



WYDZIAŁ ELEKTROTECHNIKI AUTOMATYKI
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ

INFORMATYKA

ROK STUDIÓW: II

ROK AKADEMICKI: 2018/19

SYMULACJA RUCHU SAMOCHODÓW
NA II OBWODNICY KRAKOWA

KATARZYNA NYZNAR

IZABELA PACHEL

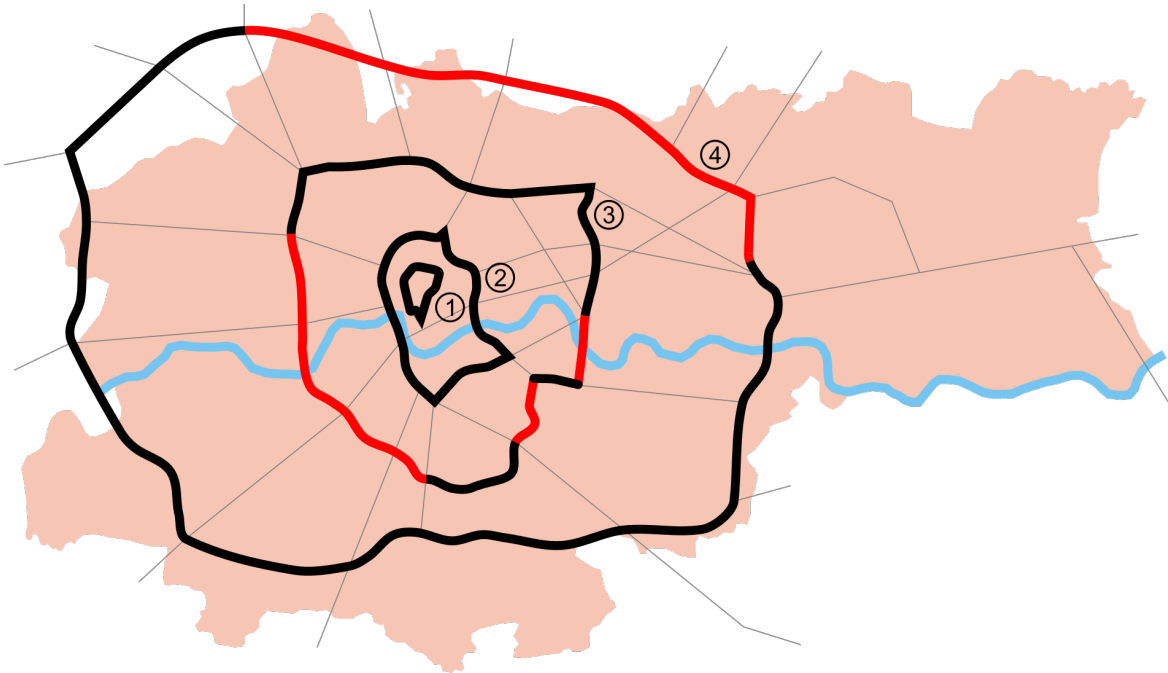
PRZEMYSŁAW BIELECKI

Spis treści

1	Wstęp	3
2	Idea automatów komórkowych	3
3	Model Nagela-Schreckenberga	4
4	Zmodyfikowany model Na-Sch	6
5	Reprezentacja	7
6	Algorytm	8
7	GUI	9
8	Podsumowanie i dalsze kierunki rozwoju	14

1 Wstęp

Celem niniejszej pracy jest stworzenie symulacji samochodowego ruchu miejskiego na II Obwodnicy Krakowa w oparciu o model Nagela-Schreckenberga z wykorzystaniem automatów komórkowych. W modelu uwzględniona ma zostać specyfika ruchu pojedynczego pojazdu w sytuacjach takich jak redukcji prędkości, zmiany pasa ruchu, wjazdu/zjazdu z obwodnicy oraz zbliżania się do sygnalizacji świetlnej.



Rys. 1: Obwodnice Krakowa

2 Idea automatów komórkowych

Automat komórkowy jest systemem składającym się z pojedynczych komórek, znajdujących się obok siebie. Każda z komórek może przyjąć jeden ze stanów, przy czym liczba stanów jest skończona. Stan komórki zmieniany jest synchronicznie zgodnie z regułami mówiącymi, w jaki sposób nowy stan komórki zależy od jej obecnego stanu i stanu jej sąsiadów.

Automaty komórkowe, których struktury opisane są przez siatkę komórek oraz ich stany, przejścia i reguły tych przejść, są modelami matematycznymi. Tworzą one środowisko dla większych dyskretnych klas modeli, ponieważ wszystkie opisujące je struktury przyjmują wartości dyskretne.

Każdy automat komórkowy składa się z n -wymiarowej regularnej, dyskretnej siatki komórek, każda komórka jest taka sama, cała przestrzeń siatki musi być zajmowana w całości przez komórki ułożone obok siebie. Każda z nich posiada jeden stan ze skończonego zbioru stanów. Ewolucja każdej komórki przebiega według tych samych ściśle określonych reguł lokalnych (jednorodność), które zależą wyłącznie od poprzedniego stanu komórki oraz od stanów skończonej ilości komórek - sąsiadów. Ewolucja następuje w dyskretnych przedziałach czasowych, jednocześnie dla każdej komórki (równoległość). W automacie komórkowym komórka jest automatem.

3 Model Nagela-Schreckenberga

Za główny wzorzec modelu posłuży nam model Nagela-Schreckenberga (zwany krócej Na-Sch), klasyczny model oparty na automatach komórkowych opisujący ruch samochodowy na autostradzie. W modelu Na-Sch przyjęty rozmiar komórki wynosi 7.5m.

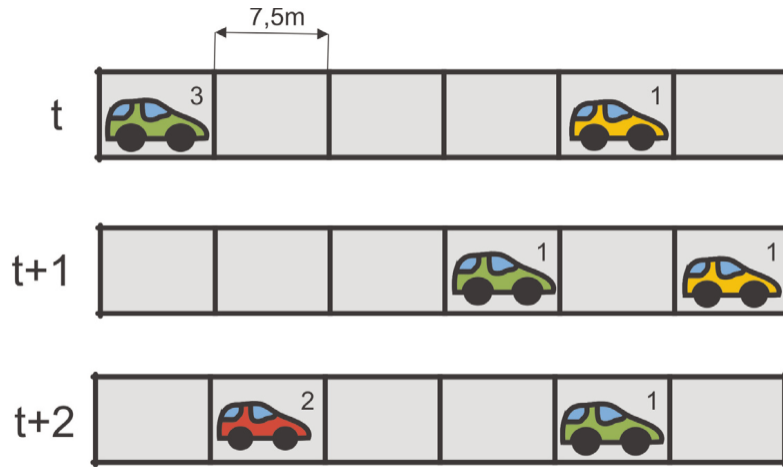
Model Na-Sch jest opisywany przez następujące reguły:

- Każda komórka może być pełna (zawierać pojazd) lub pusta (nie zawierać pojazdu)
- Każdy pojazd ma przypisaną prędkość v_{ki} , ograniczoną z góry przez v_{max}
- Prędkość jest liczbą komórek, o które przesunie się pojazd w kolejnym kroku
- Przyspieszenie: $v(t+1) = \min(v(t) + 1, v_{max})$, gdzie $v(t)$ to prędkość aktualna
- Hamowanie: $v(t+1) = \min(v(t), g(t) - 1)$, gdzie $g(t)$ to liczba pustych komórek między pojazdami
- Element losowy: prawdopodobieństwo p , że zajdzie $v(t+1) = \max(v(t) - 1)$, jeżeli $v(t) \geq 1$

- Ruch: $x(t + 1) = x(t) + v(t)$

W każdym kroku dla każdego pojazdu wszystkie powyższe reguły sprawdzane są w następującej kolejności:

- przyspieszenie
- sprawdzenie czy nie nastąpi kolizja i ewentualna redukcja prędkości
- losowe zmniejszenie prędkości o 1 z danym prawdopodobieństwem
- przemieszczenie pojazdu



Rys. 2: Ruch w modelu Na-Sch

W związku z tym, że model Na-Sch odwzorowuje zachowanie pojazdów na autostradach, jest zawodny w przypadku ruchu miejskiego. Przyspieszenie pojazdów przyjmuje zbyt duże wartości, a hamowanie służy jedynie zapobieganiu kolizjom i zachowaniu odstępów między pojazdami. Dla ruchu w mieście jest to podejście zbyt niedokładne, należy uwzględnić skrzyżowania, sygnalizację świetlną, zjazdy itp. Naturalnie, pojazdy muszą zmniejszać prędkość zbliżając się do przeszkody lub przed wykonaniem skrętu. Dlatego w naszej implementacji model Na-Sch wymagał wprowadzenia pewnych modyfikacji.

4 Zmodyfikowany model Na-Sch

W celu zwiększenia dynamiki poruszania się samochodów, zmienione zostały zasady ich hamowania. Wprowadzone także zostały nowe reguły kierujące ruchem. Przedstawiają się one w następującej kolejności:

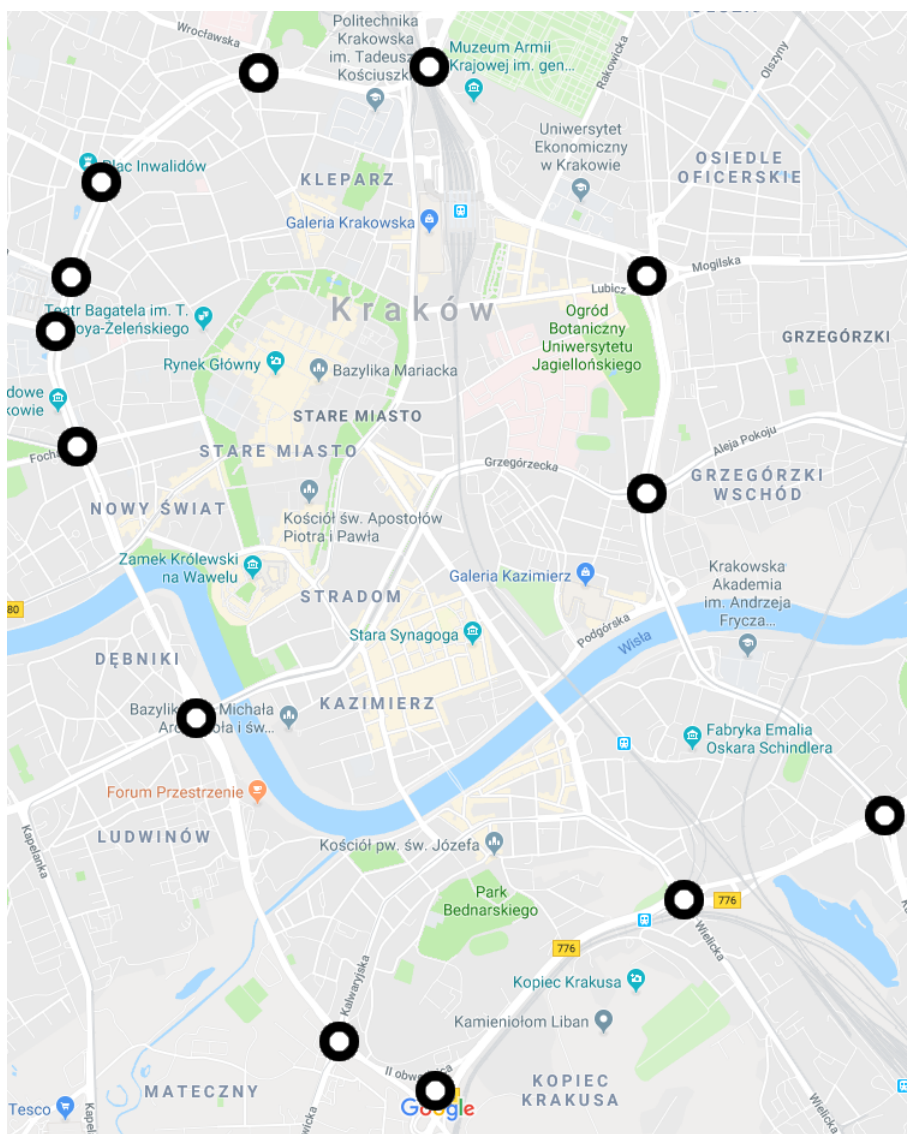
- sprawdzenie, czy samochód zbliża się do docelowego zjazdu, jeśli tak, to:
 - redukcja prędkości
 - zmiana pasa w kierunku zjazdu (jeśli jeszcze na nim nie jesteśmy)
- sprawdzenie, czy samochód zbliża się do czerwonych świateł, jeśli tak - redukcja prędkości, jeśli nie - przyspieszenie (zgodne z regułami Na-Sch)
- sprawdzenie bezpiecznej odległości od samochodu przed nami, jeśli wystąpi kolizja:
 - próba zmiany pasa
 - jeśli niemożliwa jest zmiana pasa - redukcja prędkości
- losowe zmniejszenie prędkości o 1 z danym prawdopodobieństwem
- przemieszczenie pojazdu

W przypadku konieczności hamowania w celu uniknięcia kolizji, samochód przyjmuje prędkość znajdującego się przed nim pojazdu. Zapobiega to niepotrzebnemu zatrzymywaniu się samochodów i blokowaniu obwodnicy oraz zwiększa dynamikę ruchu.

Ponadto pod uwagę wzięte zostały też różne preferencje kierowców co do prędkości. Każdy samochód otrzymuje losowo generowaną prędkość maksymalną, której nie będzie przekraczał. Ma to na celu wierniejsze oddanie ruchu miejskiego, w którym nie każdy kierowca postępuje w ten sam sposób.

5 Reprezentacja

Sieć dróg została przedstawiona jako graf skierowany, którego węzły odpowiadają skrzyżowaniom, a krawędzie odcinkom drogi. Każdy węzeł odpowiada za generowanie nowych samochodów oraz usuwanie tych, które opuszczają obwodnicę. Węzły są wyposażone w dodatkowy atrybut reprezentujący sygnalizację świetlną. Zmienia on swoją wartość w równych odstępach czasu. Uwzględnione zostało 14 głównych skrzyżowań II obwodnicy, co widać na rysunku poniżej.

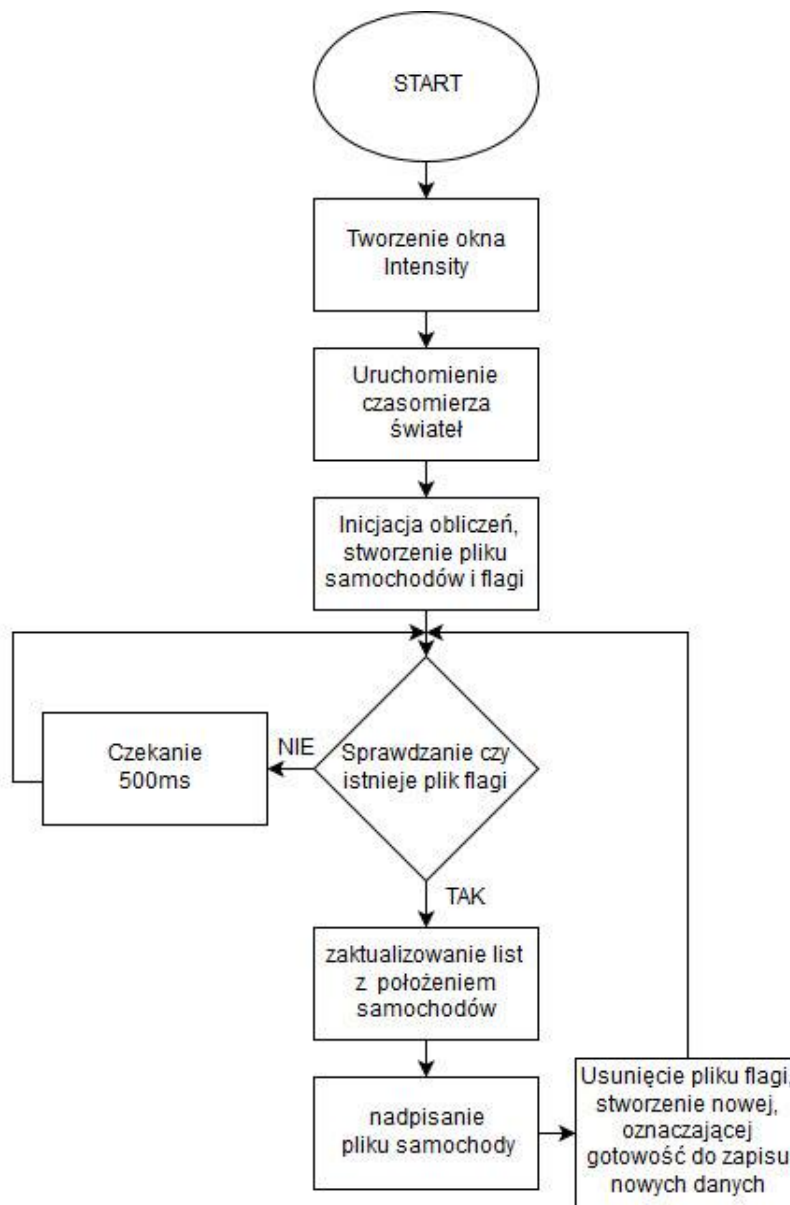


Rys. 3: Skrzyżowania uwzględnione w programie

6 Algorytm

Językiem programowania zastosowanym w algorytmie jest Java. Na początku tworzone jest okno *Intensity*, dzięki któremu użytkownik będzie miał możliwość wejścia w interakcję z programem, tj. zmiany natężenie ruchu samochodów oraz zobaczenia aktualnej liczby samochodów znajdujących się na obwodnicy. Następnie, po uruchomieniu timera odpowiadającego za zmianę świateł, inicjowane są obliczenia pozycji samochodów na drodze oraz tworzone są pliki potrzebne do komunikacji międzyprogramowej. W kolejnym kroku program wchodzi do pętli, gdzie pozostanie aż do zakończenia pracy. W pętli tej sprawdzany jest stan (istnienie) pliku - flagi, ustawianego przez program odpowiedzialny za wyświetlanie obwodnicy, gdy gotowy jest przyjąć nową porcję danych. Jeśli flaga jest ustawiona, następuje aktualizowanie pozycji samochodów, nadpisywanie pliku ze stanem obwodnicy i ustawianie flagi dla GUI. Jeśli nie - ponownie sprawdzamy stan flagi po odczekaniu 0.5s.

Schemat blokowy działania algorytmu:



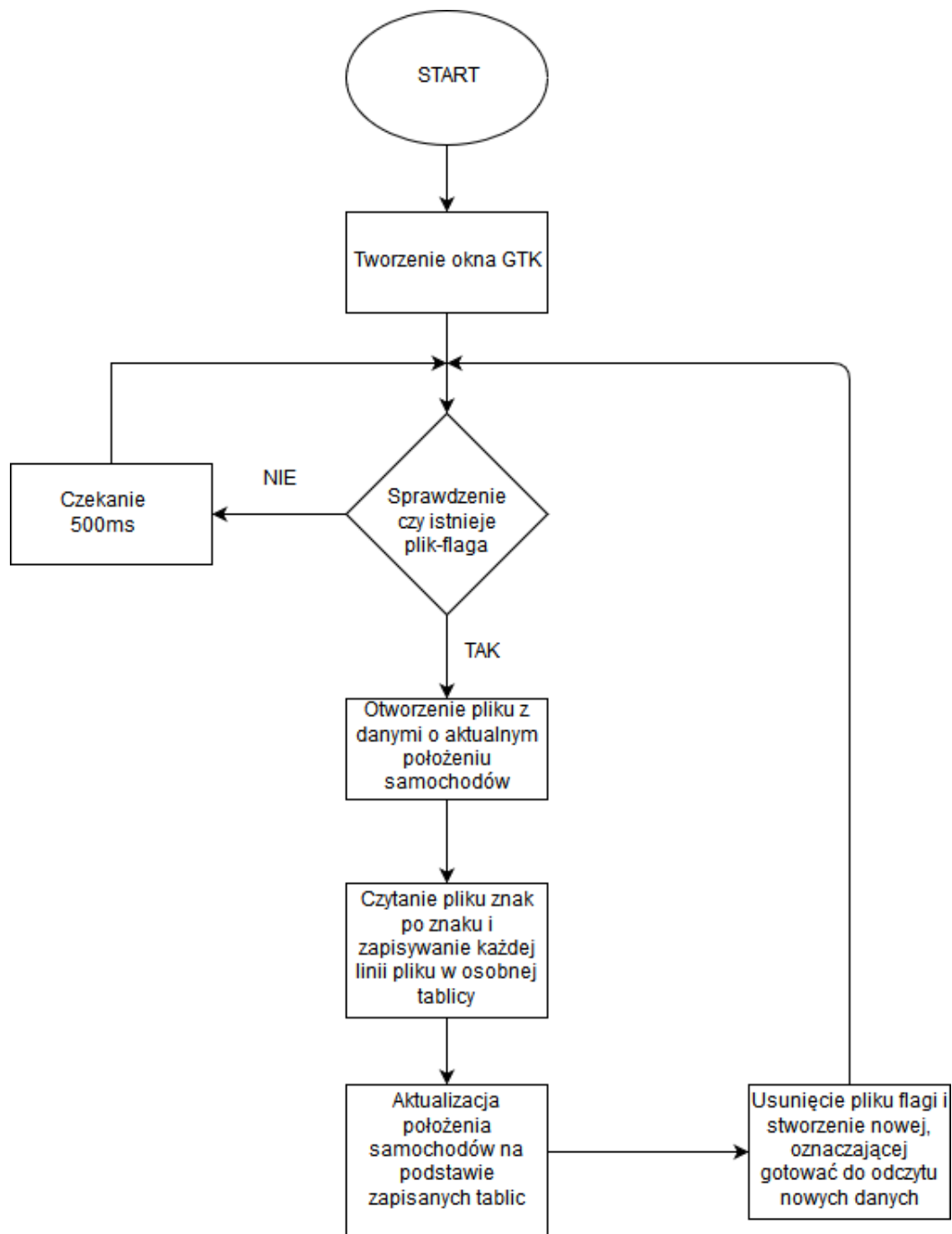
Rys. 4: Schemat blokowy działania programu

7 GUI

Interfejs graficzny programu został zaimplementowany z wykorzystaniem bibliotek Cairo oraz GTK. Program odpowiedzialny za GUI w pierwszej kolejności tworzy okno a następnie „drawing area”. Następnie łączy sygnał draw z funkcją odpowiedzialną za rysowanie oraz ustawia właściwości okna (tytuł, rozmiar, pozycja w której się pojawi itp.). Następ-

nie w pętli głównej programu wywoływana jest funkcja `g_timeout_add()` wywołująca funkcję `car_movement()` co 500ms. W funkcji `car_movement()` sprawdzany jest status pliku-flagi. Jeśli jest odpowiedni funkcja otwiera plik z danymi o pozycji samochodów na obwodnicy. Jeśli flagi nie ma funkcja wraca do głównej pętli po czym jest ponownie wywoływana po 500ms. W przypadku spełnienia warunków odczytu otwierany jest plik wygenerowany przez główny program. Następnie wartości z pliku są zapisywane do sześciu tablic typu `int` o długości odpowiadającej długości obwodnicy (tj. 1666 komórek). Każda tablica odpowiada za jeden pas ruchu. Po Przepisaniu danych program GUI usuwa stary plik-flagę i tworzy nowy, informując główny program że może rozpocząć generowanie nowego pliku z danymi. Następnie wysyłany jest sygnał do `drawing_area` w celu przerysowania obszaru na sposób odpowiadający nowej zawartości tablic. Rysowanie przebiega na podstawie wcześniej wyliczonych współrzędnych – obwodnica jest podzielona na proste odcinki o zmierzonej długości (w metrach). Długość następnie jest dzielona przez 7.5 (długość jednej komórki). W wyniku w/w działań otrzymujemy długość odcinka w komórkach. Następnie liczone jest przesunięcie `x` i `y` na danym odcinku. W kolejnym korku długość w pionie i poziomie jest dzielona przez ilość kratek. W ten sposób otrzymujemy kolejne pozycje na obwodnicy. Ponieważ jest to sposób z różnych przyczyn (nie każdy odcinek da się podzielić równo przez 7.5, `x` i `y` muszą być całkowite itp.) przybliżony w ostatnim kroku trzeba dodać odpowiedni offset żeby dopasować wyświetlane samochody do wcześniej narysowanych obrysów obwodnicy. W tym miejscu należy również dodać że z powodów czasowych nie byliśmy w stanie wykonać dokładnie powyższego algorytmu. Całość działa przy pomocy wygenerowanego wcześniej zestawu kilkuset plików testowych.

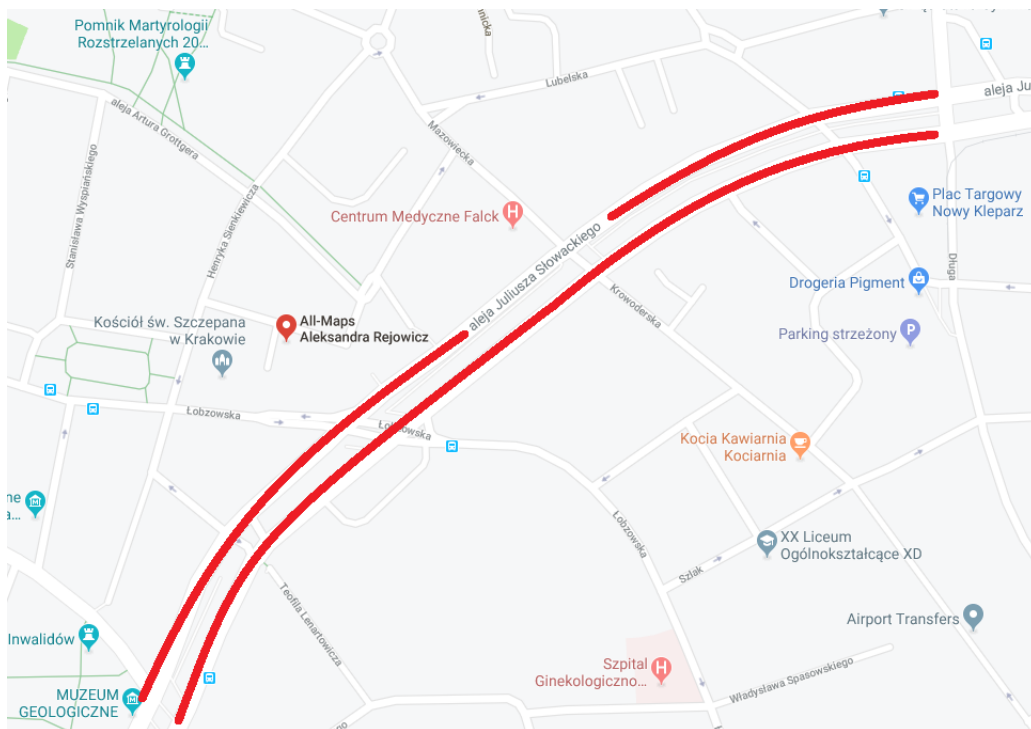
Schemat blokowy działania interfejsu przedstawia się następująco:



Rys. 5: Schemat blokowy działania interfejsu

Na ten moment zaimplementowane zostało wyświetlenie fragmentu drogi od Nowego Kleparza do ulicy Czarnowiejskiej, jednak trzeba zaznaczyć, że obliczenia w tle wykonywane są dla całości obwodnicy. W osobnym oknie mamy możliwość regulacji natężenia

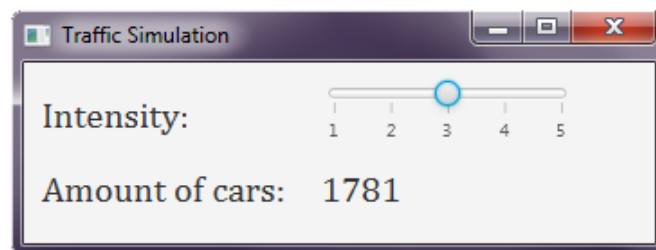
ruchu określanego prawdopodobieństwem generowania się samochodów na skrzyżowaniach. Najmniejsza wybrana wartość odpowiada prawdopodobieństwu 50%, natomiast największa - ok. 83%. Wyświetlana jest także informacja o obecnej liczbie samochodów.



Rys. 6: Wyświetlany fragment obwodnicy



Rys. 7: Graficzna reprezentacja fragmentu obwodnicy



Rys. 8: Regulacja natężenia ruchu na obwodnicy

8 Podsumowanie i dalsze kierunki rozwoju

Celem projektu było stworzenie symulacji ruchu miejskiego na II Obwodnicy Krakowa w oparciu o automaty komórkowe oraz model Nagela-Schreckenberga. Model ten docelowo został stworzony, aby symulować ruch pojazdów na autostradzie. Celem realistycznego przedstawienia zachowania samochodów w dużym natężeniu ruchu, został on odpowiednio zmodyfikowany, a zachowanie samochodów dostosowane do poruszania się po mieście.

Kolejną fazą rozwoju projektu będzie możliwość wyświetlenia dowolnie wybranego fragmentu obwodnicy (przedstawienie całości jest niemożliwe ze względu na dużą skalę). Użytkownik będzie miał możliwość wybrania z mapy obwodnicy pożądanego fragmentu, który następnie zostanie wyświetlony w osobnym oknie. Ponadto zmienione zostaną zasady generowania nowych samochodów. Każde skrzyżowanie charakteryzowało się będzie daną przepustowością, której wartość pobierana będzie z oficjalnych źródeł.

Bibliografia

- [1] Tomasz Lewowski, *Ruch drogowy, Model Nagela-Schreckenberga*.
- [2] Jarosław Wąs, Rafał Bieliński, Bartłomiej Gajewski, Patryk Orzechowski, *Problematyka modelowania ruchu miejskiego z wykorzystaniem automatów komórkowych*.
- [3] Krzysztof Małecki, Mateusz Rokita, Jarosław Wątróbski, *Wykorzystanie automatów komórkowych w modelowaniu ruchu drogowego*.
- [4] Ewa Dudek-Dyduch, Jarosław Wąs, *Formalizacja automatów komórkowych w zagadnieniach symulacji dynamiki pieszych*.
- [5] C. Burstedde, K. Klauck, A. Schadschneider, J. Zittartz, *Simulation of pedestrian dynamics using a two-dimensional cellular automaton*.
- [6] Andreas Schadschneider, *Traffic flow modelling*, <http://www.thp.uni-koeln.de/~as/Mypage/traffic.html>
- [7] Kai Nagel, Michael Schreckenberg, *A cellular automaton model for freeway traffic*, <https://pdfs.semanticscholar.org/17a6/135c5b7af91238989514f4c709c8146031d5.pdf>
- [8] Antonina Chechina, Natalia Churbanova, Marina Trapeznikova, *Multilane Traffic Flow Modeling Using Cellular Automata Theory*, https://www.epj-conferences.org/articles/epjconf/pdf/2018/08/epjconf_mmcp2018_06003.pdf
- [9] David Hartman, *Head Leading Algorithm for Urban Traffic Modeling*, <http://www.scs-europe.net/services/ess2004/pdf/log-13.pdf>
- [10] Esser J., Schreckenberg M., *Microscopic Simulation of Urban Traffic Based on Cellular Automata*.