maciej Źądło

Spis treści:

1. Wstęp
2. *Idea automatów komórkowych*
3. *Model Nagela-Schreckenberga*
4. *Model zmodyfikowany*
5. *Reprezentacja*
6. *Algorytm*
7. *GUI*
8. *Podsumowanie i dalsze kierunki rozwoju*

1. Wstęp:



Praca powstała aby ukazać symulację ruchu samochodowego na II Obwodnicy Krakowa. Do jej wykonania użyto modelu Nagela-Schreckenberga przy wykorzystaniu automatów komórkowych. Model uwzględnia takie sytuacje jak pora dnia, zmiana prędkości ruchu pojazdu znajdującego się przed nim, warunki pogodowe na drodze, docelową destynację kierowcy, charakter kierowcy czy sygnalizację świetlną. Model zmodyfikowany został również o sytuacje specyficzne dla ruchu miejskiego - zatrzymywanie się na skrzyżowaniach, poruszanie się w korku, zmiana pasa na taki, który umożliwi opuszczenie obwodnicy, gdy samochód zbliża się do swojego punktu docelowego.

2. Idea automatów komórkowych:

Automat komórkowy jest systemem składającym się z sieci dyskretnych, pojedynczych, sąsiadujących ze sobą komórek. Wszystkie elementy są tych samych rozmiarów. Każda komórka przyjmuje jeden ze skończonej liczby stanów. Zmiana stanu komórki zależy od ustalonych reguł oraz obecnej sytuacji - stanów komórek sąsiadujących. Zmiany następują w dyskretnych odstępach czasu, równoległe dla wszystkich komórek.

3. Model Nagela-Schreckenberga

Przy realizacji naszego projektu wykorzystaliśmy model Nagela-Schreckenberga (w skrócie Na-Sch). Jest on oparty na wyżej opisanych automatach komórkowych, przy pomocy których opisuje ruch samochodów po ulicy. W przyjętym przez nas modelu użyliśmy takich reguł:

- Rozmiar komórki wynosi $d = 7.5\text{m}$.
- Przyspieszenie - $v(t+1) \rightarrow \min(v(t) + 1, v_{\text{max}})$, gdzie $v(t)$ jest aktualną wartością prędkości.
- Hamowanie - $v(t+1) \rightarrow \min(v(t), g(t) - 1)$, gdzie $g(t)$ to ilość pustych komórek między pojazdami.
- Losowe hamowanie - prawdopodobieństwo p na to, że $v(t + 1) \rightarrow \max(v(t) - 1)$, gdy $v(t) \geq 1$.
- Ruch - $x(t + 1) = x(t) + v(t)$.

4. Model zmodyfikowany

Aby model nasz lepiej reprezentował sytuację rzeczywistą, wprowadziliśmy kilka reguł, których podstawowy model Na-Sch nie posiada:

- Sprawdzanie czy samochód zbliża się do czerwonego światła, jeśli tak to samochód zwalnia.
- Sprawdzanie czy następnym skrzyżowaniem jest docelowe dla danego samochodu - w takim wypadku taki samochód musi zmienić pas na taki, który umożliwi opuszczenie obwodnicy.
- W przypadku natrafienia na samochód uniemożliwiający jazdę z obecną prędkością, sprawdzamy czy jest możliwa zmiana pasa w celu uniknięcia redukcji prędkości.

5. Reprezentacja

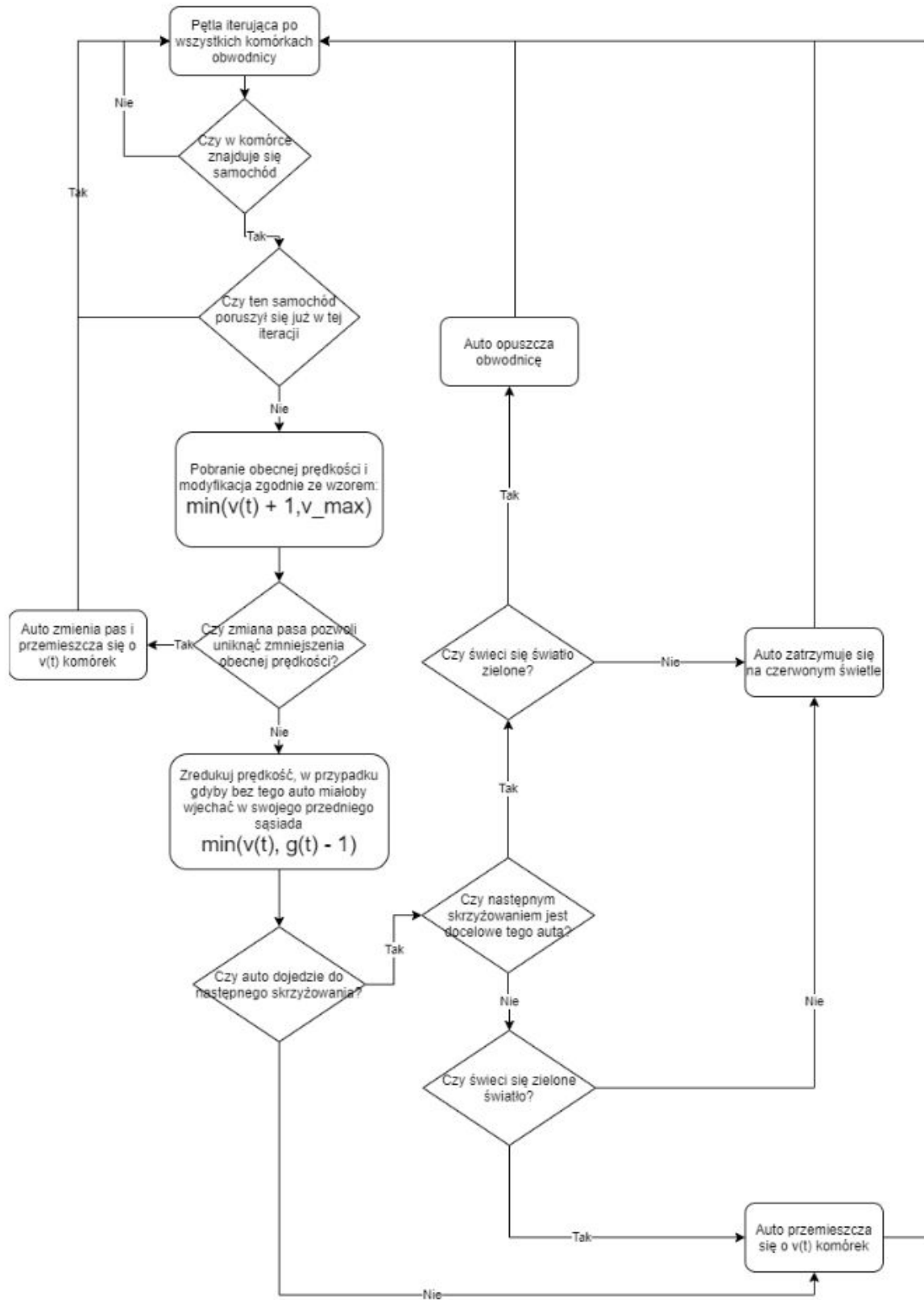
Druga obwodnica została przez nas przedstawiona jako dwie tablice dwuwymiarowe, każda opisująca ruch w przeciwnym kierunku. Kilkadziesiąt komórek specjalnych reprezentuje skrzyżowania, na których pojawiają się nowe auta.

Wykorzystaliśmy Mapy Google do znalezienia rzeczywistych proporcji długości dróg na ulicy.

6. Algorytm

Do implementacji algorytmu wykorzystaliśmy język programowania Java. Na początku tworzymy puste komórki, następnie są one jednokrotnie losowo uzupełniane (ilość początkową aut ustala użytkownik przy wybieraniu pory dnia). Stworzony obiekt auta zawiera losowo wygenerowany cel - jedno ze skrzyżowań obwodnicy, oraz osobowość kierowcy - agresywny kierowca będzie miał mniejszą szansę na losową redukcję prędkości.

Od tego momentu w każdym odcinku czasu będą zmieniały się stany komórek w ustalony sposób:



7. GUI

Do stworzenia graficznego interfejsu użytkownika wykorzystana została biblioteka javafx. Przy pomocy narzędzia scenebuilder udostępnionego od firmy gluonhq zaimplementowane zostały podstawowe edytowalne pola dla użytkownika. Po przekonwertowaniu danych z formatu XML frontend przekazuje dane do funkcji update() znajdującej się w pliku MapController, który odświeża rysowany widok co 1 sekundę. Dane prezentowane są za pomocą odczytu danych z dwóch tablic dwuwymiarowych, a obraz jest wyświetlany za pomocą zdjęć oraz kwadratów (wszystko w formacie 63x63). Każda z kratek reprezentuje 7.5 metra długości, zgodnie z modelem NS. Możliwymi danymi do wprowadzenia są:

- aktualna pogoda,
- pora dnia,
- ilość aut chcących wjechać na obwodnice przy wykonaniu każdej pętli.

Interfejs menu:

Druga Obwodnica Krakowa

— □ ×

DRUGA OBWODNICA
KRAKOWA

Pora dnia

Wieczór

Pogoda

Słońce
Deszcz
✓ Śnieg

Ustaw ilość aut chcących wjeżdżać na poszczególnych skrzyżowaniach

Rondo Matecznego

30

Kamieńskiego i Tischnera

15

Podgórze SKA

32

Kuklińskiego

24

Rondo Grzegórzeckie

53

Rondo Mogiłskie

34

Wita Stwosza i Aleja 29 Listopada

30

Nowy Kleparz

30

Plac Inwalidów

30

Czarnowiejska i aleja Adama Mickiewicza

30

Reymonta i aleja Adama Mickiewicza

30

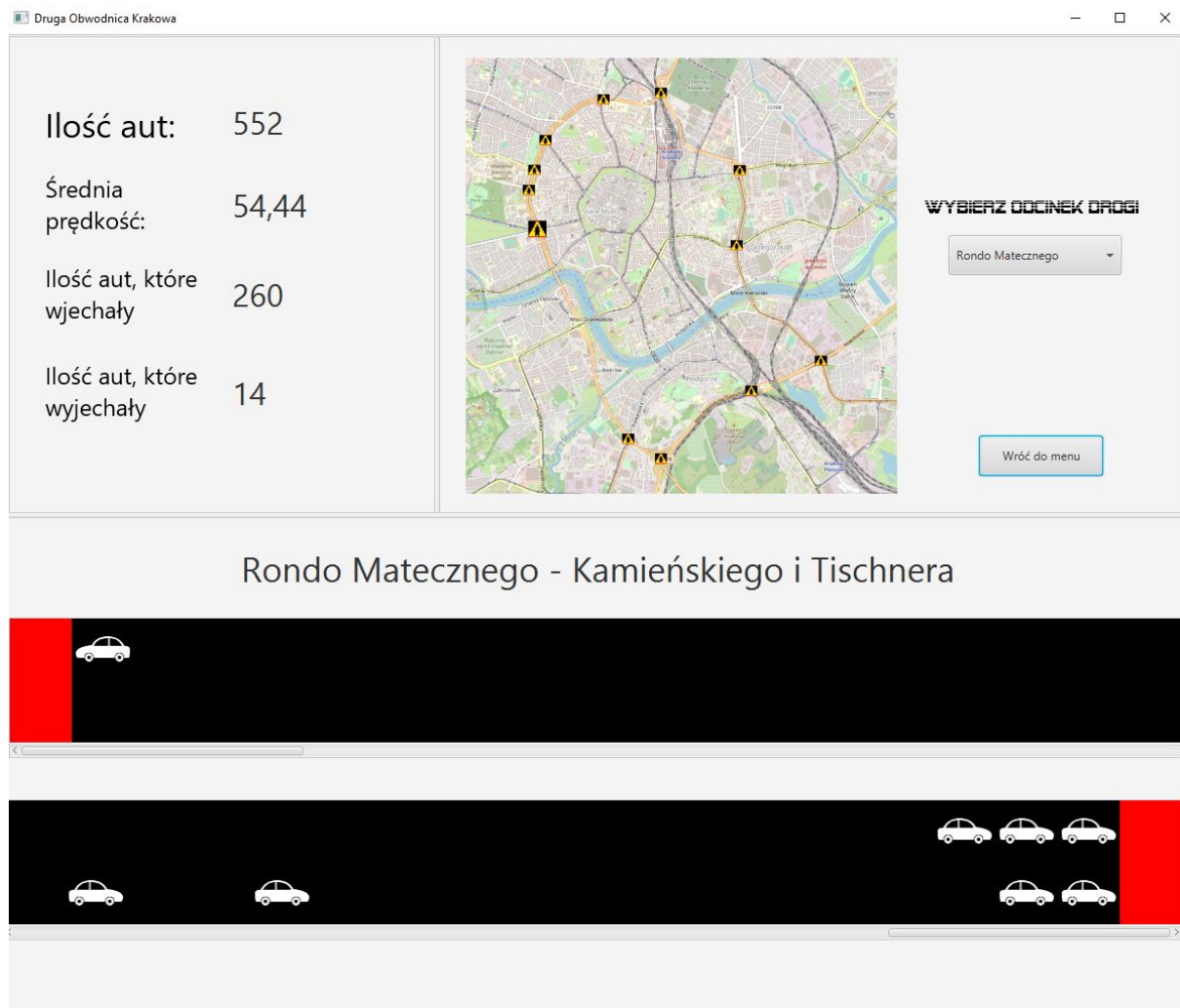
Stadion Cracovii

30

☒ Użyj wartości w zależności od wprowadzonych danych zamiast pory dnia

Rozpocznij Symulację

Interfejs animacji:



8. Podsumowanie i dalsze kierunki rozwoju

Wnioski:

- W wypadku znacznie zwiększonej ilości dołączających aut (np. mecz Wisły albo Cracovii) obwodnica zaczyna znacząco się przeciążać. Oczywiście wszystko w zależności od tego gdzie auta będą chciały dołączyć do obwodnicy. Dlatego też nasze rozwiązanie umożliwia użytkownikowi własnoręczne przeprowadzanie analiz danych poprzez wpisanie spersonalizowanych wartości.
- W wypadku zwiększonej ilości aut na obwodnicy prędkość ruchu zdecydowanie zwalnia. Stąd też bardzo ważnym jest przeprowadzanie stałych analiz danych, aby zoptymalizować czas trwania świateł.

- Gdy na drodze są mniej sprzyjające warunki pogodowe (deszcz, śnieg) jest bardziej prawdopodobne, iż kierowcy będą zmuszeni zredukować swoją prędkość, co powoduje spadek prędkości średniej na całej obwodnicy.